updated for 1.3

Design Made Easy with Inkscape®

A practical guide to your journey
from beginner to pro-level vector illustration

CHRISTOPHER ROGERS

‹packt›

Design Made Easy with Inkscape

A practical guide to your journey from beginner to pro-level vector illustration

Christopher Rogers

packt

BIRMINGHAM—MUMBAI

Design Made Easy with Inkscape

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Group Product Manager: Rohit Rajkumar

Publishing Product Manager: Ashitosh Gupta

Content Development Editor: Abhishek Jadhav

Technical Editor: Simran Ali

Copy Editor: Safis Editing

Project Coordinator: Manthan Patel

Proofreader: Safis Editing

Indexer: Manju Arasan

Production Designer: Vijay Kamble

Marketing Coordinator: Nivedita Pandey

First published: May 2023

Production reference: 1070423

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-80107-877-1

`www.packtpub.com`

I'd like to quickly thank all those who provided feedback and advice on this book. This includes my friend and mentor Martin Owens (aka Doctormo), who has worked with me to fix many of the bugs that I've identified during the writing process and make Inkscape a significantly better user experience. Ultimately, it's his work and the work of others in the project that make Inkscape an easier software to teach and learn from. Please do consider supporting his Inkscape development work here: https://www.patreon.com/doctormo.

I'd also like to thank my folks, Charles and Starrlet Rogers, for always supporting my creative endeavours, my family and friends for believing in me, my girlfriend Aless for encouraging me to complete this process, and my sister Lisa for providing some financial assistance and the ever-important question "How's the book coming?".

A big thanks also goes to the Open Document Foundation for their work on LibreOffice Writer, which enabled me to write this book and communicate with my publisher on Linux, using only free and open source software.

Lastly, dear reader, I'd like to thank you for buying this book to help support my work in the Inkscape project and teaching activities on Inkscape's various social media platforms. It's my pleasure to help you learn one of the greatest design tools gifted to the world by Free and Open Source Software (FOSS) developers, contributors, and the greater Inkscape community.

Christopher Rogers

Contributors

About the author

Christopher Rogers (also known as **C. Rogers**) is an international artist, designer, and animator, presently working out of London, England. American born, he started his career after university making small websites and advertising media for the Coloradoan newspaper.

Through his freelance contracting work, he began working with Ashtead Retail & Wholesale Ltd (`tuff-luv.com`, `alstoncraig.com`) with headquarters in Weybridge, England. He was eventually sponsored by this company to come to London and work as a full-time graphics and product designer.

It was in Leeds that he attended his first Inkscape Hackfest and subsequently joined the project as a contributor, then eventually was elected to the board of directors (also known as Inkscape PLC). He has served on the board of Inkscape for the past five years as of the writing of this book and produced three of Inkscape's annual release videos (1.0, 1.1, and 1.2), which can be found on YouTube on the official Inkscape channel.

He has produced videos and teaching materials for Inkscape for the University of Exeter and helped with the development of new Inkscape features over the years. He loves to help others get to grips with powerful FOSS in the name of creative freedom.

You can check out his latest design portfolio here: `https://crogersmedia.com/crogers_portfolio.pdf`.

About the reviewer

Stephanie Hardy is a 2D digital artist and animator. She has always had an interest in drawing and cartoons. In 2014 she completed an associate degree in computer animation. She has expertise with the Adobe Creative Suite and has used Inkscape since 2017.

In 2020 she started freelancing and has loved it ever since, now having a career doing what she loves all from the comfort of her home. Her work includes logos, cartoon animations, mascots, coloring pages, t-shirt designs, and more!

Table of Contents

3

Drawing Shapes with the Shape Tools 51

4

Automatic Shape Alignment in Inkscape 67

5

Node Editing – Modifying Your Shapes with Nodes and Curves 91

Part 2: Advanced Shape Editing

9

Clips and Masks 211

10

Automation with Clones and Linked Files 225

Part 3: Inkscape's Power Tools

11

Organization Using Layers 243

12

Live Path Effects 259

13

Filters and Extensions 275

14

Vectorizing with Trace Bitmap 293

15

Document Properties, Pages, Exporting, and Printing 307

Preface

With the power and versatility of the Inkscape software, making charts, diagrams, illustrations, and UI mockups with infinite resolution becomes enjoyable, and if you're looking to get up to speed with vector illustration in no time, this comprehensive guide has got your back! *Design Made Easy with Inkscape* is easy to follow and teaches you everything you need to know to create graphics that you can use and reuse forever for free!

You'll benefit from the author's industry experience as you go over the basics of vector illustration, discovering tips and tricks for getting professional graphics done fast by leveraging Inkscape's powerful toolset. This book teaches by example, using a great variety of use cases, from icons and logos to illustration, web design, and product design. You'll learn about hotkey charts and take a best-practice approach developed over 10 years of using Inkscape as a design tool in production. What's more, this book also includes links to free graphics resources that you can use in all your projects.

Whether you're a new user or a professional, by the end of this book, you'll fully understand how to use Inkscape and its myriad of excellent features to make stunning graphics for your projects.

Who this book is for

This book is for aspiring designers, developers, and anyone who wants to leverage the power of Inkscape for vector illustration. No prior knowledge of graphics applications or vector and raster graphics is required. This book simplifies Inkscape for the laymen; all you need to know is how to use a mouse and keyboard.

Wait, why is Inkscape free?

I'm glad you asked! There's always some catch, isn't there? Well, the truth is, Inkscape is different. It's made by volunteers and even paid developers whose primary goal is to make software that everyone can use for free forever. The driving force for the development of Inkscape isn't to make millions of dollars for some company but to make really good software in the spirit of sharing. That's pretty much it! Inkscape is free because every member of the Inkscape project has core beliefs that user freedoms are of the utmost importance.

That's cool, but why should I use Inkscape?

The aforementioned core beliefs also give Inkscape many advantages over alternatives, such as the following:

- **Inkscape respects your privacy**: Inkscape will never collect personally identifiable information or try and push ads on you. The project is funded by donations and volunteer work and by self-funded developers, so no advertising revenue is needed. Such data collection is also seen as an invasion of privacy by many in the community, so it's kept entirely out of Inkscape as a matter of principle.
- **Inkscape breaks down international boundaries**: Inkscape is available in every country of the world in a multitude of languages, with a lively and active community of contributors. No government in the world can shut down Inkscape because the source code is available everywhere for everyone to use however they like!
- **Inkscape breaks down financial barriers**: Inkscape is free for everyone to use, regardless of income. Moreover, Inkscape can be used to make money, learn design, and raise the quality of life for those who would otherwise be blocked by paywalls imposed by proprietary software vendors.
- **Inkscape is completely cross platform**: Because of its open source code, users of macOS, Windows, and Linux can compile Inkscape code into software that works the same across all three platforms. This means that you can work together regardless of the operating system you use. Thus, Inkscape is far better for collaboration than proprietary alternatives, many of which only cater to one or two operating systems.
- **Inkscape is software made by designers**: Some of our contributors are professional designers who help make Inkscape a better production tool, often requesting features unavailable in other software. This is originally what got me interested in Inkscape – it had features that were significantly better than even the most expensive paid programs.

  To this day, there are still features that I'd love to see in Adobe Illustrator that Inkscape has had for years. I've helped add some of these features to Inkscape to save myself time in graphics production, but more importantly, to save everyone time!
- **Inkscape is free software**: Not simply free as in no-cost, but free as in freedom. This comes from Inkscape's license, called the **GNU Public License** (**GPL**). GPL software allows you to download, edit, use, and share Inkscape's code for any purpose. Moreover, everyone who contributes to Inkscape must provide the source code under the same license. This ensures Inkscape will always be free and can never be locked down or put behind a paywall.
- **Inkscape gives you control over your assets**: Since there's no paywall, graphics assets designed in Inkscape will always be free to open in Inkscape in an industry-standard SVG format, which is portable to other graphics applications.

- **Inkscape is more fun**: In addition to getting all the new features for free, you can even help test and improve brand-new features in development. You don't need to be a programmer to do this and many new features are just fun to use. We will see a ton of them in this book, but the fact is more are added every year and it's really exciting to be part of bringing that fun to people worldwide.
- **Inkscape is a community-driven project**: Inkscape is used by millions of people all over the globe! No matter what your skill level is, it's easy and fun to be part of the community. The project has live chat boards to ask questions and thousands of active members across most major social media networks, including Facebook, Mastodon, Twitter, and so on.

  There are even chat rooms where you can talk to the developers. Best of all, everyone is friendly and respectful. In fact, it's a requirement! All of Inkscape's forums are moderated to make sure everyone has a good time. It's a great way to share art, tips and tricks, and design knowledge. Come be a part of the fun!

What this book covers

Chapter 1, *The Inkscape Interface*, gives you a tour of Inkscape's menus, dialogs, parts, and controls.

Chapter 2, *Moving and Styling Shapes*, introduces you to shapes in Inkscape, transforming and styling them with stroke and fill.

Chapter 3, *Drawing Shapes with the Shape Tools*, shows you how to use Inkscape's shape tools and options.

Chapter 4, *Automatic Shape Alignment in Inkscape*, helps us understand how to align shapes relative to each other and the page automatically.

Chapter 5, *Node Editing – Modify Your Shapes with Nodes and Curves*, helps us understand how to make custom shapes with Inkscape's node editing tools.

Chapter 6, *Fast Shape Editing with Path Operations and the Shape Builder Tool*, uses open and closed shapes as tools to make more complex combinations of shapes quickly.

Chapter 7, *Using Text in Inkscape*, helps us understand how to add titles and paragraphs and wrap text around and through shapes.

Chapter 8, *Advanced Shading and Coloring*, deals with coloring techniques to achieve depth, shadows, and textures with gradients, blurs, and blending modes.

Chapter 9, *Clips and Masks*, uses shapes and images to cut and mask other shapes non-destructively.

Chapter 10, *Automation with Clones and Linked Files*, helps us understand how to set up super templates where one change to an object changes them throughout the document.

Chapter 11, *Organization Using Layers*, helps us understand how to show, hide, and organize parts of complex files using named layers.

Chapter 12, *Live Path Effects*, adds amazing non-destructive effects to your shapes for blazing-fast complex shape generation.

Chapter 13, *Filters and Extensions*, helps you to make your own filters and use extensions from the Inkscape community, added on to Inkscape for all manner of illustration fun.

Chapter 14, *Vectorizing with Trace Bitmap*, converts raster (pixel) images to vectors for advanced editing and super-fast illustrations.

Chapter 15, *Document Properties, Pages, Exporting, and Printing*, dives deep into getting the most out of your files from Inkscape.

To get the most out of this book

Although you can follow most of the lessons in this book with older versions of Inkscape, I've written this book specifically for the brand-spanking-new version, which is 1.3 at the time of writing. My aim is to give you the most value out of reading this book. There is a ton of new time-saving awesomeness in 1.3 that makes it worth the upgrade if you happen to be using an older version.

Software/hardware covered in the book	**Operating system requirements**
Inkscape 1.3	Windows, macOS, or Linux

You can download the latest version of Inkscape at `https://www.inkscape.org`, which contains installation instructions for your particular operating system.

Download the example files

You can download the example files for this book from GitHub at `https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape`. If there's an update to the code, it will be updated in the GitHub repository.

We also have other bundles from our rich catalog of books and videos available at `https://github.com/PacktPublishing/`. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots and diagrams used in this book. You can download it here: `https://packt.link/57GQC`.

Conventions used

There are a number of text conventions used throughout this book.

`Code in text`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "The width value of your rectangle might be `300` pixels, cm, inches, and so on."

Bold: Indicates a new term, an important word, or words that you see onscreen. For instance, words in menus or dialog boxes appear in **bold**. Here is an example: "Select **Save as** from the **File** menu."

> **Tips or important notes**
>
> Appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, email us at `customercare@packtpub.com` and mention the book title in the subject of your message.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit `www.packtpub.com/support/errata` and fill in the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packt.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `authors.packtpub.com`.

Share Your Thoughts

Once you've read *Design Made Easy with Inkscape*, we'd love to hear your thoughts! Please select `https://www.amazon.in/review/create-review/error?asin=1801078777` for this book and share your feedback.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere?

Is your eBook purchase not compatible with the device of your choice?

Don’t worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don’t stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below

`https://packt.link/free-ebook/9781801078771`

2. Submit your proof of purchase
3. That’s it! We’ll send your free PDF and other benefits to your email directly

Part 1:
Finding Your Way Around

This part will introduce readers to the Inkscape interface and show the various tools and dialogs, what they do, and how to use them.

There are the following chapters in this section:

- *Chapter 1, The Inkscape Interface*
- *Chapter 2, Moving and Styling Shapes*
- *Chapter 3, Drawing Shapes with the Shape Tools*
- *Chapter 4, Automatic Shape Alignment in Inkscape*
- *Chapter 5, Node Editing – Modify Your Shapes with Nodes and Curves*

1
The Inkscape Interface

There's no doubt about it, Inkscape is a powerful design application. As such, there is a multitude of tools and functions to get the results you're after. When you open Inkscape for the first time, it can seem a bit overwhelming. Worry not – we'll have you up to speed in no time!

In this chapter, you will learn about the various parts of the Inkscape user interface, what they are used for, and how to navigate them. Knowing your way around will dramatically speed up your work in Inkscape, as you will spend less time searching for the tools you need, which of course leaves more time for using them.

By the end of this chapter, you should have a good idea of where to find things easily in Inkscape.

In this chapter, we will cover the following topics:

- Navigating the interface
- The Menu bar
- The Tool bar and Tool control bar
- The Desk and Page area
- The Swatch Palette
- The Color Indicators
- The Layer Info Indicator
- The Status bar
- The Docking area and dialogs
- The Commands bar
- The Snap control bar
- The Display Transform Control

Technical requirements

To make the most of this chapter, you'll need a computer running a recent version of Linux, macOS, or Windows.

You will, of course, also need a fresh copy of Inkscape 1.1, which you can download from the Inkscape website at `www.inkscape.org` (simply click the **DOWNLOAD** button and follow the instructions). As mentioned in the *Preface*, Inkscape is free to download and can be used for many purposes. Neat, huh?

I also highly recommend using a mouse as there are some functions tied to the mouse that are hard (or impossible) to do with a touchpad. Laptop trackpads are notoriously terrible for your wrists, and while they may suffice for casual mousing on the go, they are less precise and much harder on your wrists than a mouse for click-heavy graphic design tasks. As a laptop addict in university, I destroyed my wrists learning this lesson the hard way.

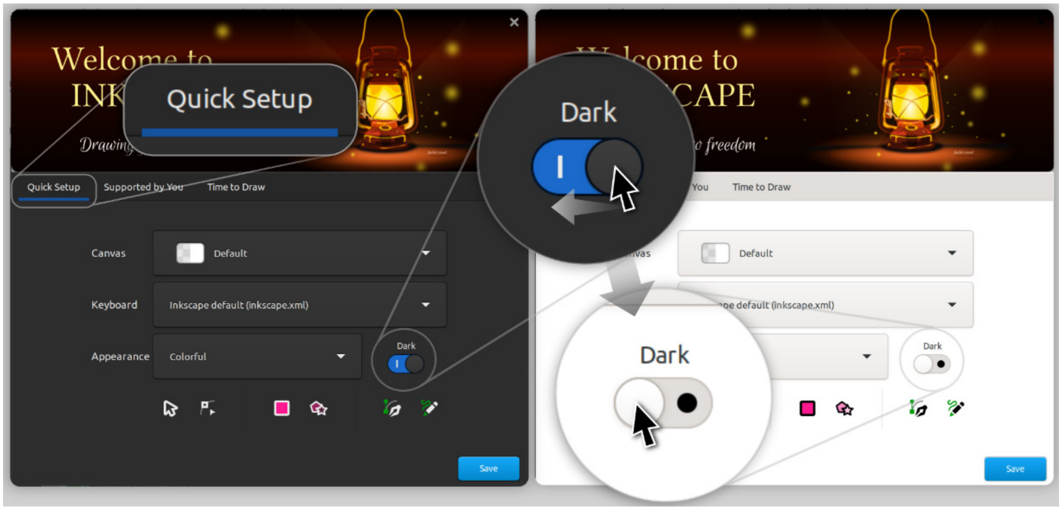
So, you can save yourself the medical bills and pain by investing in a good-quality mouse now. While I tend to recommend a Logitech mouse, even the cheapest off-brand mouse is better than the most expensive trackpad. If you can manage to get one with a third mouse button, then you get some nice extra functionality, such as being able to grab/pan the canvas just by holding it down.

Navigating the interface

The Inkscape user interface is broken up into panels that are organized like shelves for all the stuff you'll be using throughout this book. We will provide plenty of screenshots so that you don't have to memorize all the parts at this point. Think of this more as a guided tour than something to memorize and a reference for when you can't quite remember what that thing was called. You know... that thing. The thingy with the buttons... Yeah, that one.

But before we dive into the parts of Inkscape, it will help a little to make sure we're looking at the *same* Inkscape. I've opted to use light mode for the figures in this book, because it looks better in print, it saves ink (which is also eco-friendly), and smaller print tends to be easier to read.

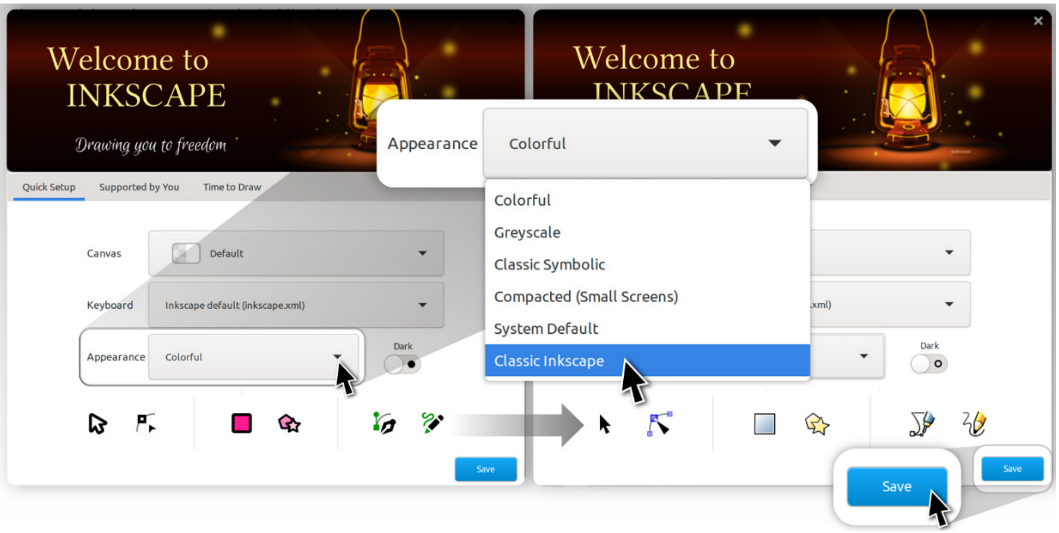
By default, Inkscape's got a brand-new, sexy dark mode. It's fine to leave it that way if you like it – just be aware that the icons will be slightly different than the ones in this book. For those keen on having things look the same, you can switch out of **Dark mode** in the **Startup** dialog area (also known as the **Welcome Screen** area) by flipping the **Dark mode** switch, as shown in *Figure 1.1*:



Canvas: Default
Keyboard: Inkscape default (inkscape.xml)
Appearance: Colorful

Figure 1.1 – Switching out of dark mode in the Startup dialog (Welcome Screen)

I've also opted to use the classic Inkscape icons because they are easier to see and also make this book usable for legacy versions of Inkscape that predate the new icon themes. You can get them looking the same as they are in this book by using the **Appearance** dropdown beside the **Dark** mode switch, as shown in *Figure 1.2*:



Canvas: Default
Keyboard: Inkscape default (inkscape.xml)
Appearance: Colorful

Figure 1.2 – Changing the icon theme to Classic Inkscape

That certainly looks more like it! However, there's one more gotcha; depending on how wide your screen is, Inkscape may look slightly different than in some of the screenshots in this book. For example, the laptop that I'm using to write this book is 3,840 x 2,160px, meaning it's a widescreen format of ratio 16:9. The main difference between this and Inkscape's default view is that the Commands bar is on the side of the screen rather than taking up space at the top.

Inkscape is smart enough to recognize my screen type and give me a wider view instead. Your view may be a bit different, so for consistency, you might want to switch the view to **Wide** instead. You can do this by clicking the **View** menu at the top of the screen and choosing **Wide**, as shown in *Figure 1.3*:

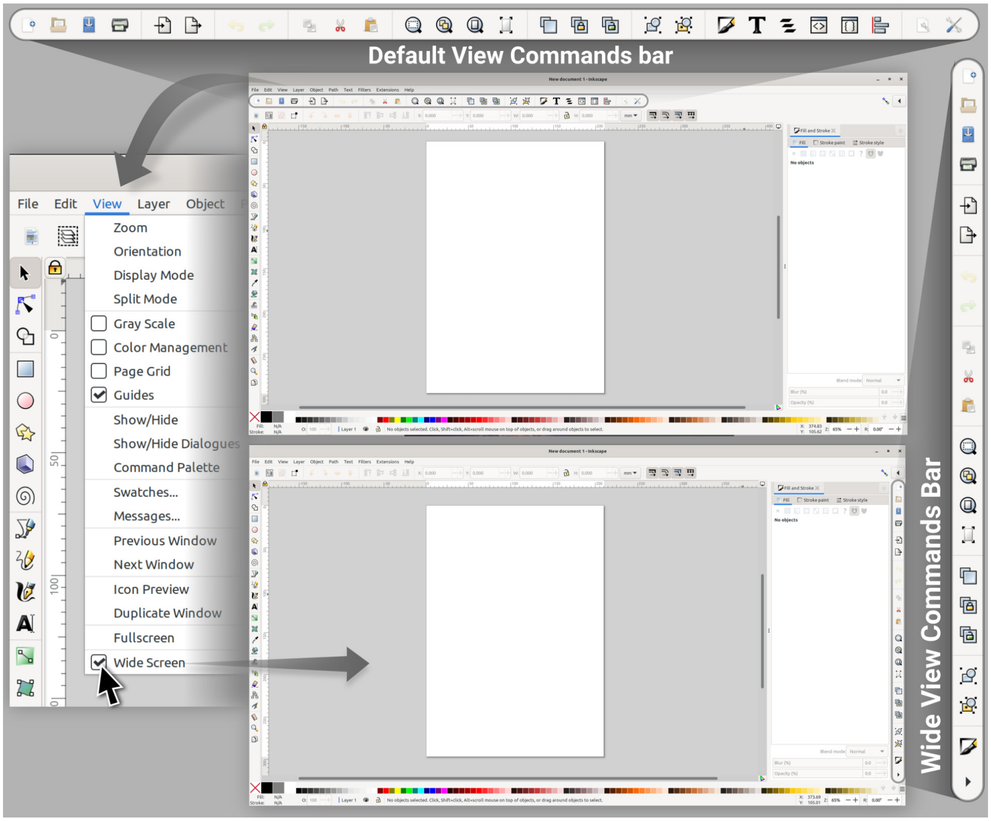


Figure 1.3 – Switching from the default view to the Wide view

Switching views is entirely optional and will not affect anything except the location of the Commands bar, which we will go over but not use very much as it's far more efficient to use hotkeys for most of the items in this menu.

It also gets in the way of the much more useful Tool control bar, which we will use extensively, so I recommend pushing it to the side to preserve your vertical screen space for drawing and reduce clutter.

Have a look at *Figure 1.4* for a breakdown of the parts. Here, I've cut up the **Wide Screen** interface into bits, spaced out so that you can see the different areas we'll be discussing in this section better:

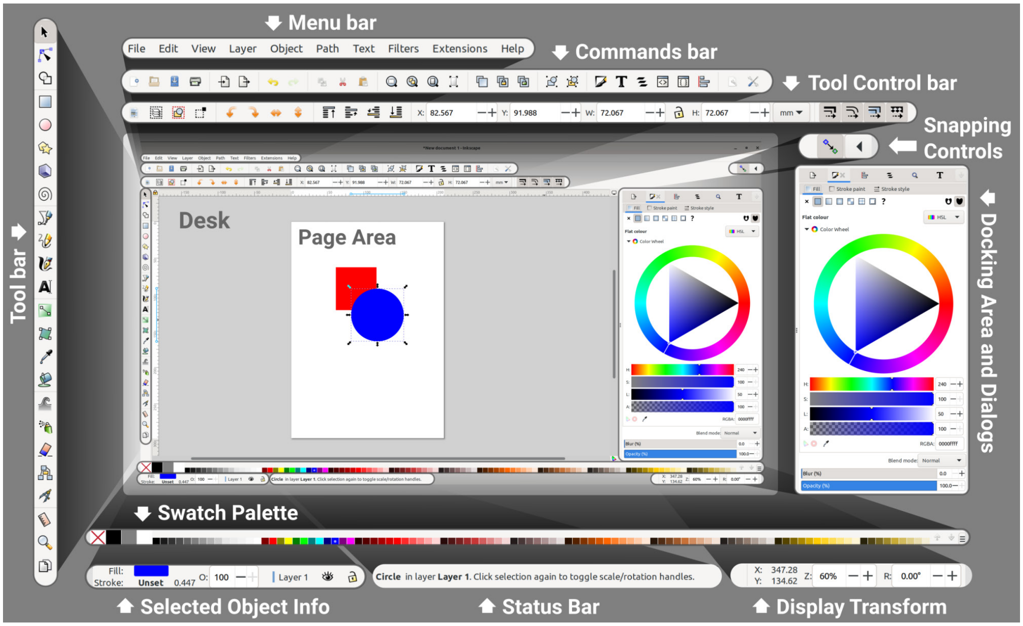


Figure 1.4 – Inkscape interface

Before we dive into the specifics of the various parts of the interface, it's worthwhile going over some conventions that Inkscape uses to assist you in finding your way around:

- **Hover to discover**: Hover your mouse cursor over an icon or box for a moment to see what it does (see *Figure 1.5* for an example of hovering to discover more information):

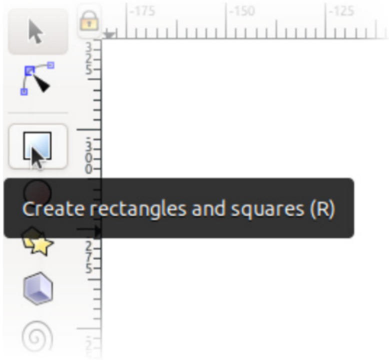
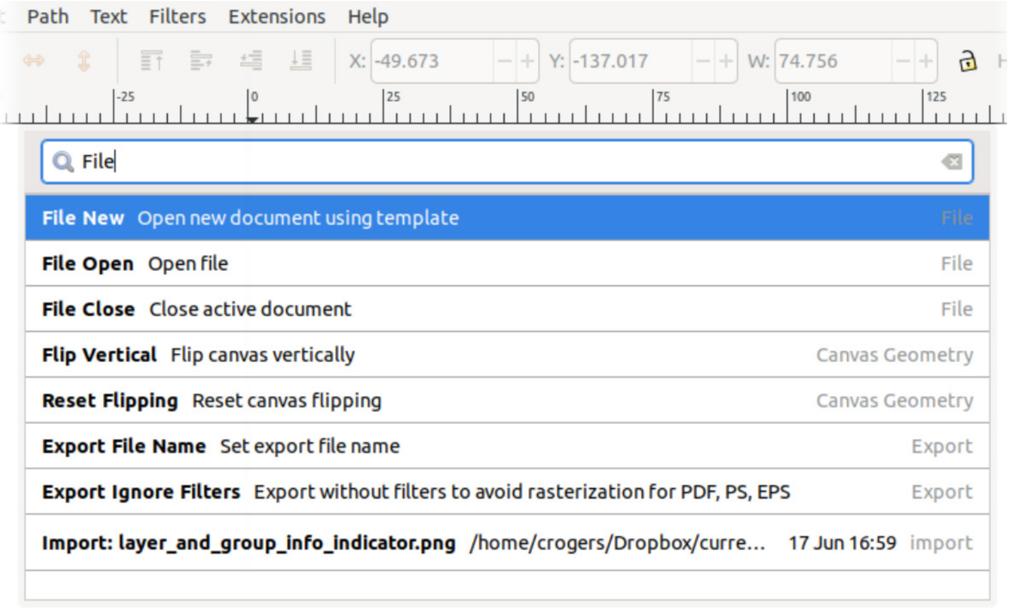


Figure 1.5 – Hovering over a button shows its function

- **Hotkeys**: A hotkey is a keyboard key (or a combination of them) that you can use to do something instantly rather than clicking through menus or looking through the icon bar. Inkscape will generally list the hotkey associated with the item, be it in the Menu bar or the hover information (*Hover to discover*, as mentioned in the previous bullet point).

  The more you memorize these hotkeys, the faster you can work and the less time you will need to spend clicking around. Eventually, you'll be able to *think about doing something and do it simultaneously* via hotkeys. A little time invested in doing this during your projects will save you untold amounts of time forever after.

- **Instant Search**: As of Inkscape 1.1, there is a new Command Palette that can be called up with the *Shift* + *?* hotkey. This is a search bar where you can type in what you want, and Inkscape will show you the results right away. This is a great way to find things quickly, and well worth remembering the hotkey. Got a question? Tap the *Shift* + *?* hotkey, as shown in *Figure 1.6*:



File

File New Open new document using template — File

File Open Open file — File

File Close Close active document — File

Flip Vertical Flip canvas vertically — Canvas Geometry

Reset Flipping Reset canvas flipping — Canvas Geometry

Export File Name Set export file name — Export

Export Ignore Filters Export without filters to avoid rasterization for PDF, PS, EPS — Export

Import: layer_and_group_info_indicator.png /home/crogers/Dropbox/curre... 17 Jun 16:59 — import

Figure 1.6 – The Command Palette in action

- **Buttons**: These require only a single click to perform the action. Some examples of these are the **Save** and **Load** buttons contained in the Command Palette, and the + and – keys located inside the Spin Boxes, which allow you to increment the value in the box.

- **Toggles**: These have the same appearance as buttons, but instead of performing a one-time function, they activate or deactivate something. They appear often in the Tool control bar, where you can see which tool or snapping options are active at a glance. They have two states – *on* and *off*.

When toggled on, the background of the icon will go a few shades darker to indicate that the button is depressed. Pressing it again when it's in this state toggles it off again, and the dark background will disappear. See *Figure 1.7* for examples of toggles and buttons in Inkscape:

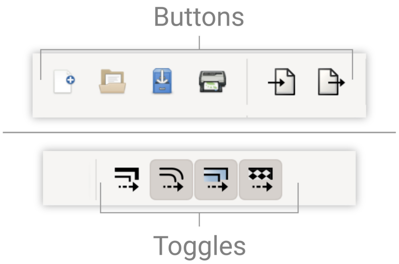
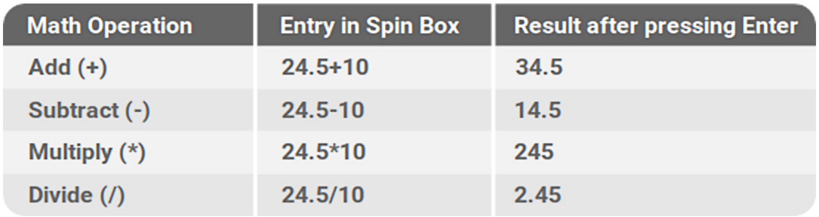


Figure 1.7 – Toggles versus buttons

- **Spin Boxes**: These are boxes that contain numeric values that can be incremented or decremented using the + and – buttons contained in them. Pressing the + button increments, while – decrements. You can also click inside the box to edit the value manually.

Here are a few neat tricks to use with Spin Boxes:

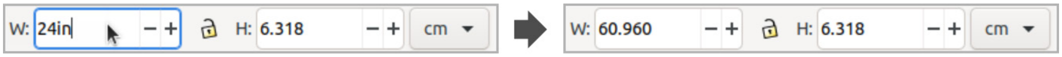
- Double-click inside the Spin Box to highlight the integer part or the decimal part of the number. You can then start typing a new value and it will overwrite the highlighted part of the number.
- Triple-click inside the Spin Box to highlight the entire number. As before, you can type in a new value. You can also copy and paste the value by using the *Ctrl* + *C* and *Ctrl* + *V* hotkeys.
- Calculate by using math operators in the Spin Box. That's right! You can add, subtract, multiply, and divide right in the Spin Box and Inkscape will change your math into a number once you hit the *Enter* key. No calculator is required! Some examples can be seen in *Figure 1.8*:



Math Operation	Entry in Spin Box	Result after pressing Enter
Add (+)	24.5+10	34.5
Subtract (-)	24.5-10	14.5
Multiply (*)	24.5*10	245
Divide (/)	24.5/10	2.45

Figure 1.8 – Spin Box math operations table

- Convert from one measurement into another by adding the units to the number. For example, if you are working in centimeters (cm) and are making something that needs to be 24 inches, you can convert it into cm just by typing `24in` into the Spin Box and tapping the *Enter* key when you're done:

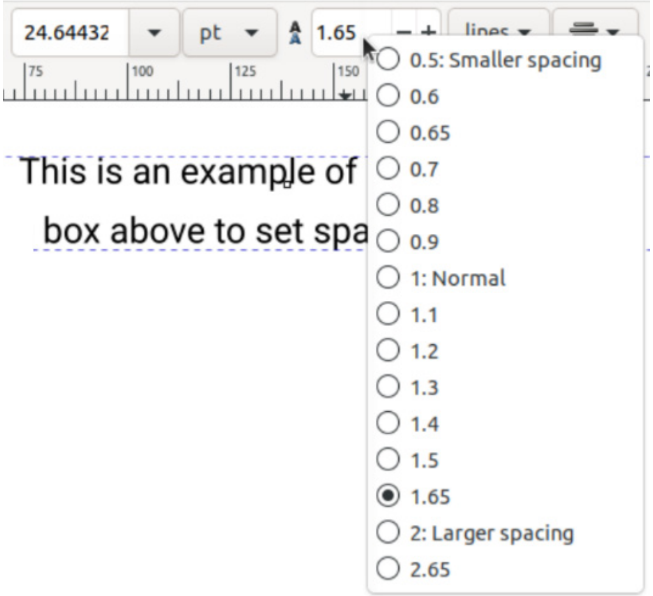


W: 24in H: 6.318 cm

W: 60.960 H: 6.318 cm

Figure 1.9 – Spin Box converting inches into centimeters

- Select a preset by right-clicking the Spin Box and choosing a value from the menu that pops up:



24.64432 pt 1.65 lines

This is an example of

box above to set spa

Figure 1.10 – Right-clicking a Spin Box pops up useful default options

So, with all these little tips and tricks for getting around, are we ready for the grand tour? Great! Let's start with an old familiar friend, the Menu bar.

The Menu bar

You're probably familiar with this bar as it's a staple of most desktop applications. Whenever you want to find something, it's usually organized into menus at the top of the screen (see *Figure 1.11*):

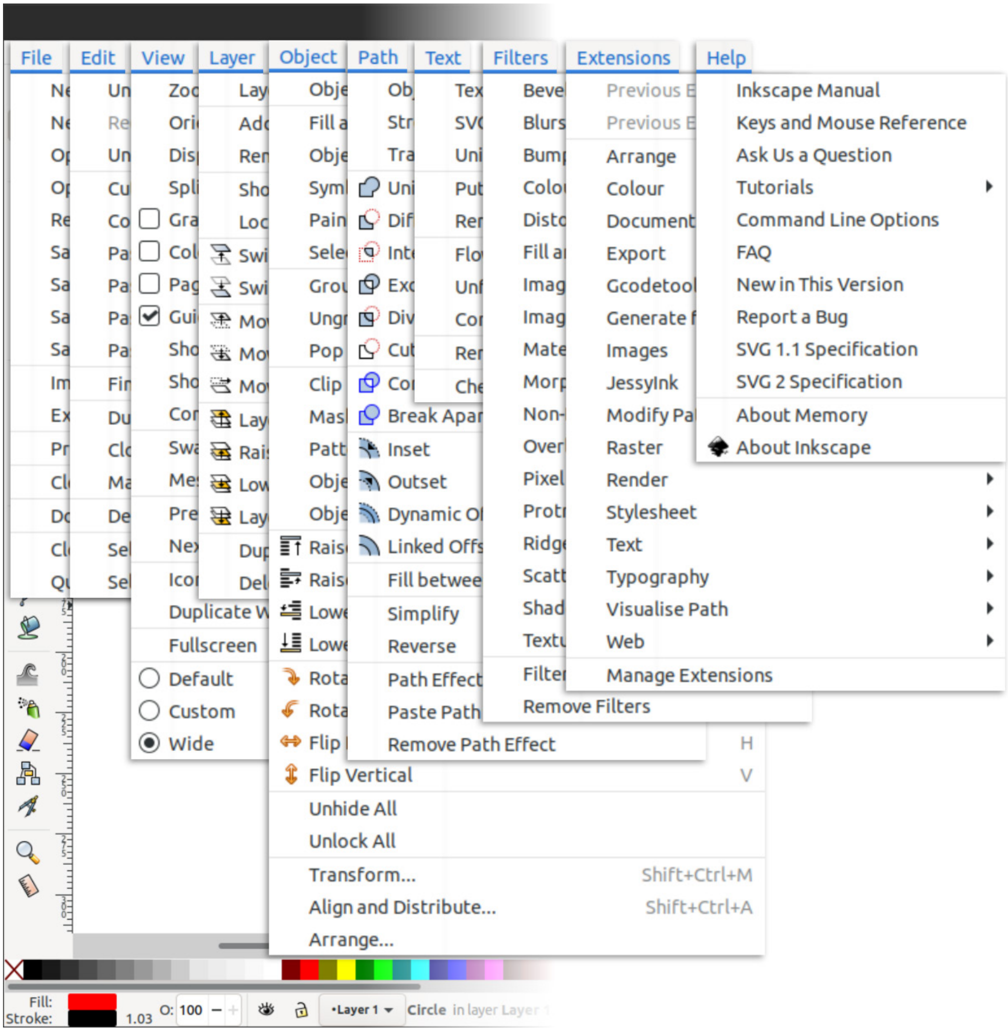


Figure 1.11 – Inkscape's Menu items

While this certainly seems like a lot of information, you will see that everything is sorted logically, and once you get used to where things are, it will become second nature. For now, here are some examples of the kinds of things you can find in each menu:

- **File**: Contains entries related to loading and saving files, printing, cleaning up, and changing the properties of your document.
- **Edit**: This menu contains copy, paste, and select actions, as well as the **Inkscape Preferences** dialog.
- **View**: This menu features zoom controls, display modes, and a variety of customization options to show and hide pieces of the Inkscape user interface.

- **Layer**: This menu houses layer controls and shortcuts. This menu also brings up the **Layers** dialog. You will likely use the **Layers** dialog more often than the items in this menu, but they are still a good reference for the hotkeys to raise and lower layers, and so on.
- **Objects**: This menu includes a myriad of useful tools for interacting with objects, including raising and lowering them in the object stack (more on that later). This menu also brings up the **Objects** dialog, the **Align and Distribute** dialog, and the **Arrange** dialog, all of which we will cover in later chapters.
- **Path**: This is a veritable toolbox of functions for editing the nodes and curves (lines/strokes) of your shapes, as well as for converting from an object into a path to make shapes editable. There are also useful path operations to add, remove, and cut shapes using other shapes. We'll spend a good deal of time in this menu as it contains fast ways to get the shapes we want to make without the need to edit them node by node.
- **Text**: This menu can be used to bring up the **Text and Font** dialog, attach text to paths, flow text through objects so your paragraphs take on the shape of non-rectangle objects, and, of course, the all-important **Check Spelling** dialog.
- **Filters**: In addition to constructing shapes, Inkscape makes it possible to add effects to them. It's best to use the effects in this menu sparingly for a variety of reasons. However, there are some neat effects here and the ability to build your own, so we'll have some fun with that later.
- **Extensions**: Inkscape supports extensions written in the programming language Python. Writing these is beyond the scope of this book, but I'll take you on a guided tour of the fabulous extensions that come with Inkscape so you can do awesome things such as generate barcodes and QR codes, change the colors of your shapes and objects as you would do in photo-editing software, and so on.
- **Help**: This menu contains a great collection of quick links for learning Inkscape basics, getting help, and even chatting with Inkscape community members in real time on the Inkscape chat server. The manual is a bit outdated at the time of writing this book but contains a lot of useful information nonetheless.

Note

In this book and online forums, it's common to use > to indicate the way to get to a menu item. For example, instead of saying "go to the top Menu bar and click on the **Edit** menu, then click on the **Paste Size** option, then click on the **Paste Width** option," we instead say **Edit** > **Paste Size** > **Paste Width**.

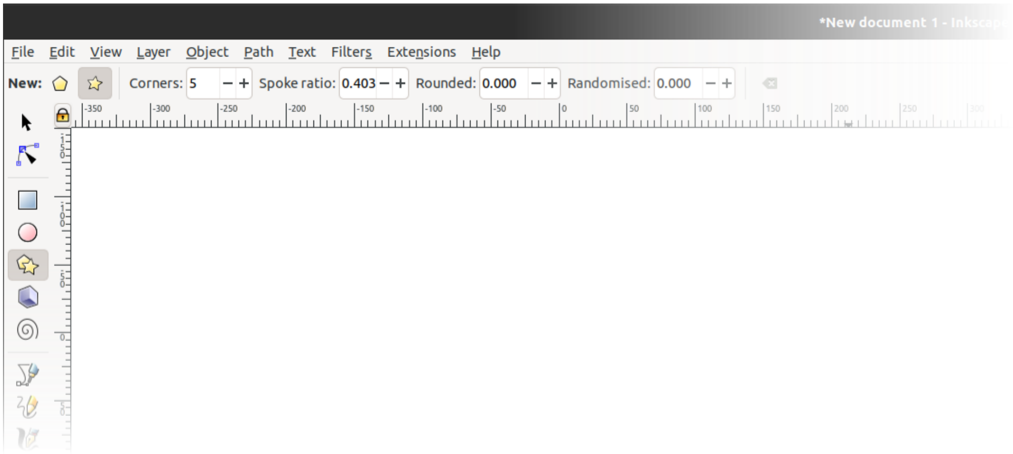
Much easier to read, isn't it? It's also easier to type and remember, and generally saves everyone involved a ton of time.

That about wraps it up for the Menu bar. Next, we'll have a look at the Tool bar and Tool control bar.

The Tool bar and Tool control bar

These two go hand in hand. The tools are on the left, and the controls for the tools are at the top. When you choose a tool from the bar on the left, the Tool control bar will display all the options for that tool. For example, if you choose the **Star and Polygon** tool, the Tool control bar area displays the number of points, whether it's a star or a polygon, and even the rounding of the points! Changing to the **Text** tool shows all the various fonts, letters, line and character spacing, and so on.

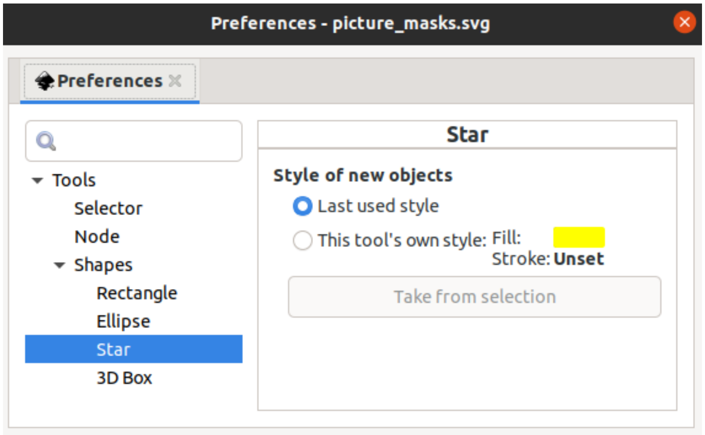
When the **Star and Polygon** tool is selected, the Tool control bar shows options for the tool, such as how many corners the star has, as shown in the following screenshot:



Corners: 5 Spoke ratio: 0.403 Rounded: 0.000 Randomised: 0.000

Figure 1.12 – The Star and Polygon tool

Double-clicking on a tool in the Tool bar will bring up options for that tool (see *Figure 1.13*), including **Last used style**, which you'll probably want to turn on so that Inkscape remembers the fill color and stroke thickness of the last shape you drew, the last font you used for text, and so on. Double-clicking a tool brings up the **Preferences** dialog for that tool:



Star

Style of new objects

● Last used style

○ This tool's own style: Fill: 🟨
Stroke: **Unset**

Figure 1.13 – Star tool Preferences dialog

We'll be exploring the various options for each of these tools in detail in the next chapters, but for now, let's move on to the main event – the **Desk and Page areas**.

The Desk and Page areas

This area, shown in *Figure 1.14*, is where you'll be drawing with the various tools in the Tool bar. The Page area, by default, is an A4-sized page but can be changed by selecting **File** > **Document Properties** or pressing *Ctrl* + *Shift* + *D*:

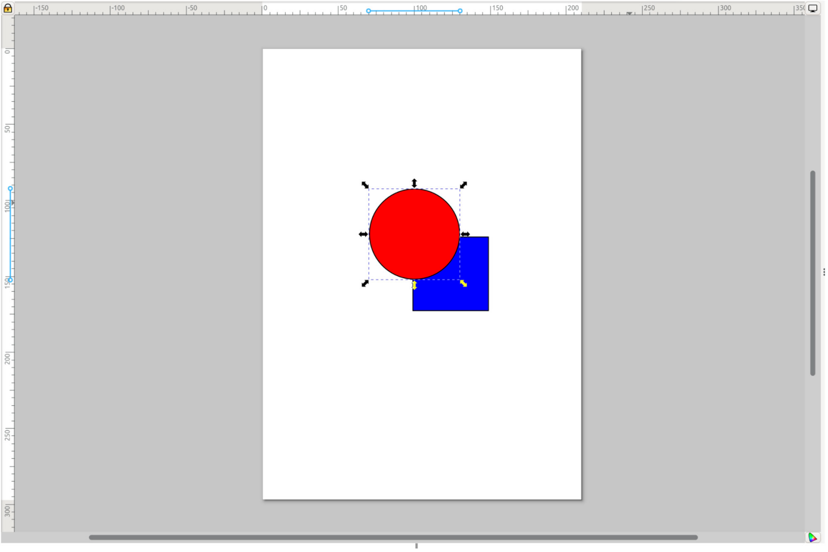


Figure 1.14 – The Desk area with a few shapes drawn on the Page area

Although you can draw anywhere in the **Desk** area, the **Page** area is a good place to keep your main graphics. It's what will appear in the thumbnail preview of your file browser and web browser, or what someone will see when they open that PDF you saved. It's also the region that is exported when choosing the **Page** option in the **PNG Export** dialog (more on that later).

Here are some tips for navigating the **Desk** and **Page** areas:

- **Zoom in and out**: Hold the *Ctrl* key and use the mouse wheel to zoom in and out. If your computer supports a touch screen, you can also *pinch to zoom*, or use the + and – keys as well. There is also a **Zoom** tool in the Tool bar that looks like a magnifying glass, but with all these great ways to zoom without switching tools, I recommend the other methods.
- **Panning the view**: Moving the view from side to side or up and down is called **panning**. Sure, you can drag the tiny scrollbars at the bottom and left-hand side of the canvas, but you could also just hold the spacebar down, which will grab the canvas and move it as you move the mouse.

  Holding the middle mouse button also works if you have one, but once you get used to the spacebar method, it's going to be your go-to method of panning. You will wish your other applications did the same after a while.
- **The context menu**: What if you didn't have to move your mouse off the canvas to access some of the Menu bar items? Even better, what if you could just right-click on the canvas, or an object on the canvas and instantly be offered a subset of actions just for that kind of object? Well, that's exactly what the context menu is for! Simply right-click on the **Desk**, **Page**, or any selected object and you will be presented with a choice assortment of useful things, as you can see in *Figure 1.15*:

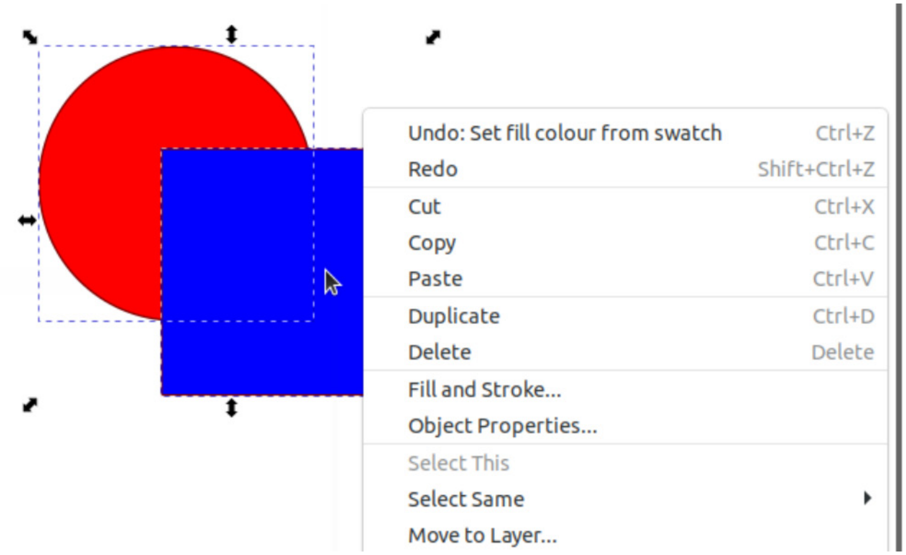


Figure 1.15 – Right-clicking this selection brings up the context menu for what is selected

So, now that you know how to get around the **Desk** area, let's have a look at some ways you can color your shapes with the Swatch Palette.

The Swatch Palette

As shown in *Figure 1.16*, the Swatch Palette is good for quick coloring. Clicking one of these will fill any selected shape with that color:

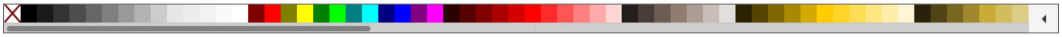


Figure 1.16 – The Swatch Palette

Likewise, holding *Shift* and clicking a swatch will assign a stroke of that color, or you can right-click a swatch and choose **Set stroke** from the popover menu. Clicking the hamburger menu (the one with three lines) on the far right will give you option to see more colors and switch between palettes. Also, the up and down arrow buttons to the left of the hamburger menu will scroll up and down the available swatches in the currently selected palette, as will hovering the mouse cursor over the swatches and using the mouse wheel.

In Inkscape 1.3, you may notice that the first swatches on the left are larger than the rest. These are pinned colors, and you can add and remove them from this area to save your favorites. Say, for example, you really like the brightest red, and you'd like it to be pinned. Simply right-click on the red swatch and choose **Pin color** from the menu that pops up, as shown in *Figure 1.17*.

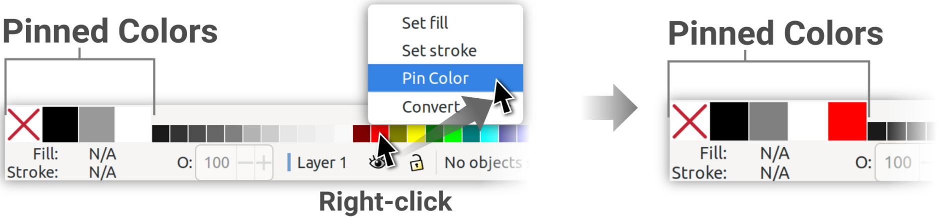


Figure 1.17 – Pinning colors to the Pinned Colors area

The color indicators

The color indicators indicate the **Fill and Stroke** (outline) colors for the selected shape. These indicators will be blank (and show **N/A**, which is short for **not applicable**) until a shape is selected. Clicking the color bar will bring up the **Fill and Stroke** dialog for more options, as well as a variety of color pickers to choose custom colors.

Right-clicking the color bars will bring up a menu of nice presets, as well as the ability to copy, paste, and swap colors between fills and strokes, as shown in *Figure 1.18*:

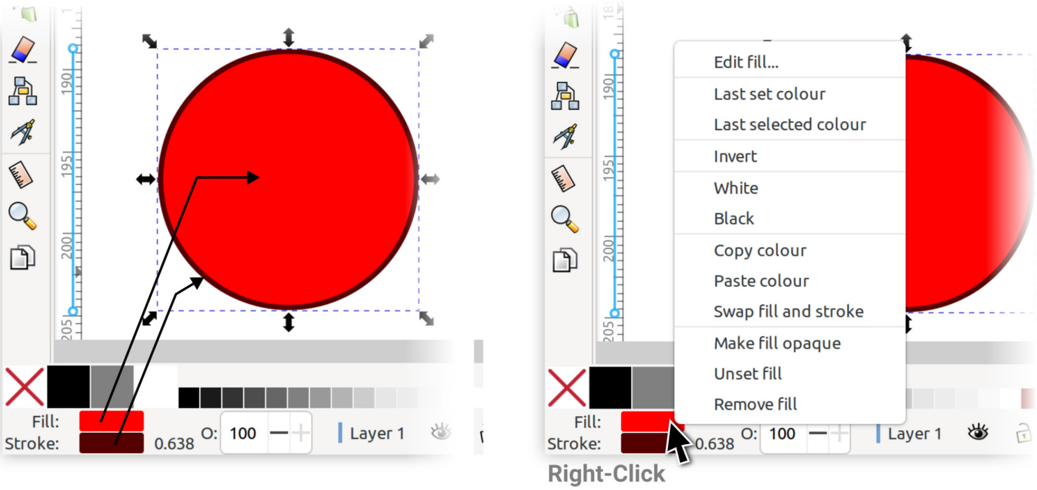


Figure 1.18 – Fill and stroke indicators (left) and the options you get after right-clicking the Fill or Stroke color bars (right)

Next to the **Stroke** color bar is a number representing the stroke's width (`0.638` in this case). Clicking that number will open the **Stroke Style** tab in the **Fill and Stroke** dialog, where you can change its value.

The **O:** value to the right of the stroke's width in the box is the **opacity** value, which controls the transparency/opacity of the selected object. This value corresponds to the opacity slider at the bottom of the **Fill and Stroke** dialog. You can use either one to change this opacity value.

Those are the basics of the color indicators, but you'll notice there are more options directly to the right. Let's see what the Layer Info Indicator is all about, shall we?

The Layer Info Indicator

These are just a few quick-access tools (see *Figure 1.19*) for working with the currently selected layer or group. These controls are also available in the **Layers** dialog and **Objects** dialog, which we will discuss later:

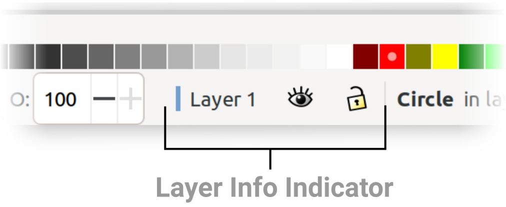


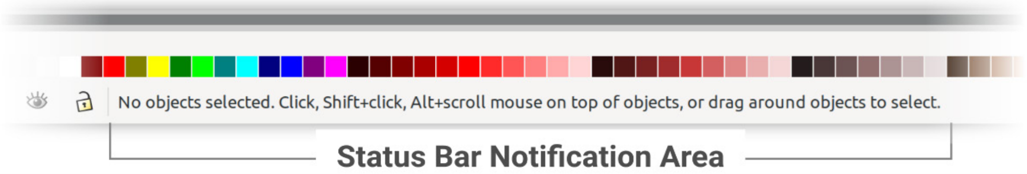
Figure 1.19 – Layer Info Indicator

Clicking the **Eyeball Toggle** icon will hide or show the current layer or group, and the **Lock Toggle** icon will lock or unlock it. *Locking* a layer means you cannot select it, which is nice for working on top of background shapes that you don't want to accidentally select. You can also click the layer name, and the **Layers and Objects** dialog will appear with the current layer and object selected.

Further to the right is a space filled with a message – this is the **Status bar**.

The Status bar

A much-overlooked area of the interface, this readout shows useful information about the current state of Inkscape (see *Figure 1.20*):



No objects selected. Click, Shift+click, Alt+scroll mouse on top of objects, or drag around objects to select.

Figure 1.20 – The Status bar – an invaluable source of information

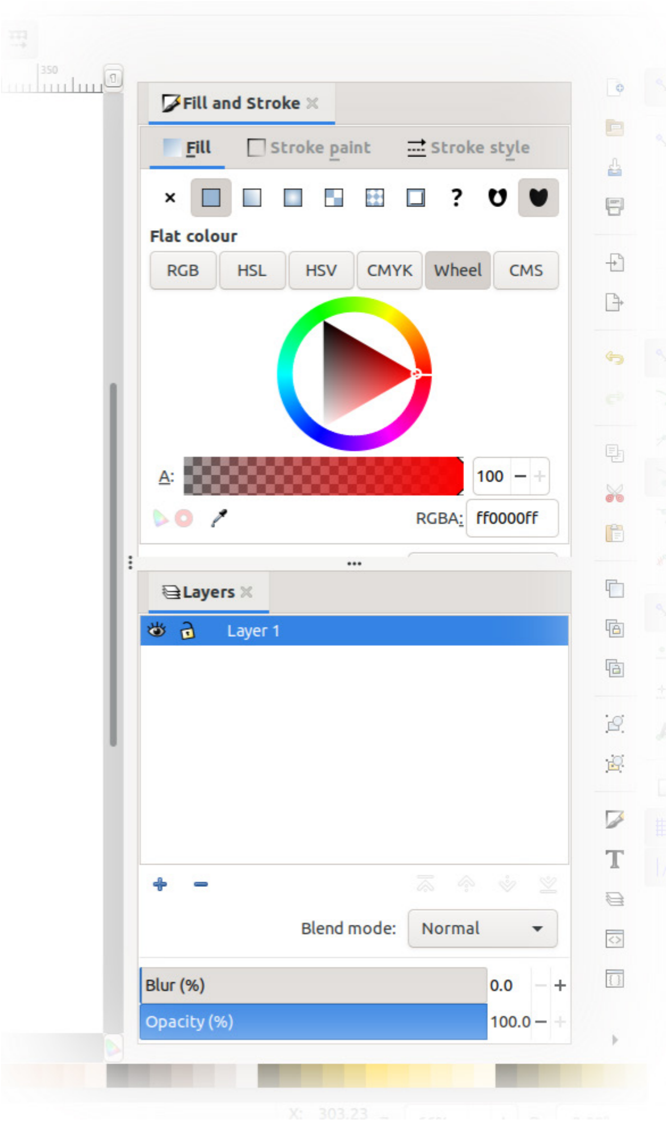
This area shows different things, depending on what's selected:

- *When no shapes are selected*, the Status bar contains useful tips about the active tool.
- *When an object is selected*, the Status bar shows what the object type is and information about the object.
- *When a path is selected*, the Status bar shows how many nodes are in the current path.
- *When a group is selected* the Status bar shows how many objects and paths the group contains, and whether it contains other groups.
- *When an image is selected*, the Status bar shows the dimensions of the image in pixels, and whether the image is linked or embedded in the document.
- *When something goes wrong* – for example, you've tried to subtract one shape from another using path operations, and one of the objects isn't a path. Inkscape will show an error message here reminding you that one of your shapes is an object and will not work for this operation until you convert it (more on this later).

Now, let's take a look at the Docking area and dialogs.

The Docking area and dialogs

This area, featured in *Figure 1.21*, will show all dialogs you call up, and provides a customizable space where you can keep one or more of them handy while you work:



A: 100

RGBA: ff0000ff

Blend mode: Normal

Blur (%) 0.0

Opacity (%) 100.0

Figure 1.21 – The Docking area, shown here with the Fill and Stroke and Layers and Objects dialogs

When opening Inkscape for the first time, you may not see any dialog boxes. There are various ways of summoning these dialogs, such as clicking the **Fill and Stroke** color bars (as mentioned previously), or by using equivalent options in the Menu bar (**Object** > **Fill and Stroke**, for example).

You can click and drag the tabs at the top of each dialog to rearrange them in the Docking area, and as of Inkscape 1.1, you can even dock them to the other side of the screen next to your Tool bar if you wish.

Notice the three dots on the left-hand side of the dialogs. This is a handle, which you may use to resize the dialogs. Handles are also present between each dialog for the same purpose.

Tip

At the time of writing, developers are working hard to shrink the area needed for some of the wider dialogs. Meanwhile, you can also drag out a dialog into a floating window and then close it. Inkscape will remember your preference and will float the dialog when you call it up again so that it does not constantly use up space in your dock. The *F12* key also toggles the Docking area *on* and *off*, so if it's encroaching on your drawing space, you can banish it temporarily and pull it back out when you need it.

Moving further to the right, there is another bar of icon buttons, though you will probably not need these as often. Let's have a look at the Commands bar.

The Commands bar

This bar (see *Figure 1.22*) contains some of the commonly used functions in various menus in the Menu bar, such as save/load, undo/redo, and even calls up some of the dialogs. I never use these, but they are a nice touch for those who aren't used to selecting things out of the Menu bar or using hotkeys (which is my preferred method). Recall from earlier in the chapter that this bar may be at the top by default, depending on your screen dimensions.

The Command bar is the button-clicky way to do things you should be doing with hotkeys:

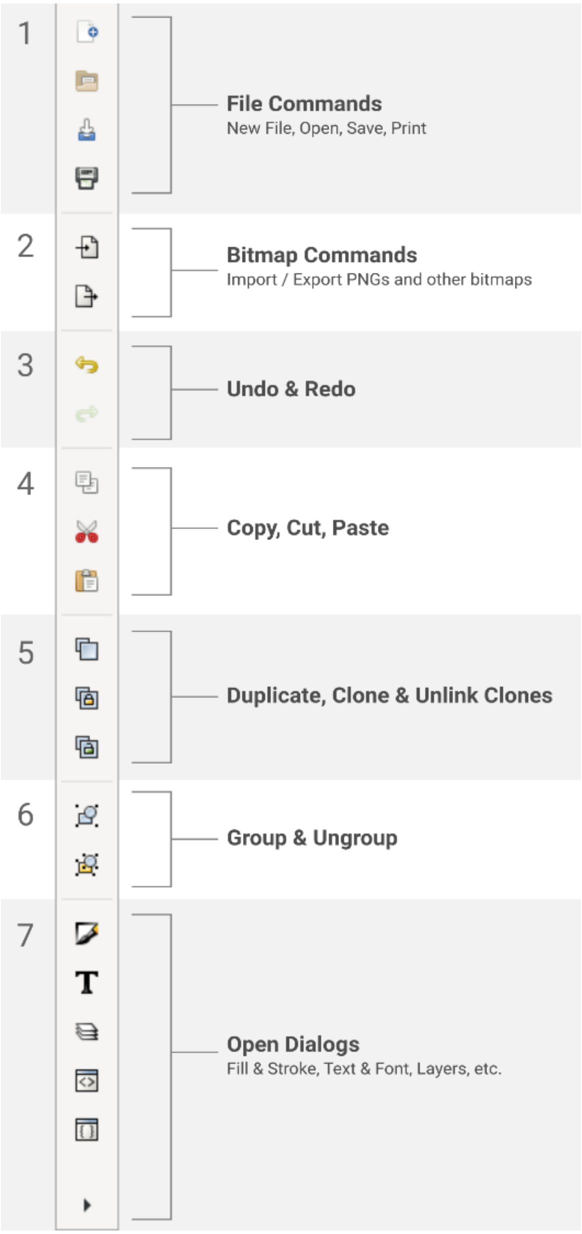

1
File Commands
New File, Open, Save, Print
2
Bitmap Commands
Import / Export PNGs and other bitmaps
3
Undo & Redo
4
Copy, Cut, Paste
5
Duplicate, Clone & Unlink Clones
6
Group & Ungroup
7
Open Dialogs
Fill & Stroke, Text & Font, Layers, etc.

Figure 1.22 – The Commands bar

The arrow at the bottom can be clicked to show more dialogs that do not fit on the screen. Depending on your screen resolution, you may or may not see this arrow.

Well, we're here at the final bar of buttons and about as far right as we can go without falling off the screen. These are toggles, as mentioned previously, and comprise the Snap control bar.

The Snap control bar

When you move your objects around the page or draw new shapes, the cursor can automatically snap to various parts of the drawing area or other shapes that you have already drawn. This can be handy for alignment, as we will see in later chapters.

Figure 1.23 shows typical snapping behavior – while you are dragging an object to a new location, Inkscape takes the closest point (or line) on your shape to your cursor and suggests places to align or snap it to a nearby shape. You can then simply release the mouse button and the shape will be perfectly aligned:

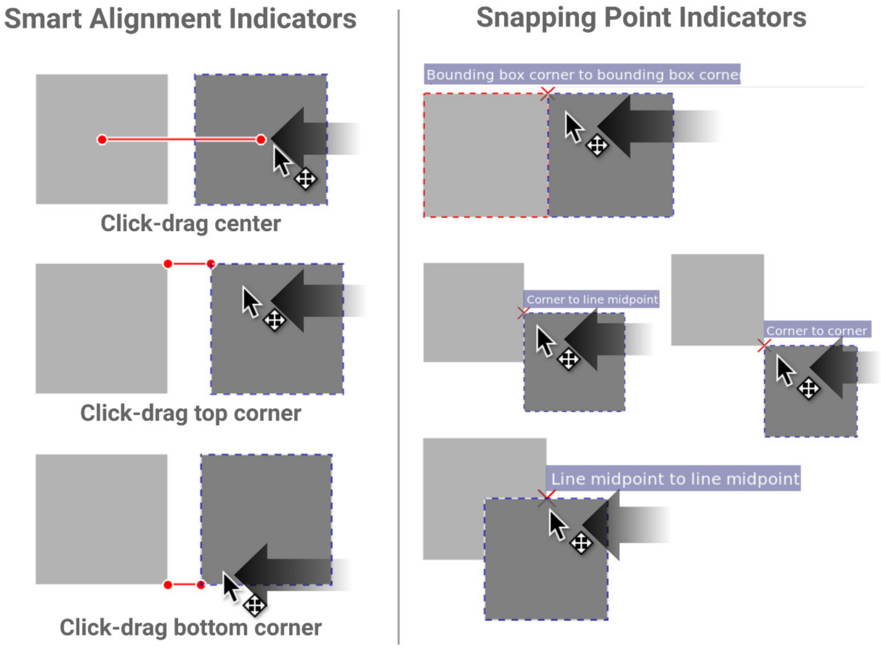
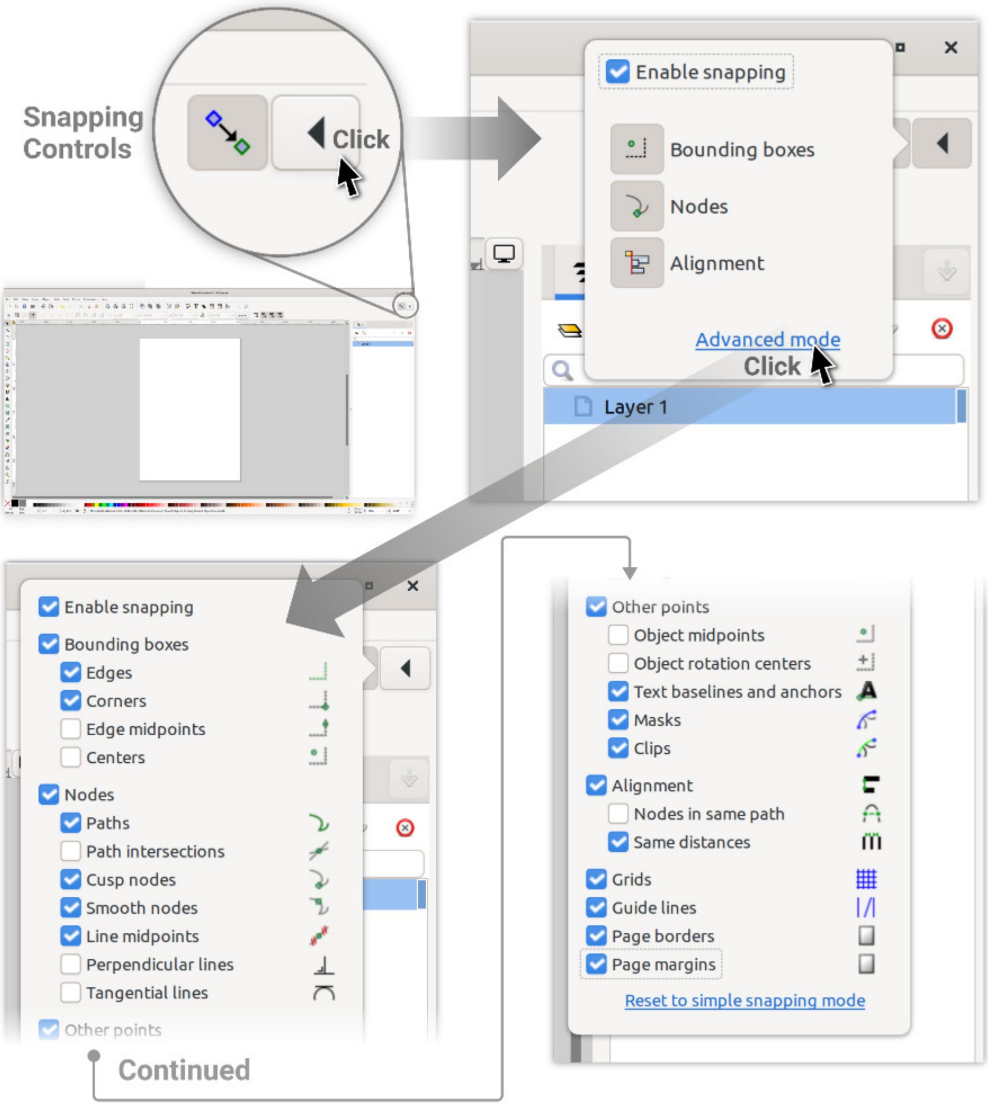

Smart Alignment Indicators
Click-drag center
Click-drag top corner
Click-drag bottom corner
Snapping Point Indicators
Bounding box corner to bounding box corne
Corner to line midpoint
Corner to corner
Line midpoint to line midpoint

Figure 1.23 – When you move one object close to another, a snapping point appears

A snapping indicator will appear when an object snaps to another so that you can see what Inkscape is snapping to before you release the mouse button. You can control what Inkscape uses to snap using the **Snapping Controls** toggle.

Clicking the **Snapping Controls** toggle enables and disables snapping, while the arrow beside it pops up the **Simple Snapping** options by default. Clicking the **Advanced Options** link at the bottom shows a full list of all snapping options in Inkscape, as shown in *Figure 1.24*:



☑ Enable snapping
Bounding boxes
Nodes
Alignment
Advanced mode

☑ Enable snapping
☑ Bounding boxes
☑ Edges
☑ Corners
☐ Edge midpoints
☐ Centers
☑ Nodes
☑ Paths
☐ Path intersections
☑ Cusp nodes
☑ Smooth nodes
☑ Line midpoints
☐ Perpendicular lines
☐ Tangential lines
☑ Other points

☑ Other points
☐ Object midpoints
☐ Object rotation centers
☑ Text baselines and anchors
☑ Masks
☑ Clips
☑ Alignment
☐ Nodes in same path
☑ Same distances
☑ Grids
☑ Guide lines
☑ Page borders
☑ Page margins
Reset to simple snapping mode

Figure 1.24 – Expanding the Snapping controls to Advanced mode for all snapping options

The following summarizes what these checkboxes do:

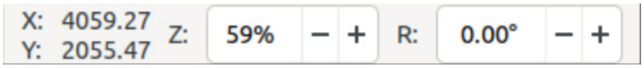
- **Enable Snapping**: This button will toggle all snapping *on* and *off* so that you can activate or deactivate snapping easily. You can also use the % key to toggle this *on* and *off*.
- **Bounding Boxes**: When you select an object, a dashed rectangle appears around it. This is called the **bounding box**, and this checkbox turns snapping on/off to this invisible boundary. The indented checkboxes in this section allow you to control which parts of the bounding box Inkscape snaps to.
- **Nodes**: Nodes are the points between each line of your shape. If you've converted your object into a path, this section allows you to snap one object to another by one of its nodes. We will cover more about nodes and paths in *Chapter 5*, *Node Editing – Modifying Your Shapes with Nodes and Curves*. Like the **Bounding Box** section, the indented checkboxes let you tell Inkscape what kinds of nodes you want to snap to.
- **Other Points**: This is a catch-all for those snapping options that don't fit nicely into one of the other groups – for example, snapping to the baseline of a line of text and anchors, or an object's rotational center or midpoint. We will learn more about them later when we talk about rotating and transforming shapes.
- **Snap to Grid**: By default, the grid doesn't show up in Inkscape, so turning this option on will snap objects that you are moving to the invisible grid. If you want to see the grid, simply tap the # key (which looks a little like a grid), or toggle it by selecting **View** > **Page Grid**.
- **Guide Lines**: You can create snapping lines by clicking on a ruler and dragging it outward onto the **Desk** or **Page**. Click and drag from the top ruler to make horizontal lines, and from the top-left ruler to make vertical lines. You can also click and drag from the top-left corner, where the rulers converge to drag out a 45-degree snapping line.

  Then, all you need to do is use this snapping toggle to snap your objects to your newly created guides. Use the > key to toggle these lines *on* and *off* or select **View** > **Guides**. You can also remove a guide by dragging them back to the ruler, or simply hovering your cursor over one and hitting the *Delete* key.
- **Page Borders**: Yep, you can even snap your shapes to the border of the page using this toggle.
- **Page Margins**: If you've set page margins with the **Pages Tool** property, you can snap objects to them.

In the final section of this chapter, we'll look at **Display Transform Control**.

The Display Transform Control

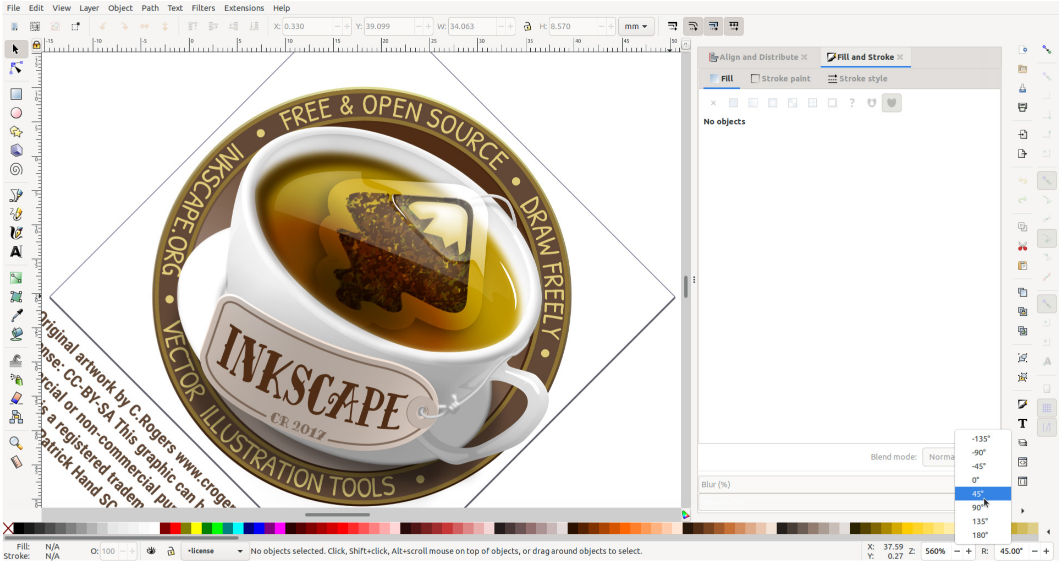
This area, shown in *Figure 1.25*, is located at the bottom-right corner of the Inkscape window and contains a manual control for the **Desk** view. It includes a readout of the X and Y cursor position when it's over the **Desk** (measured from the top-left corner of the page), a zoom control, and a view rotation control:



X: 4059.27
Y: 2055.47
Z: 59%
R: 0.00°

Figure 1.25 – Display Transform Control

Right-clicking the zoom or rotation controls gives some nice preset options, as shown in *Figure 1.26*, some of which are also in the **View** menu:



No objects

Blend mode: Norma

Blur (%)

No objects selected. Click, Shift+click, Alt+scroll mouse on top of objects, or drag around objects to select.

X: 37.59
Y: 0.27
Z: 560%
R: 45.00°

Figure 1.26 – Right-clicking the R Spin Box gives a nice set of preset rotations

For the most part, I recommend using hotkeys and mouse buttons for the navigation described in the The *Desk and Page area* section earlier in this chapter, but if you're lost in some blank area of the **Desk**, or the display is rotated at a strange angle and you don't know how you got there, this bar allows you to get back where you were quickly.

Summary

In this chapter, we looked at the parts of the Inkscape user interface, how to navigate the **Desk** area, and identify Spin Boxes, menus, buttons, and toggles. We also went over some neat tricks to use Inkscape as a calculator. Then, we covered some hidden features, such as the right-click default options in Spin Boxes, the sneaky but useful context menu, and a myriad of ways to zoom without using the **Zoom** tool.

Knowing the contents of this chapter will serve you well in your creative journey with Inkscape, saving you time and building confidence using Inkscape as a tool for creating your graphics. Even better, you can help other users find things they never knew existed to help save them time as well. Sharing is caring in the Inkscape community after all.

Now that we have our footing, we will take the next steps and start making some great stuff in Inkscape. Let's get into it with *Moving and Styling Shapes* in the next chapter!

2

Moving and Styling Shapes

So, now we can get into drawing, right? Well, almost. To prepare us for our shape drawing activities, let's go over some shape basics.

In this chapter, we will learn about shapes in Inkscape, plus how to style them with **Fill and Stroke**. We'll then cover how to use the **Transform** tool to move, rotate, and resize our shapes, and finish this chapter by aligning and distributing our shapes in relation to each other to save us a ton of work.

After all, we want to spend more time drawing, and less time moving things around manually. Fortunately, Inkscape has us covered, as we will soon see.

In this chapter, we will cover the following topics:

- Introducing shapes in Inkscape
- Using the **Select and Transform** tool
- Moving shapes
- Cutting, copying, pasting, and deleting shapes
- Resizing shapes
- Rotating shapes
- Raising and lowering shapes
- Grouping shapes
- Flipping shapes

Technical requirements

As mentioned in the previous chapter (which you should go back and read if you skipped it), this chapter assumes you've taken my advice and gotten yourself a good wired or wireless mouse to use with Inkscape.

We will also be relying on hotkeys for moving/panning around the canvas, and many more hotkeys will be mentioned along the way. It's a good idea to use them as you go through the practice lessons and try out other things in this chapter. The more you use them, the more you will remember them, and you will spend less time searching and more time drawing and constructing your vector masterpiece.

You can download the practice files for this chapter from GitHub at

`https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape/tree/main/Chapter02`

Introducing shapes in Inkscape

In this section, we'll introduce you to shape concepts. We'll begin with the basics of raster images (made of pixels) versus vector images (made of geometric shapes), drawing shapes, and then styling shapes with **Fill** and **Stroke**.

Raster versus vector images

Have a look at the two shapes in *Figure 2.1*. Both are the Inkscape logo, but they are different. The image on the right is called a **raster image**. It's nothing more than a grid of pixels, each of a specific color that makes up the image. You can tell this because when you zoom in, you can see the jagged edges of the pixels:

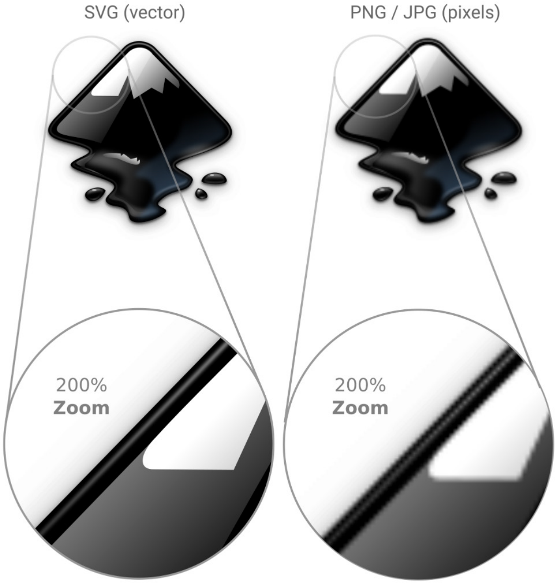


Figure 2.1 – Inkscape logo in vector and raster formats

In contrast, the image on the left is a **vector graphic** – that is, it's made up of points (also called **nodes**), and lines connecting them. Instead of storing and displaying pixels of this graphic, your computer can re-draw a vector graphic each time you zoom in so that the edge is always crisp, with no jagged pixel edges visible.

This is handy for a variety of reasons:

- **Vector images take up less space**: Since we are only storing points, curves, and fill information, we don't need to save every single pixel to recreate the drawing. This keeps the file size very small. For example, the Inkscape logo shown here is only 36.5 KB, which will load instantly in most browsers, even on a very poor connection. As we will see in later chapters, Inkscape saves in **Scalable Vector Graphic** (**SVG**) format, which is a web standard like `.jpeg` or `.png`; many websites utilize this to keep page load times down.
- **Vector images are infinitely scalable**: This means that no matter how much you zoom in or scale them up, the file size does not increase. If you had a printer that could print across the surface of the moon (see *Figure 2.2*), you could give it that same 35.5 KB file, and it could render the entire 2,750 square kilometers (about 1,708 square miles) of it with perfect crispness, down to a tiny fraction of a millimeter:

Figure 2.2 – The Inkscape logo, printed across the moon [Moon photo by Dom Le Roy - pexels.com/@domleroy]

So, now that we know how very useful vector shapes can be, let's talk about how to draw a shape and style it to our liking.

Drawing your first shape

Of course, before we can color and style our shapes, we will need to draw one. So, draw yourself a brand-spanking-new rectangle by selecting the **Rectangle** tool, and then by clicking some location on the canvas and holding the left mouse button down while dragging (seen *Figure 2.3*):

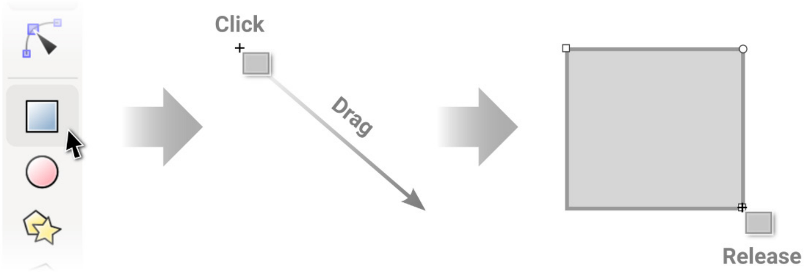


Figure 2.3 – Drawing a shape

Additionally, you can hold the *Ctrl* key while dragging to make a uniform shape, such as a square. Depending on your Inkscape settings, you may see that your new rectangle is styled differently than the one in *Figure 2.3*. That's okay, though. Let's see how to style it to our liking with **Fill and Stroke**.

Fill and Stroke

All vector shapes in Inkscape have a **Fill**, which is the color that the object is filled with, and a **Stroke**, which is the outline color and thickness of the object. In *Figure 2.4*, the **Fill and Stroke** properties are shown, and correspond to the values in the **Color Indicator** area:

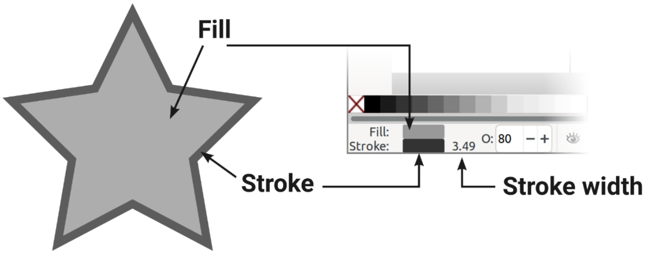
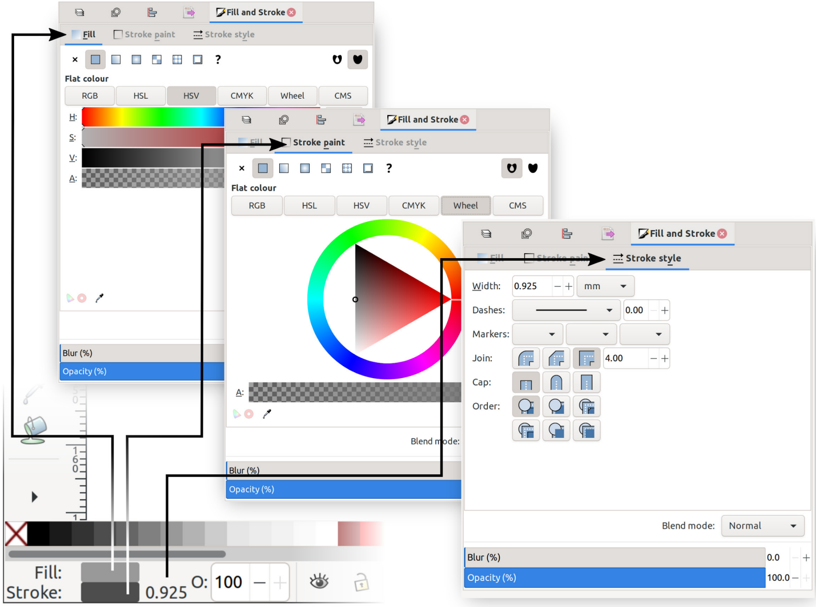


Figure 2.4 – A star shape with fill and stroke corresponding to color and stroke indicators

As mentioned in the previous chapter, you can set a fill color on the selected object by simply clicking a color in the Swatch Palette at the bottom of the screen. Holding the *Shift* key while clicking the swatch changes the **Stroke Color**.

Double-clicking the fill or stroke bar in the **Color Indicator** area will bring up the **Fill and Stroke** dialog with more options. Likewise, clicking the **Stroke** width value will bring up the **Stroke** style, allowing you to make the stroke thicker or thinner, as well as some other options. Here's a side-by-side view of all the **Fill** and **Stroke** options:



Width: 0.925 mm

Dashes: 0.00

Join: 4.00

Blend mode: Normal

Blur (%) 0.0

Opacity (%) 100.0

Fill:

Stroke: 0.925

O: 100

Figure 2.5 – Fill and Stroke dialog sections brought up by clicking on the color indicators

You may have noticed that I've used different color mixers in *Figure 2.5* for **Fill** and **Stroke paint**. It's worth noting that you can choose **Wheel** as your color mixer for either fill or stroke paint. I've included it here to showcase it as an option since the default is usually HSV or HSL color mixers with slider bars.

Help! I can't change my stroke width...

If you are using an older version of Inkscape (before 1.1), you may encounter a problem where the stroke width field is grayed out or reverting to 0 if you try and change it. If this occurs, *simply set a color for your stroke first* by either holding *Shift* and clicking a swatch, or right-clicking a swatch and choosing **Set stroke**. Old Inkscape will be like *What stroke, bruh?* if there's no stroke set. Set one and you'll find the stroke is now behaving correctly.

So, we've got an idea of some shape basics, and how to style them. But after you've drawn and styled your rectangle to perfection, you're going to want to move, rotate, and transform them in various ways. Let's get into how to sling our shapes around like pro Inkscape users!

Using the Select and Transform tool

While making your fantastic art with the shape tools, no doubt you'll want to do things such as select, move, rotate, and delete them. We've already drawn one rectangle, so let's repeat this a few more times and draw three rectangles side by side, similar to what's shown in *Figure 2.6*:

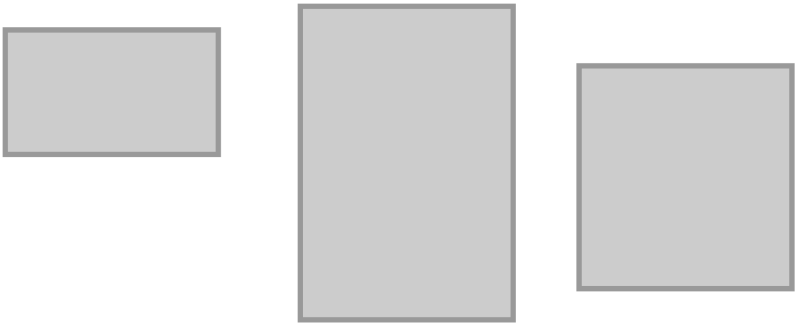
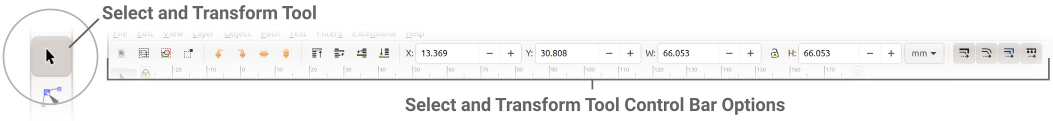


Figure 2.6 – Three rectangles. They don't have to be exactly like this example; any three will do

You will notice that when you have made a rectangle, it's already selected by default and features a dashed line and handles around the object. This is so you can easily change the color, round the corners, and more. But before we get into the fancy stuff, let's go over basic selection using the **Select and Transform** tool.

Often referred to as the **Arrow** tool, the **Select and Transform** tool is the one that looks like a mouse cursor arrow pointer, in the top-most position on the toolbar. Click to activate it and note that the Tool control bar changes when the **Select and Transform** tool is active. This gives us access to a variety of useful functions:


Select and Transform Tool
File Edit View Layer Object Path Text Filters Extensions Help
mm
Select and Transform Tool Control Bar Options

X: 13.369 Y: 30.808 W: 66.053 H: 66.053

Figure 2.7 – The Select and Transform tool

This tool is useful for various common shape selection and transforming operations. We'll take a look at some of the specific features in the following sub-sections.

Selecting and deselecting individual shapes

With the **Select and Transform** tool selected, all we have to do to select a shape is click it, as shown in *Figure 2.8*:

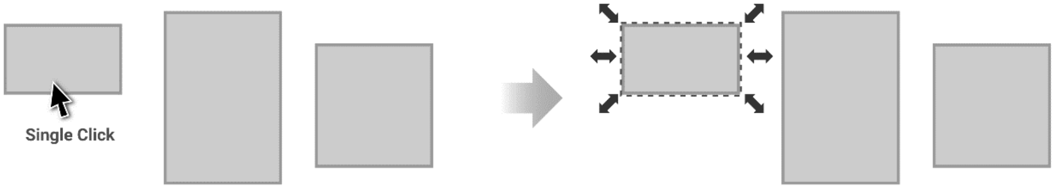


Figure 2.8 – Selecting a shape with a single click

You may have noticed that when you draw your rectangles (and indeed any shape), it's selected automatically. But maybe we don't want it selected. We can deselect it by simply clicking on the canvas or another shape, or by tapping the *Esc* key on the keyboard, as shown in *Figure 2.9*:

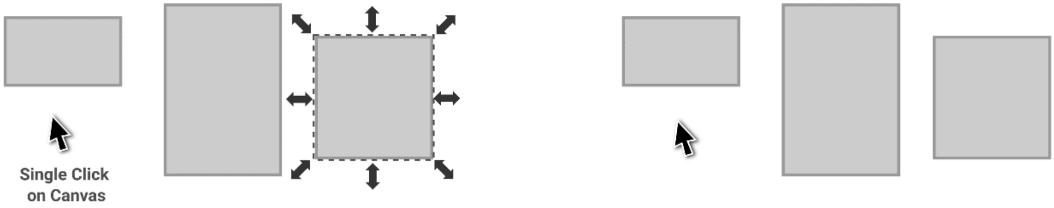


Figure 2.9 – Deselecting by clicking a blank area of the canvas

Simple, right? But wait, there's more!

To select more than one of your rectangles, simply hold down the *Shift* key on your keyboard and click each of the other two rectangles. This will add them to the selection, as shown in *Figure 2.10*:

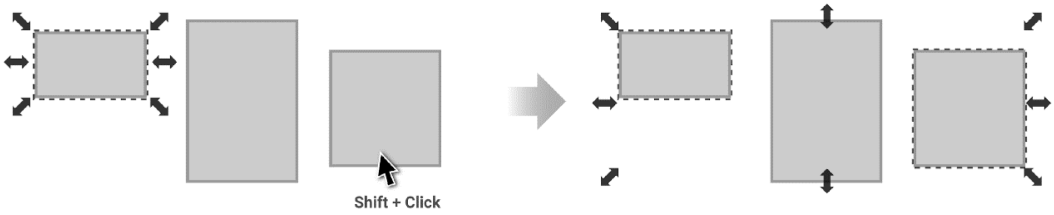


Figure 2.10 – Adding to your selection by holding Shift and clicking

To remove a selected object from the selection, just hold the *Shift* key once more and click on a selected shape to remove it from the selection. Note that you can also deselect a shape you may have accidentally selected by simply *Shift*-clicking it again. This is a rather nice feature as it means you don't have to start your selection all over again to omit it.

Selecting multiple shapes all at once

The task of shift-clicking to add each shape to a selection is all well and good for small quantities of shapes. But if there are many shapes, this method quickly becomes tedious. That's where **Rubber-band select** comes to the rescue!

You can click and drag a selection box around objects to select them all at once by holding *Shift*; everything that is completely inside that box will be added to the selection. The selection box appears as a dashed line as you drag the mouse; this dashed line is called a **rubber band**, due to the nature of being able to stretch the box around things.

Note that this works without holding *Shift* if you click and drag starting on a blank section of the canvas. If there's no blank area of the canvas where you are working, clicking and dragging may accidentally grab an object, since that's also how you move objects. Holding *Shift* in this instance tells Inkscape you're adding to a selection instead of moving an object. It's generally better to hold *Shift* every time you make a rubber-band selection to get in the habit. This way, you will never accidentally move something you didn't intend to.

This method is explained in *Figure 2.11*:

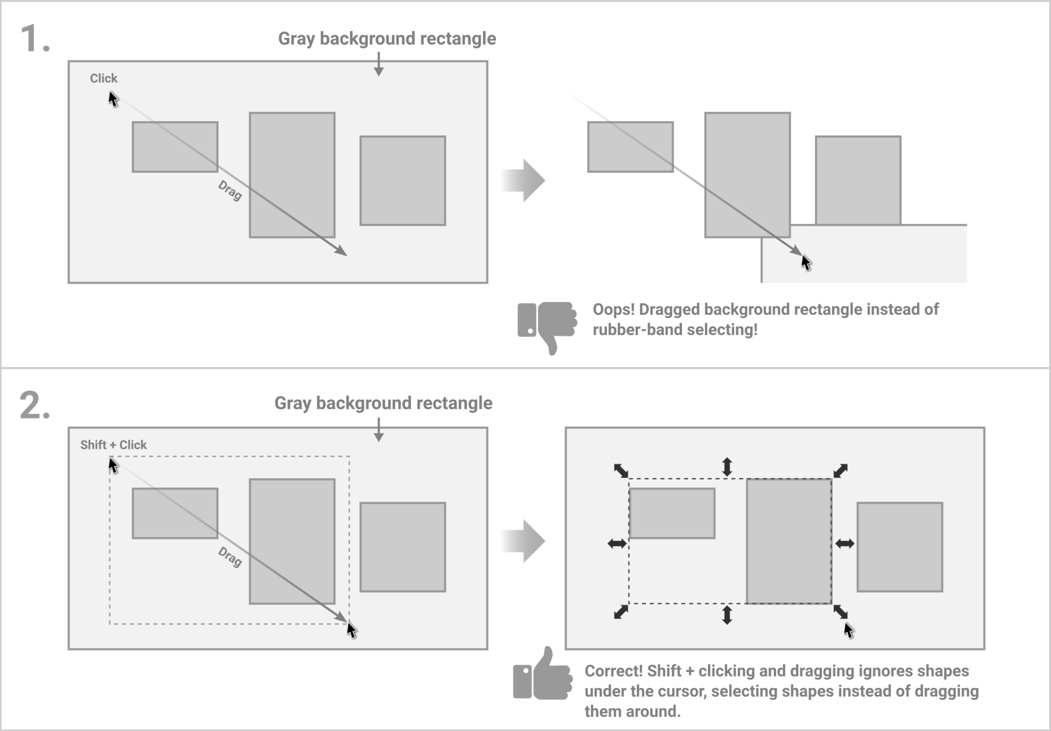

1.
Gray background rectangle
Click
Drag
Oops! Dragged background rectangle instead of rubber-band selecting!
2.
Gray background rectangle
Shift + Click
Drag
Correct! Shift + clicking and dragging ignores shapes under the cursor, selecting shapes instead of dragging them around.

Figure 2.11 – Rubber-band selection while holding the Shift key

Make sure you hold *Shift* while dragging a selection box if you start over an object. Yes, you'll forget at first, but after a while, you'll do it automatically without having to think about it.

Selecting everything

Sometimes, you just want to select everything. Inkscape uses the standard *Ctrl* + *A* hotkey to select all shapes in the current context. You can also use **Edit** > **Select All** or the **Select all** button in the Tool control bar, as shown in *Figure 2.12*:

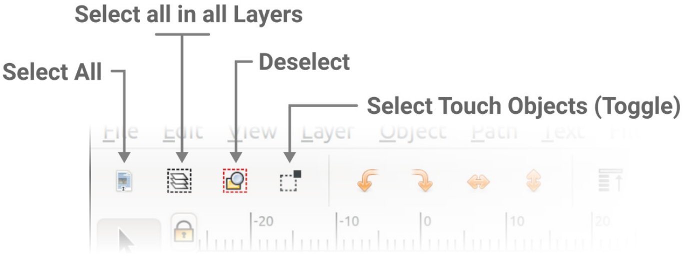

Select all in all Layers
Select All
Deselect
Select Touch Objects (Toggle)
File Edit View Layer Object Path Text
-20
-10
0
10
20

Figure 2.12 – The various selection buttons in the Tool control bar, with the Select tool activated

If you are *inside a group*, *Ctrl* + *A* will select everything inside the group. If you are *not inside a group*, *Ctrl* + *A* will select everything on the canvas. We will go over how to use groups to limit the selection in *Chapter 6*, *Fast Shape Editing with Path Operations and the Shape Builder Tool.*

For completeness, **Layers** will also limit the selection to the current layer. However, to select all objects in all layers, **Edit** > **Select All in All Layers** will select everything in your document that isn't locked (more on that when we discuss layers, also in *Chapter 6*, *Fast Shape Editing with Path Operations and the Shape Builder Tool*).

Fancy selecting with the Alt key

We've explored a few ways to select now, but sometimes, the situation calls for fancier selection options. Let's look at a couple of scenarios.

Selecting by using Touch-Path Select (holding Alt and dragging the mouse)

Sometimes, a rectangle rubber-band is just the wrong shape for what you want to select. Fortunately, if you hold the *Alt* key down, you can drag a red selection line along the shapes you wish to select. Every shape it touches will be added to the selection when you release the mouse cursor, as shown in *Figure 2.13*:

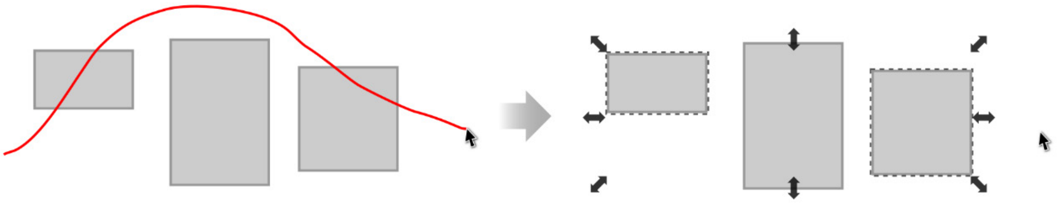


Figure 2.13 – Holding Alt while dragging

Keep in mind that this only works if there is *nothing* already selected. Holding the *Alt* key while dragging a selected object will move that object no matter where the cursor is on the page (see the *Moving shapes* section later in this chapter)! Simply hit the *Esc* key a few times to deselect if there's something selected; then, the **Touch-Path Select** option will work.

Selecting an object behind another object using X-Ray Select (holding Alt and rotating the mouse wheel)

Sometimes, you'll have objects stacked on top of each other, and you'd like to select a shape that is hidden under another object. To do this, hold the *Alt* key and click the top shape until the shape under it is selected.

Another way to do this is to hold the *Alt* key and scroll the mouse wheel, which will perform an **X-Ray Select** action that makes all objects under the cursor temporarily transparent, while the selected object is opaque. Turning the mouse wheel will select each object in turn. Simply release the *Alt* key to select the opaque object shown.

A demonstration of this effect is shown in *Figure 2.14*:

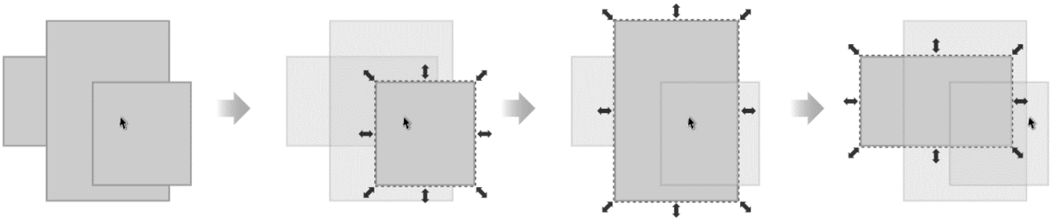


Figure 2.14 – Selecting using X-Ray Select mode

To deselect everything, you can either click on an empty space on the canvas or tap the *Esc* key.

Selecting objects with the same attributes

Say, for example, you have a big cluster of shapes, and you want to select only the rectangle objects. Or maybe you'd like to select all shapes that are the same color? What about all the objects with a black stroke? Inkscape has you covered with **Edit** > **Select Same**, as shown in *Figure 2.15*:

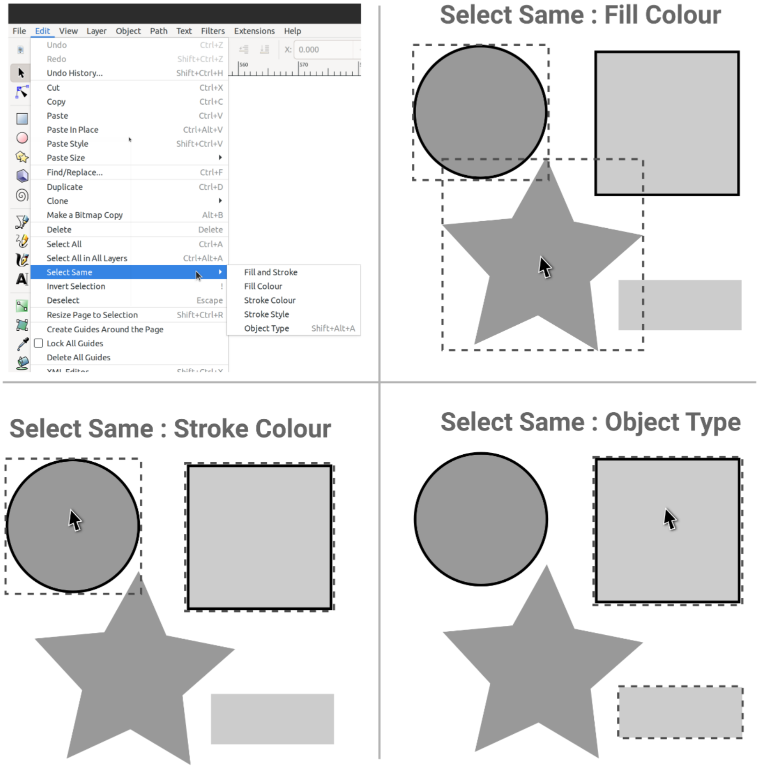


Figure 2.15 – The results of different Select Same operations

Note that the mouse cursor indicates which shape was selected before the operation.

Invert selection

Maybe you'd like to select everything except what's already selected. You can select what you don't want to be selected first and choose **Edit** > **Invert Selection** (you can also hold the *Shift* key and press the *!* key (*Shift* + *!*).

Now that we've had our fill of how to make selections, let's move on to how to move shapes around the canvas.

Moving shapes

Once you have selected your shapes, you can move them by simply click-dragging them with the mouse cursor. You can also hold the *Alt* key to force drag without having to click and hold the selected objects. This is particularly useful when the selected object you want to move is behind another object and avoids changing the selection with that initial click.

As you can see in *Figure 2.16*, while holding *Alt*, you can click anywhere on the canvas to move the selected objects. In this example, our selected rectangle is behind other shapes, so we hold the *Alt* key to make sure we only move what's already selected:

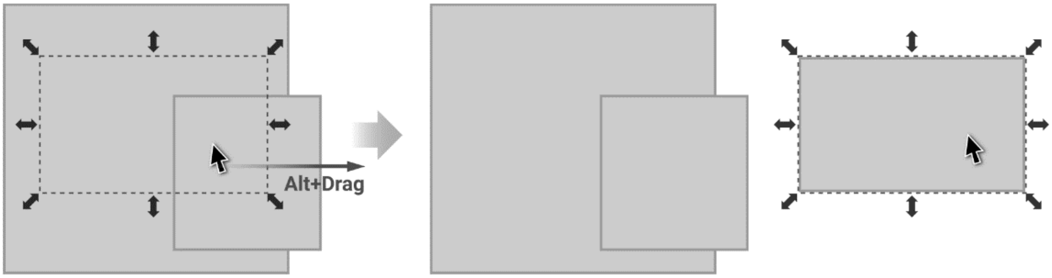
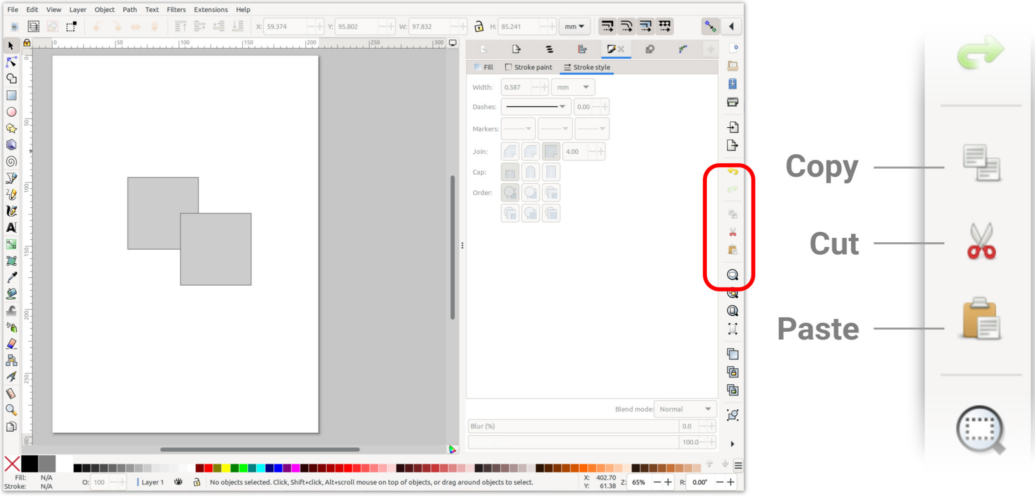


Figure 2.16 – Force dragging with the Alt key

We've got a pretty good grip of moving shapes around in various ways, but what if we want more of the same shape... or maybe we're just tired of our shape and want it gone entirely? Let's have a look at how to copy, cut, paste, and delete shapes in Inkscape.

Cutting, copying, pasting, and deleting shapes

There's nothing too new here if you're familiar with these operations in other software. Inkscape supports the standard hotkeys for **Cut** (*Ctrl* + *X*), **Copy** (*Ctrl* + *C*), and **Paste** (*Ctrl* + *V*). Note that there are buttons for these functions in the Commands bar, as shown in *Figure 2.17*. I recommend using the hotkeys instead as they are far faster to use:



X: 59.374 Y: 95.802 W: 97.832 H: 85.241 mm

Width: 0.587 mm

Dashes: 0.00

Markers:

Join: 4.00

Cap:

Order:

Blend mode: Normal

Blur (%) 0.0

100.0

Fill: N/A
Stroke: N/A

O: 100

Layer 1

No objects selected. Click, Shift+click, Alt+scroll mouse on top of objects, or drag around objects to select.

X: 402.70 Y: 61.38 Z: 65% R: 0.00°

Figure 2.17 – The Copy, Cut, and Paste buttons in the Commands bar

Additionally, Inkscape supports **Paste in Place** (*Ctrl* + *Shift* + *V*), which will paste an object in the same location as the copied object (useful for copying/pasting between layers and groups, which we will see in *Chapter 5*, *Node Editing – Modifying Your Shapes with Nodes and Curves*).

Duplicating (**Edit** > **Duplicate** or *Ctrl* + *D*) is another way to **Paste in Place** without the need to copy first.

You can also drop duplicates of a selected object as you are dragging it by tapping the spacebar along the way, as shown in *Figure 2.18*. This is by far my favorite way to duplicate objects, and I use it constantly out of habit:

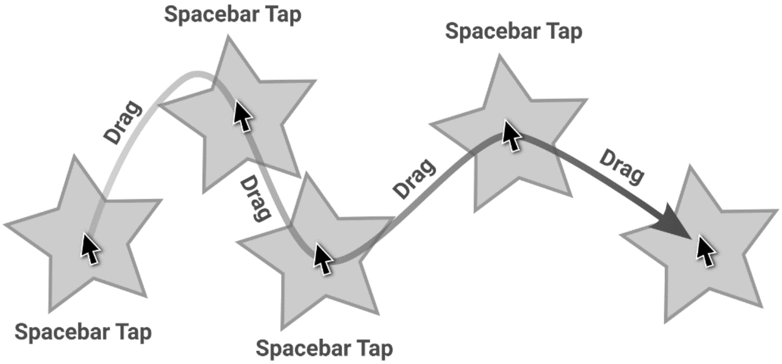


Figure 2.18 – Dropping/stamping duplicates with the spacebar

Deleting selected shapes is just a matter of tapping the *Delete* key. This differs from **Cut** in that the deleted object is not copied to the system's clipboard.

Next up, let's move on to resizing our shapes.

Resizing shapes

With a freshly drawn rectangle (or another object), when you click on the **Select and Transform** tool, you will notice handles at the corners change to resizing arrows. Grabbing these will allow you to resize the shape or shapes you have selected. Note that holding the *Ctrl* key while dragging a handle will preserve the aspect ratio of the shape in much the same way as it does when you are drawing shapes.

This means, for example, if you resize a square, it will stay a square rather than distort into a rectangle, as shown in *Figure 2.19*:

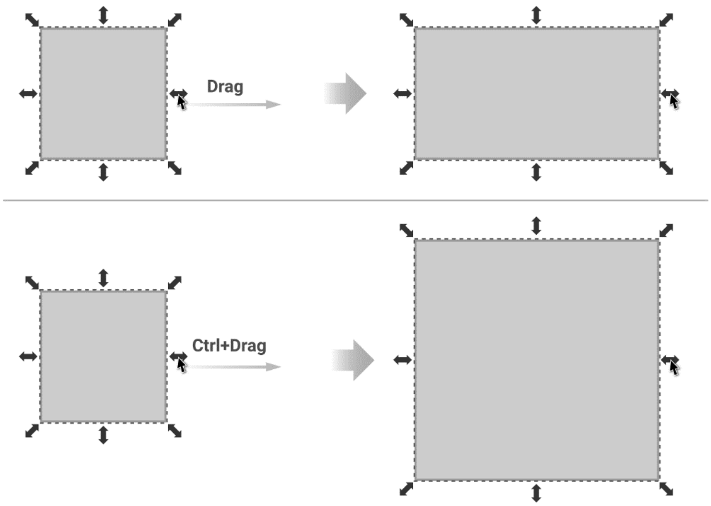


Figure 2.19 – Dragging resize handles to resize in the direction indicated by the arrow handles

This is far from the only way to resize shapes! Let's explore a few more useful options, shall we?

Resizing the height and width values with the Select and Transform tool

You may have noticed the height (**H**) and width (**W**) values in the Tool control bar when you have a shape selected (see *Figure 2.20*). You can use these values to manually set the height and width of the selected shape, and even choose the units (mm, inches, pixels, and so on) you want to resize with as well:

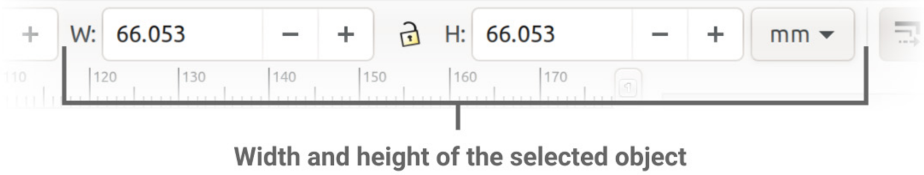


Figure 2.20 – The width and height fields in the Fill and Transform Tool control bar

A relatively new feature in Inkscape is the ability to select one of the scaling handles as the point of reference for scaling (or moving) an object with the Tool control bar options. By default, Inkscape will resize the shape from the top-left corner. However, say we want to scale our rectangle from the bottom-right corner instead.

Simply click to select the bottom-right scaling handle, and you'll see a very thin cross appear behind it running up and down, left and right to the edge of the page. Now, when you use the **W:** and **H:** values in the Tool control bar to scale, Inkscape will scale them from that reference point instead of the middle, as shown in *Figure 2.21*:

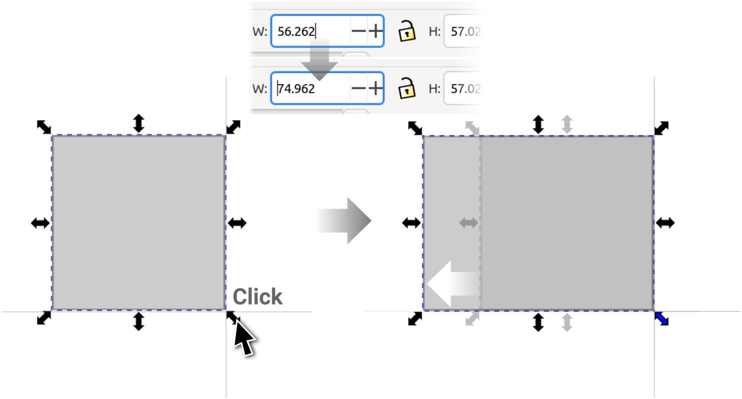


Figure 2.21 – Scaling via the Tool control bar with the lower-right scale handle selected

That's all well and good, but what if we'd like to size objects to the same height and width without all the manual labor of copy/pasting height and width values? We can! Let's explore how in more detail.

Resizing shapes with Paste Size

But what if you want a shape to be the *same* size as another shape? For example, maybe you want it to be the same height as another object that you have drawn previously.

Let's take our original three rectangles and see how we can copy and paste the size of one object onto others. To begin, let's say we want all our rectangles to be the same height as the middle one. Select the middle rectangle and copy it. Then, select the other two rectangles and choose **Edit** > **Paste Height** > **Paste Height Separately**.

This will make all the rectangles the same height, as shown in *Figure 2.22*:

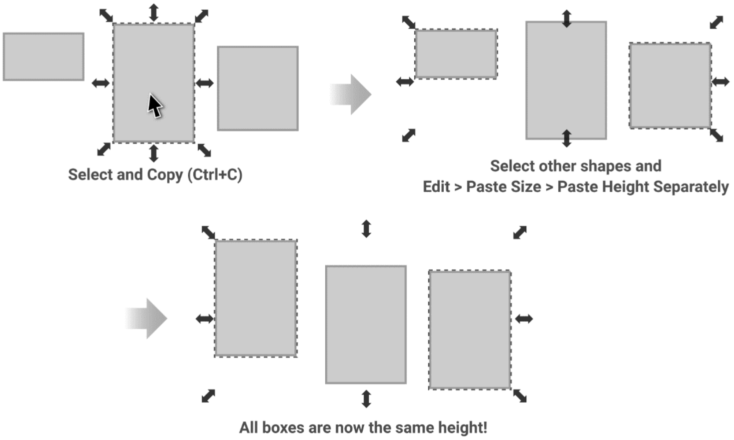

Select and Copy (Ctrl+C)
Select other shapes and
Edit > Paste Size > Paste Height Separately
All boxes are now the same height!

Figure 2.22 – Using the Paste Height Separately function

When you're in the **Paste Size** submenu, you will notice there are quite a few other options for pasting size:

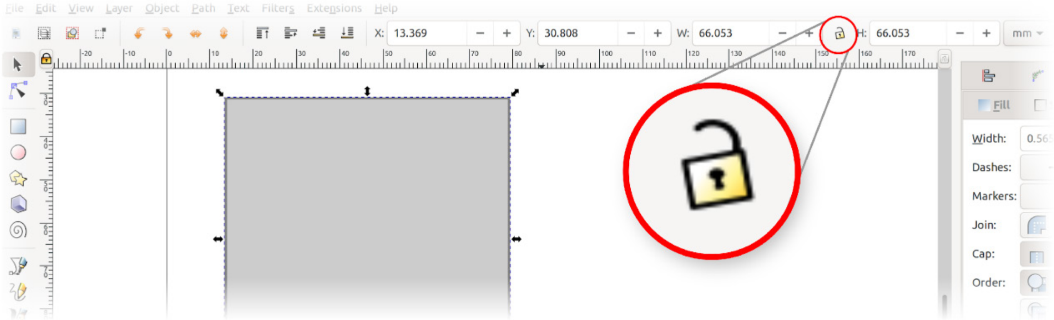
- **Paste Size**: Takes one or more selected shapes and fits the entire selection down to the *size* of the selected shape
- **Paste Width**: Takes one or more selected shapes and squishes the entire selection down to the *width* of the selected shape
- **Paste Height**: Takes one or more selected shapes and squishes the entire selection down to the *height* of the selected shape

These work okay with a single selected shape, but I've never once run into a situation where I needed to squish multiple selected shapes into the area of the copied shape.

The other three options in the submenu are far more useful:

- **Paste Size Separately**: Takes all selected shapes and resizes them to be the same *size* as the copied shape
- **Paste Width Separately**: Takes all selected shapes and resizes them individually to the same *width* as the copied shape
- **Paste Height Separately**: Takes all selected shapes and resizes them individually at the same *height* as the copied shape

However, most of the time, you will likely want to rescale objects *proportionally* without squishing or stretching them. What should you do? Where's the **Paste Height Separately Proportionally** option? Let me introduce you to my very good friend the **Lock Proportions** button, which is making its debut in *Figure 2.23*:



X: 13.369 Y: 30.808 W: 66.053 H: 66.053 mm

Figure 2.23 – The Lock Proportions button, located in the Tool control panel

This tiny button makes the **Paste Size** options suddenly much more useful. For example, clicking it into the locked position before clicking **Paste Height Separately** will make all of your rectangles the same height without stretching them.

Maybe you have three adorable Jack-o'-Lantern illustrations, but you want to make them all the same height in your Halloween party file. *Figure 2.24* shows what happens if you leave proportions unlocked versus locked:

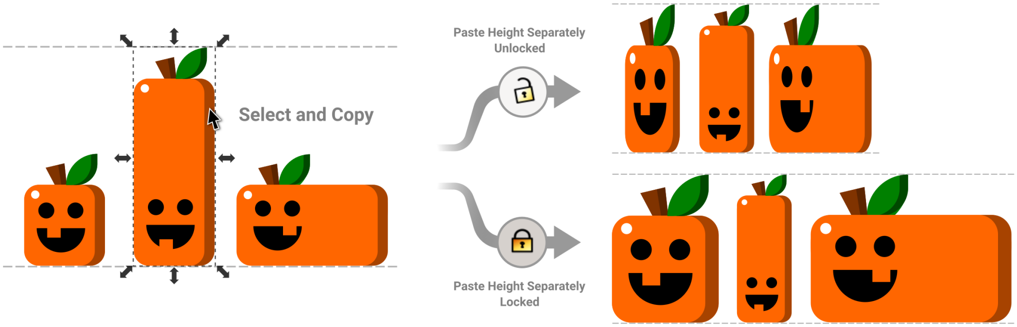

Select and Copy
Paste Height Separately Unlocked
Paste Height Separately Locked

Figure 2.24 – Unfortunate pumpkin stretching cured by locking proportions

As you can see, stretching isn't very good. It looks like a mistake, or at the very least lazy design. The moral of the story is it's best to toggle that **Lock Proportions** button on when using **Paste Size**.

Scaling styles

Now, we know how to scale shapes... but what about the strokes on those shapes? Is there a way to scale the object and leave the stroke as-is? The answer, my dear readers, is yes!

In illustration work, it's often important to keep a consistent stroke, so Inkscape offers you a way to lock the stroke width so it stays the same no matter how big or small you make your illustration during resizing. Take, for example, the adorable child with the flower in *Figure 2.25*:

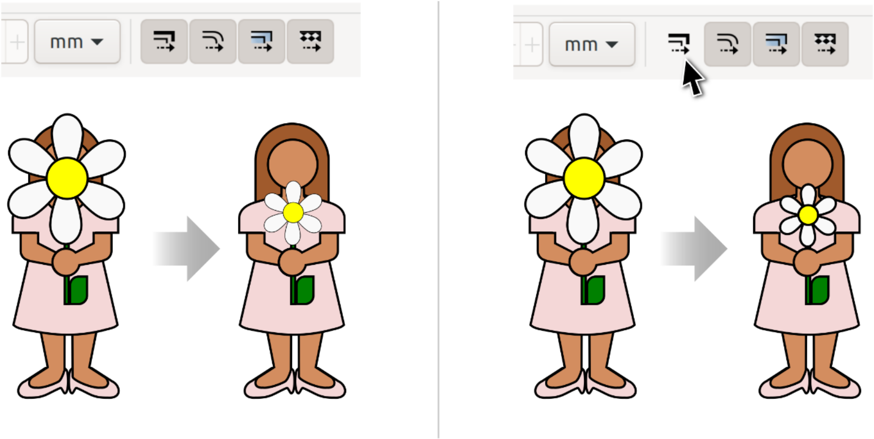


Figure 2.25 – Scaling the flower top with Stroke Scaling on (Inkscape's default) versus off

As you can see, she has a bit of a problem with that massive flower. We could do her a favor by making the top of it smaller, but when we do, our stroke also gets smaller (shown on the left of *Figure 2.25*), which looks terrible. By toggling the **Scale Stroke** button off (it's in the Tool control bar), we can scale, and Inkscape will keep our stroke the same width, as shown on the right of *Figure 2.25*.

It's the same deal for toggling off the other style buttons. Toggling off **Gradient Scaling/Moving** will scale the shape and leave the gradient in place. Toggling off **Pattern Scaling/Moving** will leave the pattern in place as you scale the shape, and although we haven't discussed corner rounding on rectangles yet, toggling off **Corner Scaling** will keep those rectangle corners the same shape regardless of how big or small you resize your rectangle.

Well, have we had enough of resizing? Good. Let's hurry along and get to rotating our shapes.

Rotating shapes

Ready to rotate? Good! Just like with resizing, this section will cover various ways to rotate.

We'll start by selecting one of our rectangles. Notice the resize handles at the corner? Well, clicking your selected rectangle again will change them to rotate handles. *Shift* + *S* will toggle this too, and a rotation pivot will appear in the middle of the rectangle as well. You can move this pivot around and it will allow you to rotate the shape from a different point.

Rotation and rotation after moving the pivot point are shown in *Figure 2.26*:

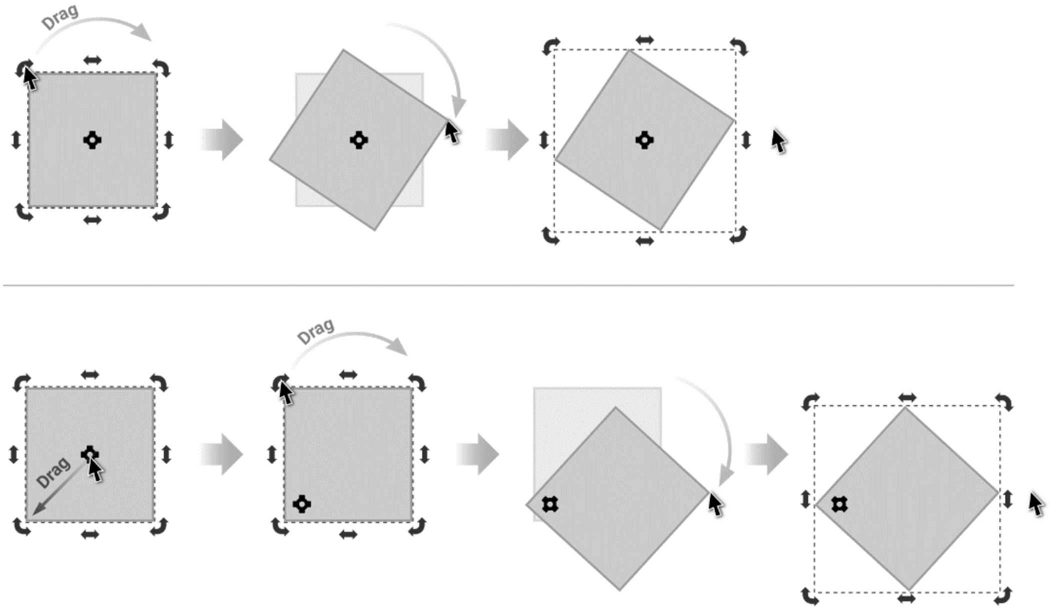
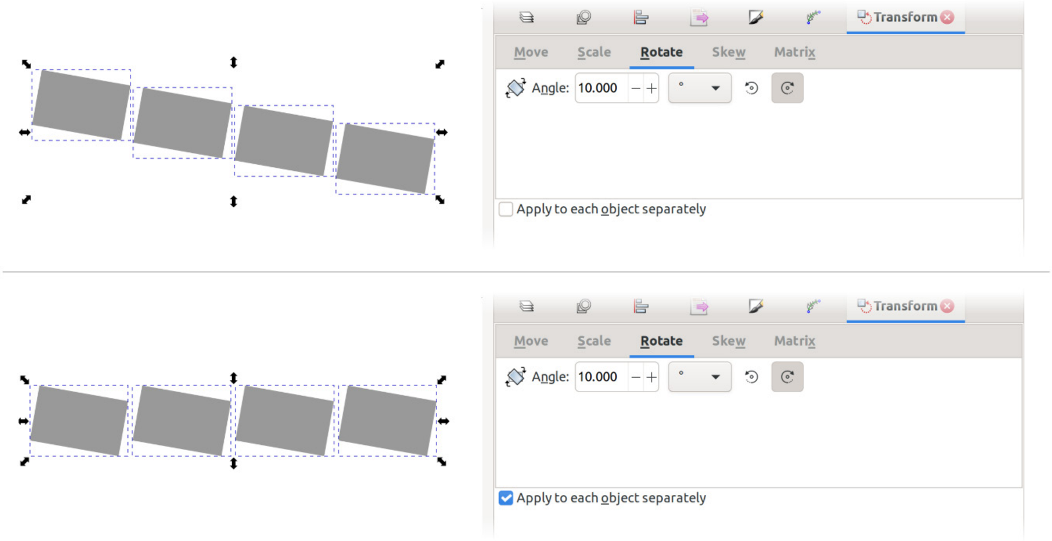


Figure 2.26 – Rotation (top) versus moving the pivot point and rotating (bottom)

While rotating, you can hold the *Ctrl* key or *Alt* key to lock the rotation to 15-degree increments, which is useful for some quick precision.

Another useful trick is to hold the *Shift* key while rotating. This will rotate the shape from the opposite corner of the handle you're using. For example, if you grab the top-right rotation handle and hold *Shift*, Inkscape will start rotating around the bottom-left of the selection box. Releasing *Shift* at any time during the rotation will resume rotation from the pivot point.

If you need to rotate a shape to a precise amount that isn't a multiple of 15, you can also bring up the **Transform** dialog with **Object** > **Transform** and choose the **Rotate** tab. Here, you can enter any value you desire and also choose whether to rotate all selected objects together or rotate them separately by ticking the checkbox shown in *Figure 2.27*:



Angle: 10.000
□ Apply to each object separately

Angle: 10.000
☑ Apply to each object separately

Figure 2.27 – Rotating 10 degrees with the Transform dialog (together and separately)

You'll notice there are other options in the **Transform** dialog to **Scale**, **Move**, **Skew**, and so on. I recommend experimenting with these as they can be useful in certain circumstances. However, rather than inundate you with a detailed description of every option, let's move along to raising and lowering shapes.

Raising and lowering shapes

By default, Inkscape will draw every new shape you make on top of the last shape. But what if we'd like to draw a new rectangle and move it *under* one or more of the rectangles we have already drawn? To do this, we can simply select the rectangle we want to move down behind the others and click the **Raise** or **Lower** button in the Tool control bar.

Figure 2.28 shows what happens to our rectangle when we use each of these buttons to raise and lower our selected shape:

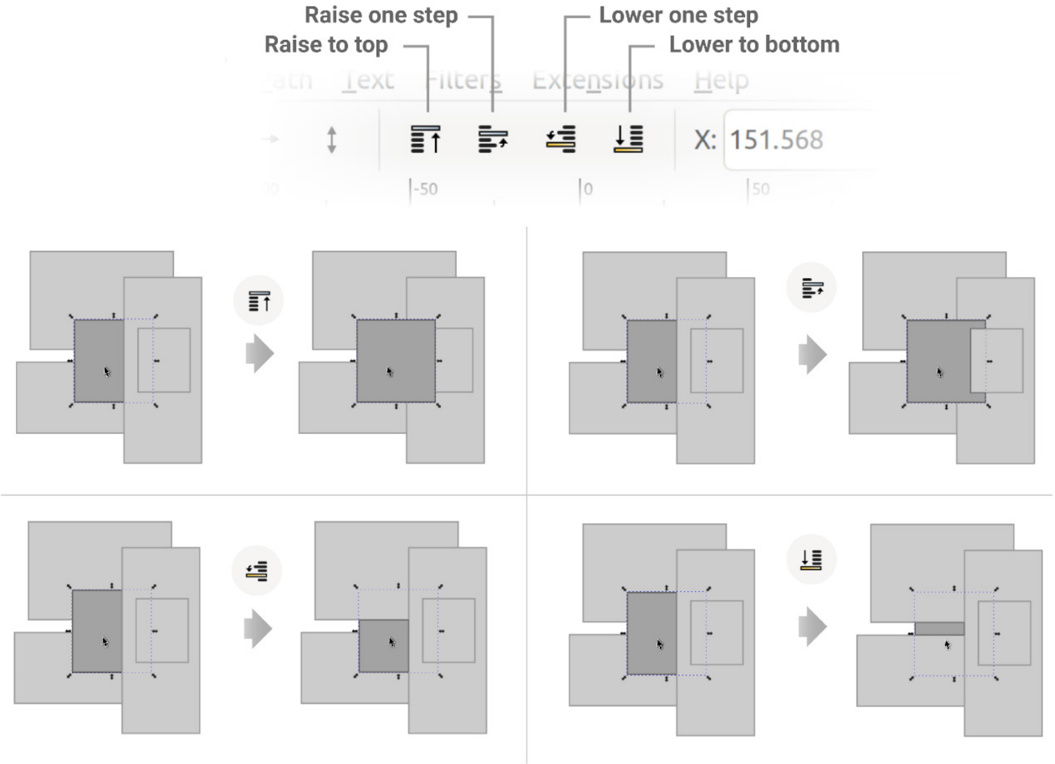


Figure 2.28 – The Raise and Lower buttons, and the effects they have on our selected rectangle

You can also use the *Page Up* and *Page Down* hotkeys to raise and lower the selected object without having to click buttons, which is my preferred method for doing so. Likewise, the *Home* and *End* keys will move the selected object to the very top or the bottom of all the other shapes, respectively.

Notice that the icons for the raise and lower operations appear to form a stack as if the shapes were viewed from the side. Indeed, we will learn that all shapes in Inkscape form a stack, which we will explore more thoroughly when we talk about the **Layers and Objects** dialog in *Chapter 11*, *Organization Using Layers*.

For now, you can think of your shapes like a stack of cutout paper shapes, all overlapping each other from a top-down perspective. *Figure 2.29* shows what this might look like if you could somehow view our top-down overlapping rectangle stack from the side to see the stack of paper objects:

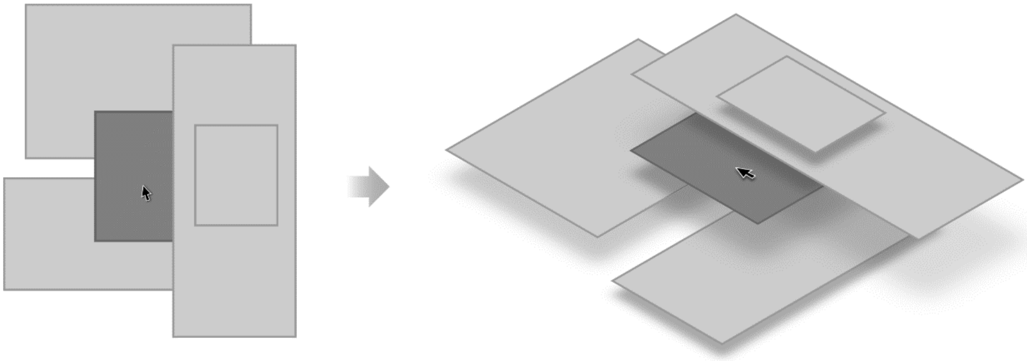


Figure 2.29 – The object stack visualized as a stack of overlapping paper shapes

The more you use Inkscape, the more this will seem natural. So, if this is odd to you now, don't worry. It's just a way to visualize. The important takeaway is that you can raise and lower objects and make them overlap in different ways.

We can raise and lower shapes, but what if we want to group them? After all, it does get tedious re-selecting shapes over and over. Let's see how to temporarily glue shapes together with groups!

Grouping shapes

Sometimes, you'd like to make objects stick to each other. Take, for example, a window made of five rectangles. We'd rather not have to select all the shapes to move them as one piece. Grouping allows us to select the whole bunch of them with a single click as if they were one object.

To group selected shapes into one object, we can choose **Object** > **Group** by using the **Group** button in the Commands bar or simply press *Ctrl* + *G*, as shown in *Figure 2.30*:

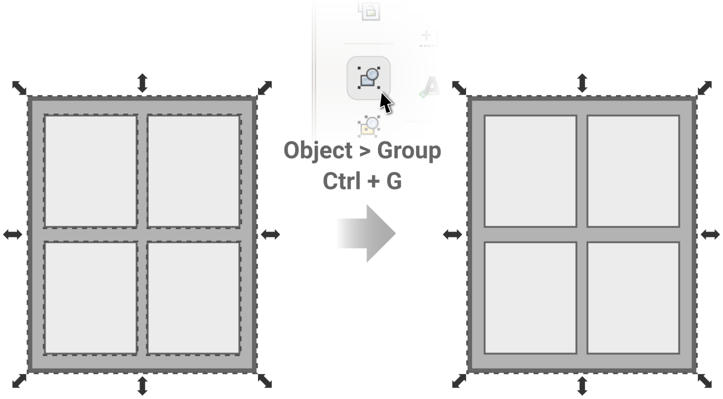


Figure 2.30 – Grouping combines selected shapes into one object

Now, if we deselect our shapes, we can click just once on any of the shapes in the group and move, scale, and rotate the whole thing.

But what happens if we want to move or recolor one of the windowpanes? We could ungroup all our window shapes with **Object** > **Ungroup** (or press *Ctrl* + *Shift* + *G* or use the **Ungroup** button in the Commands bar), but the good news is that we don't have to destroy the group to change things. We can simply double-click on our group, which lets us jump *into* it. This makes it possible to select our pane and make the changes, as shown in *Figure 2.31*:

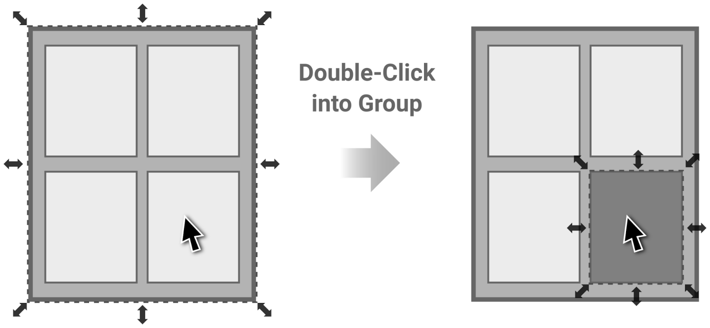


Figure 2.31 – Double-clicking into the group to change one of the panes to a different color

We can then tap *Esc* twice to exit the group, or just click anywhere outside the group. The next time we single-click on an object in the group, it will behave as we want and select the entire group.

Neat, huh? This simple feature is quite powerful. We can even make sub-groups inside groups, which makes grouping a powerful tool for organizing an illustration. But what if we'd like to mirror our shape? Let's learn how to flip them around.

Flipping shapes

Flipping a rectangle isn't very exciting since it's symmetrical. Let's use a line of elephants instead to demonstrate the idea of flipping.

We'd like these elephants to be a bit more social and talk to each other instead of just standing in line. To flip one of them around, we could use the **Flip Horizontal** or **Flip Vertical** buttons in the Tool control panel, but why go through that trouble when we could just use *H* and *V* keys (respectively) on the keyboard to do the same job?

Figure 2.32 shows the result:

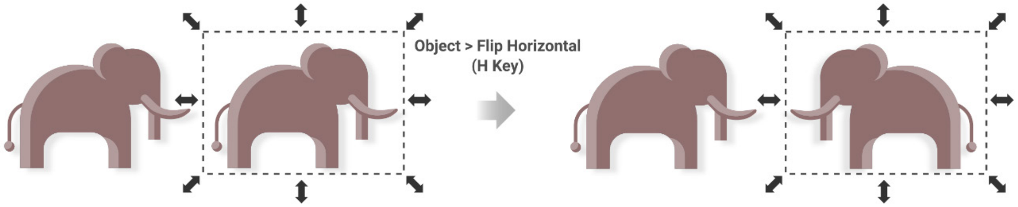


Figure 2.32 – Flipping an elephant

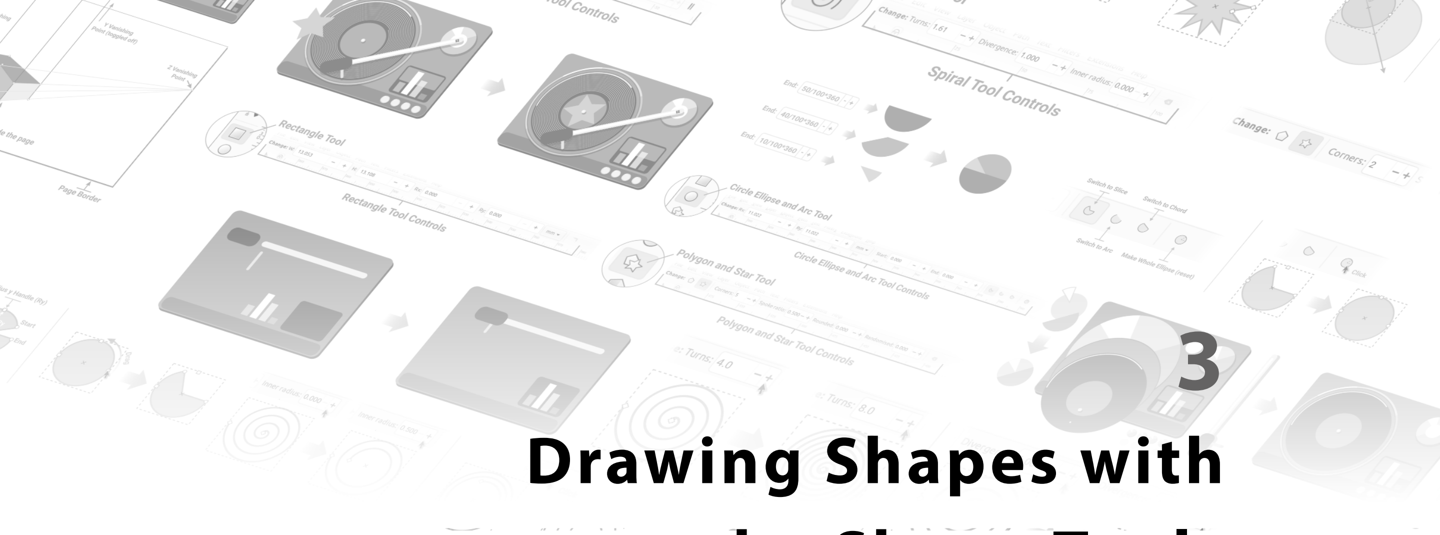
That's right! Simply select one of our elephants, hit the *H* key, and a full-on pachyderm conversation is ready to take place. Just don't ignore the elephant in the room!

Similarly, tapping the *V* key will flip our elephant on its back, but why you would want to do that to our poor elephant is beyond me.

Summary

Well, we sure covered a lot in this chapter! We learned the basics about vector shapes, how to color and style them with **Fill and Stroke**, as well as many ways to move, scale, rotate, reorder, and transform our shapes.

This information will serve us well in moving our shapes about and completing our first project at the end of the next chapter. So, let's continue and see the many wondrous shape tools we have available to work with in Inkscape.



3

Drawing Shapes with the Shape Tools

When building anything, it's helpful to know what materials and tools you have to work with and get a little practice using them. Since the previous chapter gave us a solid foundation for manipulating shapes in various ways, let's dive into what kind of shapes Inkscape can make out of the box.

In this chapter, we will cover the following topics:

- Inkscape object properties
- The Rectangle tool
- The Circle, Ellipse, and Arc tools
- The Polygon and Star tool
- The 3D box tool
- The Spiral too

Technical requirements

This chapter assumes you've read *Chapter 2*, *Moving and Styling Shapes*, and that you have a good working knowledge of how to move, resize, and rotate objects around the page. You might be able to muddle your way through the project at the end of this chapter if you haven't, but as with most things in life, that little time spent learning the basics pays off in time spent fiddling about.

Inkscape's object properties

Objects in Inkscape have special properties. These properties are different, depending on the type of object you are drawing (circle, rectangle, star, spiral, and so on).

For example, you may have noticed that the rectangles you made in the previous chapter with the **Rectangle** tool have corner rounding properties, which allow you to round the corners. These properties are displayed in the Tool control bar and have special handles displayed on the selected shape itself to adjust these values without needing to enter values.

Stars or polygons made with the **Star and Polygon** tool have an adjustable number of points as well as point rounding, plus other useful features such as randomization, which we will soon see when exploring these drawing tools.

Likewise, circles have **Start** and **End** angles, which are useful for making pie charts, as we will soon explore when we look at the **Circle**, **Ellipse**, and **Arc** tools.

Figure 3.1 shows various types of Inkscape objects and their respective properties:

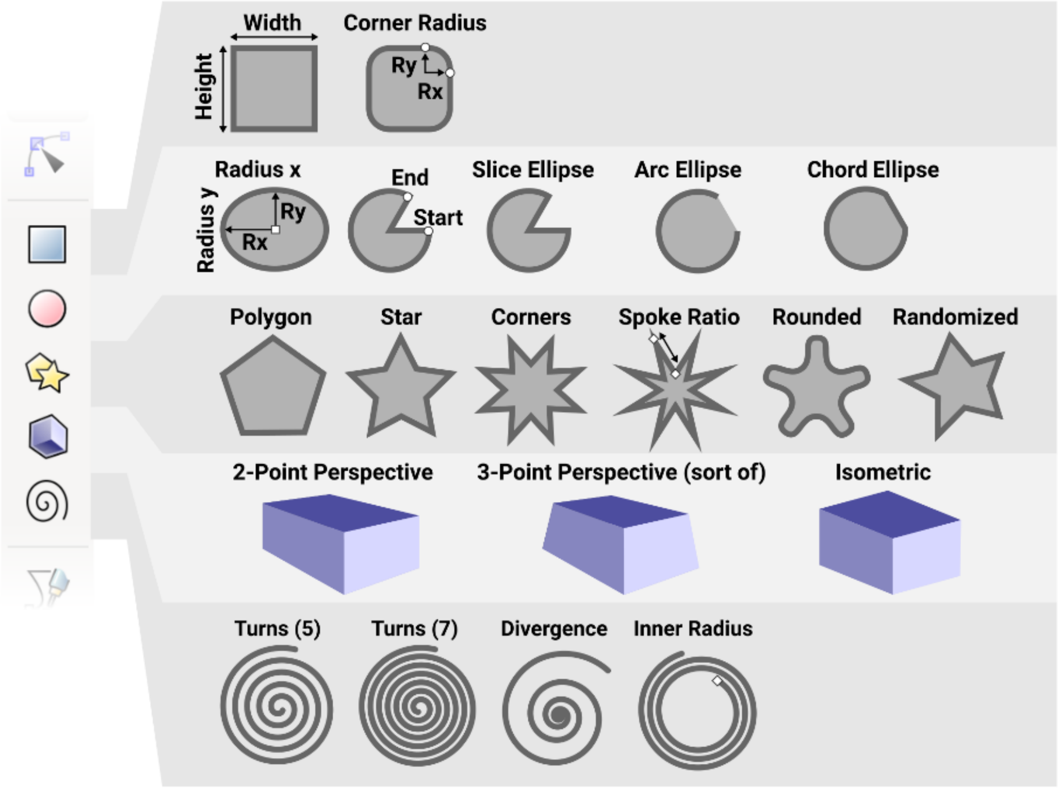


Figure 3.1 – Different object types have different properties that can be edited in the Tool control bar when you select an object tool

Now that we've got an idea of the basic shapes we can use, let's take a closer look at the tools Inkscape gives us to make and modify these fantastic shapes.

The Rectangle tool

Okay, so you drew some rectangles in the previous chapter. Let's build on that by going through the various options for your brand-spanking-new rectangles. It will come as no surprise that these **Rectangle** tool options are located in, yes, you guessed it, the Tool control bar, as shown in *Figure 3.2*:

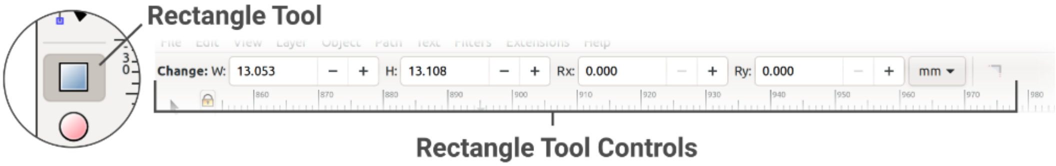


Figure 3.2 – The Rectangle tool icon and the corresponding Tool control bar options

Here, you have some manual control over the width (**W**) and height (**H**) of the rectangle, as well as two **Corner Radius** values (**Rx** and **Ry**) that correspond to the **Corner Rounding Handles** property on the top-right corner of the shape, as shown in *Figure 3.3*:

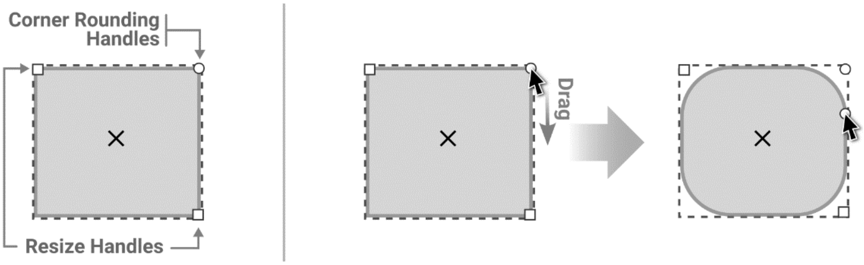


Figure 3.3 – Using the circular handles to drag the sharp corners into rounded ones

Dragging either of these handles will round the corner of your rectangle evenly on all four corners. You can also drag the other handle afterward to make an elliptic corner instead, though I have to admit that I've never wanted to do this in all my years as a designer.

It's 100% ugly 100% of the time. If you happen to move both of the handles accidentally and are stuck with an elliptical corner, simply hit the **Make Corners Sharp** button in the Tool control bar and it will reset it for you, as shown in *Figure 3.4*:

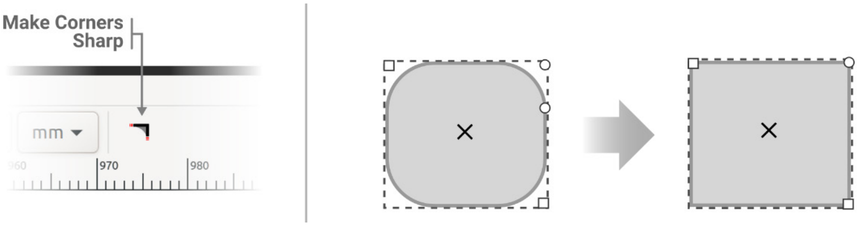


Figure 3.4 – Clicking the Make Corners Sharp button, located in the Tool control bar, resets the rounded corners to sharp

You can also hold the *Ctrl* key and drag one of the handles to round the corners symmetrically.

The handles at the top left and bottom right will scale the height and width of the rectangle *even if the rectangle is rotated*. That is to say, the rectangular shape will be preserved.

As exciting as rounded rectangles are, maybe you want a circle or ellipse instead...

The Circle, Ellipse, and Arc tools

Let's make some circles, ellipses (elongated circles), and arcs. For briefness, we'll just refer to this tool as the **Circle** tool, even though we know from the title that we can make other shapes with it too.

We'll approach making circles the same way we did making rectangles. Click and drag one corner out until your shape is sized to your satisfaction. If you want to drag a circle out from the center instead of the corner, simply hold the *Shift* key, and the circle you create will be dragged out from the center. Additionally, hold the *Ctrl* key (so, *Shift* + *Ctrl*) and you'll get a perfect circle too, as shown in *Figure 3.5*:

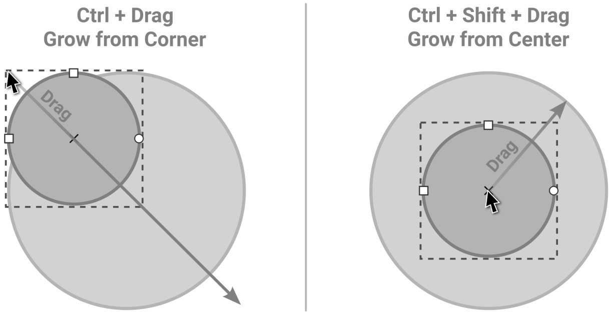
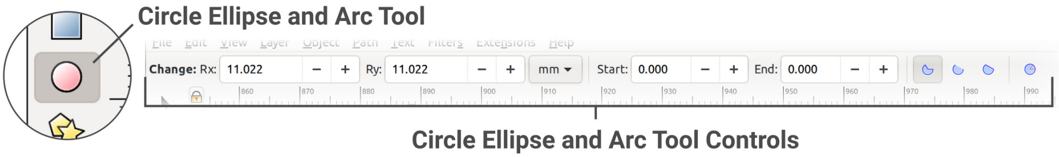


Figure 3.5 – To resize or create perfect circles, hold Ctrl, or Ctrl + Shift to resize from the center

Similar to the rectangle, there are some special properties and options we can play with in the Tool control bar (*Figure 3.6*):


Circle Ellipse and Arc Tool
File Edit View Layer Object Path Text Filters Extensions Help
Circle Ellipse and Arc Tool Controls

Change: Rx: 11.022 − + Ry: 11.022 − + mm Start: 0.000 − + End: 0.000 − +

Figure 3.6 – The Circle tool button and Circle tool options in the Tool control bar

Notice that **Rx** and **Ry** are present, but this time, they correspond to the *X* and *Y* radius of the entire shape (thus controlling the size of the ellipse).

Additionally, there are two values, **Start** and **End**, which are angle values in degrees. You may remember from your grade-school geometry class that angles can be represented by degrees, such as 90 degrees for a quarter circle, 180 degrees for half a circle, and 360 degrees for a complete circle.

As you can see in *Figure 3.7*, using these degree values also allows you to make easy pie charts and other semi-circular shapes by dragging the start and end handles:

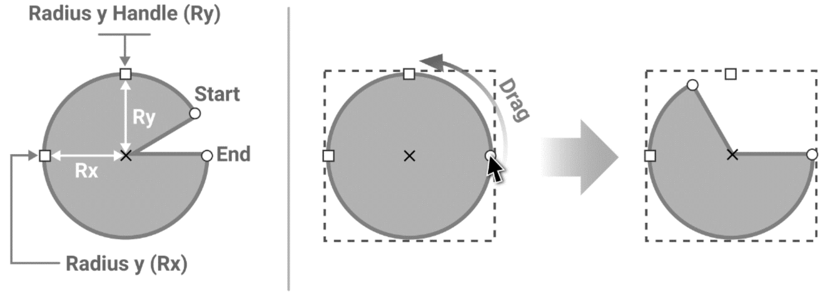


Figure 3.7 – (left) Parts of the Circle corresponding to the values in the Tool control bar. (right) Dragging the Start and End handles

You can switch the style of the shape using the corresponding icons in the Tool control bar. You can also reset the ellipse/circle to make it whole again by clicking the **Make Whole Ellipse** button, as shown in *Figure 3.8*:

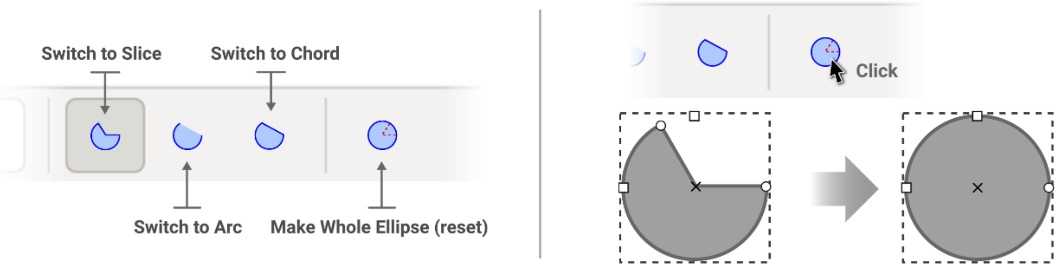


Figure 3.8 – Shape type option toggles in the Tool control bar. Here, you can switch between Slice, Arc, Chord, and Whole Ellipse

Seems easy enough, right? Let's use a real-world example and make a brand-spanking-new pie chart. You can use the `P/100 * 360` equation to get your pie piece, where `P` is the percent of the pie chart you want to fill up. Let's take an example where we have a pie chart with pieces representing 50%, 40%, and 10%. The following equations, when typed into the **End Spin Box** area, will give the correct pie piece shapes:

- `50/100*360`
- `40/100*360`
- `10/100*360`

Figure 3.9 shows the resulting shapes:

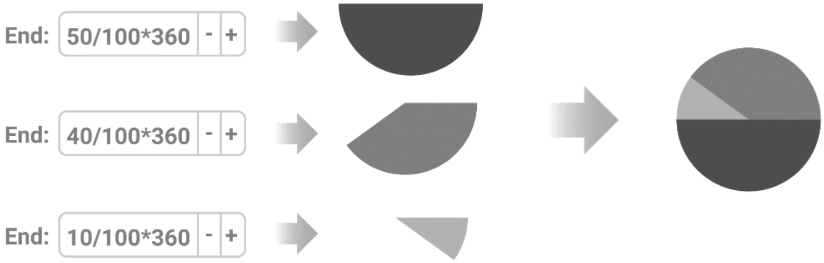


Figure 3.9 – End values of 50%, 40%, and 10%, and their resulting pie sections combined to make a pie chart

Ready for some even more interesting shapes? You need to look no further than the **Polygon and Star** tool.

The Polygon and Star tool

Talk about an object with options! The **Polygon and Star** tool is useful for making all manner of neat shapes. Here's a view of the option-packed Tool control bar for this tool (*Figure 3.10*):

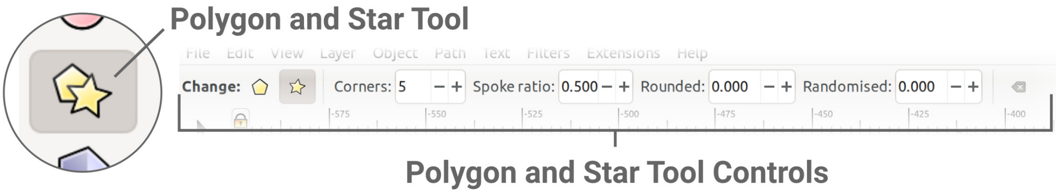


Figure 3.10 – The Polygon and Star tool and its corresponding controls

The first two toggles by the word **Change:** control whether the shape is a polygon or a star, as seen in *Figure 3.11 (left)*. The **Corners:** Spin Box controls how many corners the polygon has or how many points the star has, which you can see in *Figure 3.11 (right)*:

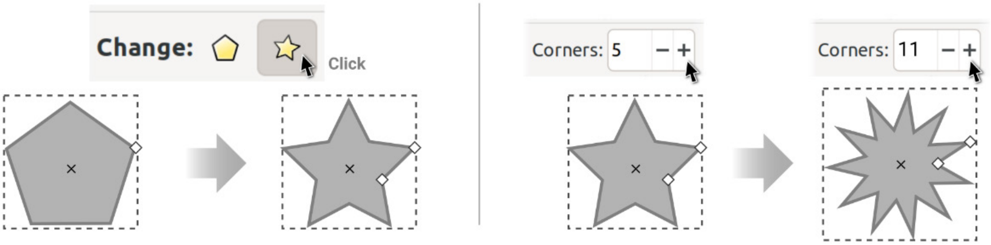
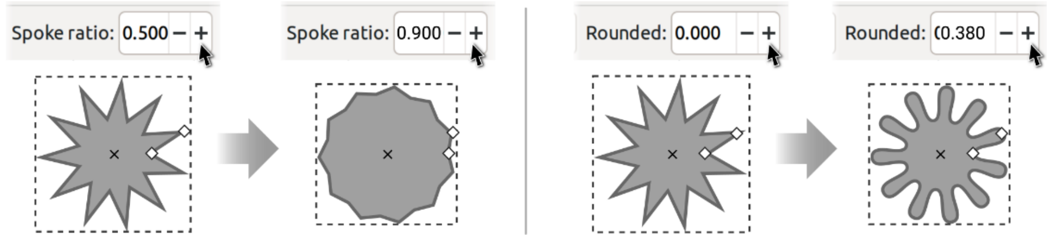


Figure 3.11 – (left) Changing a polygon into a star and back with the Change: toggles. (right) Adding/removing corners with the Corners: Spin Box controls

The **Spoke Ratio** aspect of our star controls the sharpness of the spokes, as you can see in *Figure 3.12 (left)*. Typically, though, it's easier to control via dragging the spoke handles, which allows you to manually change it to be as pointy as you like without messing with the Spin Boxes.

The **Rounded:** field is a fun one, letting you make anything from a floral or splotch shape to one of those Spirograph thingies from the 1980s. Sure, they are almost completely useless as a design element, but look me in the eye and tell me it's not fun to play with anyway.

See *Figure 3.12: (right)* for the effect of increasing the **Rounding** value of your star:



Spoke ratio: 0.500

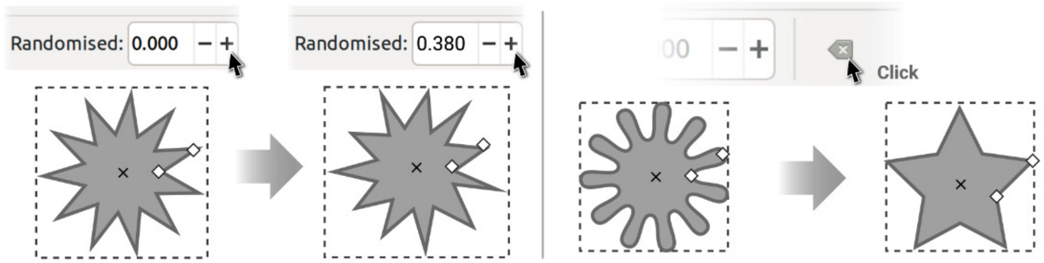
Spoke ratio: 0.900

Rounded: 0.000

Rounded: 0.380

Figure 3.12 – (left) Increasing Spoke Ratio makes shallower peaks and valleys in your star shape. (right) Increasing the rounded value creates splotches and Spirograph-like shapes

The **Randomised:** Spin Box controls the amount of randomization that's added to each point of the star to create a jitter effect. If you want your stars to be a bit less regular and look more haphazardly drawn, this setting is for you. *Figure 3.13 (left)* shows this randomization; the higher the value, the crazier the distortion gets:



Randomised: 0.000

Randomised: 0.380

Figure 3.13 – (left) The result of increasing the randomization value. (right) Clicking the Reset Shape Parameters button to reset the geometry to 5 points without rounding or randomization

To return your polygon or star to its default values, press the **Reset Shape Parameters** button, as shown in *Figure 3.13 (right)*. You can use the following modifier keys to adjust your star attributes freehand as well:

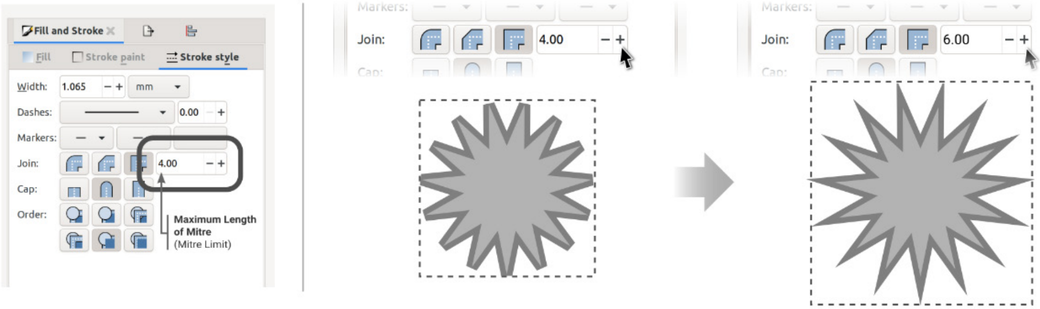
- **Snap to Angle**: Hold the *Ctrl* key when drawing your star or polygon to lock it to precise incremental angles

- **Lock Spoke Symmetry**: Hold the *Ctrl* key while dragging one of the spoke handles to make the spokes symmetrical
- **Change Rounding**: Hold the *Shift* key while dragging a spoke handle to control corner rounding
- **Change Randomisation**: Hold the *Alt* key to increase or decrease the randomization value as you are dragging the spoke handles

While drawing your stars, you may have noticed that with large amounts of spokes, the sharp tips *are no longer sharp*. That's because of something called the **mitre limit**. In Inkscape, where the stroke joins at a sharp corner, the stroke can be drawn in three different join styles: round, bevel, and mitre.

If mitre is chosen, Inkscape will make a sharp point. If the point is too long, however, Inkscape can automatically cut the corner into a bevel instead. Increasing the **mitre limit** value (the value in the Spin Box next to the mitre join toggle) will increase the distance that Inkscape will try to join your corner with a sharp mitre edge.

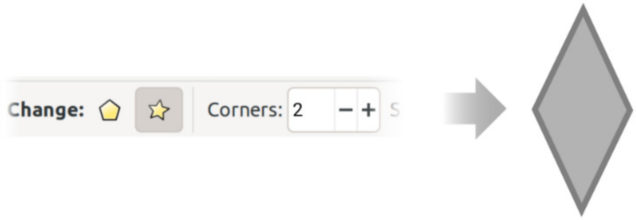
In *Figure 3.14* you can see the **mitre limit** value has been reached, and Inkscape has cropped off the corners into bevel joints instead. Increase the **mitre limit** value (maximum length of the mitre) to get nice sharp points back:



Width: 1.065 mm
Dashes: 0.00
Join: 4.00
Join: 6.00

Figure 3.14 – (left) The mitre limit value. Increasing this will put the points back on your star's stroke (right)

There's one more hidden treasure in the **Star and Polygon** tool. As of Inkscape 1.1, you can create a polygon rhombus (or diamond, if you prefer my terrible puns about hidden treasure). Simply set the **Corners** value to 2 and you get this new shape. This was not possible in previous versions and is a very handy shape to have (see *Figure 3.15*):



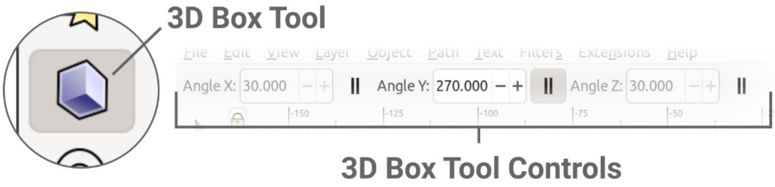
Corners: 2

Figure 3.15 – Changing to a star shape and setting Corners to 2 to get a rhombus shape

That's about it for the **Polygon and Star** tool. If you thought this tool was pretty fancy, wait until you get a load of the 3D Box tool!

The 3D Box tool

By utilizing a two-point perspective, the 3D Box tool draws… well, 3D boxes. As you can see in *Figure 3.16*, there are some rather vexing **Angle** controls and buttons with parallel lines beside them. Unlike the previous tools, changing these values will cause nothing but confusion and frustration unless you know precisely what you're doing, so I highly recommend leaving them alone and using the handles instead:


3D Box Tool
File Edit View Layer Object Path Text Filters Extensions Help
3D Box Tool Controls

Angle X: 30.000 Angle Y: 270.000 Angle Z: 30.000

Figure 3.16 – 3D Box tool and controls in the Tool control bar

Things can also get quite strange when you attempt to draw a box outside the page area, so if you're going to use this tool, I recommend drawing your first box on the page, then adjusting the vanishing points by click-dragging them to extend beyond the page borders. Then, you don't accidentally step into the fourth dimension and get a twisted mess.

Figure 3.17 shows a normal on-page 3D box (left) versus an off-page one (right):

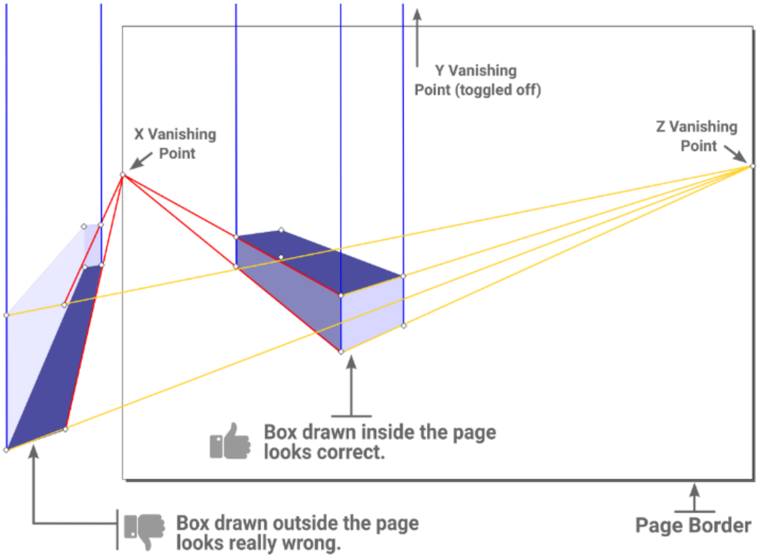

Y Vanishing Point (toggled off)
X Vanishing Point
Z Vanishing Point
Box drawn inside the page looks correct.
Box drawn outside the page looks really wrong.
Page Border

Figure 3.17 – Comparison of 3D boxes drawn on and off the page

Since the default vanishing points are on the edges of the page, drawing a box off the canvas results in a really… interesting-looking box. Draw on the page and adjust the vanishing points after the fact.

You can switch to a three-point perspective by toggling the **Y Vanishing Point** property to finite (off). It is toggled on (infinite) by default. Note that the **Vanishing Point** (**VP**) buttons feature two parallel lines as an icon on each. This is because when they are toggled on, the VP lines are parallel instead of converging – that is to say, they don't ever cross, so no vanishing point is created by the lines.

If you've worked with perspective drawings before, you have probably guessed that you can also create one-point perspective and isometric boxes by toggling on/off the other two VPs. There is an easier way to construct isometric shapes using the axonometric grid, which we will discuss later in this book.

I'm lost... what are you talking about with all this "perspective" stuff?

On the off chance that you've never heard of VPs or perspectives, and so on, I recommend reading some articles on the subject by searching for *two-point perspective* in your favorite web search engine.

You don't *need* to know any of this to use Inkscape, as this tool has rather limited usefulness in 2D illustrations. Indeed, we will be simulating a variety of 3D-looking illustration effects throughout this book, none of which involve this tool.

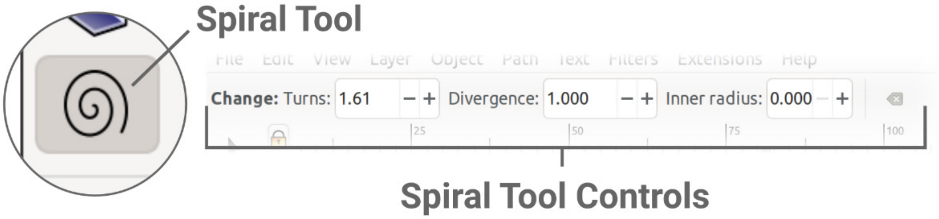
As you've seen in this section, the primary use of the **3D Box** tool is to make boxes. Those boxes can be used as the basis for package design or architectural mock-ups and other faux-3D shapes, depending on your illustration skill level. It can also help you lay out 3D scenes if you're into that sort of thing.

Full usage of this tool for illustration work is beyond the scope of this book, but it's there if you want to use it. If you're into 3D, however, I recommend grabbing a copy of Blender (from `https://www.blender.org/`), which is another free and open source software power tool, specifically designed for 3D vector art.

Now, let's finish our object tool tour with the **Spiral** tool.

The Spiral tool

Every once in a while, you may want to make a spiral. Well, your go-to for that sort of thing is the **Spiral** tool, whose Tool control bar options are shown in *Figure 3.18*:


Spiral Tool
File Edit View Layer Object Path Text Filters Extensions Help
Spiral Tool Controls

Change: Turns: 1.61 Divergence: 1.000 Inner radius: 0.000

Figure 3.18 – The Spiral tool and associated controls in the Tool control bar

The **Turns** Spin Box controls how many times the spiral goes around the midpoint (revolutions), while the **Divergence** Spin Box determines the rate at which the spiral grows between revolutions. That is to say that the spiral gap starts smaller toward the center and grows larger the farther out you go (as you would see in a nautilus shell, for example).

You can see examples of turns and divergence in *Figure 3.19*:

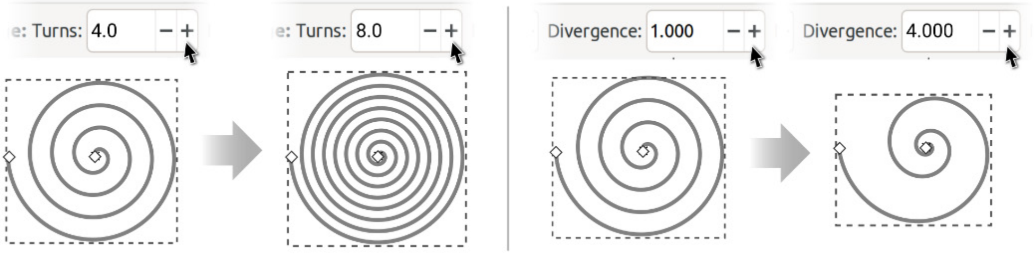


Figure 3.19 – Increasing the Turns value (left) and increasing the Divergence value (right)

Dragging the handles lets you manually adjust where the start and end points reside along the spiral.

There are also some handy modifier keys you can use to edit your spiral while dragging a handle:

- **Snap to angle**: Holding the *Ctrl* key snaps the **Start** or **End Node** property to an incremental angle around the center
- **Resize**: Holding the *Shift* key lets you resize the spiral if you're dragging the **End handle** property
- **Rotate**: Holding the *Alt* key while dragging the **End handle** property rotates the whole spiral
- **Adjust Divergence**: Holding the *Alt* key while dragging the **Start handle** property allows you to manually adjust the **Divergence** value

You can also use the **Inner radius** Spin Box to set a decimal percentage value from `0.0` to `1.0`. For example, if you wanted the spiral to start halfway from the center point to the outside, that's 50% of the distance, or `0.5`. The result is shown in *Figure 3.20 (left)*. And of course, you can reset the spiral to its defaults by clicking the **Reset shape parameters** button shown in *Figure 3.20 (right)*:

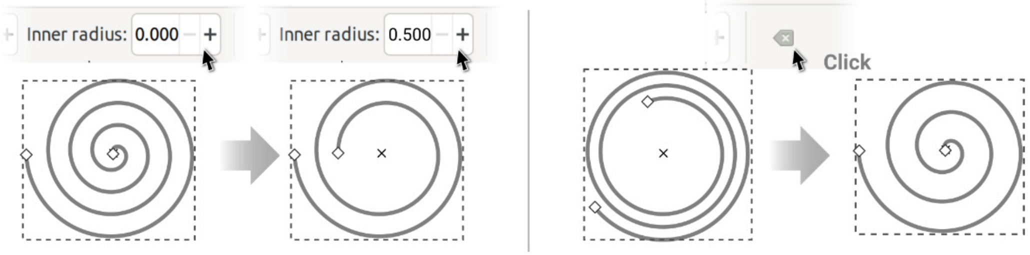


Figure 3.20 – (left) Entering a value of 0.5 for the inner radius starts the spiral at 50% of the distance from the center to the outside of the spiral (right). You can reset the spiral parameters to their defaults with the Reset shape parameters button

With that, we have wrapped up the object tools! Before we go on and learn how to further modify these base shapes, let's take a moment to make something from start to finish, using what we've learned so far in this chapter.

Try it yourself

Okay, okay, so maybe that's not the most fascinating thing to make, but we're dealing with basic shapes here, and a turntable (also known as a record player) has a variety of shapes that utilize most of the tools we've discussed so far. We'll spice it up with a modern view screen with an equalizer built right in too.

Here's how to do it:

1. Let's start with the **Rectangle** tool and draw the series of shapes shown in *Figure 3.21 (left)*. I've drawn them like this so you can see the different shapes before we move them into place to form the main body of our turntable, as you can see in *Figure 3.21 (right)*:

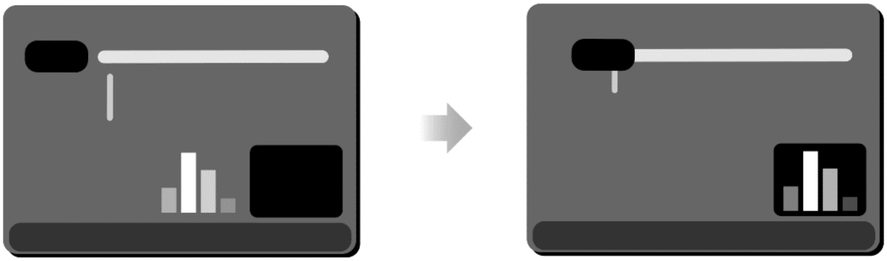


Figure 3.21 – (left) All shapes drawn with the Rectangle tool (right). Rectangles repositioned to form the tonearm, equalizer, and the front panel of the turntable

2. Add some shadows and highlights to the arm by selecting all the parts of the arm and choosing **Group** from the **Object Menu** area in the Menu bar (recall that the shorthand for this is **Object** > **Group**).
3. After we have grouped the parts, we can move them around as one object. We'll make a copy of this group with *Ctrl* + *C* and paste it with *Ctrl* + *V*. Make all the shapes in this group black by simply clicking the black color swatch in the **Swatch Palette** area.
4. To make this shadow shape semi-transparent, enter a value of `50` in the **O:** Spin Box next to **Color Indicators**.
5. Now, move the newly created shadow under the arm by tapping the *Page Down* key or by using the **Lower Selection one Step** option from the **Select and Transform** tool area.
6. Do the same thing for the reflection on the metal portion of the arm and on our equalizer window – make rectangles with 50% opacity, but this time set the color to white instead of black and move the rectangles over the arm and equalizer, respectively, instead of underneath.

The before and after are shown in *Figure 3.22*:

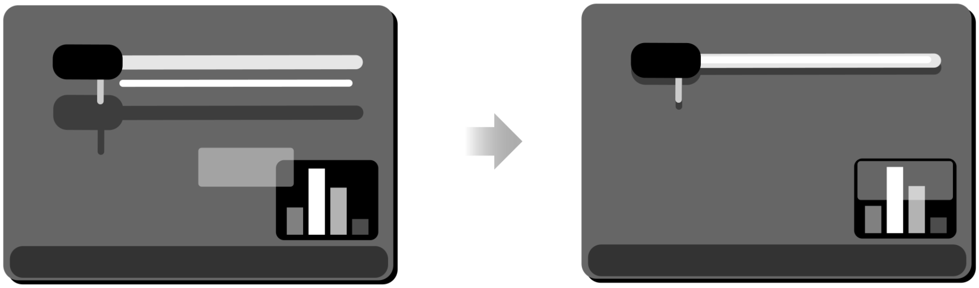


Figure 3.22 – Making the shadows and highlights for the tonearm and the equalizer

7. Next, use the **Circle** and **Ellipse** tools to add all of our round shapes. Start by making pie chart-like round metal bits, as shown in *Figure 2.23 (left)*.
8. Once we have this, we'll simply group it and duplicate it (either copy/paste or tap the spacebar to drop copies while dragging). Scaling the pie chart's shape up gives us the metal base for the record platter while scaling it down gives us metal for our four control dials, as shown in *Figure 3.23 (right)*. Note that the glossy edge on the record is a circle set to Arc mode with a white stroke, and the fill value is set to **None**:

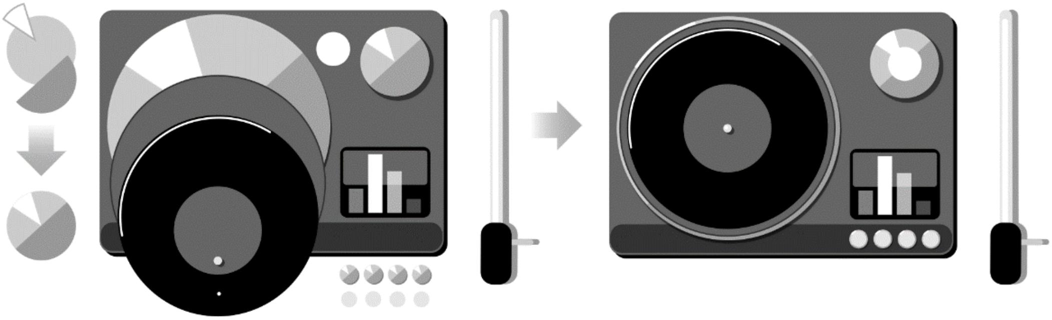


Figure 3.23 – Lots of circle and pie shapes combined to make the parts of the platter, including dials

9. Next, we'll add the track to our record with a spiral by setting the **Turns** value to `31` and the **Inner Radius** value to `0.5` (50% of the width of the spiral).
10. Set the **Stroke** option to a light gray by holding the *Shift* key and clicking a gray color swatch.
11. Adjust the stroke's width to be very slim by clicking on the value next to the **Stroke Color Indicator** area. Then, you can change the value in the **Fill and Stroke** dialog that pops up with the stroke's **Style** tab and **Width** Spin Box so that it's ready to accept your new value.
12. Change the fill to **None** by clicking on the **red X** swatch at the far left-hand side of the **Swatch Palette**.

13. Then, we can move our spiral into place, as shown in *Figure 3.24*:

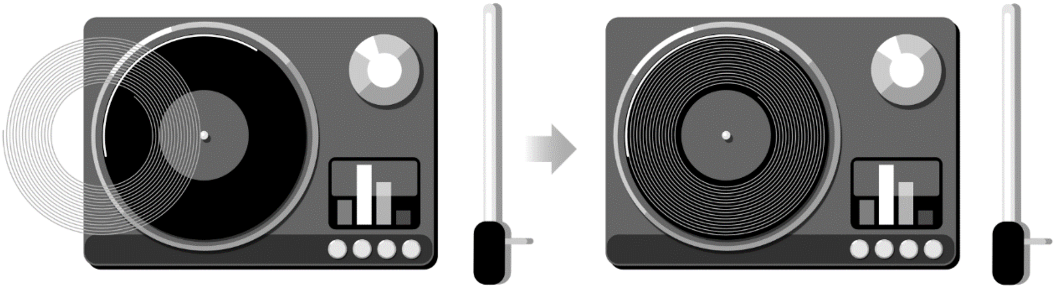


Figure 3.24 – One of the few uses for a spiral

It looks pretty good, but now, we need to attach that tonearm to the turntable. We could just rotate it approximately and then move it into place, but then the stylus position might not be where we want it, causing us to have to rotate and move it again. What we want is to move the rotational pivot to the back of the tonearm. Then, we can move it once to align the back of the tonearm with the tonearm base and rotate it into place just like a real turntable.

14. We can do that by clicking once more after we select the tonearm (or just press *Shift* + *S*) to get our rotation handles and pivot point and drag the pivot to the back of the tonearm, as shown in *Figure 3.25 (left)*:

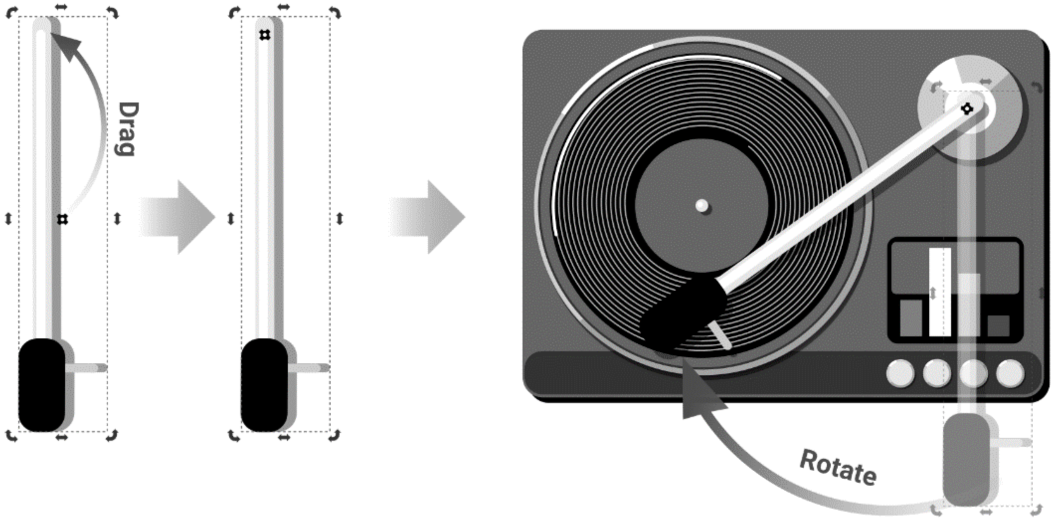


Figure 3.25 – Grouping the tonearm parts together and moving the rotation pivot to the back lets us rotate the arm the same way we would in real life

15. Move the tonearm into place on the base and rotate the arm into the exact position you want it, as in *Figure 3.25 (right)*.
16. As a final touch, we'll add a star to the record label. Yeah… drawing that probably doesn't need much explanation at this point, but here's a nice illustration to show you how it looks (see *Figure 3.26*):

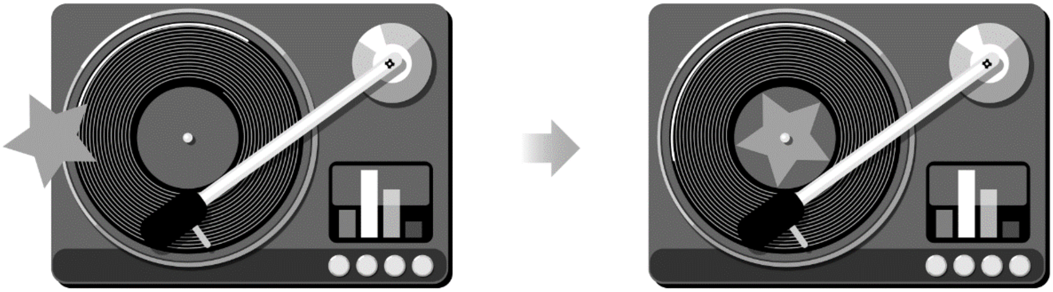


Figure 3.26 – Adding a star and admiring our finished turntable illustration

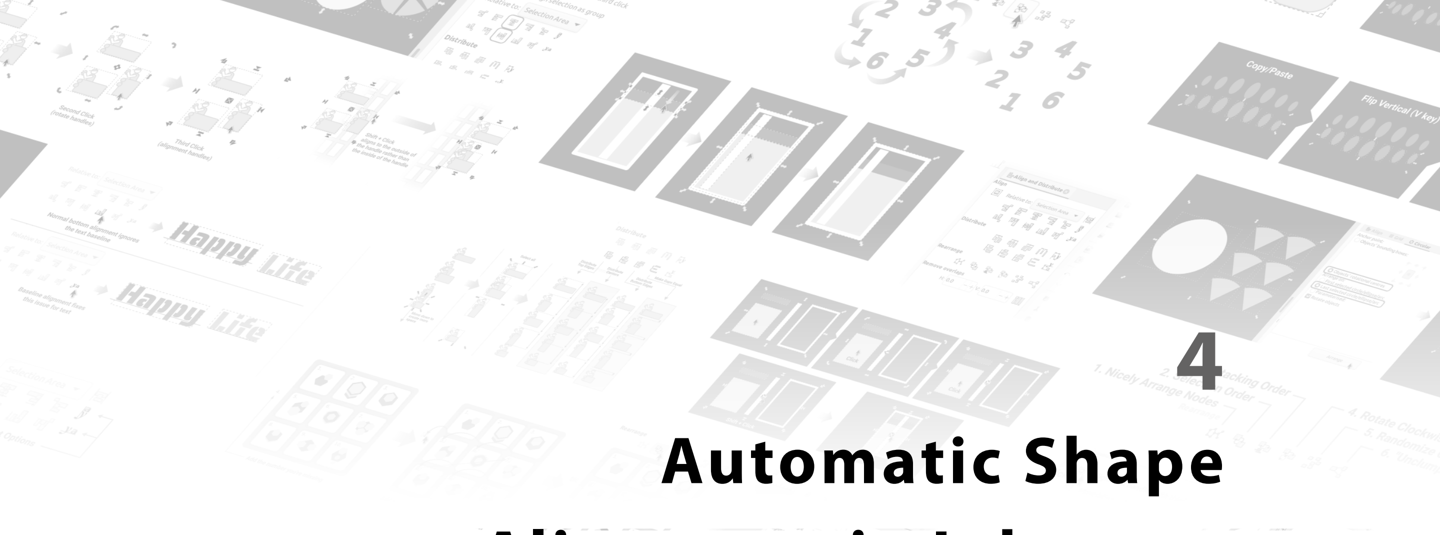
Congratulations! You've made your first full illustration in Inkscape. Let's save it using **File** > **Save**. You'll want to save it as an Inkscape `.svg` file since that's Inkscape's native format. Give it a descriptive file name such as `turntable.svg` so you can find it later.

And that's it for our illustration! Do you feel good? I certainly do. There's something about the achievement of completing a design project that's fulfilling in a way few other things in life are. Now that we have our first full illustration under our belts, we can move on to more advanced topics with confidence.

Summary

In this chapter, we learned about some basic shapes, the tools to make them, and how to modify their properties. As we saw in our practice exercise, these tools come in quite handy for making vector art and will serve us well as we continue.

While making the turntable, you may have noticed that illustration can involve a lot of manual alignment work to look correct. In the next chapter, we're going to explore some ways to be lazy and make Inkscape do all that tedious alignment work for us.



4

Automatic Shape Alignment in Inkscape

I'm going to share a little secret with you: designers are impatient people. We hate monotonous redundant work that could be better done automatically. Automation leaves us free to do the meat and potatoes of our work, which is creating stuff. So, let's feed those peas we've hidden in our napkin to the dog and make Inkscape align and space stuff out evenly for us automatically.

We will cover the following topics in this chapter:

- The **Align and Distribute** dialog
- Grid arranging shapes
- Circular arranging shapes
- Automatic spacing and alignment with Smart Snapping options

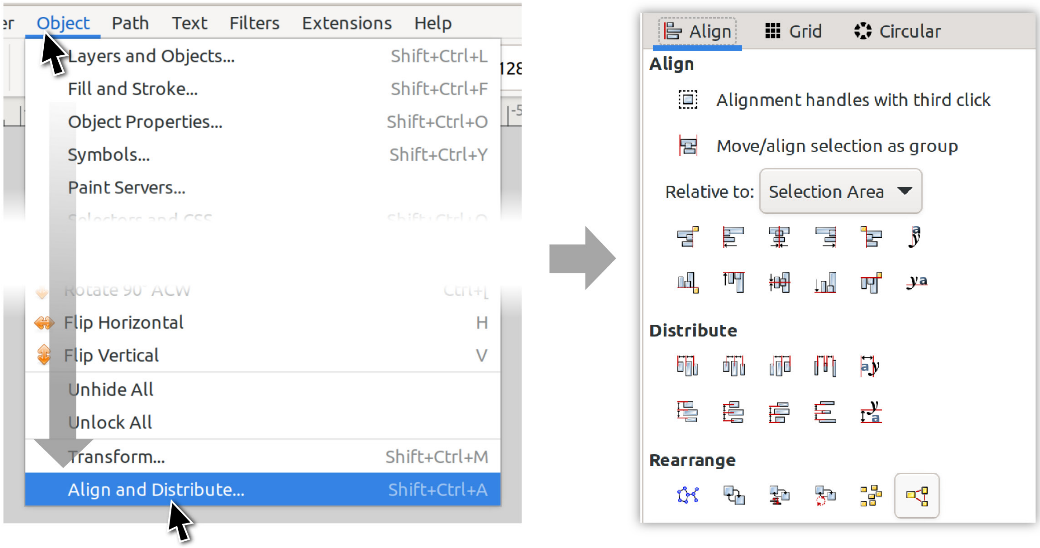
Technical requirements

You can download the practice files for this chapter from GitHub at

`https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape/tree/main/Chapter04`

The Align and Distribute dialog

If I may draw your attention to the **Align and Distribute** dialog, which can be summoned by choosing **Object** > **Align and Distribute** or by hitting the magic *Ctrl* + *Shift* + *A* hotkey combination, as shown in *Figure 4.1*:


Object
Path
Text
Filters
Extensions
Help
Layers and Objects... Shift+Ctrl+L
Fill and Stroke... Shift+Ctrl+F
Object Properties... Shift+Ctrl+O
Symbols... Shift+Ctrl+Y
Paint Servers...
Rotate 90° ACW Ctrl+[
Flip Horizontal H
Flip Vertical V
Unhide All
Unlock All
Transform... Shift+Ctrl+M
Align and Distribute... Shift+Ctrl+A
Align
Grid
Circular
Align
Alignment handles with third click
Move/align selection as group
Distribute
Rearrange

Relative to: Selection Area

Figure 4.1 – The Align and Distribute dialog

Here, we have a lot of options, but if you look closely at each of the icons, you will see they do exactly what they indicate – the **Align** section aligns all selected shapes together, while the **Distribute** section equalizes the spaces between shapes in different ways.

Let's start with the **Align** section and see some practical examples.

Say you have a building and some window washers. You'd like to align the window washers to the side of the building so that they aren't floating out over the street, menacing people with a clear defiance of gravity.

Figure 4.2 shows this scenario and some ways we might use the core **Align** buttons to achieve it:

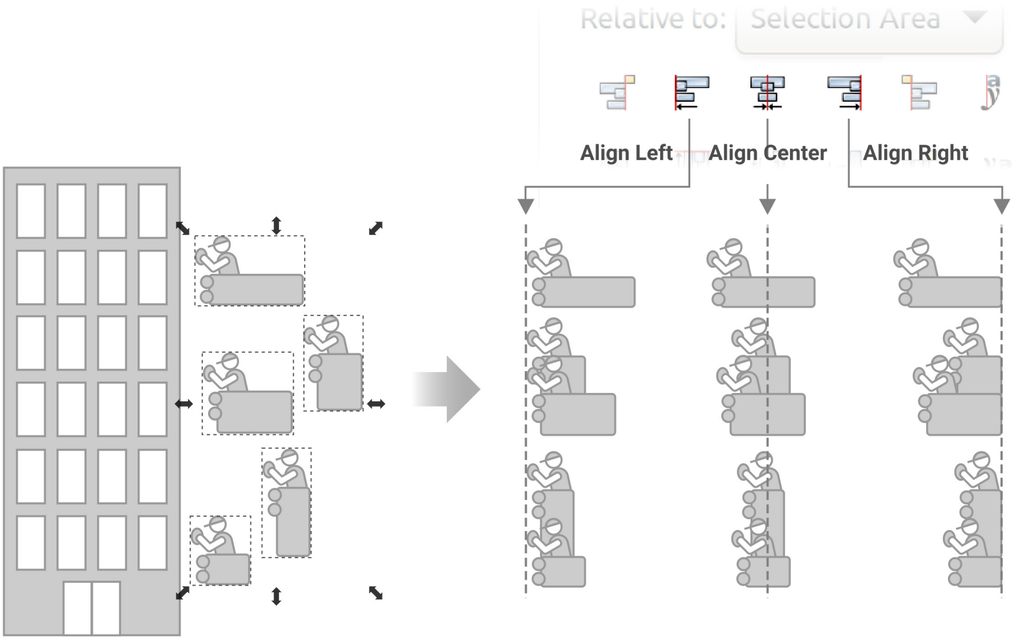


Figure 4.2 – Aligning the window washers with some of the Align options

As you can see, the **Align Left** option gets us the results we want, with all washers appearing to wash the same surface. But now, we have a different problem: it's pretty crowded.

Ideally, we'd like to spread out our window cleaners evenly so that they have room to work and don't get cranky with each other's very oddly shaped elevator boxes. For this, we will be using the **Distribute** buttons, but before we can use them effectively, we need to create more space in our selection by moving the bottom cleaner down a bit more.

You see, unlike the **Align** options, the **Distribute** options always use the selection area to determine the spacing. Thus, if we don't create more space, our washers will still be grumpy and crammed together, albeit evenly so. *Figure 4.3* begins by moving the bottom window washer down, and then shows the results of our various distribution actions on our selection:

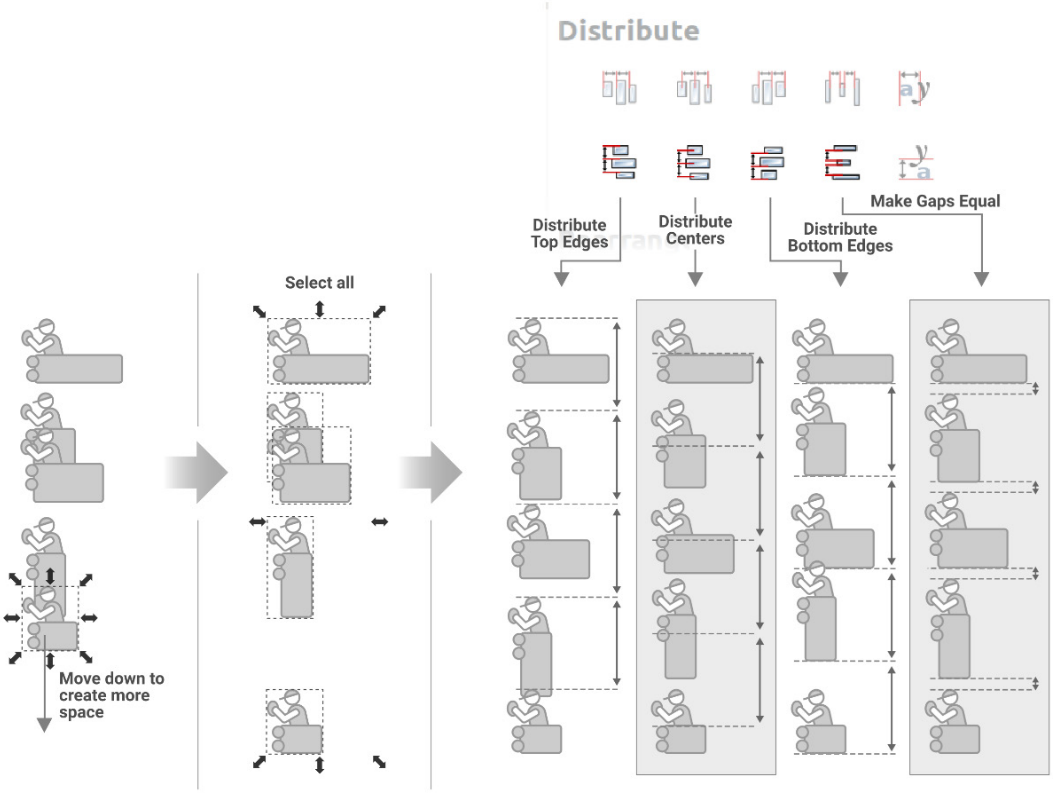


Figure 4.3 – The Distribute options and their results

As you can see, the last option, **Make Gaps Equal**, produces the best results, with each washer getting an equal amount of headroom. You may also notice that this also produces the most visually pleasing results, so much so that you can generally forget about all the other options and just try this one first. Then, if the results aren't quite right for your case, experiment with the others until you get the results you like best.

The other rows of options for **Align and Distribute** have to do with horizontal alignment and distribution, but the effect is the same. 99% of the time, you'll want to go with the **Make Gaps Equal** option. You can also use the **Align horizontal** option and then vertically position a single object to center the object in another object. Simply switch the **Relative to:** dropdown to **Last Selected**. *Figure 4.4* shows an example of aligning a smaller circle to the center of a larger circle using this method:

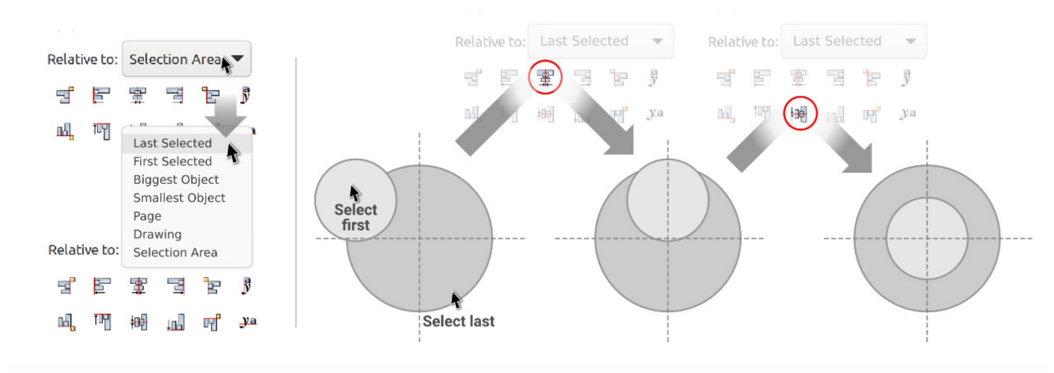


Figure 4.4 – Aligning one object to the center of another

You'll notice that the **Relative to:** dropdown has other options. We could have gotten the same results with **Biggest Object** or **First Selected** if we had selected the large circle first. I generally leave this set to the **Last Selected** option, however, because this is the easiest thing to control. Here's why.

Even if the object you want to align to was not selected last, you can press *Shift* and click it again to remove it from the selection and press *Shift* and click again to add it back to the selection, making it the last selected object. This little trick makes the **Last Selected** option the most versatile option and a good default for most projects. I also recommend **Page** for aligning things to the center of the page, naturally.

Text object alignment

Although we will discuss text in more detail when we discuss the **Text** tool, it's good to know that there are some special options for text objects. As you can see in *Figure 4.5*, these alignment options align different text objects according to the baseline of the text:

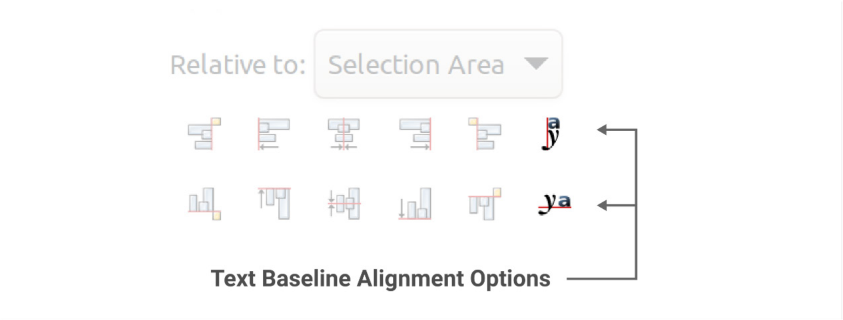


Figure 4.5 – Text baseline alignment options

This is because some characters dip below this line, and more often than not, it's the baseline of the text we want to align with, not the lowest point of a lowercase letter *y*, for example. *Figure 4.6* shows the difference between aligning with the usual bottom alignment button and the text baseline alignment option:

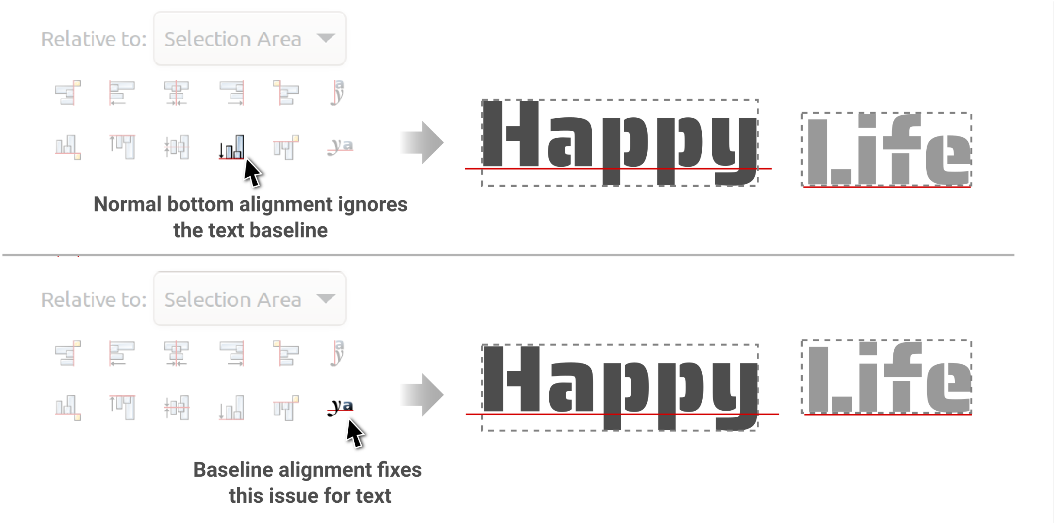

Relative to:
Selection Area
Happy
Life
Normal bottom alignment ignores the text baseline
Relative to:
Selection Area
Happy
Life
Baseline alignment fixes this issue for text

Figure 4.6 – Using baseline alignment to properly align two different text objects

All this works pretty well. But wouldn't it be nice if there were some features to align our shapes on the canvas? After all, we have handles to rotate and resize... Why can't we align with handles too? Short answer – we can!

On Canvas Alignment

Maybe you want to align things on the canvas, rather than clicking on the myriad of alignment buttons. Well, fortunately, Inkscape 1.0 and above features the **On Canvas Alignment** tool. This tool is shown in *Figure 4.7* and can be toggled on and off:

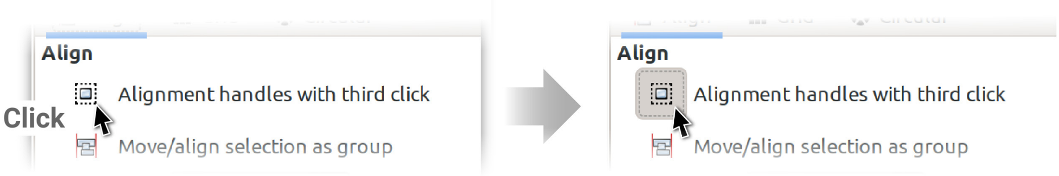

Align
Click
Alignment handles with third click
Move/align selection as group
Align
Alignment handles with third click
Move/align selection as group

Figure 4.7 – Toggling on the On Canvas Alignment option

The **On Canvas Alignment** tool adds one more mode to the **Selection** tool:

1. Click once to get the **Resize** handles.
2. Click twice to get the **Rotate** handles.
3. That third click will now get you **Alignment** handles!

How cool is that? As you can see from *Figure 4.8*, the handles are different:

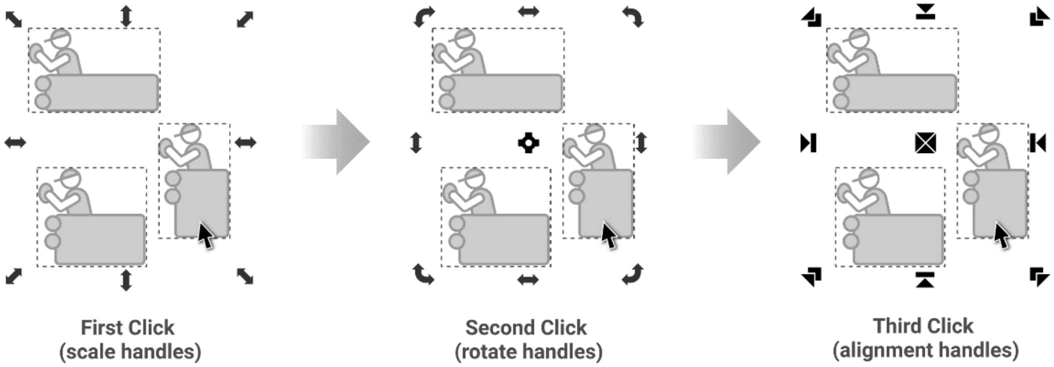


Figure 4.8 – Clicking three times on the selected group with On Canvas Alignment toggled on results in fancy new handles

As you would expect, clicking on a handle aligns the selected objects to that side of the selection. Click left to align left, and click right to align right. The top and bottom are no surprise either, but what about the middle?

By default, clicking the middle icon aligns to the center horizontally – maybe you'd rather align vertically though. To do this, simply hold down the *Shift* key while you click. Neat, huh? This is very handy when you have lots of alignment to do. That *Shift* key also does things for the other alignment handles – it aligns the objects on the opposite side of the selection.

For example, say your window washers are floating conspicuously over your building instead of on the side. *Figure 4.9* shows what happens if you hold *Shift* while clicking the right alignment handle to make them align to the outside of the building where they should be:

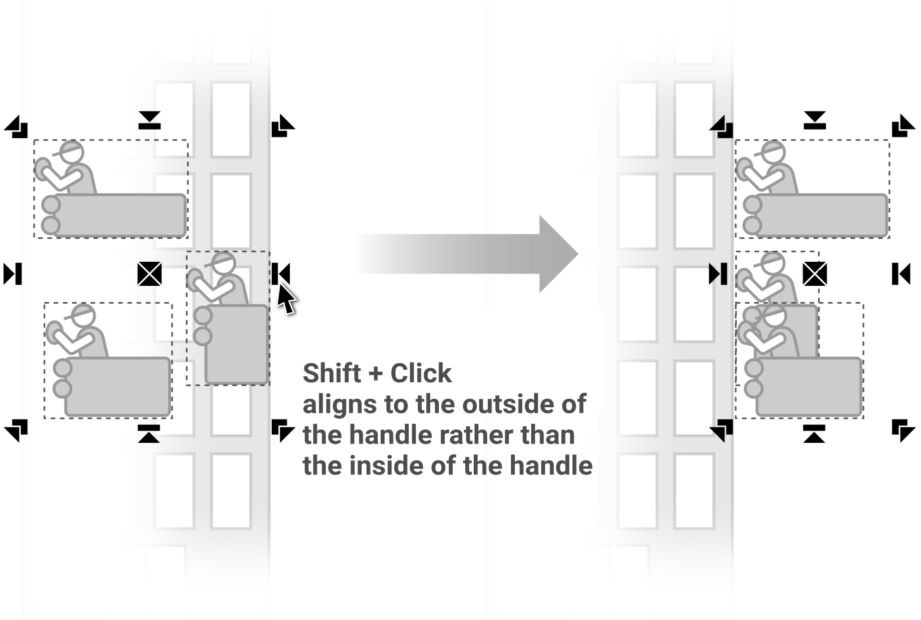


Figure 4.9 – Using the Shift key to align to the opposite side of the alignment handle

Once you get the hang of using it, you'll wonder how you ever got along without it. It's one more click in the select/rotate/align chain though, so it's also nice to be able to toggle it off when you're not using it to avoid extra clicks during your resizing and rotating activities.

Rearranging options

Below the **Alignment** icons, there's a mysterious row of somewhat cryptic icons for rearranging our objects. These can be handy for swapping things around, so let's briefly have a look at some of the options via real-world examples.

Say you have several fancy logos you've knocked out for a client. You want to make it easy for them to pick their favorite of the bunch, so you've done them the favor of numbering them, as shown in *Figure 4.10 (left)*. Only, after all that numbering and careful alignment of each subsequent number into the top left of each sample box, you realize that you missed a number! Now, you're going to have to go back and re-number everything, right?

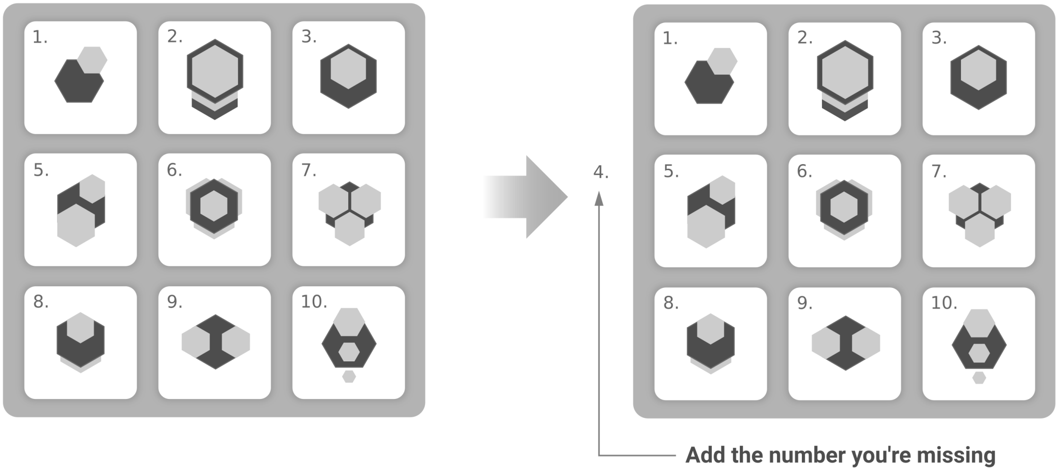


Figure 4.10 – The missing number 4 in our logo sheet

We forgot the number *4*. We'd like to shift everything over one and kick out our *10* instead. Well, our friend, the **Rearrange by Selection Order** option is here to help us. All we need to do is add the number we missed somewhere on the canvas (*Figure 4.10 (right)*), select each number we want to shift in reverse order, and then click the **Rearrange by Selection Order** button! *Figure 4.11* shows us what this looks like:

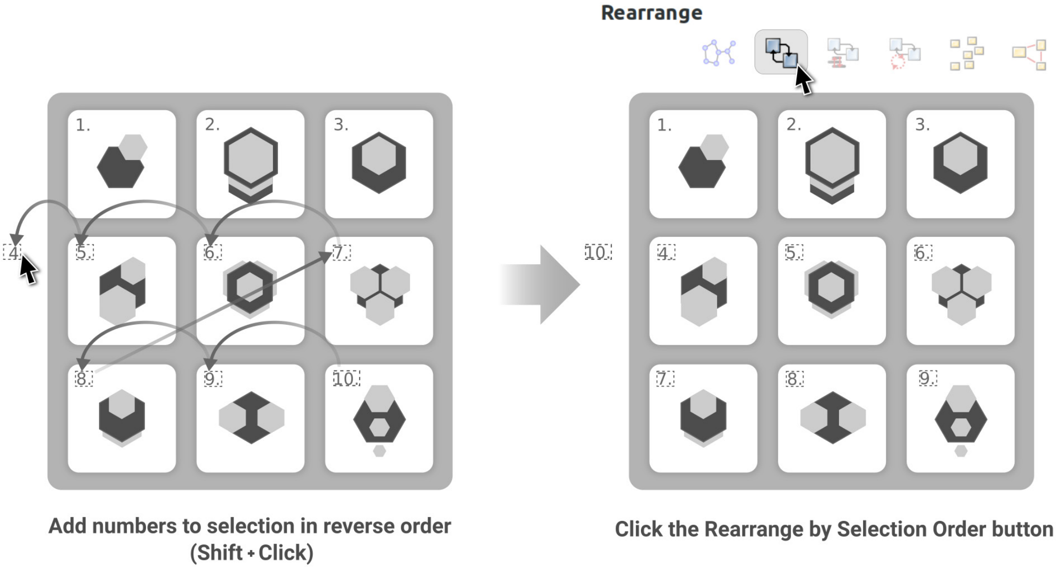

Rearrange
1.
2.
3.
4
5.
6.
7.
8.
9.
10.
10.
1.
2.
3.
4.
5.
6.
7.
8.
9.
Add numbers to selection in reverse order (Shift + Click)
Click the Rearrange by Selection Order button

Figure 4.11 – Using the Rearrange by Selection Order option to shift our numbers and give 10 the boot

You can see how this could save us potentially a lot of time, depending on how many things we have numbered.

The other options for rearranging are somewhat less useful. However, for completeness, let's go over them. *Figure 4.12* shows the other options numbered:

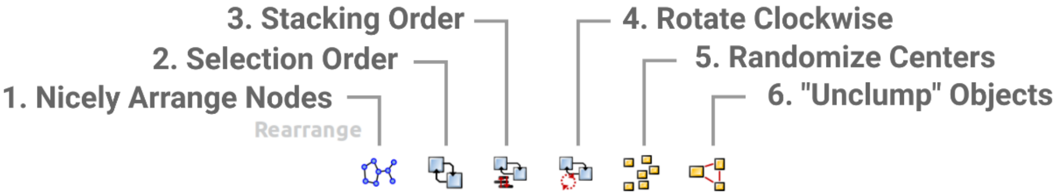


Figure 4.12 – All our rearrange options

We will briefly discuss each and what you might potentially use them for:

1. **Nicely Arrange Nodes**: This button takes a selected diagram network, made with the **Connector** tool, and makes your diagrams look nice without the need to move all the parts of it around manually, as demonstrated in *Figure 4.13*:

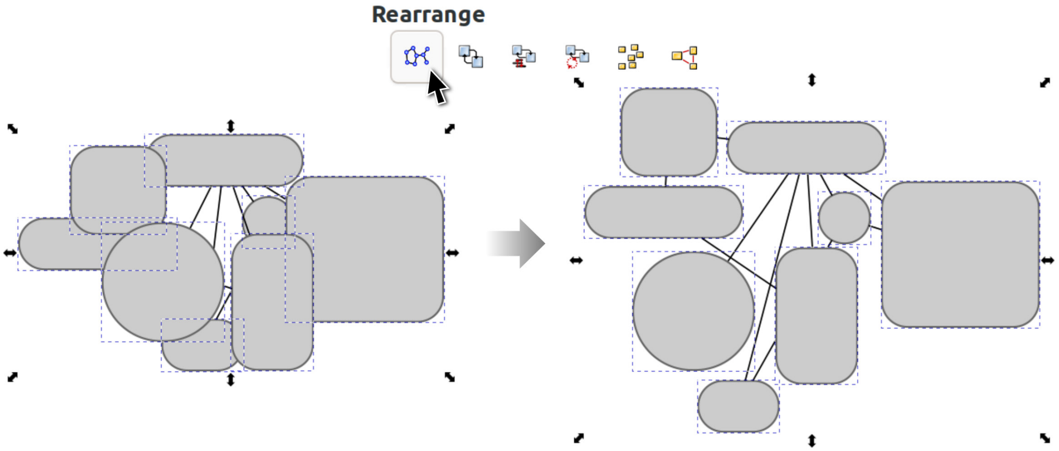


Figure 14.13 – A demonstration of a connector arrangement

2. **Selection Order**: Yep, this is the same one we just used to renumber our logos. Select your numbers in reverse order and tap this button. This is simple, effective, and likely the only one you will need regularly.
3. **Stacking Order**: Inkscape can use the stacking order of your shapes to determine the order to rearrange. **Stacking Order** arrangement allows you to control the order by rearranging items in the **Layers and Objects** dialog, by moving shapes on top or underneath one another. We'll learn more about this dialog in *Chapter 11*, *Organization Using Layers*.
4. **Exchange Around Center**: We could have used this one as well to solve our numbering problem, as our mistake occurred conveniently at the beginning of the second row. If you have shapes aligned around a circle, or a big square, or something, this could save you some trouble, as shown in *Figure 4.14*:

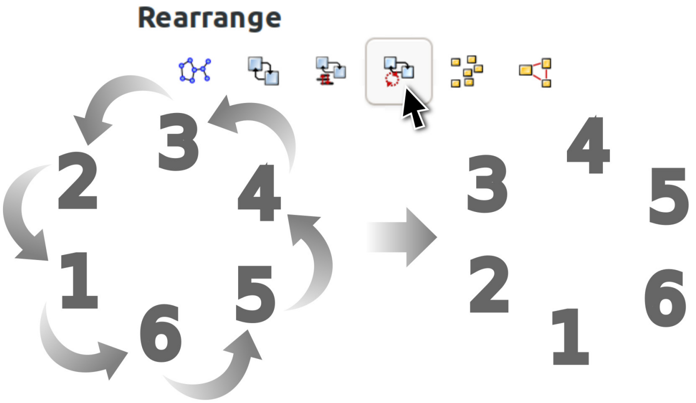


Figure 4.14 – Exchanging the positions of six objects around the center point

5. **Randomise Centers**: Maybe we don't want to reorder our shapes. You can easily add some disorder to them as well with this option. Select objects, press the **Randomise Centers** button, and watch your objects dance around in no particular order. You can then space them out evenly in the **Remove Overlaps** section just below, adding numbers for **H:** and **V:** (horizontal and vertical gaps, respectively) and clicking the **Remove Overlaps** button for a more satisfying result, as shown in *Figure 14.15*:

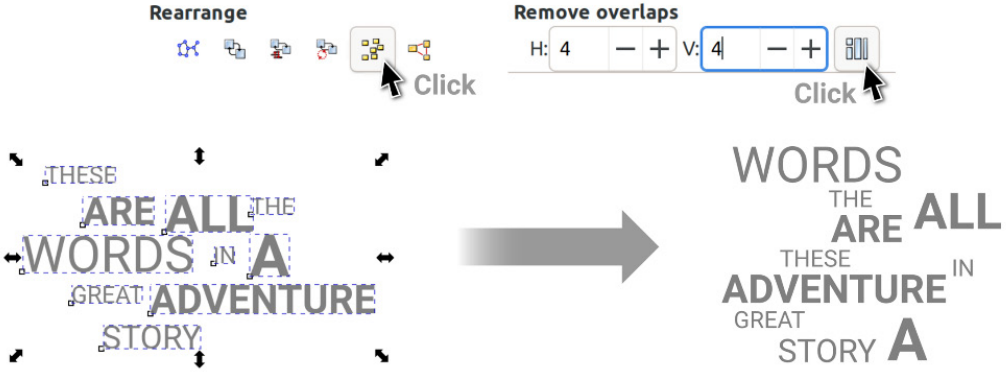


Figure 14.15 – Using Randomise Centers and Remove Overlaps to rearrange a word cloud

6. **Unclump Objects**: This doesn't *unclump* anything in the way you'd probably think. I've personally never once gotten it to satisfyingly *unclump* overlapping or close objects, for example. What it does is attempt to equalize the distance between the centers of your selected objects, by shifting them around. A much better option is to use the **Remove Overlaps** section, as previously demonstrated in *Figure 14.15*.

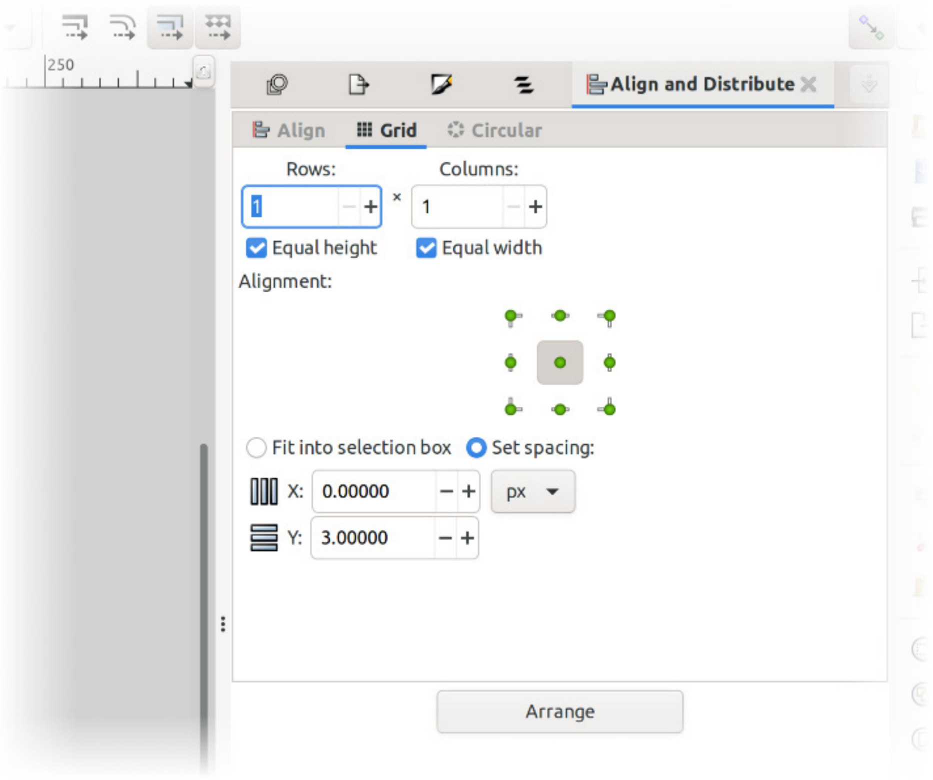
So, the takeaway here is that the clear winner of the **Rearrange** beauty pageant set is **Selection Order** as it's the option where you have the most direct control over the results.

This is all well and good for aligning the rows and columns of shapes, but what if we want to arrange our shapes in rows and columns at the same time? We might have to do many alignment operations to get the effect we want. Fortunately, we don't have to! Inkscape will do that for us.

Grid arranging shapes

In the previous section, we learned how to manually align objects to each other, but let's say we want to make a whole grid of them. Take this as an example: we'd like to display a grid of products – in this case, coffee and snacks for one of those trendy coffee shop menus.

We want to spend more time selling coffee and less time arranging things, so we'll use Inkscape's **Grid** arrange tab (formerly the **Arrange** dialog), which as of Inkscape 1.1 is now located in the very same **Align and Distribute** dialog we've been using (see *Figure. 4.16*):



Rows: 1 × Columns: 1

☑ Equal height ☑ Equal width

○ Fit into selection box ● Set spacing:

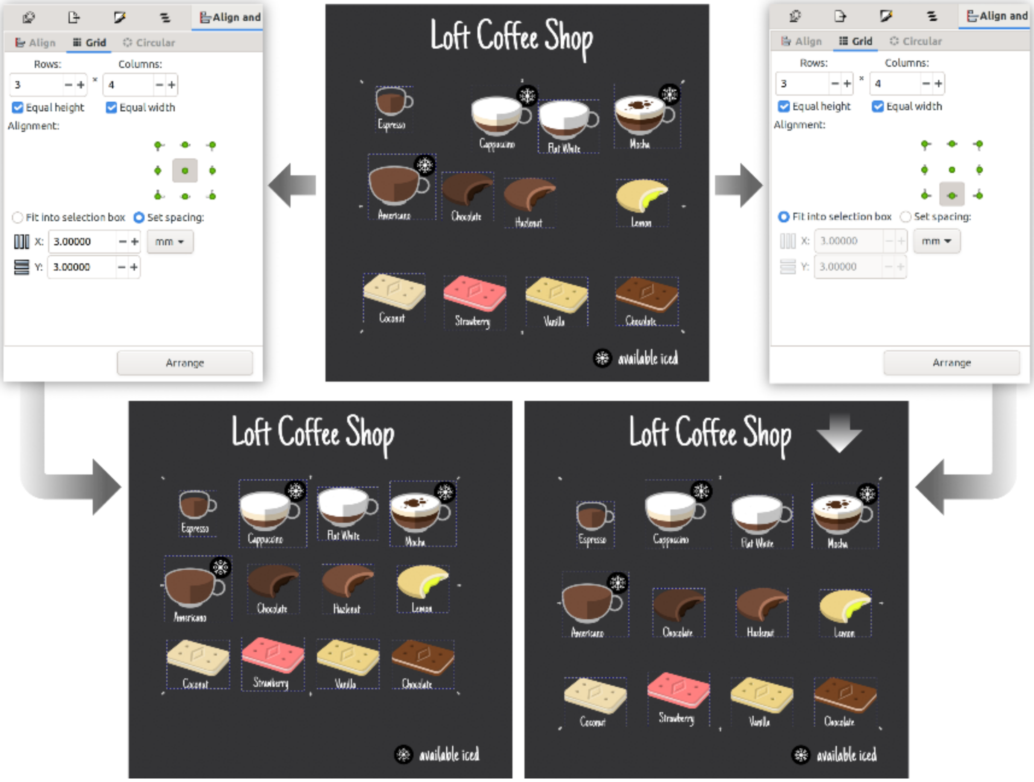
X: 0.00000 px

Y: 3.00000

Figure 4.16 – The Grid Arrange tab

With all your item graphics selected, the **Grid** tab allows you to set the number of **Rows** and **Columns** you want, the kind of spacing between items (**Equal height** and **Equal width**), which point on the objects to align to (we'll see why this is useful in a moment), and whether to **Set spacing** of the gaps between graphics to a specific measurement or **Fit into selection box**, which will save you from having to estimate the size of the gaps.

Figure 4.17 (left) shows what happens if we set the **Alignment** value to the center, and manually set our spacing to `3 mm`. It's better than before but notice how the text titles of each item aren't aligned. These graphics were made to be aligned by the bottom, so let's set the alignment to the bottom, as shown in *Figure 4.17*:



Rows: 3 Columns: 4
☑ Equal height ☑ Equal width
○ Fit into selection box ● Set spacing:
X: 3.00000 mm
Y: 3.00000

Rows: 3 Columns: 4
☑ Equal height ☑ Equal width
● Fit into selection box ○ Set spacing:
X: 3.00000 mm
Y: 3.00000

Figure 4.17 – Arranging the coffee shop menu graphics with the Grid Arrange tab

The whole collection of items is off to the right a bit as well, meaning we guessed wrong about the spacing between graphics. We also save ourselves some re-estimating by ticking the **Fit into selection box** radio button.

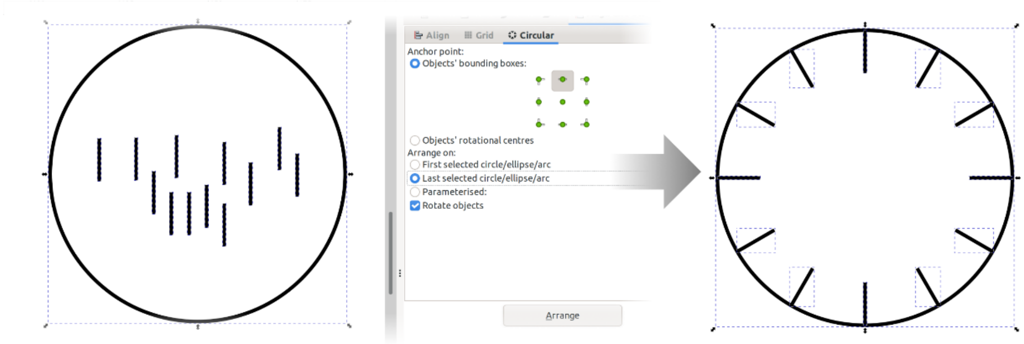
As you can see, the results are much better. We only need to manually adjust a few of the items (**Cappuccino** and **Strawberry**) that have letters that dip below the text baseline, and we're golden!

Grid arrangement is quite useful for a variety of tasks and will serve us quite well in designing with Inkscape. However, sometimes, we'd like our objects to be arranged in other ways – say, in a circle, for example.

Circular arranging shapes

Grids are all well and good, but what about a circle? That may seem like an odd pattern to arrange most of the time, but say you want to make a clock graphic or watch face. You need to arrange not only the numbers in a circle, but also all the hour, minute, and maybe even second markers. That's a lot of arranging!

So, let's pop over to the **Circular** tab in our **Align and Distribute** dialog. Starting with a circle and 12 dashes for hour markers, as shown in *Figure 4.18*, we can set **Anchor Point** to **Objects' bounding boxes** and select the top center green dot (the default is the center). This will fix the top of each line to the circle:

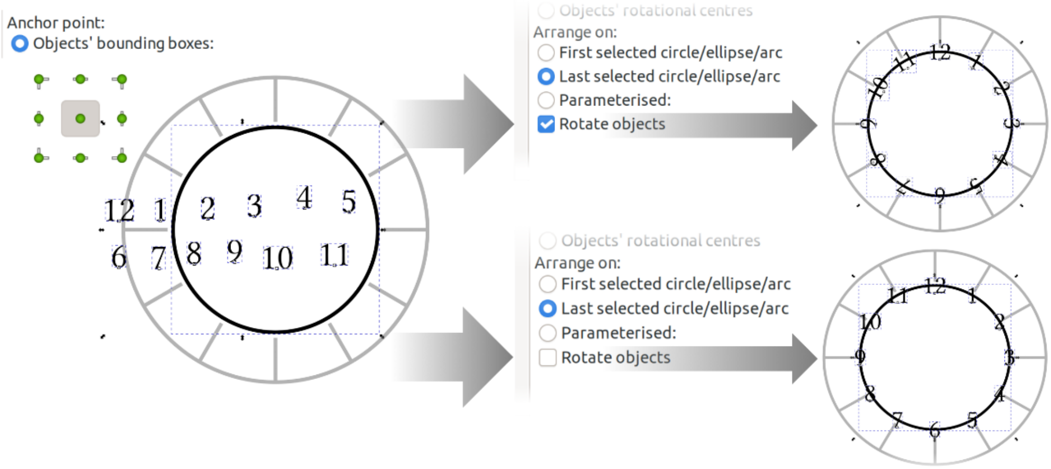


Anchor point:
● Objects' bounding boxes:
○ Objects' rotational centres
Arrange on:
○ First selected circle/ellipse/arc
● Last selected circle/ellipse/arc
○ Parameterised:
☑ Rotate objects

Figure 4.18 – An illustration of a circular arrangement

The **Arrange on:** options don't matter here since we only have one circle, but say we want to arrange a bunch of small circles on our large circle. These options help Inkscape decide which of the circles to arrange. I typically leave this set to **Last selected** since, as noted previously in this chapter, it's easy to choose which shape is the *last* in the selection simply by deselecting it and re-selecting it as part of the selection.

Finally, the **Rotate Objects** setting will automatically rotate our lines to face inwards toward the center of the circle. As you can see in *Figure 4.19*, we turn this off when arranging our numbers in a circle, lest we wind up upside-down at 6 o'clock. We'll also set **Anchor Point** to the center position:



Anchor point:
● Objects' bounding boxes:

○ Objects' rotational centres
Arrange on:
○ First selected circle/ellipse/arc
● Last selected circle/ellipse/arc
○ Parameterised:
☑ Rotate objects

○ Objects' rotational centres
Arrange on:
○ First selected circle/ellipse/arc
● Last selected circle/ellipse/arc
○ Parameterised:
☐ Rotate objects

Figure 4.19 – Circular arrangement order with numbers

Note that the selection order and rotation of the circle matter as to where the numbers start and end. As we have seen when playing with ellipses and arcs, Inkscape's circles start and end at the right-hand side by default (where the arc handles are).

This is the point where Inkscape will start aligning our numbers, so we must rotate the circle 90 degrees anti-clockwise (counter-clockwise) to move the start of the circle to the top (**Object** > **Rotate 90 ACW**).

Then, we must select the numbers in the order we want them to appear, in this case starting with 12, then 1, 2, 3, 4, 5, and so on. So, if your numbers aren't in the right places, you've likely forgotten one of these steps.

We've now covered a variety of useful ways to arrange and align things automatically, so let's get some solid practice with another illustration. Since there are many more parts to this one than we've seen in previous chapter illustrations, it will quickly become apparent how valuable it is to have Inkscape automatically align the objects for us.

Automatic spacing and alignment with Smart Snapping options

You may remember in the first chapter when we briefly looked at the snapping options that there were alignment options too. Let's revisit this and take a closer look at those options and how they work. Recall that the snapping toggle has a little arrow next to it, and we can toggle on/off the alignment options as shown in *Figure 4.20*:

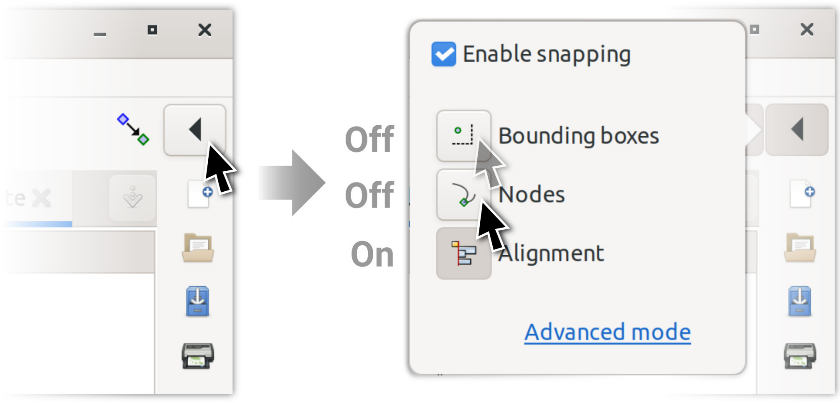


Figure 4.20 – Toggling off the Bounding Boxes and Nodes options

Normally, we'd just leave these toggled on, but for the sake of demonstration, we'd like to see the effect of *just* the alignment options. What we find is that when we have multiple shapes in a row, we can add more with the same spacing as the others, which Inkscape will helpfully detect to show snapping and alignment distances matching similar gaps it's found. *Figure 4.21* shows what happens when we drag a third rectangle closer to two others from the right:

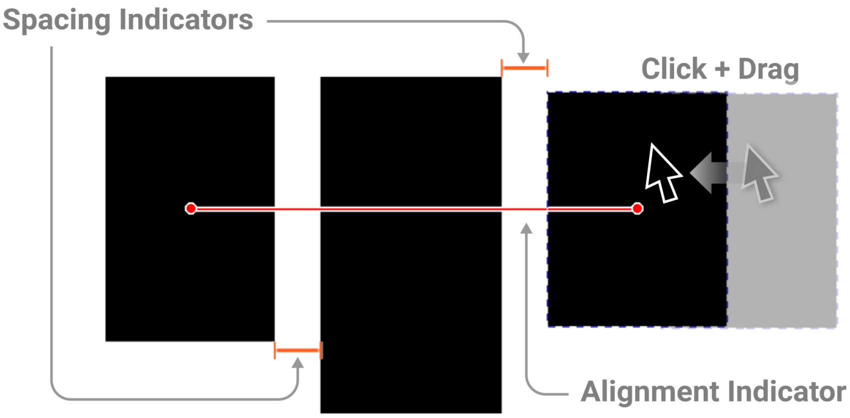


Figure 4.21 – Spacing and alignment indicators appearing while dragging

When these spacing and alignment indicators appear, you can release the mouse drag to snap the shape where the indicators… indicate.

In the **Advanced** options, there are two checkboxes – one to enable **Object snapping** and another to enable **Node snapping**, which is handy when you're working on custom shapes. We will explore more about the nodes and the **Node** tool in the next chapter. For now, let's get some practice!

Try it yourself

Remember that cool coffee shop menu we were aligning earlier? Let's add an item to the menu by illustrating a glass of juice. We'll use some simple shapes and our trusty new set of alignment tools.

Figure 4.22 shows the parts we need to make, and how they will fit together to make the result. If you need a refresher on how to make these shapes, check out *Chapter 3*, *Drawing Shapes with the Shape Tools*:

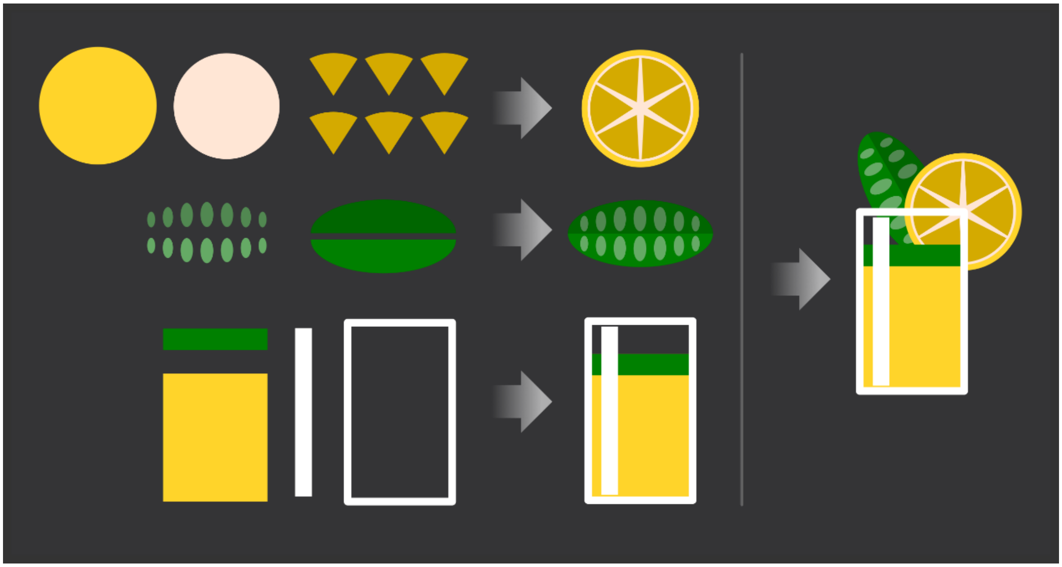
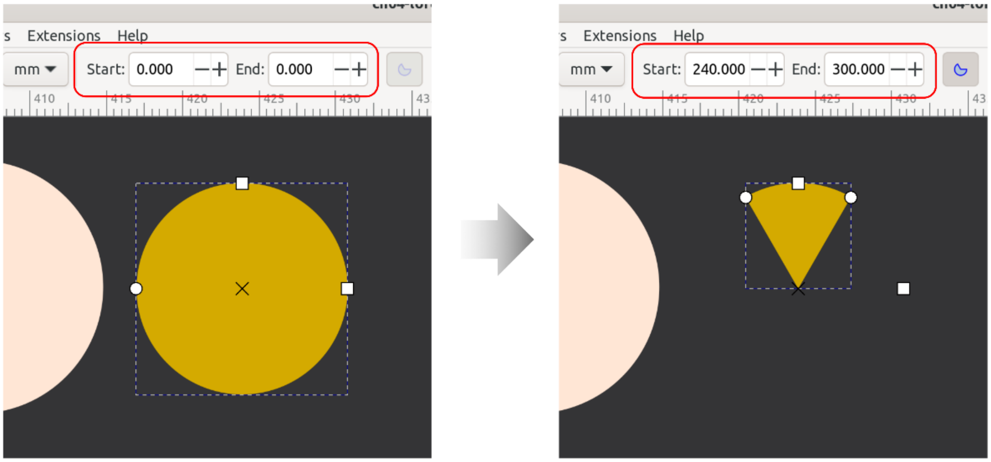


Figure 4.22 – The parts and final assembly of our practice drink illustration

Let's get right into it!

1. Draw the lemon slice circles and wedges with the **Circle and Ellipse** tool. To make this wedge, start by drawing a circle that is smaller than the beige circle, and use the circle **Start** and **End** values in the Tool Control bar to make that circle into a wedge.
2. Enter `240` for **Start** and `300` for **End**, as shown in *Figure 4.23*:



Start: 0.000 End: 0.000

Start: 240.000 End: 300.000

Figure 4.23 – Start and End values in the Tool Control bar for the lemon wedge

3. Before we make copies of the wedge, we need to move the rotational center up to the edge. This will allow us to use the **Circular** alignment dialog to rotate and align these into place for our lemon segments in *Step 5*.

4. With the **Select and Transform** tool active, select the wedge, then click again to see the rotation handles and rotation center handle. Drag the rotation center handle up to the top edge of the wedge, as shown in *Figure 4.24*:

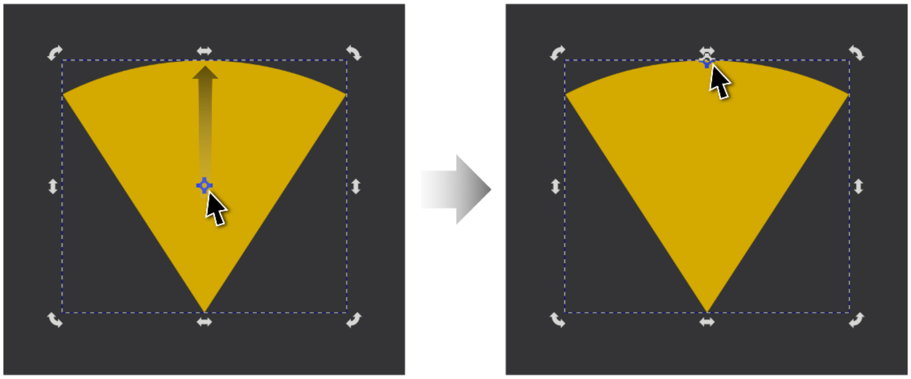
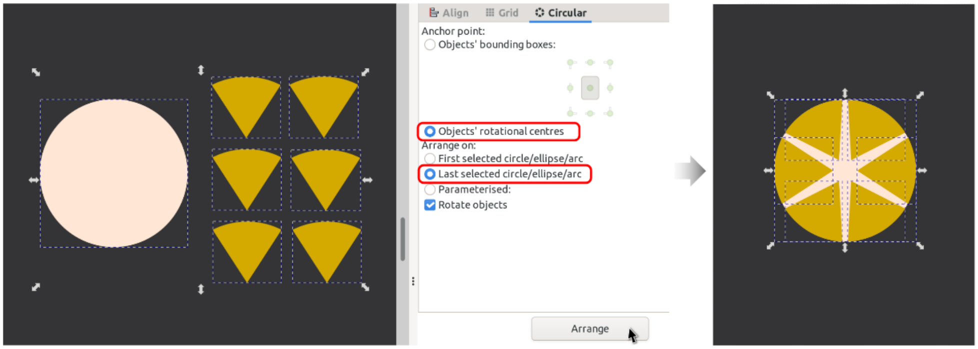


Figure 4.24 – Moving the rotational center to the top of the wedge

You can now duplicate the wedge, making five copies, and the new copies will also have the same rotational center at the edge where we moved it.

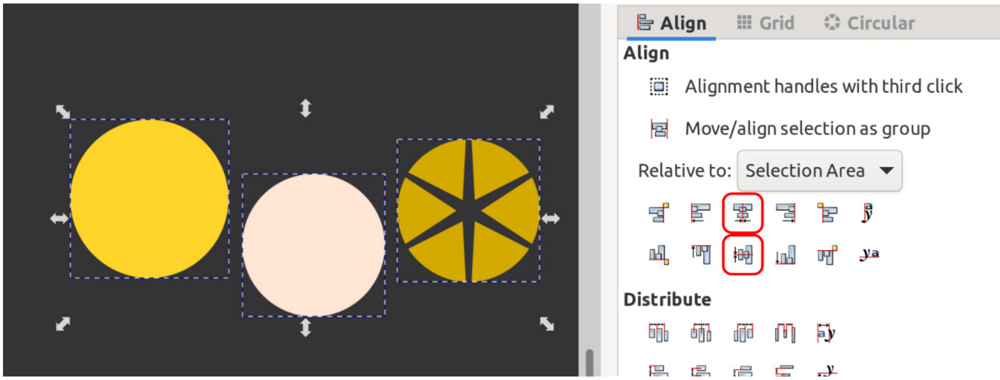
5. Now that we have our six wedges, we're going to use the **Circular** tab in the **Align and Distribute** dialog (**Object** > **Align and Distribute**). Select all six of your wedges and tick the **Objects' rotational centers** and **Last selected circle/ellipse/arc** radio buttons.
6. Then, box-select all your shapes and press *Shift* and click to deselect the circle. Then, click again to re-select it so that it is the last selected circle.
7. Click the **Arrange** button and you'll have your slices all put into a nice radial pattern around the circle, as shown in *Figure 4.25*:



Anchor point:
○ Objects' bounding boxes:
● Objects' rotational centres
Arrange on:
○ First selected circle/ellipse/arc
● Last selected circle/ellipse/arc
○ Parameterised:
☑ Rotate objects

Figure 4.25 – Using the Circular Arrange tab to rotate and align the wedges to our circle

8. Deselect the circle by pressing *Shift* and clicking in the center of it, leaving only the slices selected. We're going to group these slices by choosing **Object** > **Group** so that we can then align them with the two circles.
9. Select the two circles and the group of slices and click the vertical and horizontal align buttons in the **Align and Distribute** dialog to complete the lemon, as shown in *Figure 4.26*:



Alignment handles with third click
Move/align selection as group
Relative to: Selection Area

Figure 4.26 – The Align and Distribute options for centering the lemon segments

10. As an optional last thing, you can select **Object** > **Group** and group them all together to complete your lemon slice illustration, for easier movement later. *Figure 4.27* shows the combined effect:

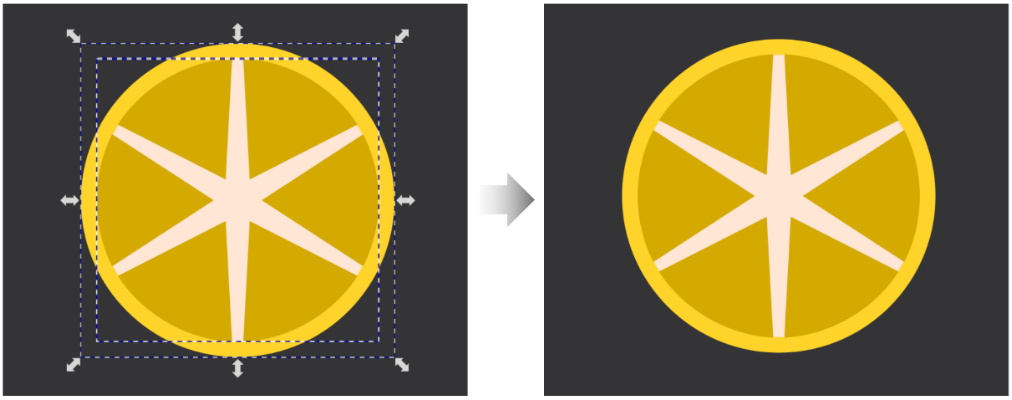


Figure 4.27 – Grouping the parts of the lemon for ease of selection and moving

11. Moving on to our leafy garnish, we'll create some more ellipses, two half ellipses, and a bunch of little ovals, as seen in *Figure 4.28*:

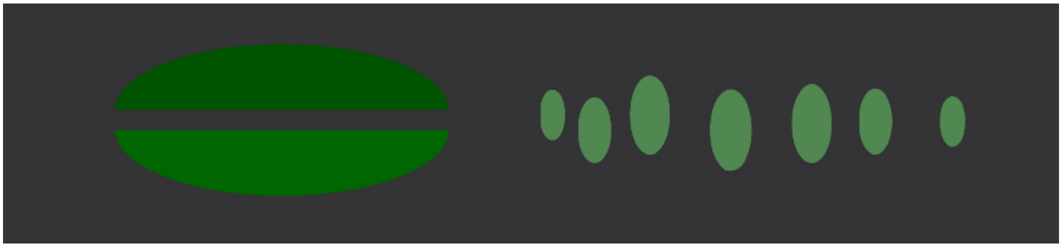
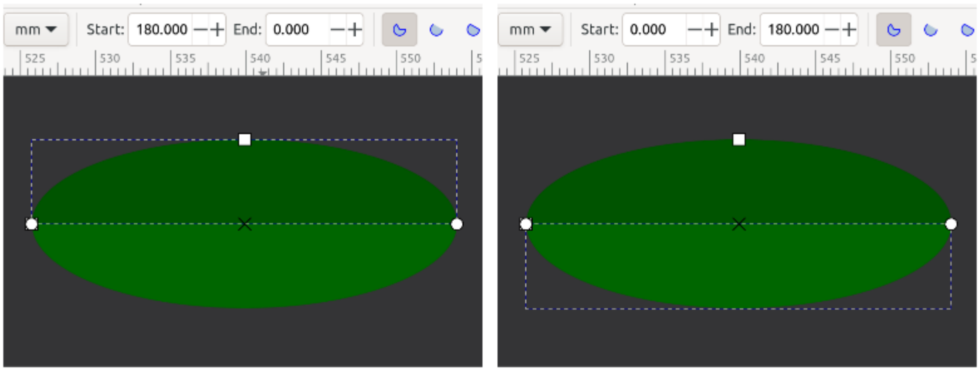


Figure 4.28 – Our leaf illustration parts

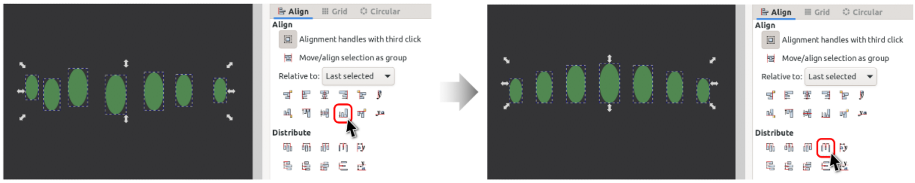
12. Let's make that leaf oval by drawing our elongated circle and setting the **Start** and **End** values like before. However, this time, we will just enter a value of `180` for **End**, leaving **Start** at `0`.
13. Duplicating this oval with **Copy** and choosing **Edit** > **Paste in Place** (or pressing *Ctrl* + *Shift* + *V*) will give us a copy of the oval directly over the first one. We can then change the **Start** and **End** values to `180` and `0`, respectively, creating the other half of our leaf and assigning it a slightly darker fill. Both are shown in *Figure 4.29* with different **Start** and **End** values:


mm Start: 180.000 End: 0.000

mm Start: 0.000 End: 180.000

Figure 4.29 – Start and End values for each half of our leaf

14. For the leaf details, we'd like the small ovals to be aligned to the bottom and then evenly spaced by evening out the gaps. Thus, we must select our row of ovals and click the **Align bottom Edges** button, followed by the **Distribute horizontally with even horizontal gaps** button, then group the ovals (**Object** > **Group** or press *Ctrl* + *G*). See *Figure 4.30* for an illustration of this process:


Relative to: Last selected

Align
Grid
Circular
Align
Alignment handles with third click
Move/align selection as group
Distribute

Figure 4.30 – Aligning and spacing our leaf ovals

15. Since our upper row is a mirror image of our bottom row, we will copy/paste it and tap the *V* key to flip it vertically. Assembling the parts results in our leaf, which we can then group, as shown in *Figure 4.31*:

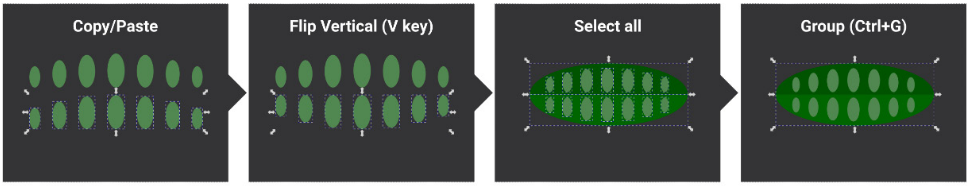


Figure 4.31 – Finishing our leaf

16. To make our glass, we're using rectangles. Let's toggle on the **On-canvas alignment** option and get some practice using it. Select all your rectangles and turn on the **Alignment handles with third click** button, as shown in *Figure 4.32*:

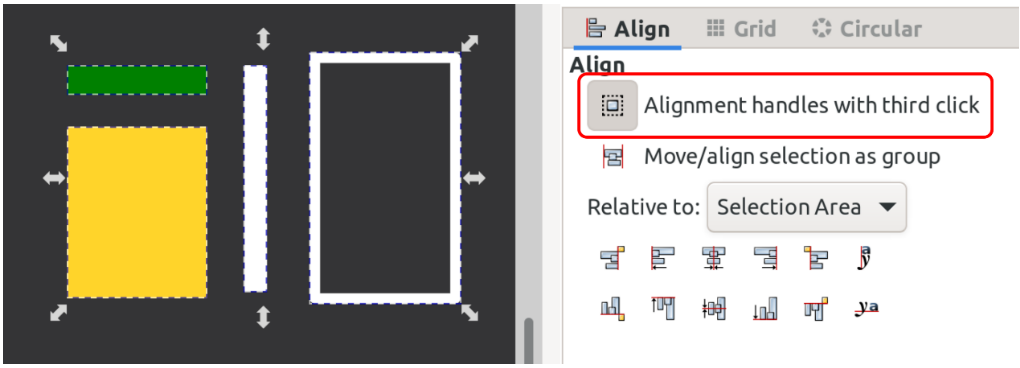

Align
Grid
Circular
Align
Alignment handles with third click
Move/align selection as group
Relative to:
Selection Area

Figure 4.32 – Turning On-canvas alignment on

17. With this button activated, you can get the alignment handles with the third click:

    I. Click one selects the shape(s).

    II. Click two changes handles to rotation.

    III. The third click gives us our on-canvas alignment handles.

18. We can then hold the *Shift* key and click the center alignment handle in the middle of the selection (it will align horizontally if you don't hold the *Shift* key). This will align the rectangles vertically to the center, as shown in *Figure 4.33*:

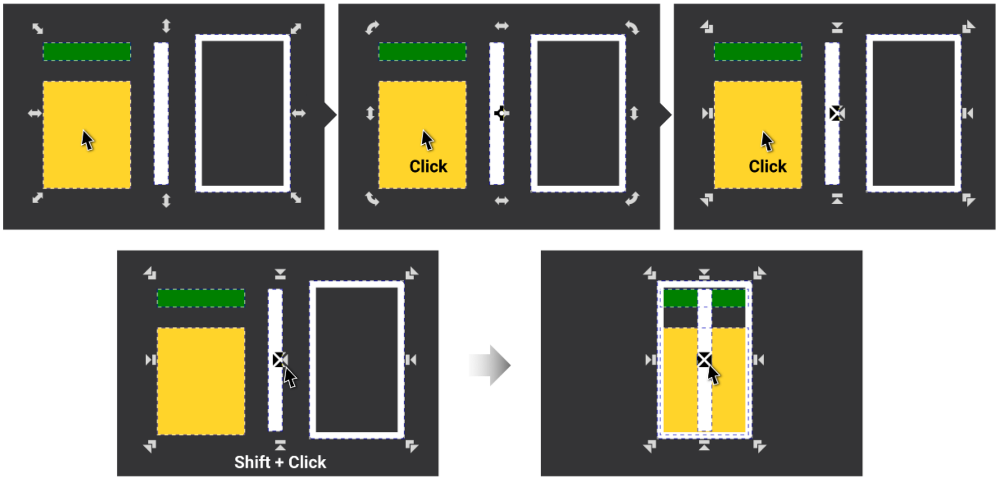


Figure 4.33 – Getting our alignment handles and using the center handle to center-align

19. All that remains is to assemble our three illustrations – lemon, leaf, and glass – into the lovely lemon-lime drink. If you were diligent about grouping your objects, this task is much easier. Just move each part into place and one final grouping of the three will give us our final result (*Figure 4.34*)!

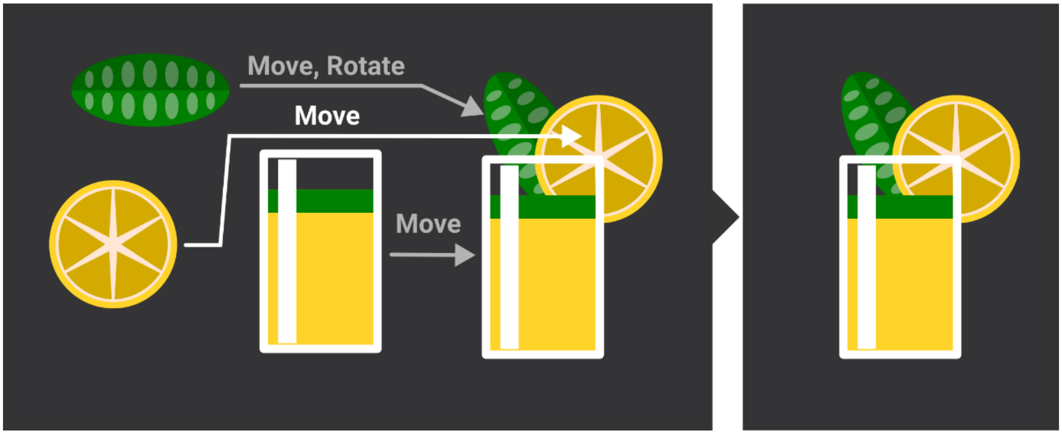


Figure 4.34 – Assembling our final drink illustration out of the parts

That wraps it up for our little practice lesson. Congratulations on completing your second illustration with Inkscape!

Summary

In this chapter, you've seen a truly epic quantity of handy alignment options and tools to sling your shapes around in visually pleasing ways. We went through the **Alignment and Distribute** dialog, which lets you align and space single lines of objects in just about any way you want, and then concluded with other options, such as arranging objects in a grid and around a circle.

With this time-saving skill set under our belts, we will move on to advanced object editing using the **Bezier Curve** tool, the **Node** tool, and the **Freehand Line** and **Calligraphy Brush** tools in the next chapter.

5

Node Editing – Modifying Your Shapes with Nodes and Curves

Up to this point, we've learned how to make a few illustrations with the shape tools. This is somewhat limiting, however. Often, we want to make completely custom shapes. This chapter is dedicated to a variety of ways to make and edit shapes to get them just the way we want them to be.

We will cover the following topics in this chapter:

- Editing shapes with the **Node** tool
- Curve handles and node types
- Breaking and joining lines
- Drawing lines and shapes from scratch
- Line styles (**Stroke Style** menu)
- Advanced curve and line options
- Making brush strokes with Stroke Shape

Editing shapes with the Node tool

Fortunately, Inkscape allows us to make virtually any shape we please with the **Edit Paths by Nodes** tool, which we will refer to from here on out as just the **Node** tool.

Before we can edit our shapes, however, we need to convert them from objects into paths, which we can then edit node by node. Take the glass we made in the previous chapter, for example – maybe we'd like to make it a bit narrower at the bottom, as shown in *Figure 5.1*:

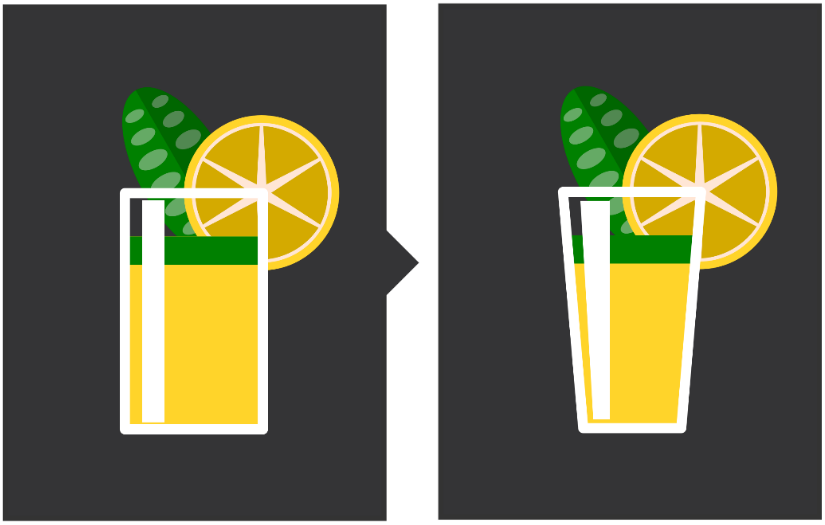


Figure 5.1 – Before and after versions of our glass

Let's begin by selecting the **Node** tool from the Toolbar area and selecting our glass. We'll ungroup the glass by pressing *Ctrl* + *Shift* + *G* or selecting **Object** > **Ungroup**, then select our glass rectangle object. Note that we have our corner rounding handles still, which are object properties. Converting from **Object** into **Path** lets us edit all the nodes instead of just the object properties, and to do that, we simply click the **Object to Path** button in the Tool control bar, as shown in *Figure 5.2*:

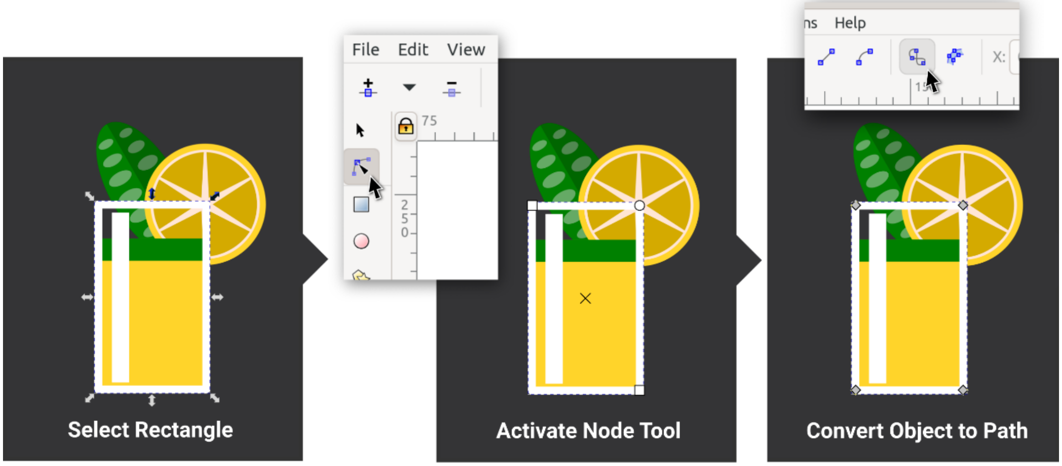


Figure 5.2 – Converting an object/shape into a path

As you can see, we now have four corner nodes that we can click-drag to resize the bottom of our glass. However, we'd like to resize the bottom of the glass and liquid evenly on both sides from the center of the glass. We can do this by first selecting both the glass and liquid rectangle, then pressing *Shift* and selecting, or dragging and selecting a box around the bottom nodes.

We get scale-transform handles with which we can resize the line between the nodes. We can hold the *Shift* key to rescale from the center (just as we did while resizing shapes) and narrow the base of our glass from both sides at once, as shown in *Figure 5.3*:

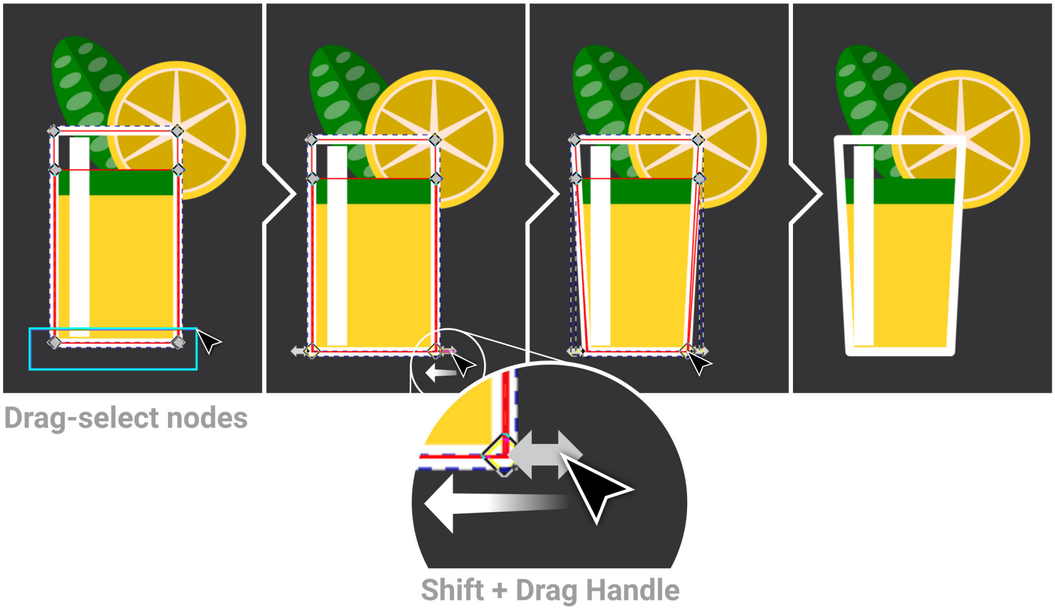


Figure 5.3 – Resizing the bottom of the glass and liquid

If you don't see the handles to resize, make sure the **Show Transform Handles** toggle is on, as shown in *Figure 5.4*:

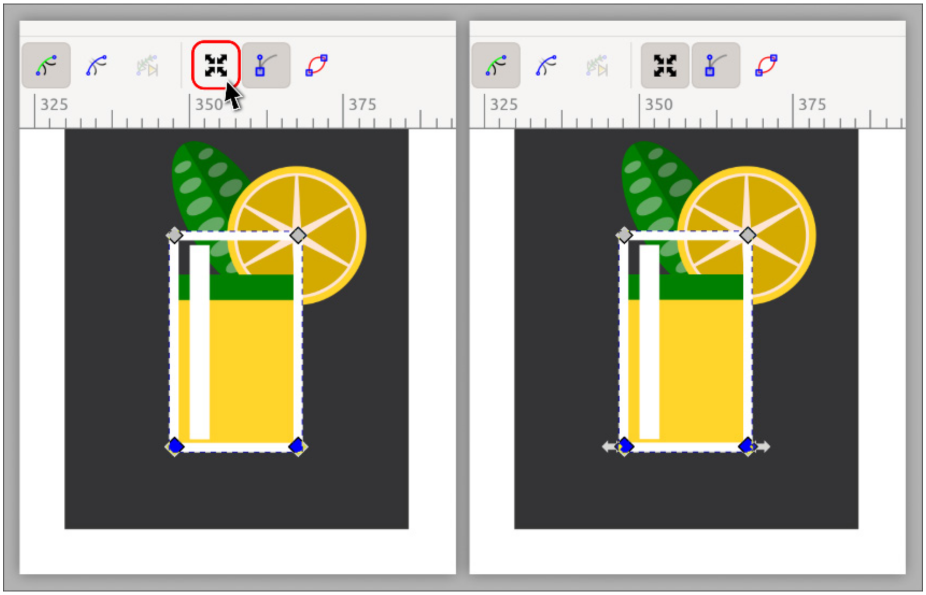


Figure 5.4 – Toggling node transform handles on

You can also click the nodes a second time to get the rotation handles, allowing you to rotate the selection as well. We will not use that here, but it's worth noting in case you need it!

Adding and removing nodes

What if we want more than four nodes? Inkscape allows you to add and remove nodes while you're using the **Node** tool by simply double-clicking the path. This will give you another node to drag around, as shown in *Figure 5.5*:

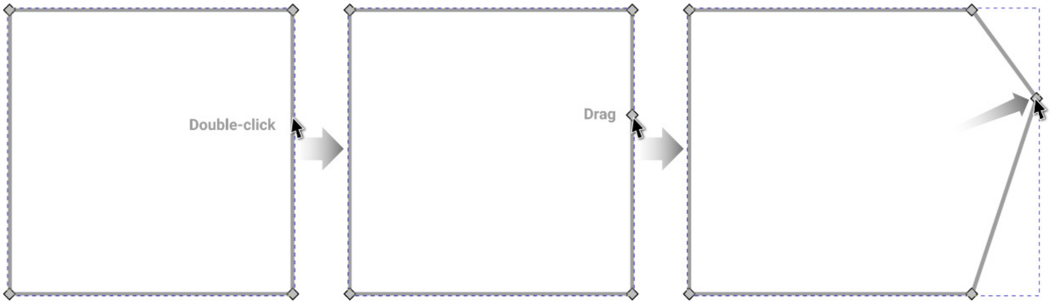


Figure 5.5 – Adding a node to a line segment by double-clicking

You can add multiple nodes at once with the handy node tools in the Tool control bar. Select a line segment by clicking on it first; then, we can use the **Insert New Nodes into Line Segment** button, also known as the **Add Node** button (the one with the + symbol above it). This will add a node right in the center of the segment, halfway between the two end nodes. Click it again to add center nodes to the two selected line segments, and so on, as shown in *Figure 5.6*:

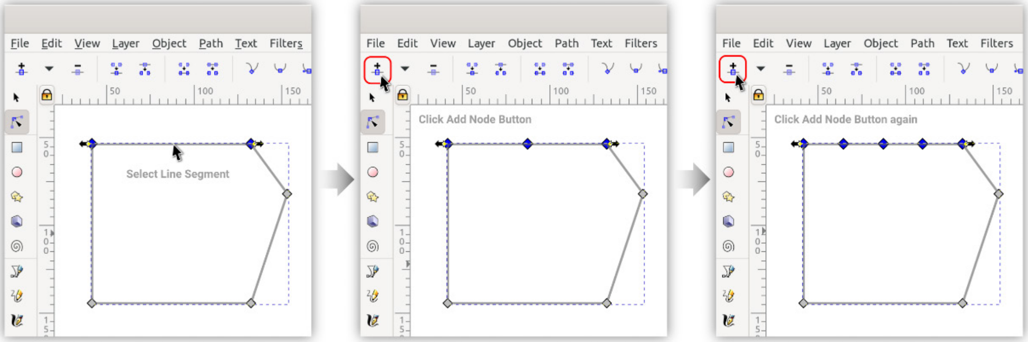

File Edit View Layer Object Path Text Filters
Select Line Segment
Click Add Node Button
Click Add Node Button again

Figure 5.6 – Adding nodes to the selected segments with the Add Node button

Removing nodes is just as easy. Simply select the node(s) you want to remove and hit the **Delete Nodes** button, press *Ctrl* + *X* (the standard hotkey for **Cut**), or tap the **Delete** key. The result of deleting a corner node is shown in *Figure 5.7*:

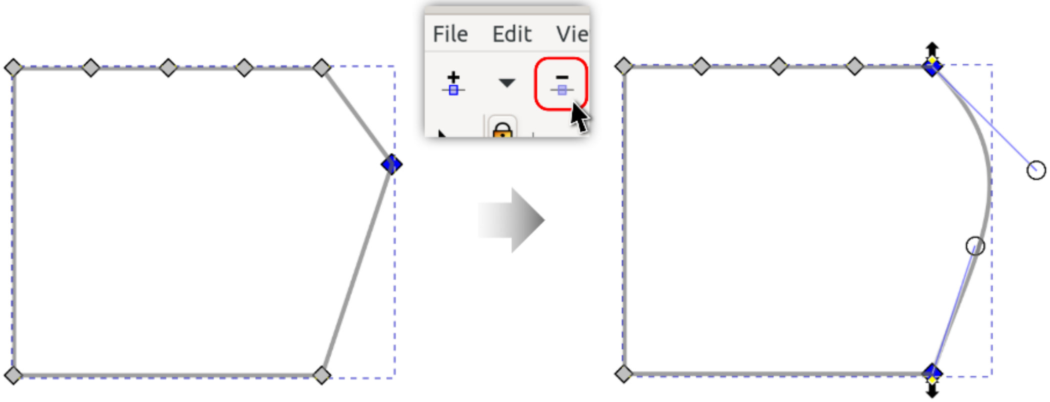


Figure 5.7 – Result of selecting and deleting a corner node

You may be asking at this point why Inkscape made a curve instead of just removing the node and restoring the vertical straight line. By default, Inkscape tries to preserve the overall shape when you delete nodes. The result is very useful for smoothing out long curves with too many nodes, or *bumpy* geometry, which we will see more of later with the **Trace Bitmap** tool.

For now, if you want your straight line back, you can tap the **Make Selected Segments Lines** button, which at the time of writing, we are in the process of renaming to the more descriptive **Straighten Lines** button. *Figure 5.8* shows the result of that operation on our selected curve:

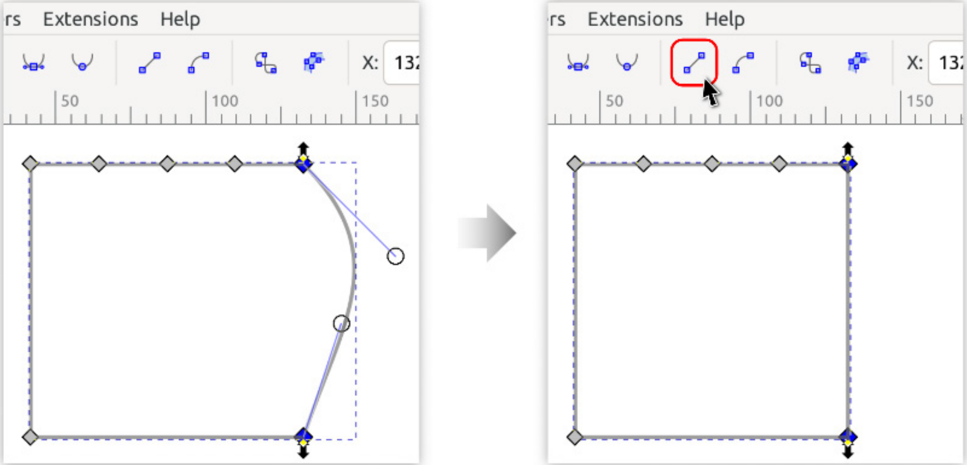


Figure 5.8 – Using the Straighten Lines button to turn the curve back into a straight line

Notice that when we do this, the curve handles disappear entirely. If you want them back, you can simply click the button directly next to the **Straighten Line** button that looks like a curve. This is the **Make Selected Segments Curves** button.

But confusingly, that's not what it does. It just adds back the curve handles, without changing the line at all, which is why in the future it will be called the **Add Curve Handles** button. Fortunately, you don't even really need this button in most circumstances. You can just click and drag any straight line into the curve shape you want, as shown in *Figure 5.9*:

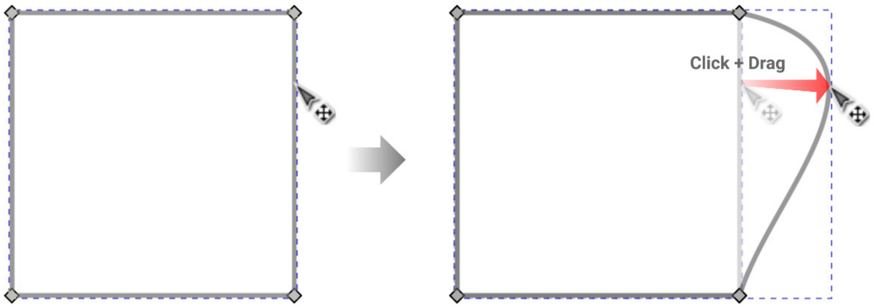


Figure 5.9 – Dragging out a curve from a straight line

What are those curve handles all about, though? Is it possible to make a curve perfectly smooth across a node? Let's explore that a bit, shall we?

Curve handles and node types

When we erased that node, we got our first taste of what a curve looks like. Curves in Inkscape are Bezier curves, made of one or two **curve handles** (also known as **control points**). When you click and drag a curve handle, you get precision control over how far your curve leans in one direction or another.

When several lines are connected to the same node, we have control over how the handles relate to each other and control the smoothness of the curve across the node. To keep us from having to manually place curve handles every time we want a smooth curve, Inkscape has different node types, which are useful for making different kinds of curves.

We can select which node type we want in the Tool control bar by selecting nodes and clicking whichever node type we want. *Figure 5.10* illustrates the different node types and their corresponding Tool control bar buttons:

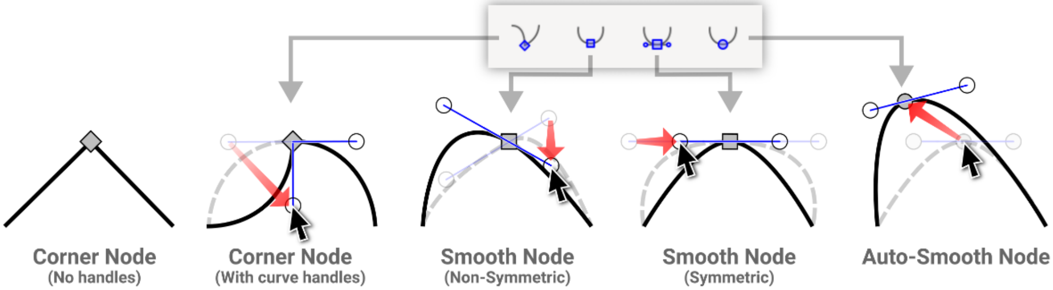


Figure 5.10 – Inkscape's different node types and how dragging one handle affects the other

As you can see, we have many options, each useful for making different kinds of curves. The **Corner** node appears as a little gray diamond and may or may not have curve handles. When curve handles are added, you can drag the handle on each side of the node freely, forming a sharp corner. For smoother options, we can select one of the two **Smooth** node options. The first is non-symmetric, which will keep the handles parallel to each other, but with different distances from the node on both sides.

Smooth Node Symmetric, on the other hand, mirrors the distance as you drag, which makes both handles equal in length. The last option is the **Auto-Smooth** node, which appears as a little gray circle. Auto smooth lets you just drag around the node itself rather than the handles, and Inkscape automatically creates smooth geometry wherever you move it.

If you grab a handle of the **Smooth** node, the node is automatically converted into a non-symmetric **Smooth** node, which preserves the curve you had before, but lets you tweak it to your satisfaction.

Let's take a break from node and line types and learn how to break our closed shapes into open ones.

Breaking and joining lines

Sometimes, we may want to break our closed path to create an opening in the shape. Take, for example, our glass. Maybe we decide that we don't want the top of our glass to have a stroke, for example.

We can select the top line of the glass and use the **Delete Segment Between Two Non-endpoint Nodes** button, which henceforth we will simply call the **Remove Line** button. As you can see in *Figure 5.11*, this removes both the line and, consequently, the stroke from the top:

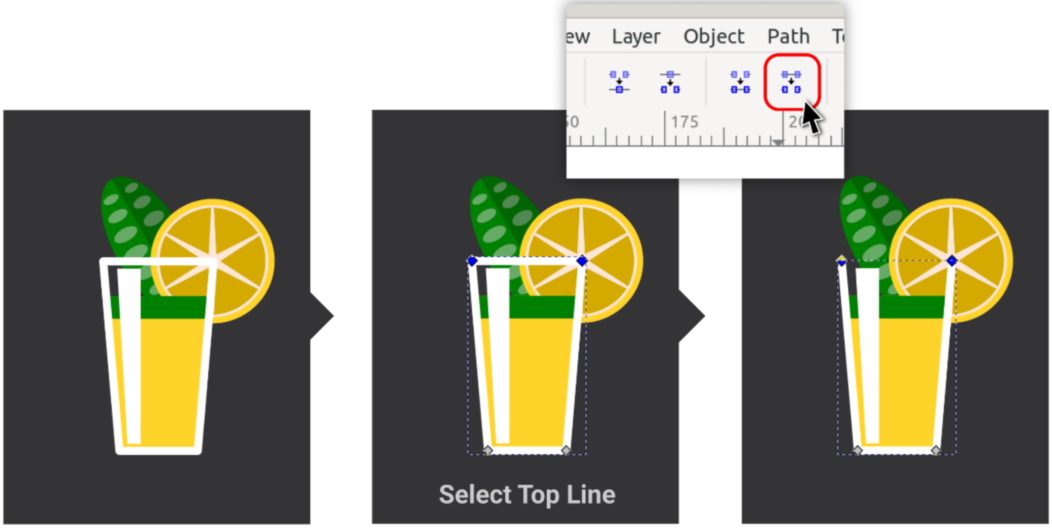


Figure 5.11 – Removing the top edge of our glass

Likewise, we can add it back with the **Join Selected End Nodes with a New Segment** button (also known as the **Add Line** button), immediately to the left of the **Remove Line** button.

You may notice that breaking the shape open in this way may or may not affect the fill of the object. This depends on which parts of the line are disconnected. When the closed shape is broken, Inkscape has to guess which parts to connect, and for this reason, I generally do *NOT* recommend assigning a fill to a broken path, and instead opt for a closed shape directly behind the stroke as a separate object, as we did when we made the fluids to fill our glass. *Figure 5.12* shows some ways the fill can go wrong in a shape, depending on how the lines are broken:

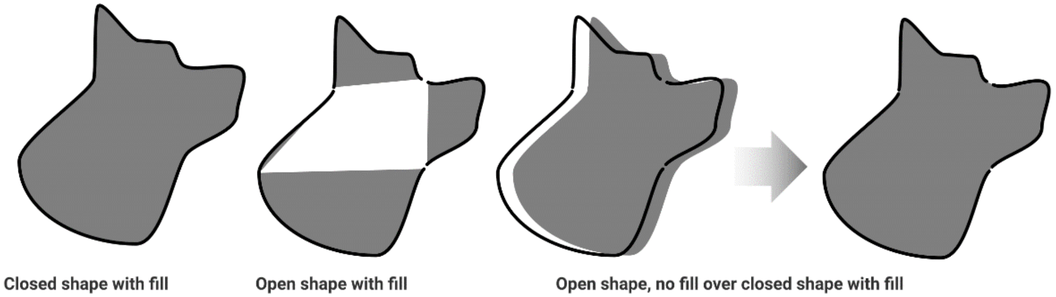


Figure 5.12 – Using two shapes to simulate a filled shape with a broken stroke

Making sure all filled shapes are closed will save you lots of potential fill-related headaches and also give you complete control over the shape of the fill. Without a line to close the gap, Inkscape can only try connecting with a straight fill, which is usually not what we want, and creates a mess, especially when we're trying to use them in boolean operations, which we will see in the next chapter. For now, let's continue making custom shapes, this time starting from scratch!

Drawing lines and shapes from scratch

We need not start with any shapes at all. Using the **Bezier Curve** tool, we can simply click in multiple places on the canvas, and Inkscape will connect those points with lines for us. Furthermore, we can complete a closed shape by clicking the same node we started with. You will notice when you move the cursor over the first node that it will be highlighted in red to indicate that clicking will close the shape. *Figure 5.13* shows this process:

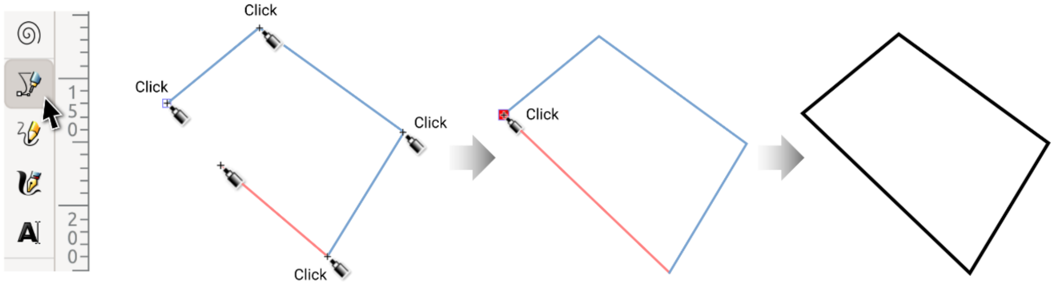


Figure 5.13 – Making an open or closed shape with the Bezier Curve tool

To make an open shape, we can hit the *Enter* key and Inkscape will end the shape at the last place we clicked.

If you misplace a node, you don't have to wait to correct it later. We can undo the last node we clicked by tapping the *Backspace* key. Inkscape will keep erasing nodes the more you tap *Backspace* until there are no more nodes left.

This is great for making a straight-edge polygon, but what if we want some curves along the way? Rather than click the next node into existence, we can click-drag out a curve, as seen in *Figure 5.14*:

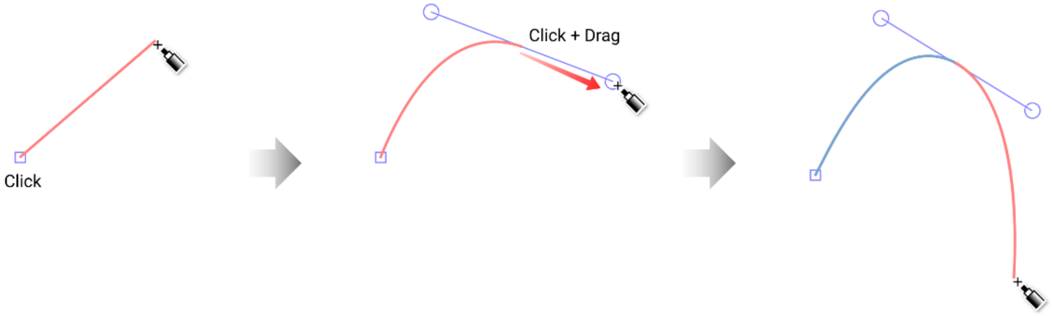


Figure 5.14 – Click-dragging out a curve

As you can see, this creates a smooth curve, with symmetrical curve handles. If we want a curve followed by a sharp corner, we can click-drag the first part of the curve where we want it and then hold *Shift* to change the node to a corner node, dragging the mouse to where we want our next curve to begin. This process, shown in *Figure 5.15*, takes a bit of practice to master, but it's a very fast way of making custom shapes:

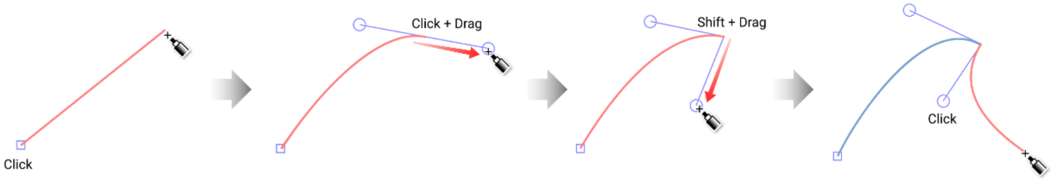


Figure 5.15 – Making sharp corners after a curve

While we're on the topic of open shapes, let's talk about line styles so that we can make our lines look just as we like.

Line styles

Thus far, we've been sort of just accepting the fact that our lines cut off sharply and are solid without any breaks unless we manually break them. However, it's possible to style our lines so that the ends are round or square capped, and our corners are round, sharp, or chamfered. We can do this through the **Stroke Style** menu.

To access this menu, we can simply click on the **Stroke width** value in the **Fill and Stroke Indicator** area in the lower-left corner of the screen. This brings up the **Stroke Style** tab in the **Fill and Stroke** dialog, as shown in *Figure 5.16*:

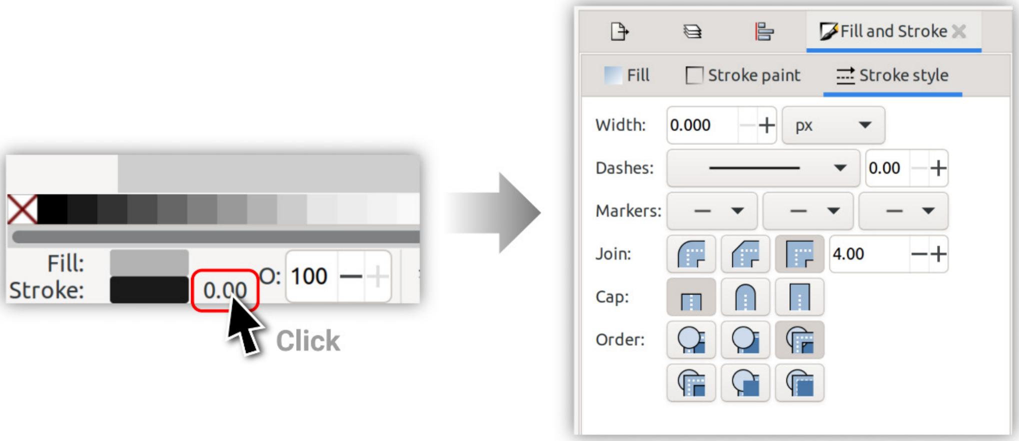


Figure 5.16 – Double-clicking the Stroke value and the resulting Stroke Style tab

Let's focus on how our corners look first. You can see the **Join** field, which controls whether our corners are sharp, rounded, or beveled, based on whichever option is toggled on.

As you can see in *Figure 5.17*, with the **Join** and **Cap** fields, we can choose the following:

- **No cap**: This stops the line at the node
- **Round cap**: This ends the line with a rounded end cap
- **Square cap**: This ends the line with a square cap, but one slightly longer than the line (good for preventing small gaps by overlapping):

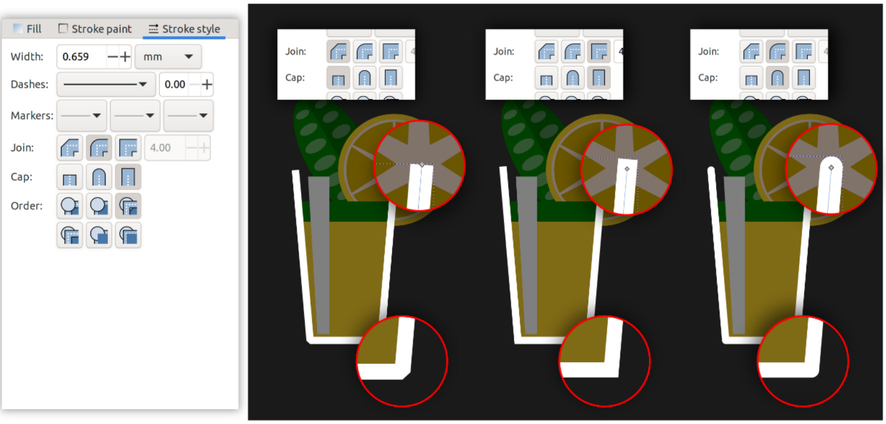
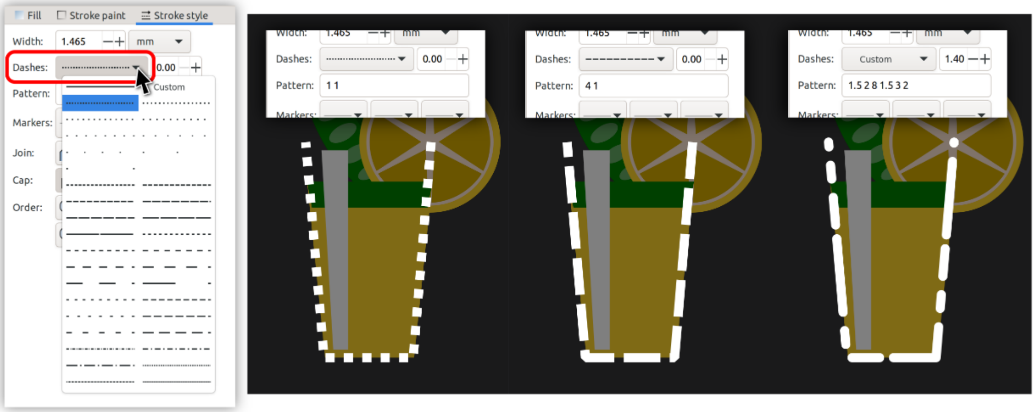


Figure 5.17 – The Stroke Style menu and three different Join (corner) and Cap types on our glass

In addition to caps, we also have the option to turn our solid line into a dashed line. *Figure 5.18* shows the **Dashes** dropdown, from which we can select some presets, as well as **Custom**:



Width: 1.465 mm
Dashes: 0.00
Pattern:
Custom

Dashes: 0.00
Pattern: 1 1

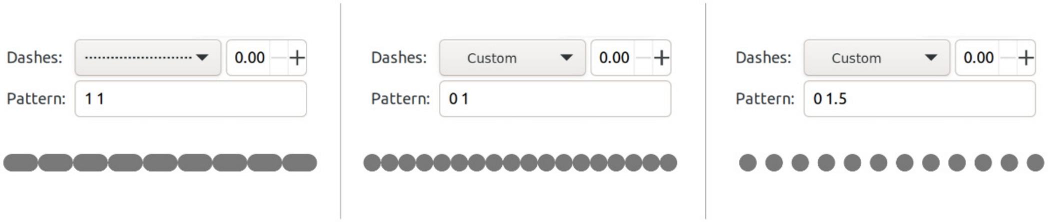
Dashes: 0.00
Pattern: 4 1

Dashes: Custom 1.40
Pattern: 1.5 2 8 1.5 3 2

Figure 5.18 – Choosing a pattern from the dropdown and the different results we can get

Note how when we choose one of these options, a **Pattern** field appears with numbers separated by spaces, representing the rhythm of the dashes and gaps. At first cryptic, the numbers in this box simply represent how long each dash is and the gap after it is. Note that you can also enter a decimal value such as `1.5` to increase the gaps in your dashes.

This is especially handy when working with rounded end caps, which tend to touch each other slightly at a gap value of `1`. Likewise, maybe you want perfectly round dots in your round-capped pattern. You can enter a value of `1`, but you will get a dashed line of `1` length with end caps added to it. Entering a value of `0` gets you *JUST* the rounded end caps (and thus a perfect circle) and gets your dots. Adding to that a value of `1.5` for a gap spaces them apart nicely. Make sure you separate the numbers in your pattern with spaces! *Figure 5.19* shows this progression of values:


Dashes: 0.00
Pattern: 1 1

Dashes: Custom 0.00
Pattern: 0 1

Dashes: Custom 0.00
Pattern: 0 1.5

Figure 5.19 – Getting a dotted, spaced line with Pattern values

As you can see, we can get some nice regular and irregular spacing for different effects. Sometimes, however, we'd like the endings and corners of our lines to feature custom shapes. For example, say we'd like to make an arrow, as shown in some of the diagrams in this book. Inkscape calls these shapes

Markers, and *Figure 5.20* shows the interface for selecting the start, end, and midpoint markers of our open shapes, with some different options:

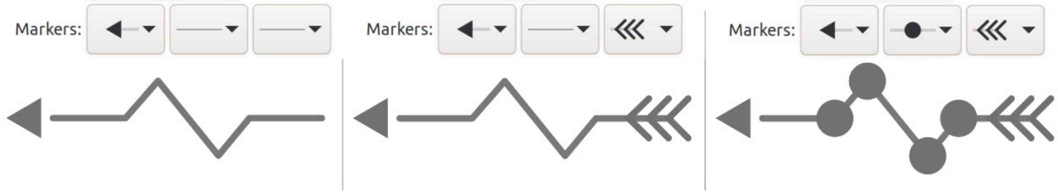
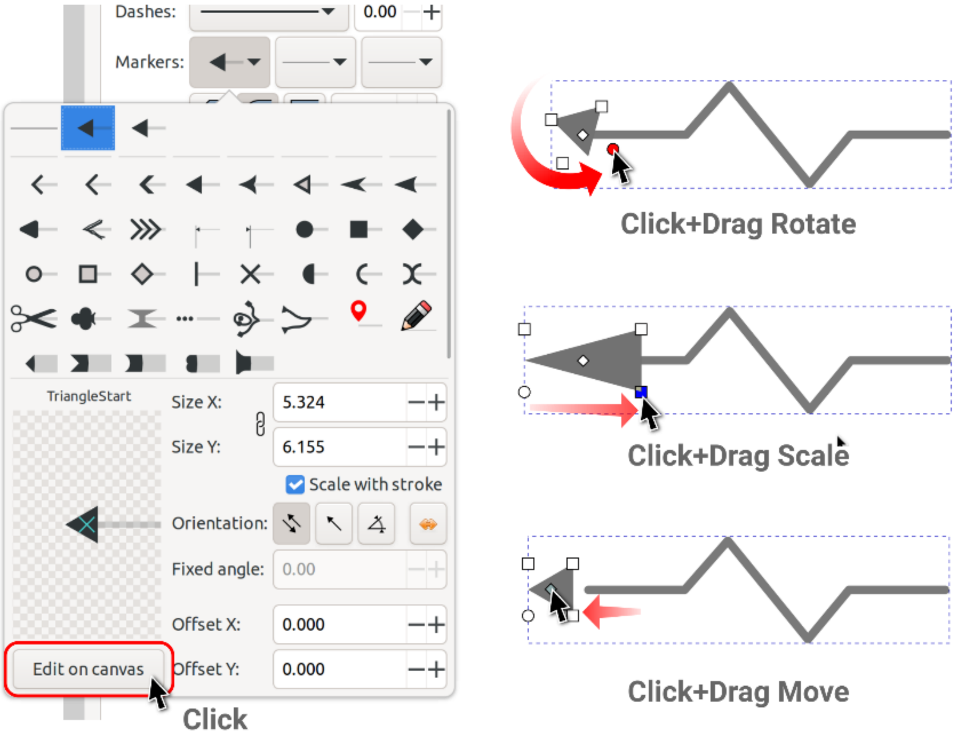


Figure 5.20 – Start, end, and middle markers

As of Inkscape 1.2, we can not only choose from a set of very nice preset markers but edit those markers on canvas, a very useful feature. You see, by default, Inkscape uses the stroke width to determine the size of the marker. This may do in some circumstances, but if you find yourself wanting some more fine control over the scale of the marker, we can select the **Edit on Canvas** option, as shown in *Figure 5.21*:


Dashes: 0.00

Markers:

TriangleStart

Size X: 5.324

Size Y: 6.155

☑ Scale with stroke

Orientation:

Fixed angle: 0.00

Offset X: 0.000

Offset Y: 0.000

Figure 5.21 – Editing the arrowhead marker on canvas

We can do the same for the middle and end markers, changing the scale and rotation in the same way we do regular shapes. Note that when we change the values on canvas, the values and options in the boxes change. Most of these are self-explanatory, but for completeness, here's what they do:

- **Size X** and **Size Y**: These control the horizontal and vertical scale of the marker. This is calculated outward from the origin (the little diamond we grab to move the marker around, which appears in the center of the marker when we are editing on canvas).
- **Scale with stroke**: This will attempt to auto-scale the marker head when you increase the stroke width. This keeps you from having to manually adjust the size of your markers every time you want a thicker or thinner line.
- **Orientation**: This has four buttons that control the direction of the line and markers:
    - **Orient along path, reversing at the start**: This is the default and ensures the start marker is oriented forward and aligned with the front of the line (the end).
    - **Orient along the path**: This is like the first option, but both the start and end markers are facing forward, aligned with the line direction.
    - **Fixed specified angle**: The third toggle in the row fixes the marker at the angle specified in the **Fixed angle** field, rather than having the marker follow the direction of the line.
    - **Flip marker horizontally**: The last button in this row flips the marker horizontally.
- **Fixed Angle**: This option is grayed out unless the **Fixed specified angle** toggle is activated. It sets the angle at which the marker will be rotated. This is automatically populated when we drag the rotation handle during on-canvas editing.
- **Offset X** and **Offset Y**: These options offset the marker from the start or end of the line, respectively. They are auto-populated when we use the diamond handle in the center of the marker to move it during on-canvas editing.

Down below the marker drop-downs is a puzzling set of buttons that look a bit like a question from an IQ test. These are options for **Draw Order**. You see, when Inkscape draws a shape for us, it needs to decide whether to draw the shape fill on top of the stroke or underneath.

To make matters more confusing, we have markers too. Should these be drawn on top of or underneath the stroke or the fill? In what **Order** should they be drawn? *Figure 5.22* shows several options for this **Order** and the effects on what our shape looks like:

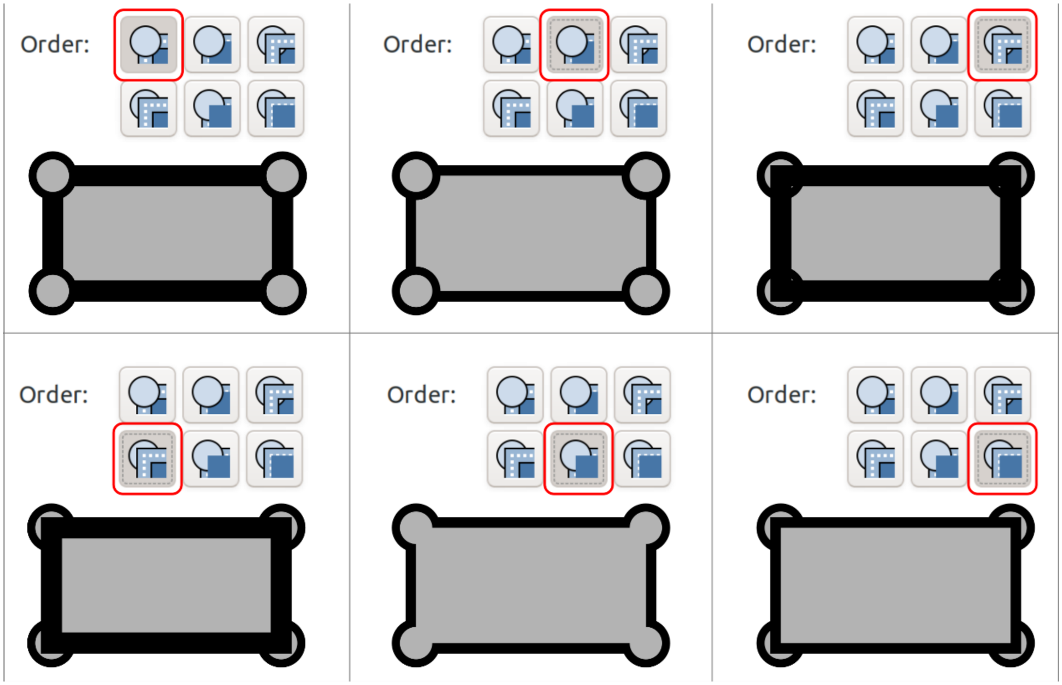


Figure 5.22 – The same rectangle with circle markers and thick strokes with different drawing orders

As you can see, the draw order makes a pretty big difference to how our shape looks. It's a good thing to keep in mind while you're working.

Before we get into some practice, it's worth mentioning two more tools that let us make custom shapes.

Classical drawing tools

Sometimes, we'd like to be able to pick a tool, just as we would a pencil, and have Inkscape make a line along the path as we draw it. Inkscape has two tools that do this.

The first is the **Freehand Drawing** tool, which is represented by a pencil icon and is located right under the **Bezier Curve** tool we looked at earlier.

Rather than click and click-drag around the canvas, we can just click-drag and Inkscape will make a line that follows the path that we drag, then stop drawing when we stop dragging and release the click. We can control the smoothness of this path with the **Smoothness** field located in the Tool control bar. As you can see in *Figure 5.23*, the higher our smoothing value, the smoother our line is:

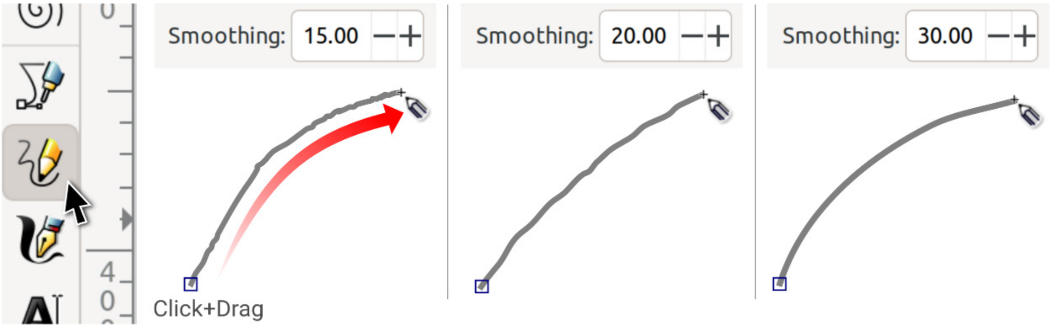


Figure 5.23 – Drawing with the Freehand Drawing tool with different smoothing values

Keep in mind that you need to have a **Stroke color** and **Thickness** set, and **Fill** set to nothing to see the preceding result. This was drawn with a mouse, so we might expect the line to be a bit wiggly as a result and can thus up the smoothing to compensate. Another method to get a smoother line is by using a graphics tablet. This is a device that includes a pen for drawing and a plastic surface that registers where you're drawing. It also registers the pressure you're applying to simulate the pressure of a real pen, pencil, or paintbrush.

Not only will this give us a much smoother line, but it will also allow us to use the **Use pressure input** button, which will add thickness to our line if we press harder, as shown in *Figure 5.24*:

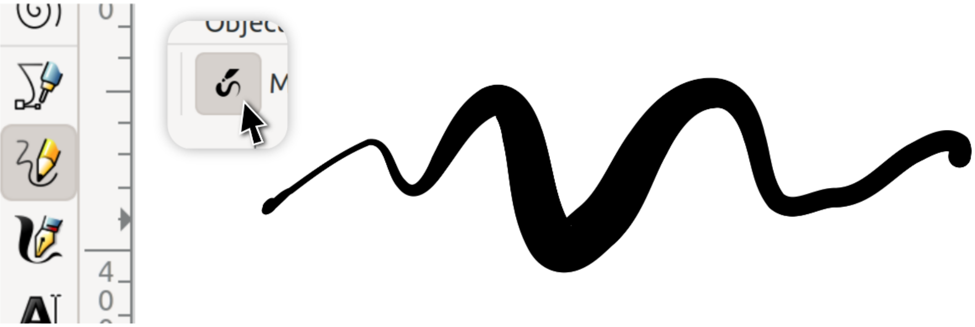


Figure 5.24 – Drawing with pressure sensitivity on

It may seem that this makes a filled shape rather than a line, but if we select our newly drawn squiggle, and activate the **Node** tool, we will see it's still just a line with some nifty extra handles to control the thickness at various points along our line. In Inkscape, this special line with thickness handles is called a **Powerstroke**. As you can see in *Figure 5.25*, we can click-drag one of these handles outwards or inwards to increase or decrease the line thickness:

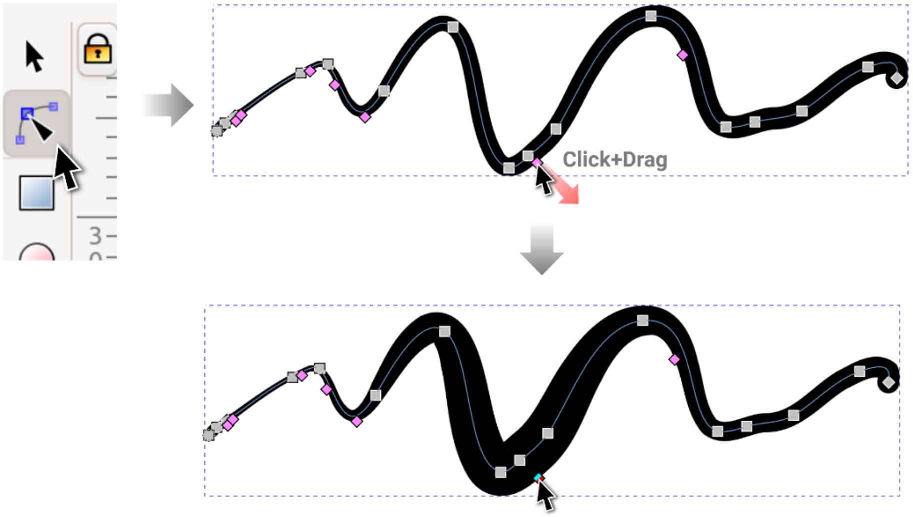


Figure 5.25 – Dragging the Powerstroke handle outwards increases the thickness

We could make all of our lines this way, but we may also just want Inkscape to make us filled shapes instead of fancy strokes. This is what the **Calligraphy Brush** tool is for. Indeed, drawing with this tool just makes filled shapes, which we can see after drawing a squiggle in the same way as before. *Figure 5.26* shows the resulting squiggle and nodes produced with the **Calligraphy Brush** tool:

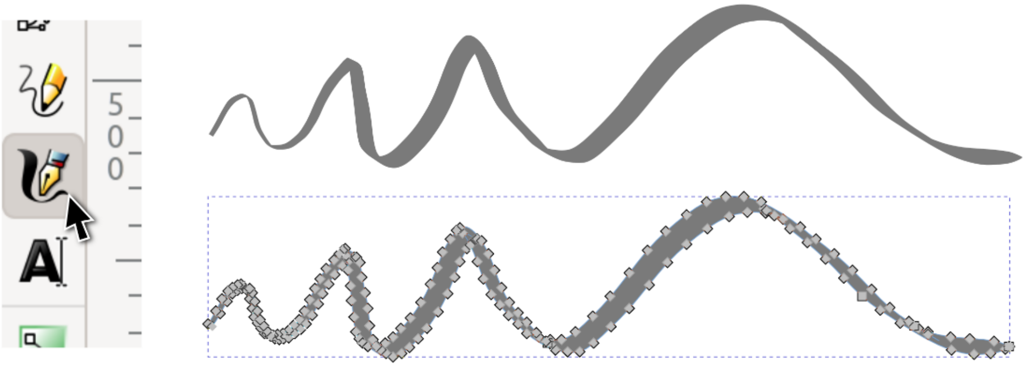


Figure 5.26 – Drawing with the Calligraphy Brush tool produces filled shapes

Notice that the shape of the brush produces a flattened calligraphy pen effect by default. There are a bunch of presets that produce different shapes, as shown in *Figure 5.27*:

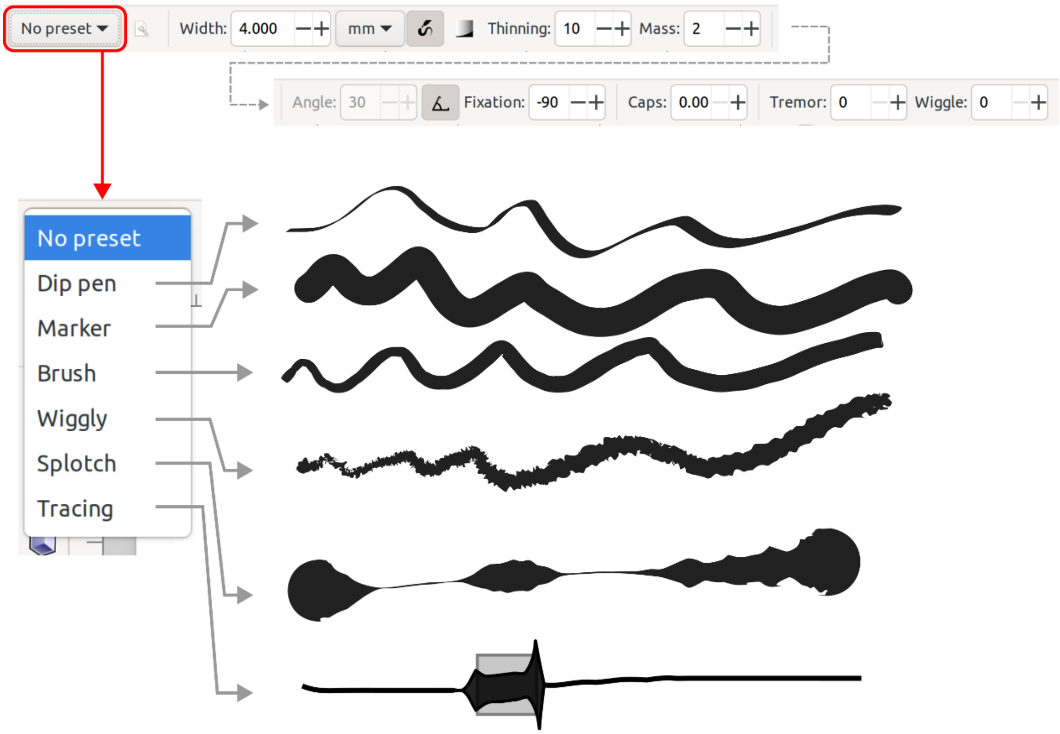


Figure 5.27 – Calligraphy Brush tool options and presets

Choosing one of these presets will change the values in the Tool control bar to achieve the look of the brush. Unfortunately, changing these values will not change the preset, but you can add your own presets by changing the values to something you like, and clicking the **Add or Edit Calligraphic Profile** button, which is directly to the right of the **Preset** dropdown. Just enter the name of your preset in the box that pops up. Your changes will be added to the preset list under that name.

Advanced curves and line options

We're nearly finished with our tour of the **Node** tool and accompanying custom shape tools. However, there are a few more kinds of curves and lines for us to explore. Let's jump back to the **Bezier Curve** tool and have a look at some of the other curve and line types. *Figure 5.28* shows the **Mode** buttons, which feature four more different types of curves/lines (the first is the **Bezier Curve** tool, which is active by default):

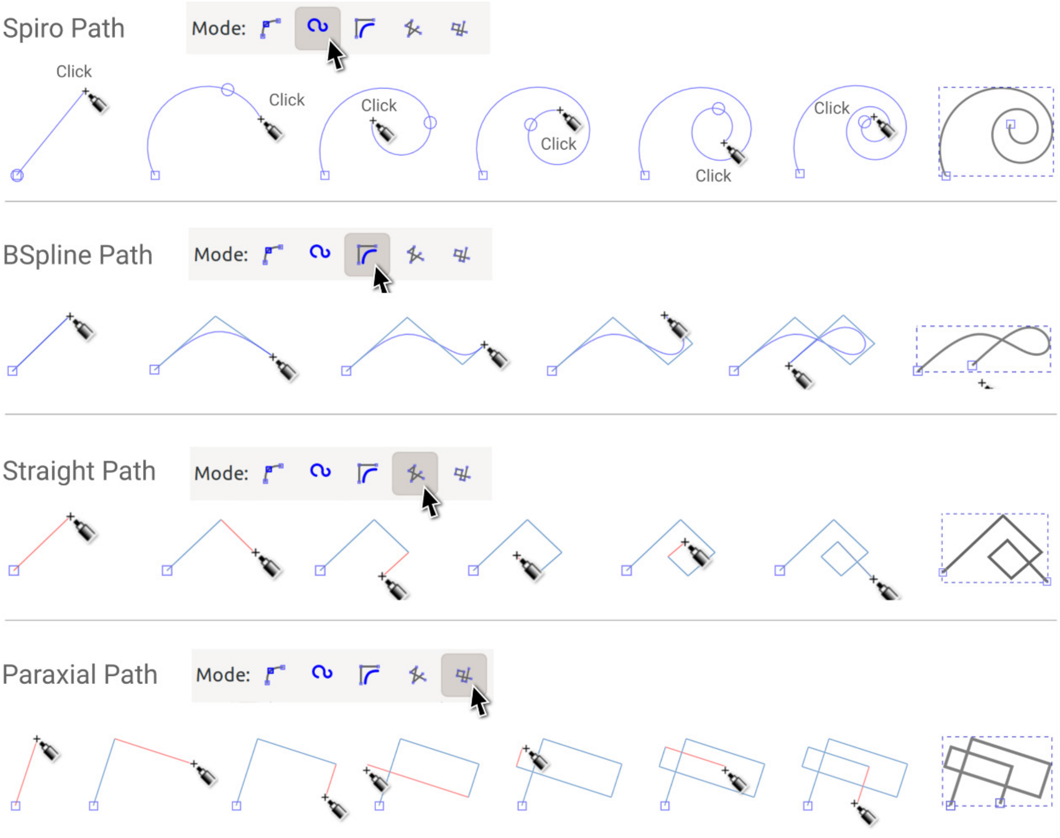


Figure 5.28 – Alternative curve and line modes

As illustrated, the different modes allow you to draw different curves and lines automatically:

- **Spiro Path**: This allows you to click around (no dragging necessary) to make a perfect spiro curve. This is great for floral flourishes, which are commonly used in graphic design.
- **BSpline Path**: This is a different kind of curve and uses points instead of handles to determine the curves. This can sometimes be simpler to work with than Bezier curves, and always creates smoothly joined lines.
- **Straight Path**: This can be handy if you never want your lines to curve. It locks out the click-drag curve options, leaving just straight lines.
- **Paraxial Path**: This is an interesting one. Your first line sets the initial angle, and every subsequent line is 90 degrees from the previous one.

Both the **Bezier Curve** tool and the **Freehand Drawing** tool feature these modes, and they also feature one more thing that we'll go over before ending this chapter with some practice.

Making brush strokes with Stroke Shape

If you're not one of the lucky people with the budget or bother to have a drawing tablet, there are still a few options that make life with a mouse much easier. Namely, we can have strokes that taper off using **Stroke Shape**.

First, let's activate the **Freehand** tool and look at the **Shape** drop-down. *Figure 5.29* shows what kind of stroke we can get with the various options contained therein:

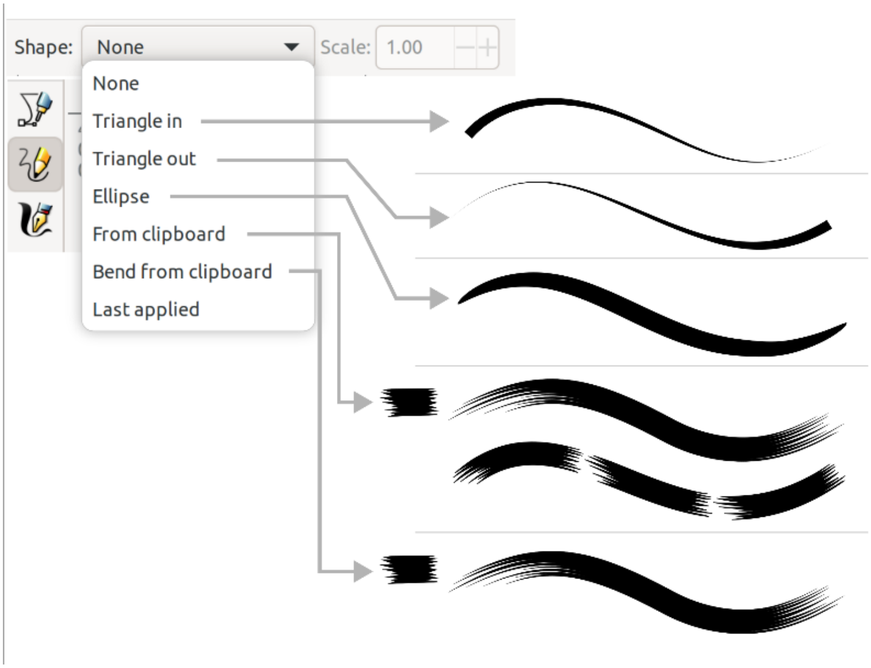


Figure 5.29 – Different shapes for drawing with the Freehand tool to simulate brush strokes

Just as we saw with **Powerstroke** earlier, these options allow you to rescale the stroke, and keep the overall shape of the simulated brush stroke.

The last two options, **From clipboard** and **Bend from clipboard**, apply two different Path effects to achieve the look of the last copied object. That is to say, you copy a shape you want to act as your brush, and then choose one of the two clipboard options. The next brush strokes you draw will look like the object that was copied.

Of the two clipboard options, my personal favorite is the **From clipboard** option, because it adds a **Pattern Along Path** effect with a lot more features, including **Repeat** and **Repeat stretch**.

We can get these same options when drawing with our **Bezier Curve** tool's **Shape** dropdown. We'll use this combined with spiro curves to create some of those nifty floral flourishes we were talking about earlier. So, without further delay, let's take what we've learned and make some stuff!

Try it yourself

Now that we have a good grip on how to draw and make different custom shapes, let's get into some practice. Having read through this chapter, you'll probably recognize some of the shapes in *Figure 5.30* that compose this nice little plant graphic we're going to build:

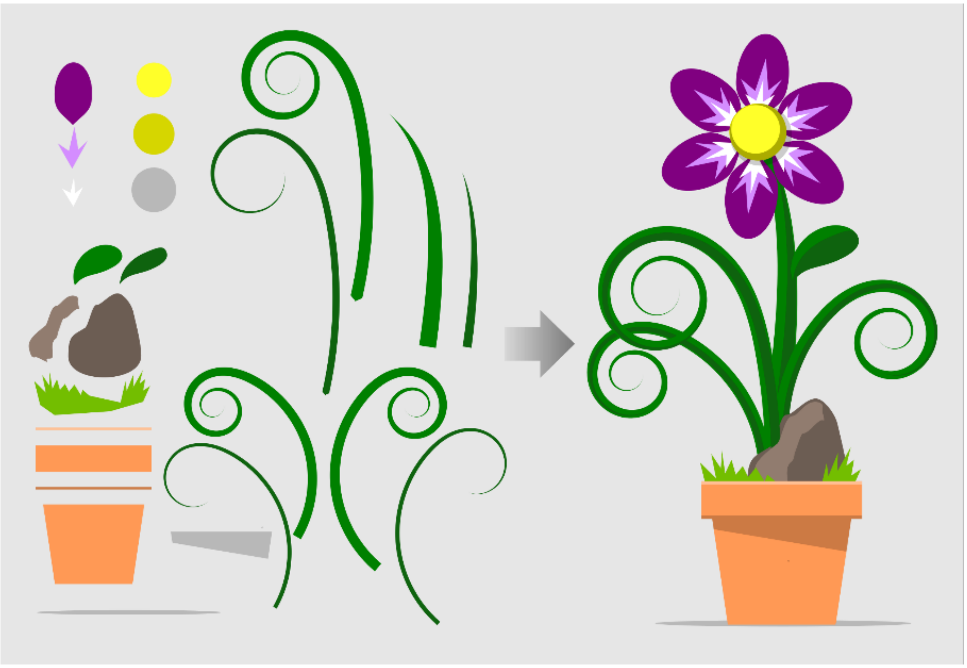


Figure 5.30 – An attractive potted plant and the parts that make it

Yes, you guessed it – we're going to use spiro curves with **Shape** set to **Triangle in** to make those attractive floral-flourish curling leaves. We'll also start with some base shapes as we did at the beginning of the chapter to save us some work, and we'll make some grass and petal details from scratch using our **Bezier Curve** tool:

1. Let's begin at the bottom with our pot. It's made mostly of rectangles, so using the **Rectangle** tool, we'll draw one rectangle to represent the base, then copy and resize it to make the various parts. We'll give the highlight and the shadow of the rim slightly different colors. We will then select the bottom rectangle, switch to the **Node** tool, and click the **Object to path** button. This process is shown in *Figure 5.31*:

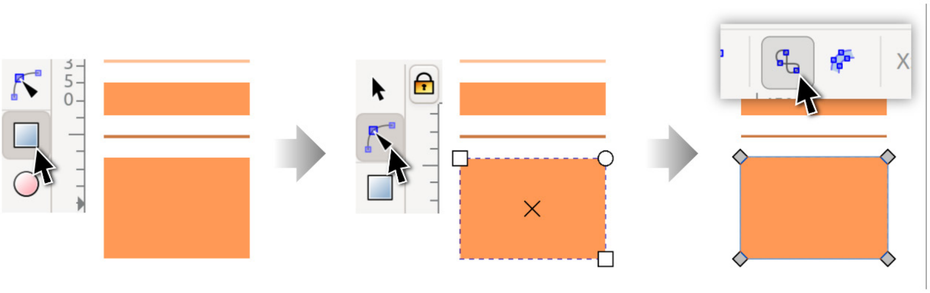


Figure 5.31 – Making our pot rectangles and converting the bottom into a path

2. Since our bottom rectangle has been converted into a path, we can use the **Node** tool to select both the bottom nodes (drag a selection box around the bottom nodes) and use the transform handles to resize the base to be thinner:

> **Tip**
>
> If you hold the *Shift* button while click-dragging the resize handle, Inkscape will resize the selected paths from the center rather than from only one node, as shown in *Figure 5.32*.

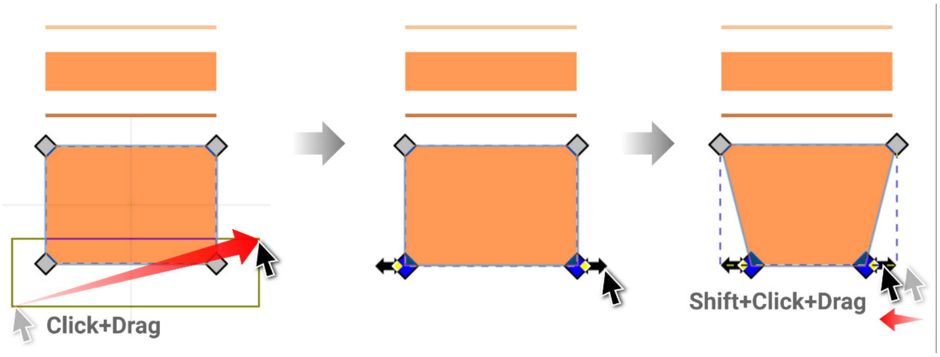


Figure 5.32 – Selecting and resizing the bottom nodes from the center

3. We can now create the shadow on our pot by selecting the pot base, copying it (*Ctrl* + *C*), and pasting it in place (*Ctrl* + *Shift* + *V*). We will then set the color to black, with an opacity of 30. Selecting one of the base nodes, we can click-drag it up the angled side of the pot, either using our snapping tools, or we can simply hold *Ctrl* + *Alt* while we click-drag to lock the node to the same angle of the line attached to it, as shown in *Figure 5.33*. We will then do the same to the node on the other side of the pot, dragging it up even more:

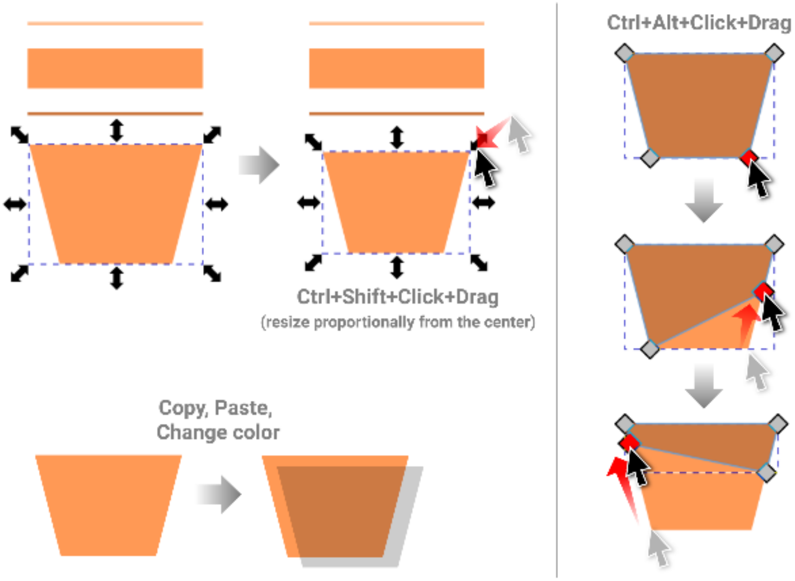


Figure 5.33 – Resizing the pot base and making the shadow

4. To finish our pot, we're going to use the **Circle** tool to make a long skinny ellipse as a base shadow for the pot and plant. We can then activate snapping and drag our shapes together to assemble our pot. This process is shown in *Figure 5.34*:

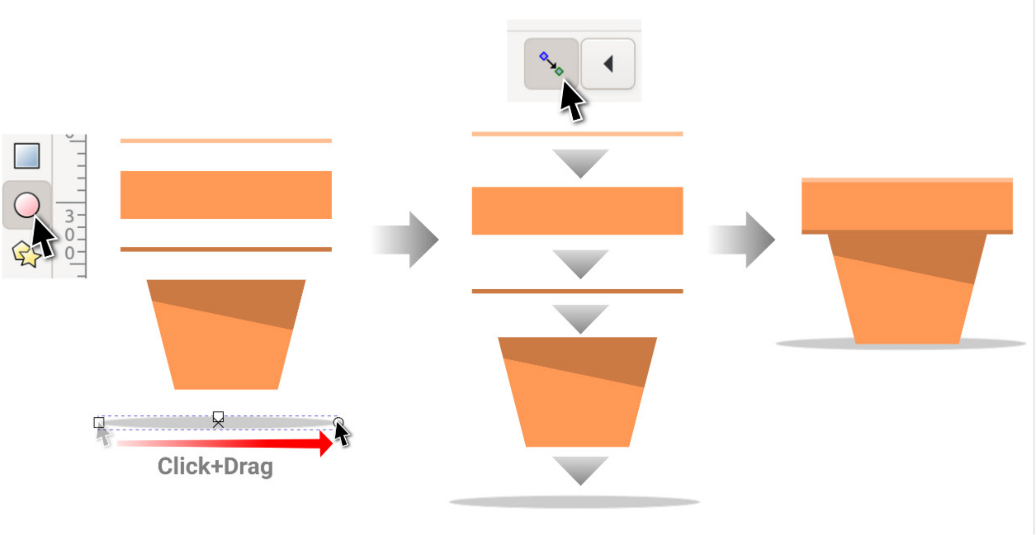


Figure 5.34 – Making the base shadow from an ellipse and assembling the pot shapes

5. Combining our pot and rock, we now have a basis to grow some grass! Just select the **Bezier Curve** tool and click a jagged line where you want your grass to go, as shown in *Figure 5.35*. When you're done, click the first node to connect it and create a solid shape, and color it green, moving it below your pot parts:

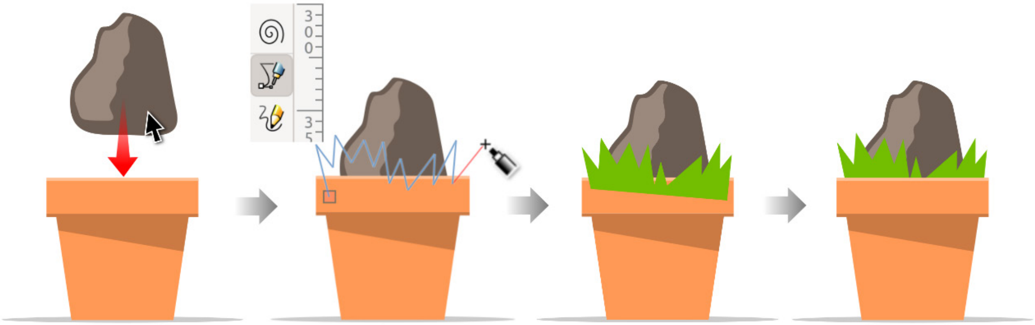


Figure 5.35 – Adding grass to our pot and rock

6. Making our stems and curled leaves is the next step, and for that, we will be using the **Bezier Curve** tool in combination with our newly discovered spiro curve and **Shape** set to **Triangle in**.

   Once we have those three things set, all we need to do is set our stroke to a green color by pressing *Shift*, clicking on a green swatch, and then drawing. It may take a few tries to see how spiro curves work. Just remember, you can always hit *Backspace* to undo the last node if it's not turning out to your liking. *Figure 5.36* shows this process, but don't worry too much about matching it perfectly. Get as creative as you like with them!

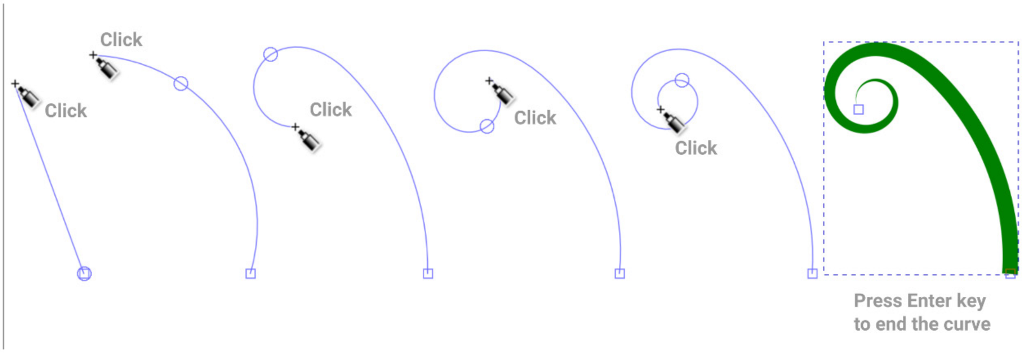


Figure 5.36 – Drawing a spiro curve stem with just mouse clicks

7. Draw three of these in different directions, starting from the base of the first. After you draw a spiro curve, you will probably want to edit it to fine-tune it. However, you will quickly see that

the original curve underneath is quite a mess of handles. Fortunately, you can turn off these handles and lines temporarily with the toggle buttons at the far right end of the Tool control bar.

What you're left with is a much more pleasant spiro curve editing experience, as you can see in *Figure 5.37*. Just don't forget to turn them back on when you're done!

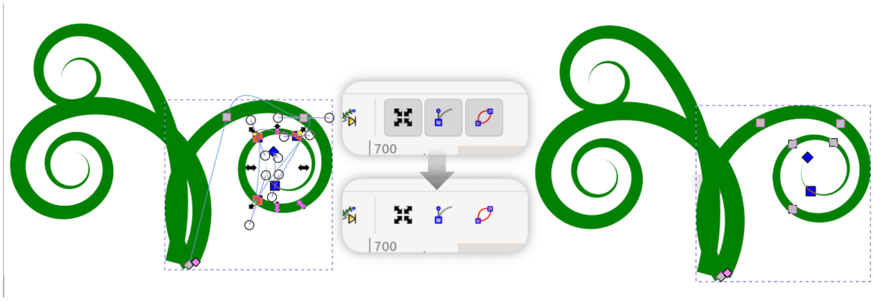


Figure 5.37 – Turning off handles to clean up the interface while editing spiro curves

8. To make the shadows on our vines, we're going to copy/paste them and change the color to black at an **Opacity** of `20`. Moving this off to the side, we can activate the **Node** tool and drag the little pink handle near the base of the stem inwards. This reduces the thickness.

   We do this to all three of our stems, and then move them back over our originals, using the **Raise** and **Lower** buttons in the Tool control bar (or *Page Up* and *Page Down* hotkeys) until the shadows are no longer overlapping each other and appear over the correct stem, as shown in *Figure 5.38*:

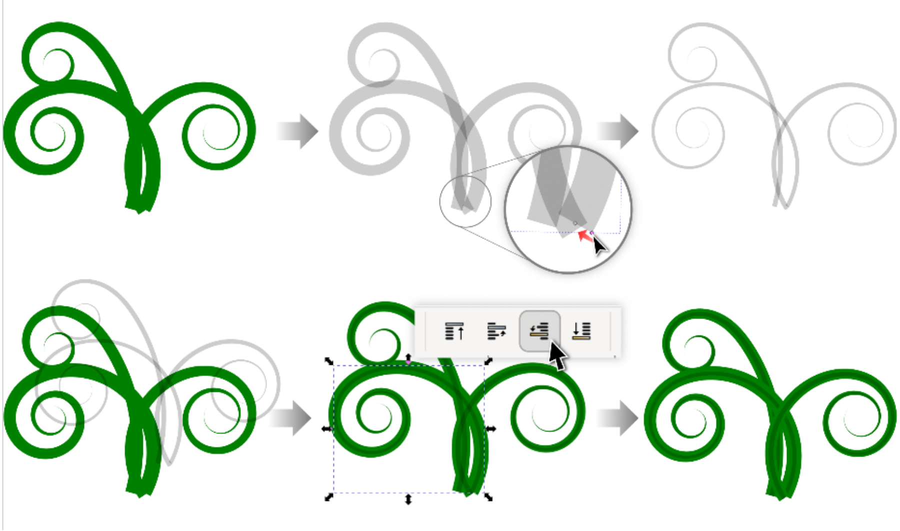


Figure 5.38 – Making shadows for our stems

9. We're on the home stretch here. Let's make the flower head out of circles with the **Circle** tool. Then, we'll start with a vertical purple oval as our petal base. Again, we'll choose the **Node** tool and convert that object into a path using the **Stroke to Path** button in the Tool control bar. We will then select the node at the bottom and convert it into a corner node using the **Corner Node** button in the Tool control bar.

   We can now adjust the handles to make them sharp and use the same technique we did to make the grass, using the **Bezier Node** tool to make our petal decorations. *Figure 5.39* illustrates this process:

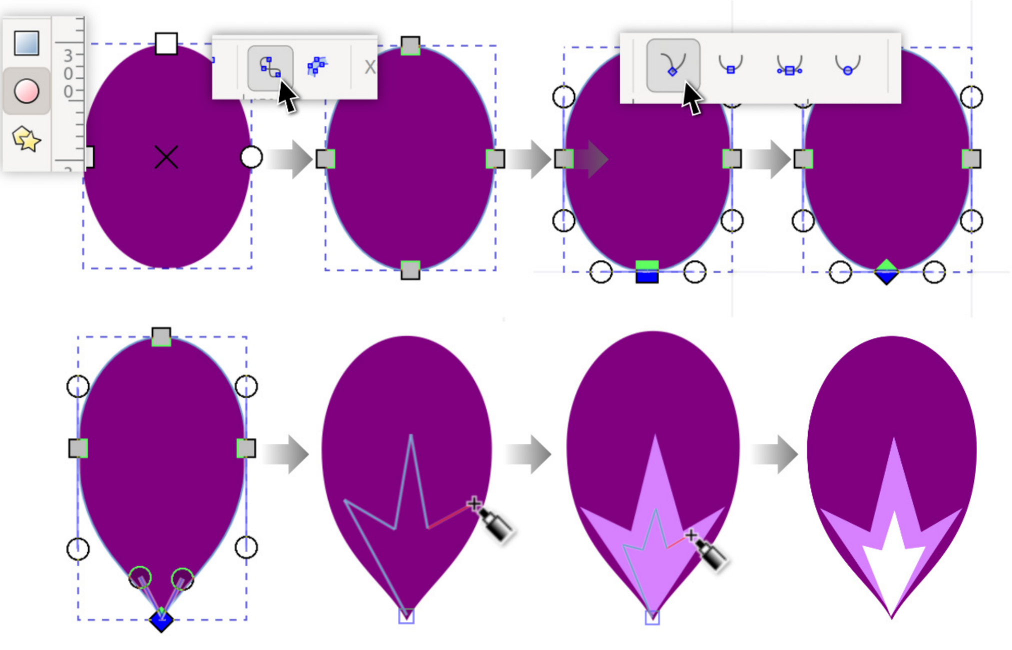


Figure 5.39 – Making a flower petal

10. With one petal made, we can group all its parts, click twice to get our rotation handles, and move the rotational pivot to the bottom center. Now, we can copy and paste in place, rotating the new petals into position while holding the *Ctrl* key to snap them to specific angles.

    Once we have six of them, we'll make some yellow circles to make our flower center. Move them over the petals and group everything together, as shown in *Figure 5.40*:

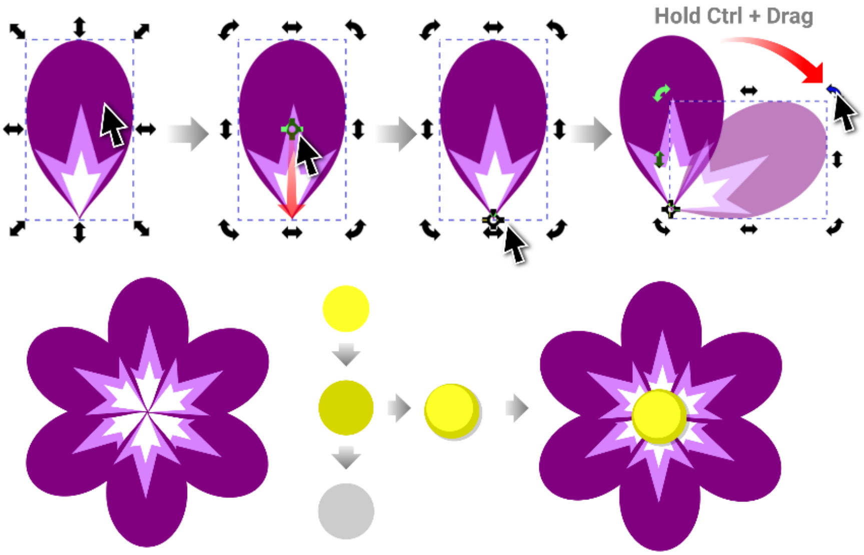


Figure 5.40 – Finishing the flower top by copying/pasting and rotating the petal group

11. Finally, we assemble our flower, stems, and pot together, as shown in *Figure 5.41*. Now, we can save our file as something memorable such as `my_amazing_flower.svg` and we're done!

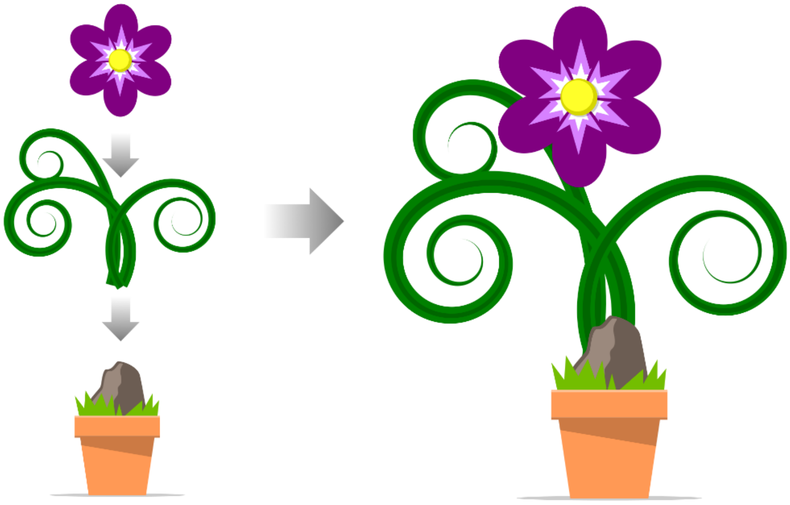


Figure 5.41 – Final assembly of flower parts

And we're done! We certainly covered a lot in this chapter, but really, we're just getting started. So, after you're finished celebrating this great progress and your new flower graphic read on for more Inkscape design fun!

Summary

In this chapter, we learned how to make custom shapes by editing paths with the **Node** tool. We also learned about the different kinds of nodes we can use (smooth, corner, auto, and so on), and played with different kinds of curves (**Bezier**, **Spiro**, and so on). We even learned how to draw paths directly on the canvas with the **Freehand** and **Calligraphy Brush** tools and apply shapes to simulate brush strokes.

In the next chapter, we will explore how to make custom shapes very quickly with **Path Operations** and the brand-new (as of Inkscape 1.3) **Shape Builder** tool.

Part 2: Advanced Shape Editing

This part expands on the previous chapters by introducing shape operations (aka Booleans), such as using shapes to cut and combine with other shapes, gradients, and patterns, and using the text tool.

There are the following chapters in this section:

- *Chapter 6, Fast Shape Editing with Path Operations and the Shape Builder Tool*
- *Chapter 7, Using Text in Inkscape*
- *Chapter 8, Advanced Shading and Coloring*
- *Chapter 9, Clips and Masks*
- *Chapter 10, Automation with Clones and Linked Files*

6

Fast Shape Editing with Path Operations and the Shape Builder Tool

In the previous chapter, we saw how we can make shapes by editing nodes and curves. This is handy but can be quite time-consuming. Wouldn't it be great if we could, say, add two shapes together to make one shape? That would certainly save us a lot of time! Well, you'll be happy to hear that we can indeed do that with what is known as **path operations**, which people also sometimes refer to as **booleans**.

We will cover the following topics in this chapter:

- What are Booleans?
- Exploring the Combine, Break Apart, Split Path, Fracture, and Flatten operations
- Inset, Outset, and Dynamic Offset Path operations
- Fast Path operations with the Shape Builder tool

Technical requirements

You can download the practice files for this chapter from GitHub at

```
https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape/
tree/main/Chapter06
```

What are Booleans?

Boolean is a term that spans the graphic design, math, and logic fields of study. In graphic design, it's a way to take two (or more) shapes and use them to create a new shape with a path operation. For example, we may want to take a circle and a square and merge them to make a new shape.

We can do this by making a square using the **Rectangle** tool, and a circle using the **Circle** tool, and then choosing **Union** from the **Path** menu, as shown in *Figure 6.1*:

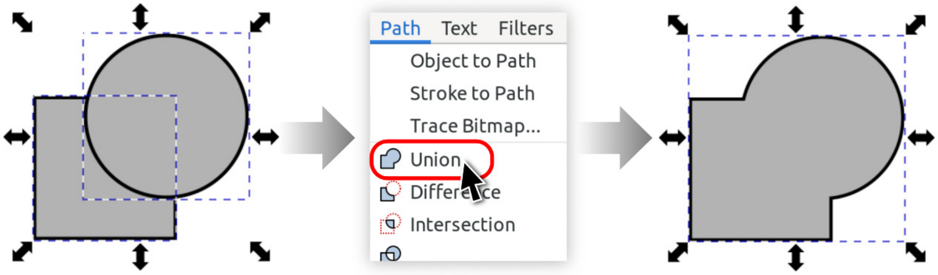


Figure 6.1 – Performing a Union operation on a circle and square

There are currently six Boolean operations in Inkscape, which are as follows:

- **Union**: Adds/merges shapes together
- **Difference**: Takes the top shape and subtracts it from the bottom shape
- **Intersection**: Takes two (or more) shapes and only leaves the intersecting parts
- **Exclusion**: Takes two (or more) shapes and cuts out the intersecting parts, leaving the non-intersecting parts
- **Division**: Takes two shapes and uses the top shape to slice the bottom shape into parts
- **Cut Path**: Uses the top shape to cut the path of the bottom shape, leaving open paths instead of closed shapes

Figure 6.2 shows these options, which are accessible from the **Path** menu:

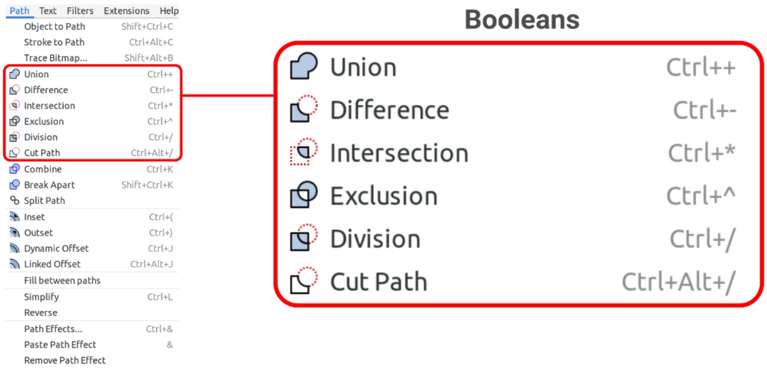


Figure 6.2 – Boolean operations in the Path menu

As you can see, the **Path** menu has a convenient icon alongside each Boolean that shows what it does.

You can select more than one shape to operate on. Say, for example, you want to combine ten shapes into one shape. Simply click-drag and select all the objects and use **Path** > **Union** or the *Ctrl* + + hotkey. *Figure 6.3* shows the results of selecting a bunch of objects and performing a **Union**:

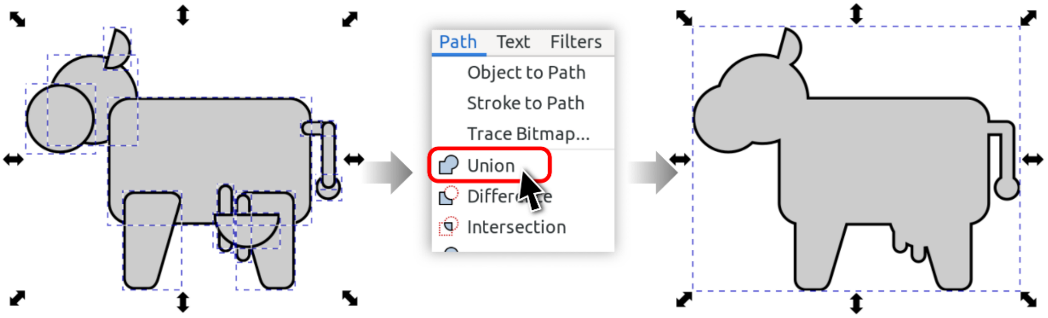


Figure 6.3 – Performing a Union on a bunch of selected shapes

The **Difference** path operation uses the stacking order of the shapes to determine the end shape. As you can see in *Figure 6.4*, **Difference** will subtract the object on top from the lower shape:

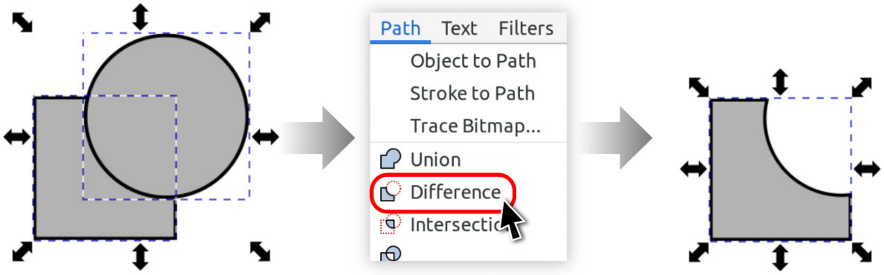


Figure 6.4 – Performing a Difference operation on a circle and square

Depending on the order in which the shapes are subtracted, different results could occur. We see this when we place the square on top of the circle and perform the same **Difference** operation (see *Figure 6.5*):

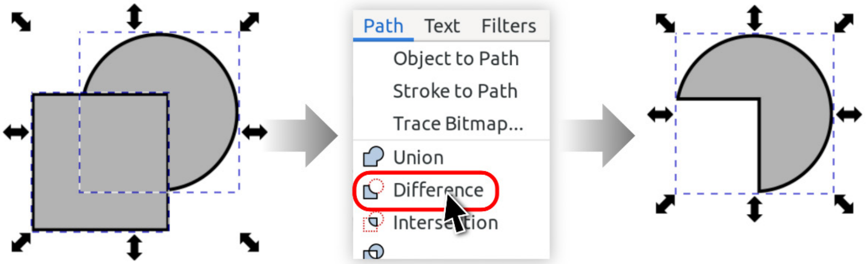


Figure 6.5 – The same Difference operation as Figure 6.4, but with the square on top

What about more than two overlapping shapes? Things work the same way, only the selection order rather than the stacking order dictates the subtraction order. *Figure 6.6* shows the results of this on a circle, square, and star *Shift*-selected in different orders and the resulting difference (whatever is selected last is the shape that remains):

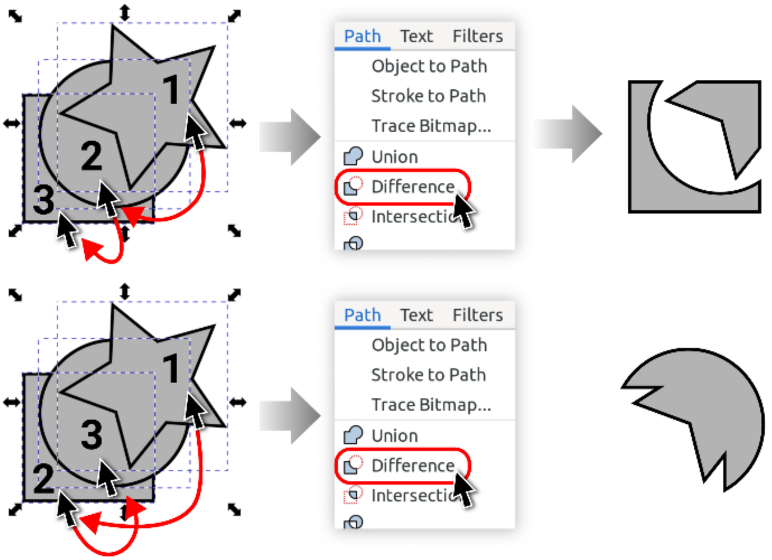


Figure 6.6 – Selection order matters! Different selection orders give different results

As you can see, in the first example, the star is subtracted from the circle first, then the resulting shape is subtracted from the square. By simply changing the order, we see far different results: the star is first subtracted from the square, and then the result is subtracted from the circle, as we can see at the bottom of *Figure 6.6*. If this is confusing, you can do what I do: select no more than two objects at a time for **Difference**. This will give you greater control over the results because you are operating on just two shapes instead of three or more.

Moving along down our list, we have **Intersection**. This modifier is useful for selecting two or more shapes and leaving *only* the parts that are overlapped/intersected by all the selected objects. *Figure 6.7* shows how we can make an interesting angular tear-drop shape with three circles and the **Intersection** path operation:

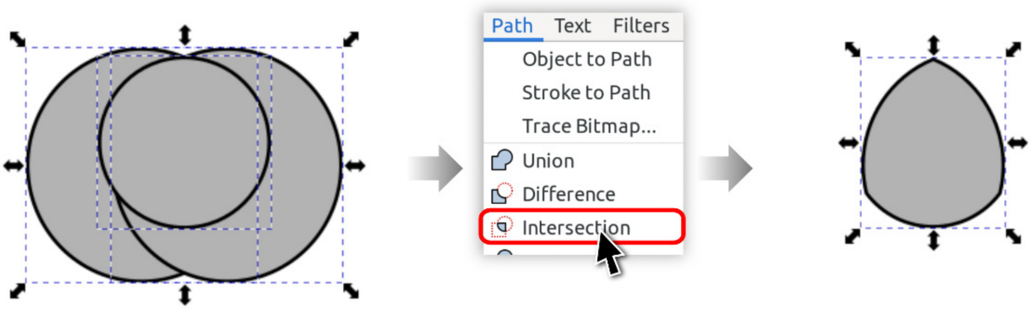


Figure 6.7 – Intersection applied to three circles

Notice that it's a bit hard to see what the resulting shape will be with solid-filled overlapping shapes. Here's a neat trick to get a preview of the result while we are still building the shape: select all shapes and set **Opacity** to `20`.

As you can see in *Figure 6.8*, the 20% opacity of each overlapping shape adds up. The more shapes overlap, the darker the shade will be. Thus, the darkest shape shows what the result of the **Intersection** operation will be:

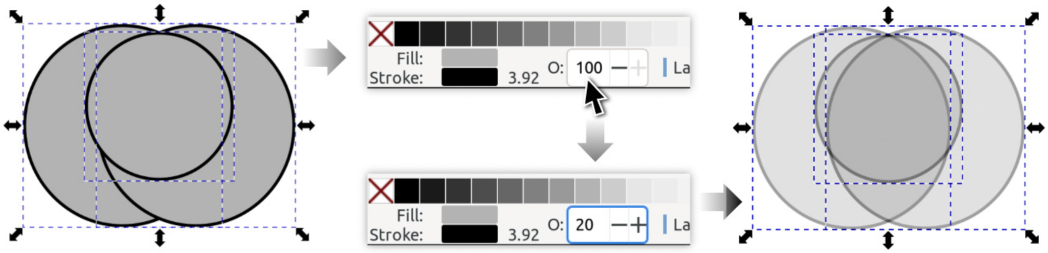


Figure 6.8 – Reducing the opacity of the shapes to reveal where the most overlapping is

What if we want the opposite of this? Say, for example, we'd like to have only the portions of the shapes that *do not* overlap. For this, we can use the **Exclusion** path operation. *Figure 6.9* shows the same three circles we used in our **Intersection** example, this time performing an **Exclusion** operation:

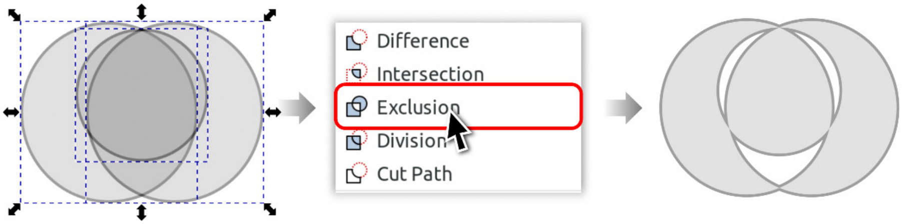


Figure 6.9 – Using Exclusion on the same three circles

Rather confusingly, the shape that results includes the area where all three shapes overlap, cutting out only the double-overlapping parts. I consider this an error, but at the time of writing, that's kind of just how this is.

My advice is to experiment with **Exclusion** and see how it works in your projects. You can avoid it altogether of course by selecting two shapes at a time and getting more predictable results.

As bad as **Exclusion** is for more than two objects, **Division** *doesn't work at all* for more than two objects. With just two objects selected, **Division** works by taking the top shape and using it like a cookie-cutter on the second shape, as you can see in *Figure 6.10*:

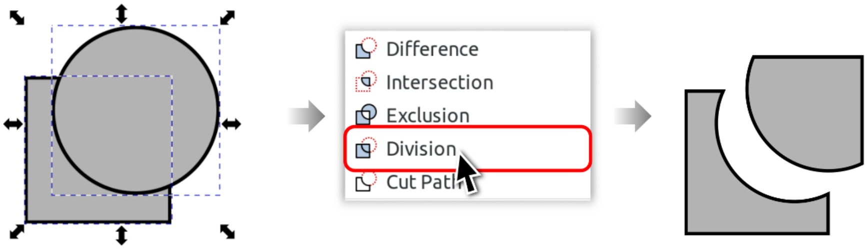


Figure 6.10 – Division operation on two objects

We run into problems when more than two shapes are selected, however. As far as I can tell, at the time of writing this book, *only two objects* are used for the operation; the rest just get deleted.

However, here, we can use another path operation to combine our cutting shapes into one shape, and then use our two-shape trick to get more predictable results. Say we want to use three lines to cookie-cut a circle into a bunch of triangular shapes. If we first use the **Combine** path operation on our three lines, it will combine them into one shape.

We can then select both our combined line shape and our circle and apply the **Division** path operation with exactly the results we want, as you can see in *Figure 6.11*:

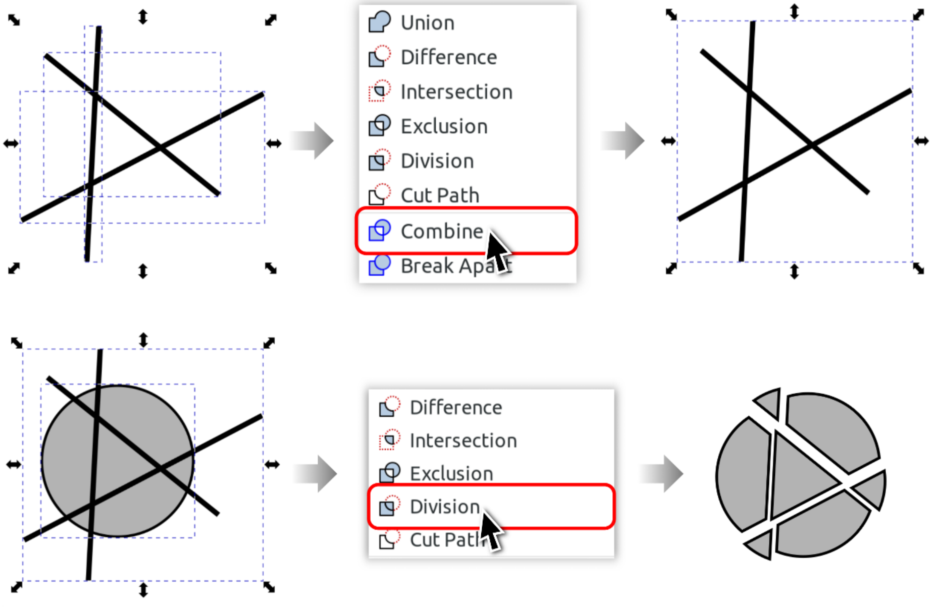


Figure 6.11 – Combining lines, then using them like a cookie-cutter on a circle with Division

We can use this cool trick again with the **Cut Path** operation, which cuts closed shapes into open paths. *Figure 6.12* shows the effect using a combined shape of seven vertical lines on the circle with the **Cut Path** operation:

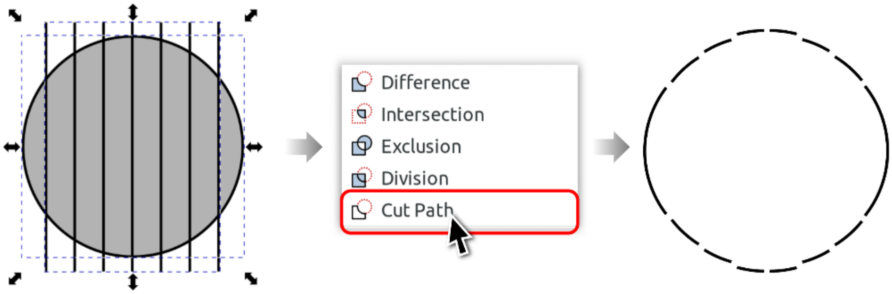


Figure 6.12 – Combined lines used to Cut Path

As you can see, everywhere the lines touched the edge of the circle, it cut the line.

We've now covered our full set of boolean operations, which make new shape paths out of multiple shapes. Sometimes, we don't want an operation to give us new geometry, though. Let's explore some other path operations that do just that.

Exploring the Combine, Break Apart, Split Path, Fracture, and Flatten operations

Maybe we want to combine paths into one shape and leave all the previous geometry alone, or do the opposite, and break apart all the paths into separate shapes. We might even want Inkscape to intelligently separate individual shapes in our paths but leave the holes and cut-out shapes intact.

Let's take a closer look at each of these operations, located just below the Booleans in our **Path** menu, as shown in *Figure 6.13*:

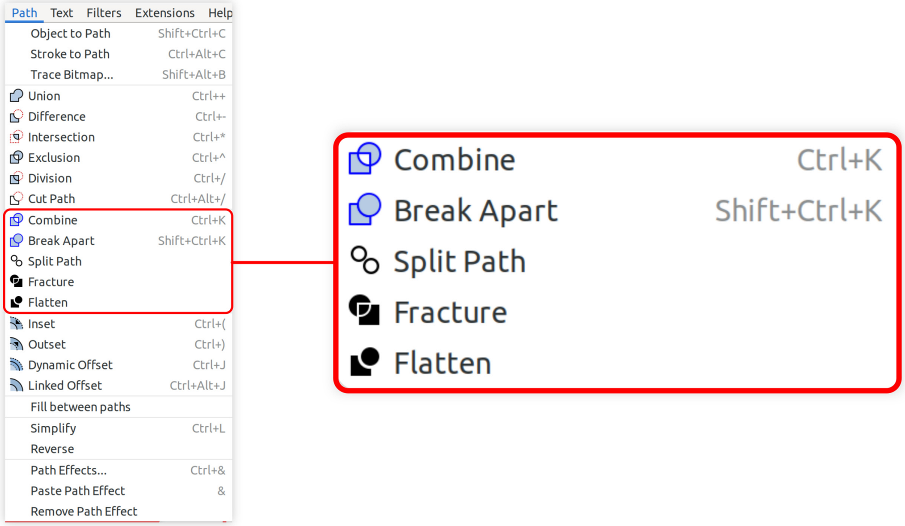


Figure 6.13 – The Combine, Break Apart, Split Path, Fracture, and Flatten operations in the Path menu

We used the **Combine** path operation to combine lines into one shape to **Cut Path** and **Divide** previously. But we can also use **Combine** with filled and even overlapping shapes, as shown in *Figure 6.14*:

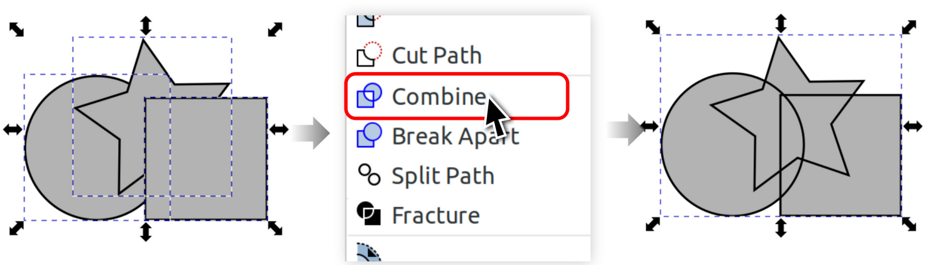
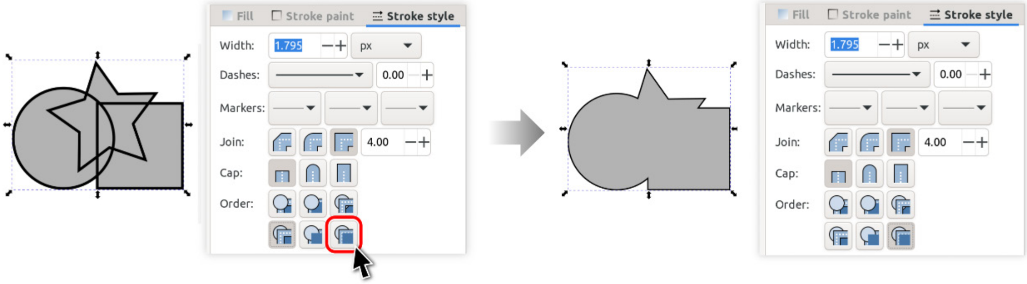


Figure 6.14 – Using Combine to join the paths into one shape without affecting the path geometry

This new combined shape could appear differently from above, though, depending on which options are checked in the **Path Style** menu. Additionally, since our three-shape path is now a single shape, any fill or stroke or style we give it will apply to all three sub-shapes. *Figure 6.15* shows two different styles applied to our new circle/star/square shape:

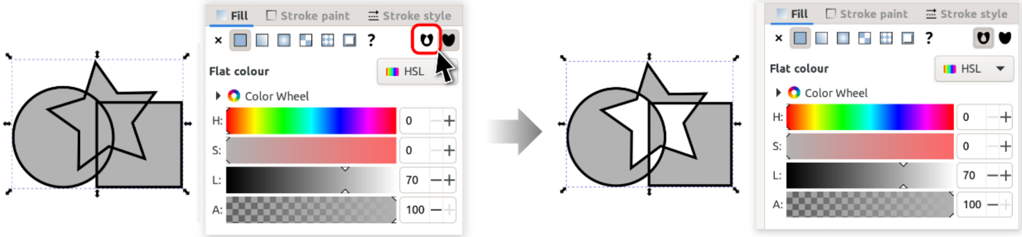


Width: 1.795 px
Dashes: 0.00
Join: 4.00

Width: 1.795 px
Dashes: 0.00
Join: 4.00

Figure 6.15 – Changing styles on combined shapes to fill on top in Draw Order

You can see that if we simply change our **Draw Order**, as we did in the previous chapter, it puts all the fills for all three shapes on top of the lines, making it look rather like we performed a union on the shapes. They are still separate paths, however. We can even change **Fill Rule** to **Even/Odd**, as shown in *Figure 6.16* on the **Fill** tab, so that the overlapping bits get reversed out:



H: 0
S: 0
L: 70
A: 100

H: 0
S: 0
L: 70
A: 100

Figure 6.16 – Changing Fill Rule to remove overlapping parts

This is handy if we want to make a letter *A*, for example, with a cutout portion represented by a triangle path inside the exterior path. Setting the mode to **Even/Odd** will cut out that part for us, as shown in *Figure 6.17*:

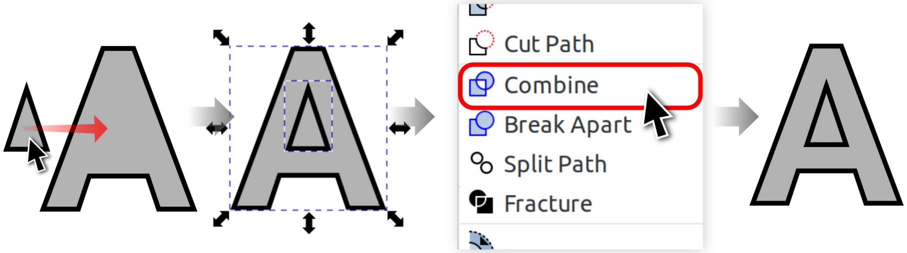


Figure 6.17 – Using the Combine path operation to turn two shapes to make an "A"

Depending on how you draw that triangle, it may or may not knock out that hole. If it doesn't, you probably drew it clockwise instead of counterclockwise. Why does this matter? Well, Inkscape takes three factors into account when deciding whether to knock out a hole in a shape or fill it:

- **Overlap**: Does the path overlap another path in the shape?
- **Draw Mode**: Is **Draw Mode** set to **Even/Odd**?
- **Path Direction**: Does the path flow in the opposite direction to the overlapping path (clockwise or anti-clockwise?)

Why all three? Put simply, there may be cases where you may want some overlapping paths in a shape to be holes and others to be filled. These three criteria ensure you have control over which paths do and do not knock out holes in your shape.

It's kind of a pain, though, right? I mean, having to remember which direction you drew a path in to get the proper result is a bit cumbersome. Well, there are two easy solutions to turn your filled paths into holes.

One is to select a node on your triangle and choose **Path** > **Reverse**. This will reverse the path and knock out your hole, providing the shape overlaps, and set the draw mode to **Even/Odd**.

The second way is to just use the **Difference** path operation on the two shapes instead of **Combine**, as we did earlier in this chapter. **Difference** will automatically combine your paths, set **Draw Mode** to **Even/Odd**, and set the path's direction to the opposite of the containing shape.

So, why not just use **Difference** all the time for holes? This is precisely what I recommend. However, if your holes aren't acting like holes, you know how to fix them. This may save you some head-scratching and re-drawing work at the very least.

So, now we know all about cutting shapes from other shapes and overlaps in our combined paths, but what if we want to do the reverse and break apart all the paths in our shape into separate shapes? For this, we can turn to our trusty **Break Apart** command directly under **Combine** in the **Path** menu.

Figure 6.18 shows a smiley face made of four shapes, three of which are holes. When we break these shapes apart, notice that the holes assume the same fill as the outer shape. This is because they no longer overlap the circle within the path, and therefore fail the overlapping requirement, as they are now being shapes:

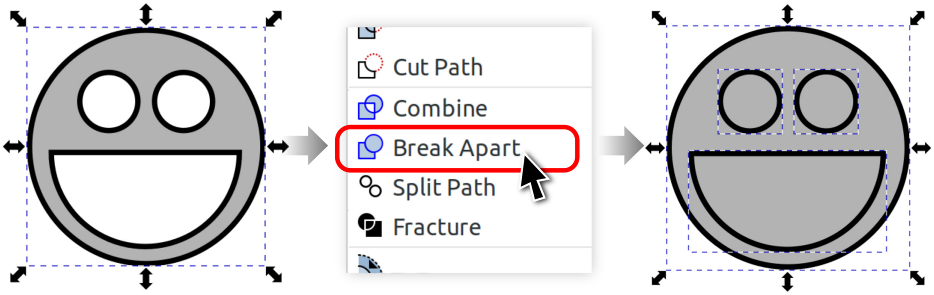


Figure 6.18 – Using the Break Apart path operation to break the mouth and eyes into separate shapes

What happens if we have three faces combined into one shape? Maybe we'd like to break them apart, but keep the holes intact. A brand new feature, as of Inkscape 1.2, is the **Split Path** operation. This will only break apart shapes that are not overlapping, as shown in *Figure 6.19*:

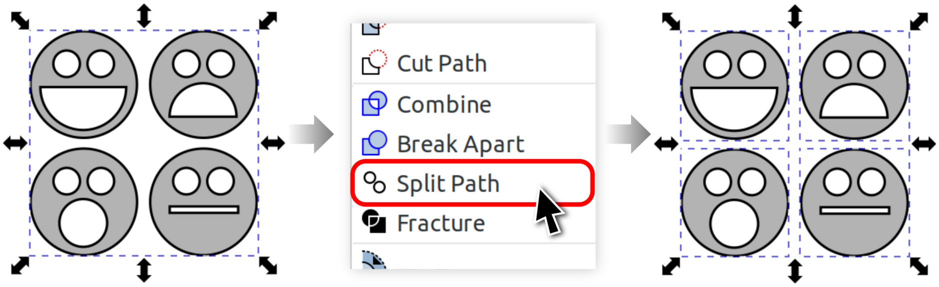


Figure 6.19 – Using the Split Path operation to split four faces into separate objects

In previous versions of Inkscape, you had to first use **Break Apart**, and then select each face with eyes and a mouth and perform a **Combine** path operation. So, you can see how this new feature saves you a lot of work.

There's one more (brand new as of Inkscape 1.3) function: the **Fracture** operation. Sometimes, we'd like to make every single overlapping part its own object. That's what **Fracture** does! *Figure 6.20* shows a set of six overlapping circles and what happens if we select all of them and use **Path** > **Fracture**:

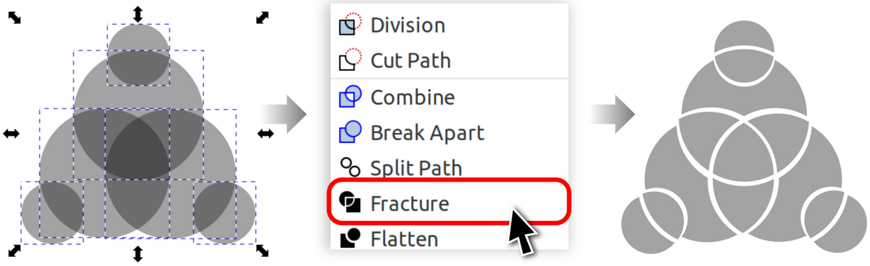


Figure 6.20 – Using the Fracture operation on overlapping shapes

As you can see, it breaks apart all the overlapping parts into their own shapes (I've spread them apart a little for visibility).

There are times, though, that we'd like to remove overlapping parts and in a sense *flatten* our shapes, leaving only what we can see when the shapes are fully opaque.

New in Inkscape 1.3 is the **Flatten** path operation (**Path** > **Flatten**). To demonstrate, let's revisit our overlapping circles, but this time in full opacity. if we select all our circles, and use the **Flatten path** operation, we get something that looks exactly the same as it did before the operation, but when we move the shapes apart, we can see that Inkscape has removed all overlapping parts, as shown in *Figure 6.21*.

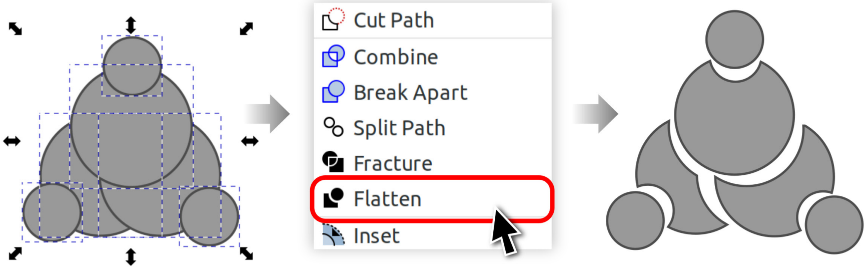


Figure 6.21 – Results (spread apart) of the Flatten path operation

This is particularly good for logo work, where you don't typically want overlapping shapes. It makes it much easier to use your illustrations for screen printing, and CNC cutting machines that require overlapping parts to be removed. Now Inkscape will do it for you!

All of these operations are great time-savers. All praise and glory to the Inkscape developers! That's not all, though! Let's have a look at some other useful path operations.

Exploring the Inset, Outset, and Dynamic Offset Path operations

Thus far, we've been learning ways to combine and break apart paths, but what if we want to expand a path outward some distance? We can do this easily with a circle just by holding *Ctrl* + *Shift* while scaling. For more complex geometry, we quickly get overlapping portions, since the shape is scaled relative to the center of the shape, and not evenly on all sides. *Figure 6.22* shows the result of duplicating and scaling operations on a circle versus more complex shapes:

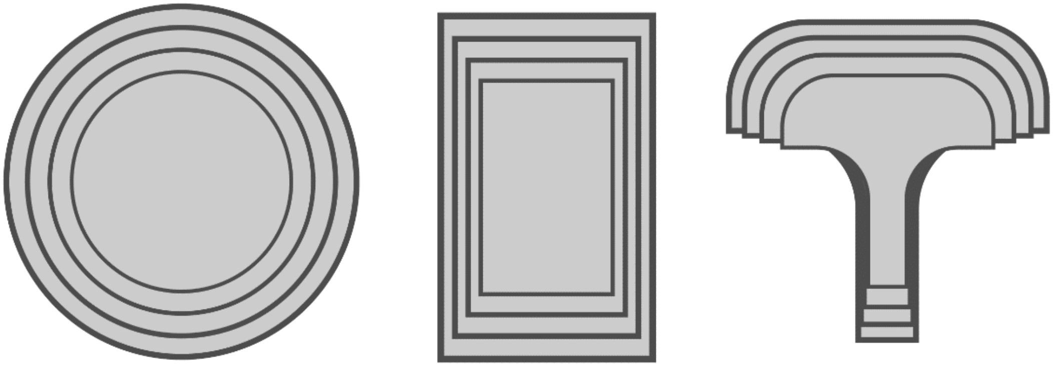


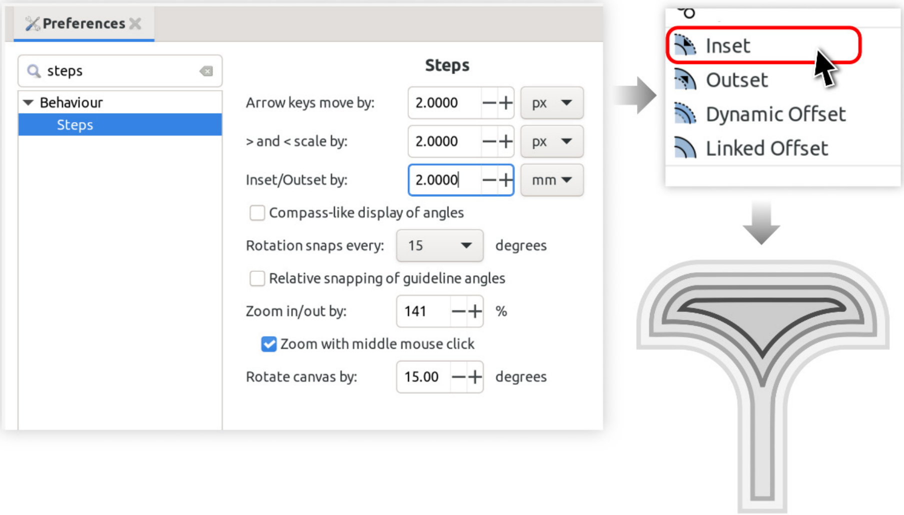
Figure 6.22 – The result of duplicating and scaling shapes

As you can see, the circle works reasonably well, but as soon as there's any variation in the shape, we run into problems. Take the rectangle, for example – it's also a simple shape, but as we scale it down uniformly, the top and bottom scale down faster than the sides.

Finally, in the case of the mushroom-like shape on the far right, we can see that anything we cut out of that shape is going to distort and even overlap when performing the same scaling operation. So, how do we scale uniformly inwards from all sides, to get the same effect as our circle? **Inset** to the rescue!

The **Inset** path operation scales inwards uniformly from each point on the shape. All you need do is select the shape and choose **Path** > **Inset**… well, almost. **Inset**, by default, will inset `2 px` at a time. This is so you can use the hotkey, which is *Ctrl* + *(*, to incrementally increase the inset. We're not stuck with `2 px` as a value, though. For our example, let's increase that value to `2mm`.

To do that, we need to open **Edit** > **Preferences**, and then type `steps` into the search box to get to the **Steps** options. We can then set the **Inset/Outset** steps to `2mm`. Now, when we tap *Ctrl* + *(*, it increases our inset or outset by `2mm` at a time, as shown in *Figure 6.23*:



Preferences

steps

Behaviour

Steps

Steps

Arrow keys move by: 2.0000 px

> and < scale by: 2.0000 px

Inset/Outset by: 2.0000 mm

☐ Compass-like display of angles

Rotation snaps every: 15 degrees

☐ Relative snapping of guideline angles

Zoom in/out by: 141 %

☑ Zoom with middle mouse click

Rotate canvas by: 15.00 degrees

Figure 6.23 – Setting Steps in preferences and using Inset on our mushroom shape

We can use the same process if we want to expand or **Outset** our path. We can use the *Ctrl +)* hotkey or choose **Path** > **Outset**, as shown in *Figure 6.24*:

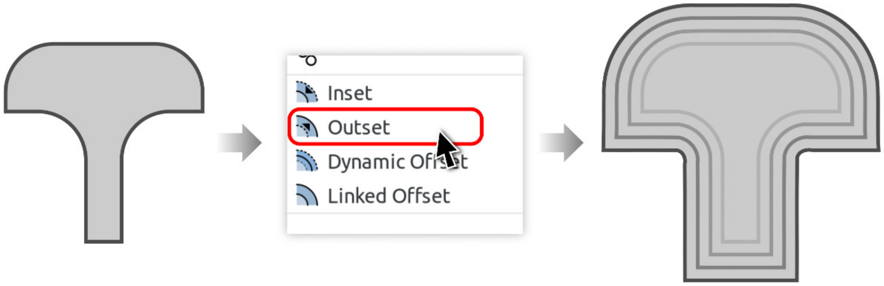


Figure 6.24 – Using Outset to expand a shape

This process works okay for set increments, but often, we'd prefer to be able to inset or outset to some increment without needing to change the **Steps** value. This is where **Dynamic Offset** comes in useful.

When we select our shape and choose **Path** > **Dynamic Offset**, we can then switch to the **Node** tool and Inkscape will replace the nodes with a convenient handle for dragging inwards or outwards to adjust the inset and outset. *Figure 6.25* shows this process and the results:

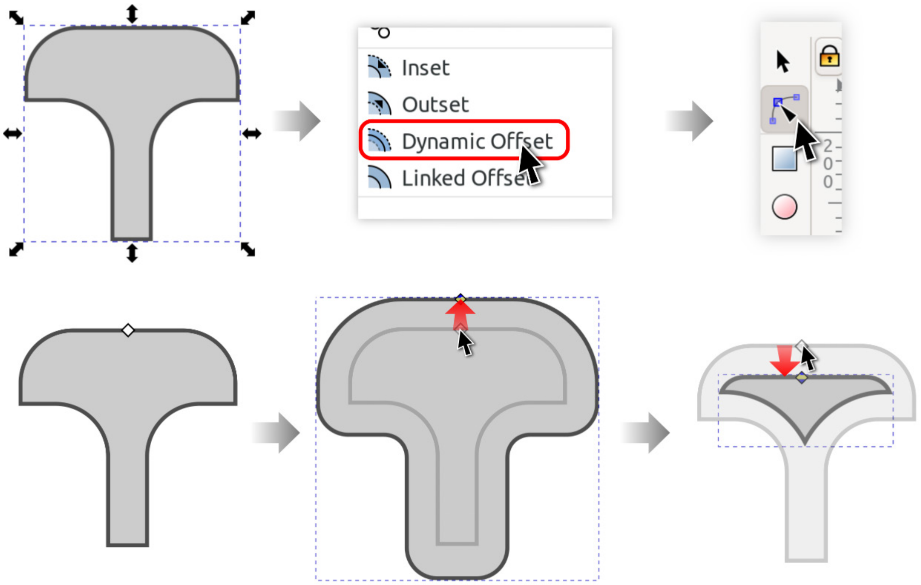


Figure 6.25 – Using Dynamic Offset to expand or contract a shape

I use **Dynamic Offset** by default for insetting and outsetting. Generally, the visual result is more important than the precise offset measurement in most of my projects.

There is another option, which we will explore later when talking about **Live Path Effects**. It's worth mentioning here that you can select your shape, choose **Path** > **Path Effects**, click the + button at the bottom of the **Path Effects** dialog, and choose **Offset**.

This will add an **Offset Path** effect to your shape and provide a panel where you can set the miter type, and offset in a variety of different measurements without the need to use **Outset** or **Inset** incrementally, as we did before. Best of all, it's non-destructive, so you can change the value at any time without affecting the original geometry.

Before we move on completely from offsets, there's one more type to discuss: **Linked Offset**. **Linked Offset** is similar to **Dynamic Offset**, but leaves the original path alone, making a copy of it with a dynamic offset handle on the copy. Additionally, whatever you do to the original path is then linked to the dynamic offset copy. That is to say, when you edit the original shape, **Linked Offset** is affected too, as shown in *Figure 6.26*:

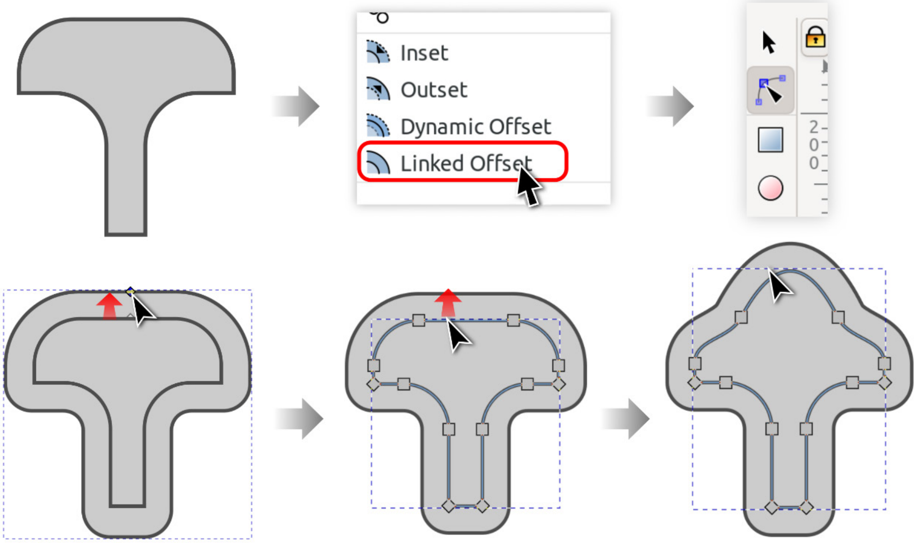


Figure 6.26 – Setting Linked Offset on a shape – editing the original moves the offset as well

Another useful operation is the **Simplify** path operation. We can use this to get rid of extra nodes that may result from our boolean operations. Take, for example, *Figure 6.27*, which shows the result of a difference operation that resulted in too many extra nodes. Using **Simplify**, Inkscape can get rid of some of these extras, making it easier to edit the shape:

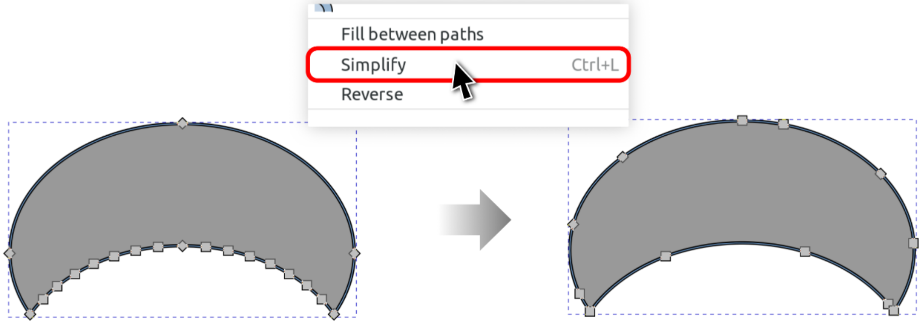


Figure 6.27 – Using the Simplify path operation to decrease the number of nodes in complex geometry

Note that this process has some flaws, as the symmetry of the remaining nodes seems to be lost. This is a known issue at the time of writing and is a problem to be fixed in the future. There also is no way to selectively reduce only part of the path, other than to select nodes and hit the *Delete* key and adjust handles to compensate.

For these reasons, I recommend only using **Simplify** when necessary. Also, note that you can reduce the threshold for simplification (how many nodes it removes) by going to **Edit** > **Preferences** > **Behavior** > **Simplification Threshold**. A value of `0.0005` or less usually works well without removing too many nodes at once.

In the interest of keeping this chapter as concise as possible, I've opted to cover the rest of the items in the **Path** menu in *Chapter 12*, *Live Path Effects*, since that topic deserves a whole chapter on its own. Before we get into our practice, let's go over a new huge time-saver when it comes to building custom shapes – the **Shape Builder** tool!

Fast Path operations with the Shape Builder tool

There's a brand-new tool on the block at the time of writing this book. Inkscape 1.3 includes the much anticipated **Shape Builder** tool and, indeed, it is an excellent time-saver when building shapes that would otherwise take many different shape operations.

Shape Builder is a simple tool for making complex shapes. It's built on the idea of selecting the parts you want to keep out of overlapping shapes. *Figure 6.26* shows three overlapping black-filled circles, each set to 20% opacity so that you can see the overlapping parts.

Now, say we just want some of the overlapping shapes and not others. By selecting the three circles, and switching to the **Shape Builder** tool, we can simply click the shapes we want and hit the *Enter/Return* key to end up with just those chosen shapes with all others removed, as shown in *Figure 6.28*:

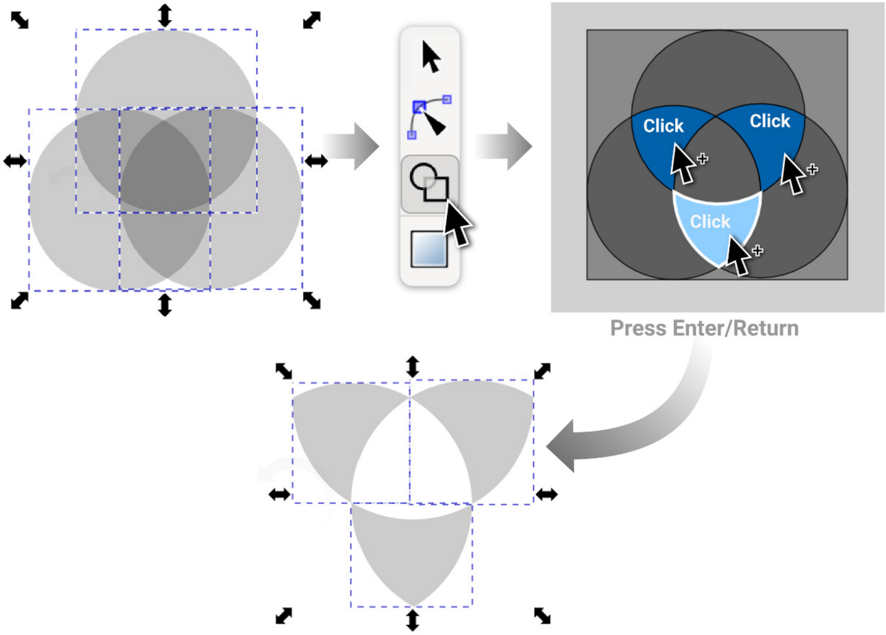


Figure 6.28 – Using the Shape Builder tool on three circles

To start building complex shapes, I highly recommend using just one fill color and setting **Opacity** to 20% on all your construction shapes. This will let you build up stacks of shapes and see where they overlap. *Figure 6.29* shows a (very messy) car that I made with simple overlapping shapes. As with our circles, I can select all the shapes, click the **Shape Builder** tool in the toolbar, and Inkscape will kick us into shape-building mode:

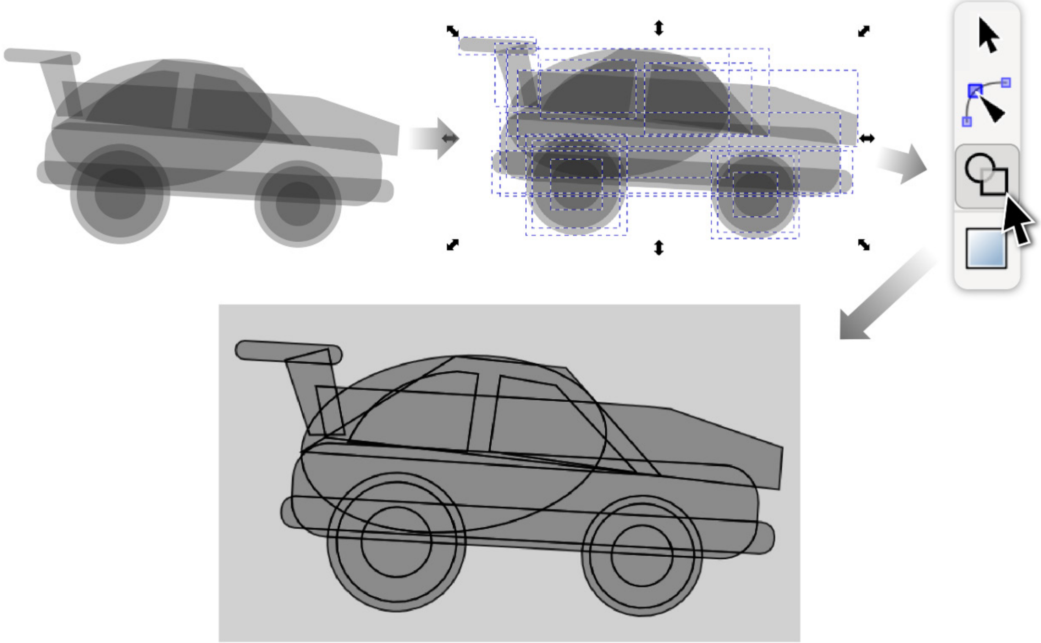


Figure 6.29 – Selecting shapes and activating the Shape Builder tool

When we move the mouse over the overlapping parts of our car, they will be highlighted, and we can simply click or click-drag over shapes to highlight what we want to keep or remove.

This works in the following ways:

- **Single click**: Add a shape to keep, but don't union it to surrounding shapes. This will result in the clicked-on shape being kept separate from the other shapes.
- **Click and drag from one shape to another**: This will union those shapes together in the final result.
- **Shift-click, or Shift-click-drag**: This removes shapes from the result.

Thus, we'd like to merge all the shapes in one of our tires, but keep the tires as separate objects in the result. We would start by clicking and dragging over all the shapes in the first tire, and then the second, as shown in *Figure 6.30*:

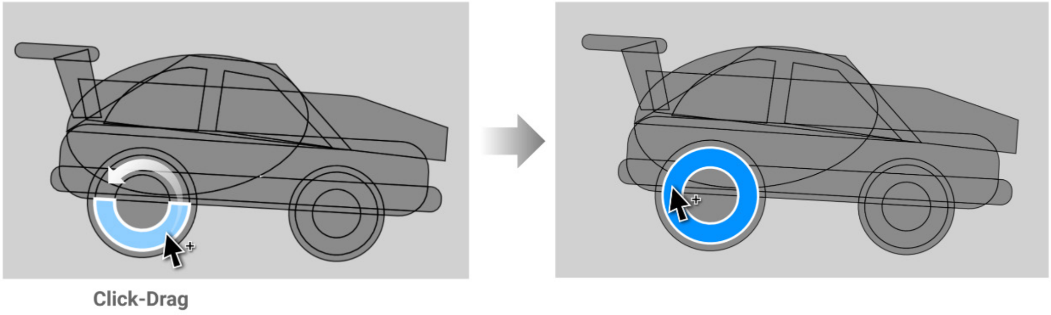


Figure 6.30 – Drag-selecting to merge all shapes in the wheel into one shape

We can also add the body of the car without the windows or extra bits of construction shapes by clicking, dragging, and starting at the tail, and moving the cursor around all the parts we want to add. We can even add the other wheel, and then just press the *Enter*/*Return* key or use the green button by **Finish:** in the Tool control bar to get our three-part car, as shown in *Figure 6.31*:

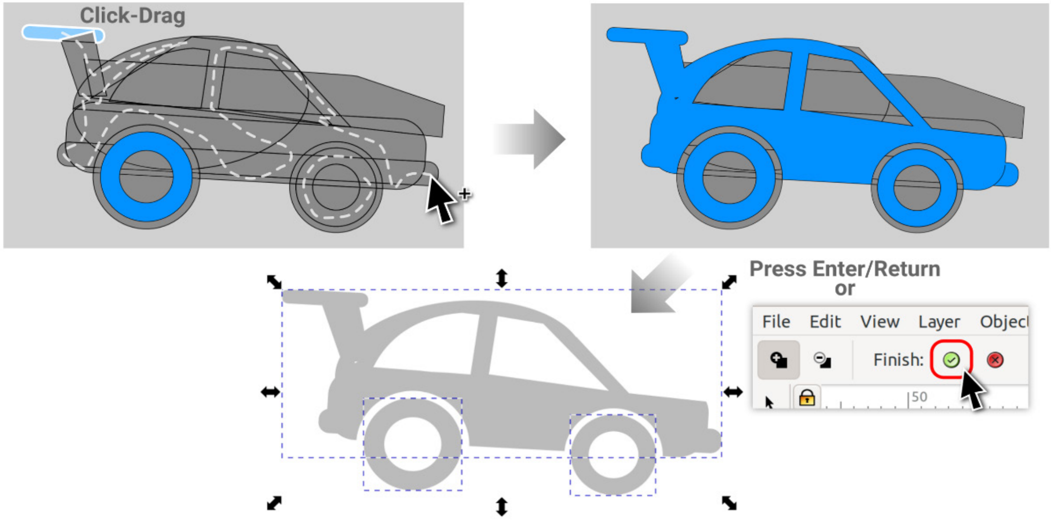


Figure 6.31 – Click-dragging to finish building our car shapes

Note that you can also remove parts in a subtractive process if that's easier. We can either hold *Shift* while clicking or click-dragging areas of the car to remove them, or we can use the **Delete** toggle in the Tool control bar to switch from adding shapes to deleting them, as shown in *Figure 6.32*:

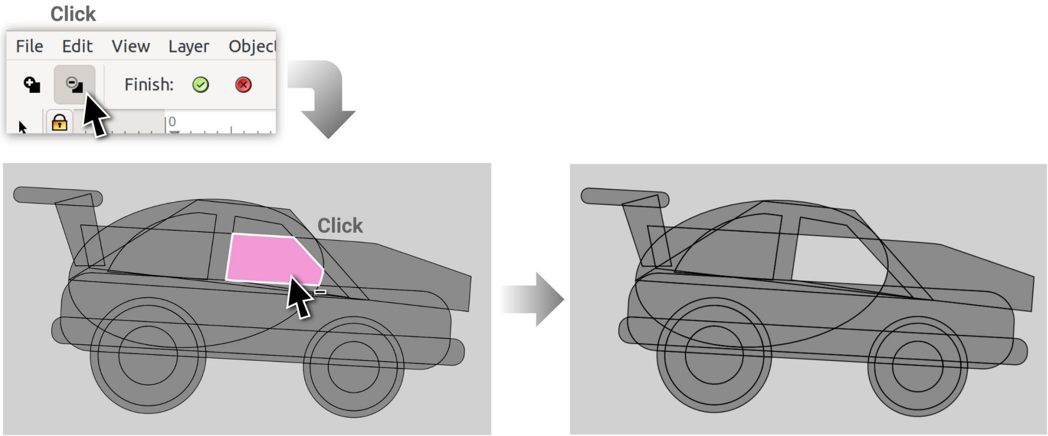


Figure 6.32 – Toggling Delete mode to subtract instead of add

While you can technically use both **Add** and **Delete** modes at the same time, it's better to choose one or the other based on whether you need to subtract or add from your overlapping shapes to get the fastest results, and just hit *Enter* or the green **Finish** button rather than wasting time going back and forth erasing and adding things during the same shape building operation.

It's also worth noting that you don't have to drag through every shape in the car body all at once to union them. You can click-drag over the tail first, release it, and then drag it from the tail to the back of the car to continue merging. Doing this incrementally has two advantages: it gives your wrist a break, and it lets you undo (*Ctrl* + *Z*) incrementally without blowing away all your careful selecting if you make a mistake!

And that's the **Shape Builder** tool! Now that we have a good grasp of the many ways to create custom shapes quickly, let's get into some practice using them.

Try it yourself

With all these great shape operation tools, we can now save ourselves loads of time making custom shapes. In this practice lesson, we'll be making some delightful non-edible mushrooms on a cloudy day, as shown in *Figure 6.33*:

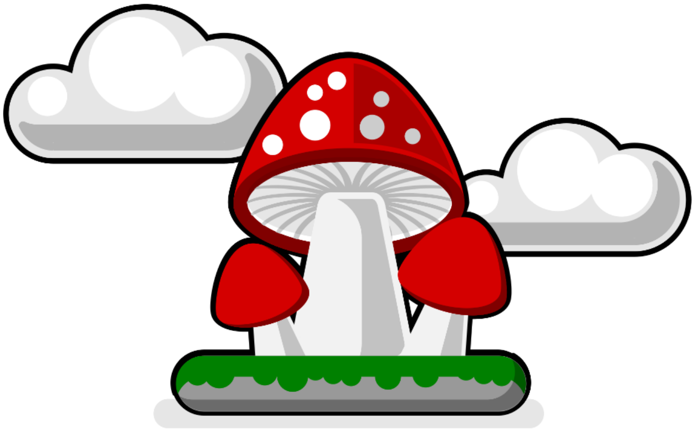


Figure 6.33 – An attractive set of non-edible mushrooms

We'll start our illustration with the clouds, which are made of circles of various sizes and a rectangle. *Figure 6.34* shows the circles, drawn with the **Circle** tool, then a rectangle drawn with the rectangle tool:

1. Simply select the circles and the rectangle and add them together with the **Union** operation in the **Path** menu or by pressing *Ctrl* + *Shift* + +:

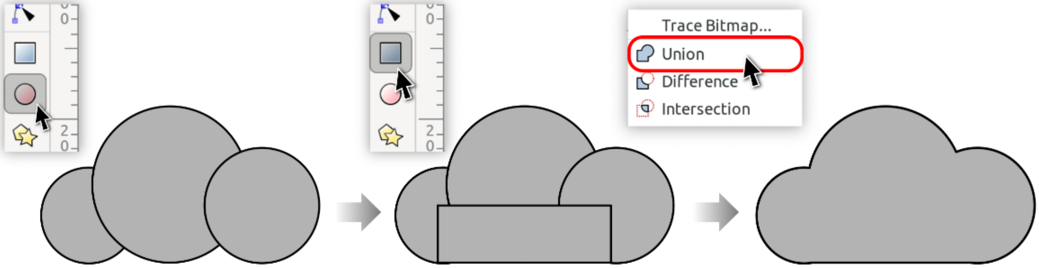


Figure 6.34 – Drawing our cloud shape with the Union path operation

2. Once we have that base shape, we can make a shadow by copying the cloud (*Ctrl* + *C*) and selecting **Edit** > **Paste in place**. This will make a copy directly on top of our cloud, which we will move up and to the left.
3. Then, we must simply select both the top and bottom cloud and perform a **Difference** operation with **Path** > **Difference** to get our shadow shape. We can then turn the shadow shape black, remove the stroke, set **Opacity** to about `20`, and then perform another **Paste in place** operation, to paste another full cloud copy over the shadow.

4. We can then move it under the shadow with the *Page Down* key, or the **Lower Selection** button in the Tool control bar. *Figure 6.35* shows this process:

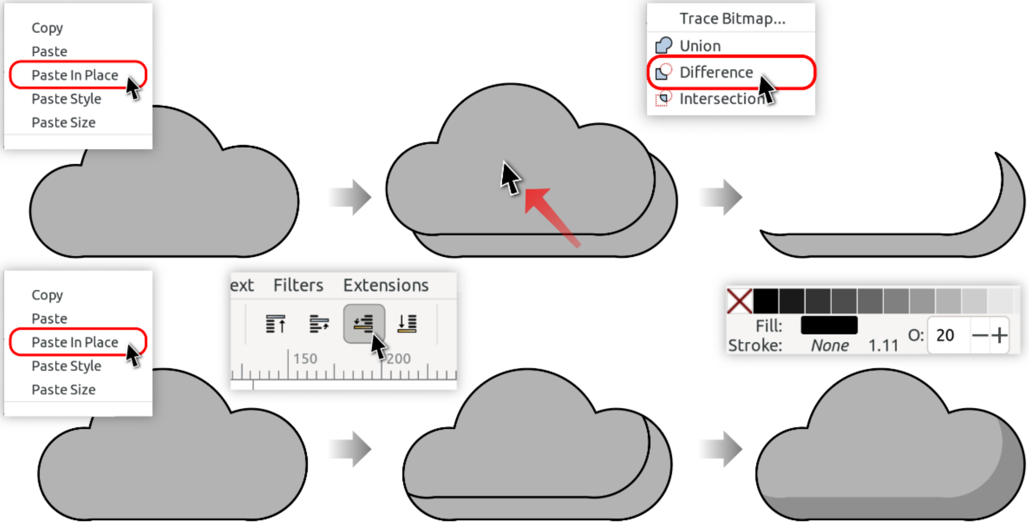


Figure 6.35 – Using the cloud shape to make the cloud shadow

We'd like to make the shadow a bit smaller so that there's some light gray of the cloud around it.

5. We can use **Dynamic Offset** to do this. Just select the shadow and choose **Path** > **Dynamic Offset** (or press *Ctrl* + *J*), then switch to the **Node** tool and pull the handle inwards. This will make a thin and rather sharp shadow, which we'd like to round out a little. We can do this by choosing **Path** > **Dynamic Offset** again and dragging the handle back out a little, as shown in *Figure 6.36*:

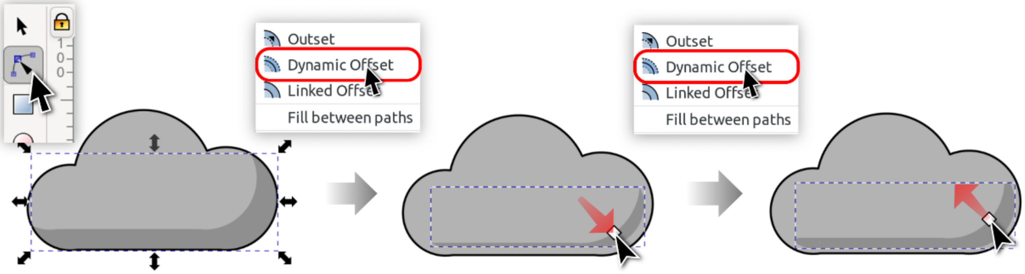


Figure 6.36 – Using Dynamic Offset to style our shadow

6. All we need to do now is add some white circles over our cloud for the highlights, and we will have our finished cloud, as shown in *Figure 6.37*. Group all the shapes for ease of selection and movement:

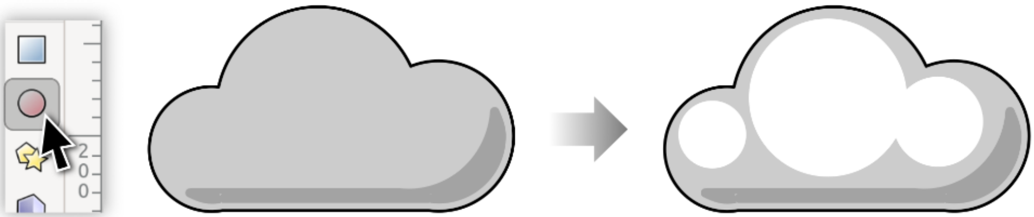


Figure 6.37 – Adding some circles for highlights

Next, we'll make our mushroom cap.

7. Start by making a red oval with the **Circle** tool, then convert it into a path with **Path** > **Object to Path** and copy it (*Ctrl* + *C*). Then, move the top node up while holding the *Ctrl* key to lock it vertically while dragging.

8. You can see this makes a rather nice shape for our cap, and we can then select **Edit** > **Paste in Place** to paste or original oval back over the base of it and make it a darker shade of red. This process is shown in *Figure 6.38*:

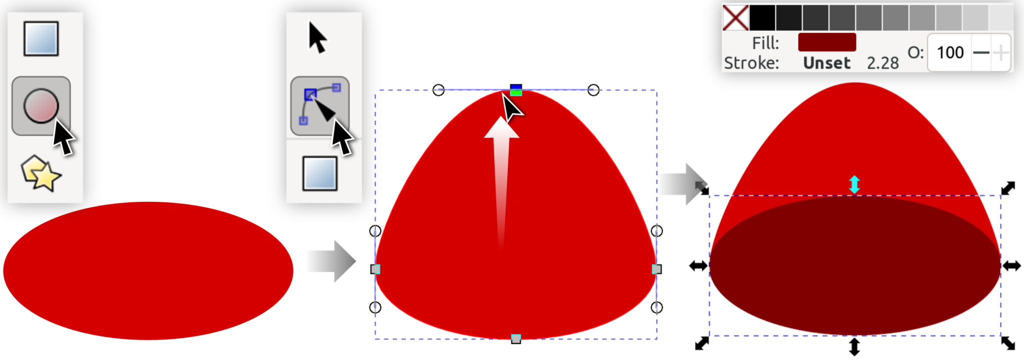

Fill:
Stroke: Unset 2.28
O: 100

Figure 6.38 – Making the top of the mushroom cap

9. Now that we have a nice initial shape, let's make the white underside of the cap by selecting **Paste in Place**, which will paste in our original oval. We can then color it white and use **Path** > **Dynamic Offset** (or press *Ctrl* + *J*) to shrink it inwards evenly, as shown in *Figure 6.39*:

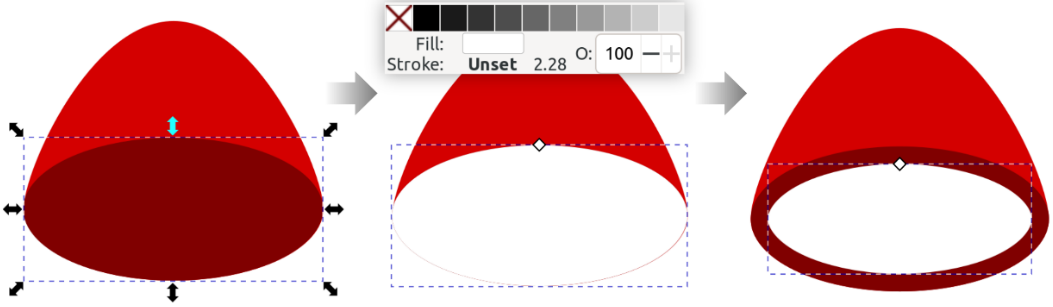


Figure 6.39 – Making the underside of the mushroom cap with Dynamic Offset

10. Let's make some gills! We can do this by first copying and pasting two copies of our oval. We'll then color them gray and give them a thin stroke so that we can see what's going on better. Then, using the **Difference** path operation (press *Ctrl* + -), we can make a curved sliver.

11. We can then click again on the resulting shape to get our rotation handles, move the origin to one end, and **Copy/Paste in place** (*Ctrl* + *Alt* + *V*) using the *[* key to rotate each pasted copy incrementally. Do this 12 times, and then **Union** them together to form a fan shape. Then, duplicate the result and flip it around to produce a mirror of the fan shape, and **Union** it with the first, as shown in *Figure 6.40*:

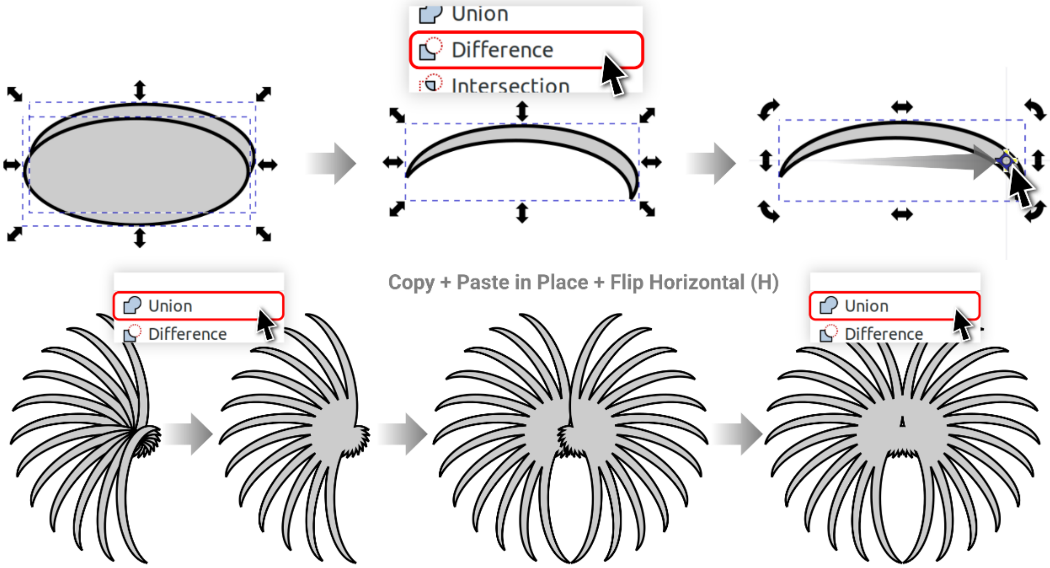


Figure 6.40 – Making the gills using the Difference and Union path operations

12. Now, all we need to do is shrink the new shape a bit, and flatten this circle of gills into an oval. We can then use a copy/pasted version of our white oval to trim this down into the underside of our mushroom cap. We can do this with the **Intersection** path operation, as shown in *Figure 6.41*, and then remove the stroke:

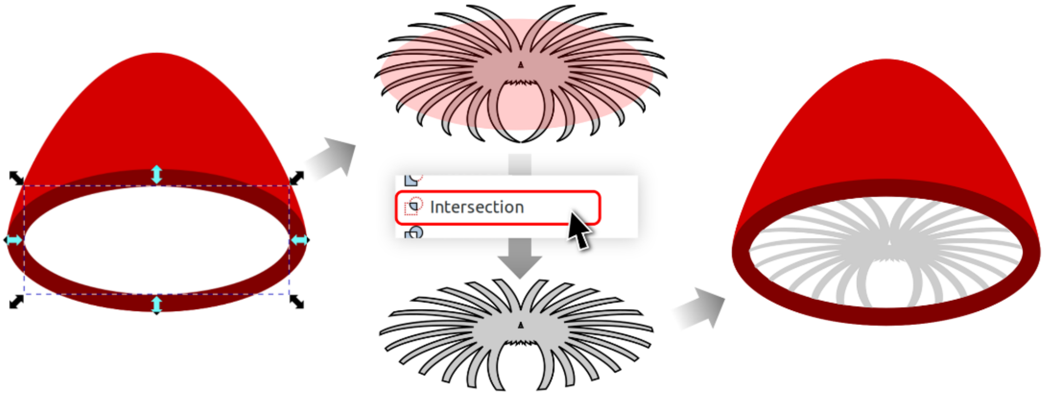


Figure 6.41 – Finishing the gills and cropping them to the white oval using the Intersection path operation

Since we are starting to get the hang of our shape operations, let's speed things up a bit. *Figure 6.42* shows how to make the remaining shapes in our mushroom illustration, and which operations were used for the shadow on the mushroom cap, the stalk, and the rock with moss.

Note that I have given some of the path operation shapes a red color with `30%` **Opacity** so that it's easier to see what's going on:

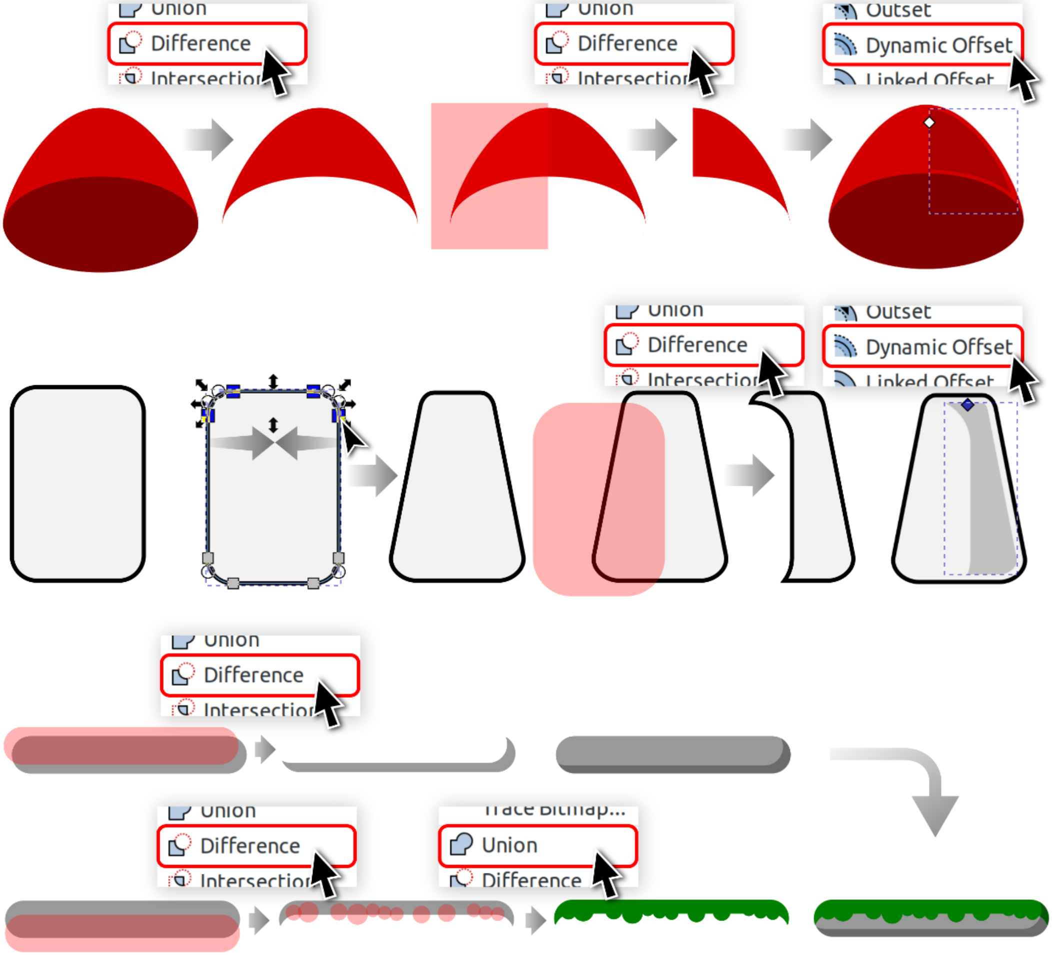


Figure 6.42 – Making the shadow on the cap, the mushroom stalk, and the stone base with moss

Reuse some shapes that we've made already for the other mushrooms; we can now bring all the pieces together to form one illustration. But before we finish, we'd like to add a black outline to the whole mushroom illustration so that it matches our clouds.

13. We can easily do this by copying/pasting all the shapes in the illustration, ungrouping any groups with **Object** > **Ungroup**, or pressing *Ctrl* + *Shift* + *G* (we may need to do this a few times to make sure all our shapes are ungrouped), and then use **Path** > **Union** to make them a single shape.

14. Then, we can color it black, move it behind our original mushroom illustration, use **Path** > **Dynamic Offset** (press *Ctrl* + *J*), and switch to the **Node** tool as before to pull out a thicker black border, as shown in *Figure 6.43*:

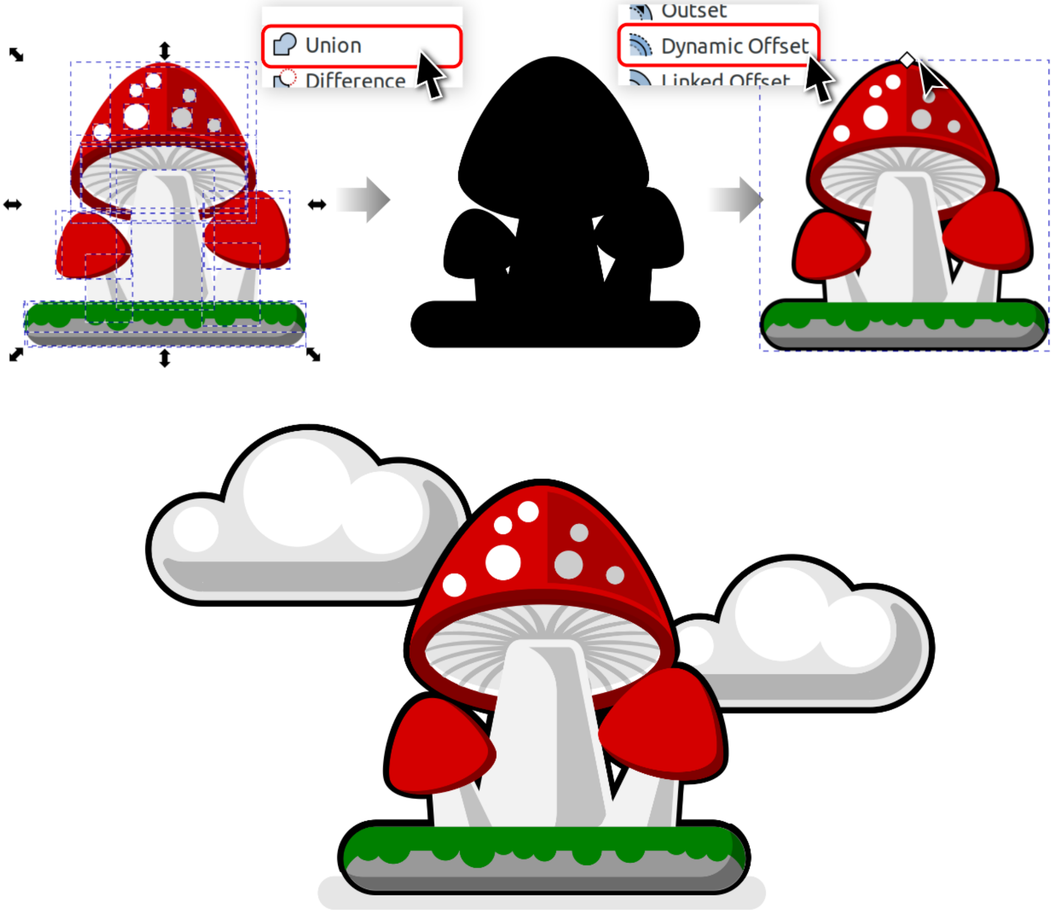


Figure 6.43 – Using Ungroup, Union, and Dynamic Offset to make our black outline

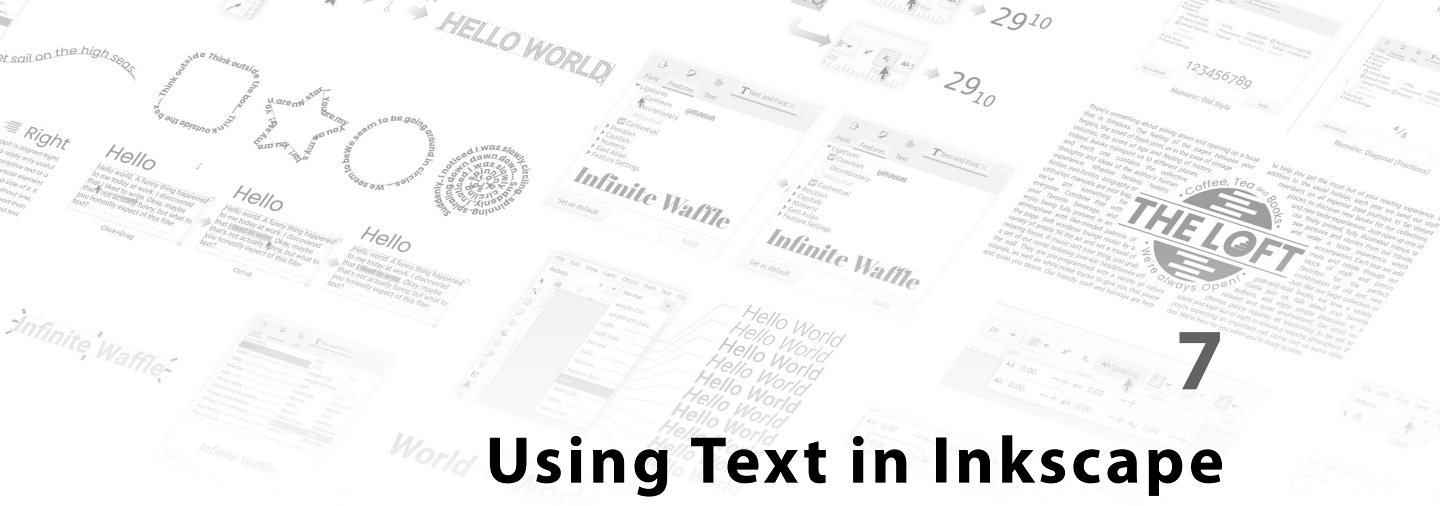
15. Slide in those clouds and maybe add a black rounded corner rectangle at `20%` **Opacity** to the base of the rock for a completed mushroom illustration!

Summary

We covered a lot of useful tricks in this chapter. We've gained the ability to add, subtract, combine, and break apart shapes with the boolean operations, and set offsets with the offset path operations. We even learned a bit about how Inkscape makes holes in things with reversed paths.

All this will serve us well in our vector illustration work for making fast custom shapes without all the manual labor of drawing them from scratch. But what about letters and numbers? Even with great tools such as the Booleans and offsets, it would take ages to make a title or caption, and eons to make a whole paragraph of text!

So, let's leave custom shapes behind for the moment and move on to using the **Text** and **Font** tools in the next chapter.



7

Using Text in Inkscape

We've been making a lot of custom shapes up to this point. Now, we're ready to start adding some text to the mix. This could take the form of fancy titles and labels for our graphics, or entire paragraphs of text. Inkscape can do it all, and this chapter is dedicated to adding and styling text in its various incarnations.

We will cover the following topics in this chapter:

- Installing new fonts
- The **Text** tool
- Font Collections
- The **Text and Font** dialog
- Understanding variable fonts
- Text on a curve
- Flowing paragraphs into shapes
- Spell check and other handy **Text** tools

Installing new fonts

Throughout this chapter, I'll be using a variety of fonts to demonstrate, but where did I get all these wonderful fonts and how did I get them into Inkscape?

At the time of writing this book, I generally get all my fonts from Google's massive font website: `https://fonts.google.com/`. The reason for this is that Google meticulously curates the best free and open fonts, which can be used professionally anywhere without you having to pay license fees. Additionally, the SIL Open Font License in which all of these fonts are included ensures that you can use, edit, and extend the character set for your purposes.

This ultimate flexibility is why I choose fonts from this list rather than proprietary fonts. There are thousands of excellent fonts in this archive, freely available to download. Google has a very nice site with a great search engine for finding just what you want.

As an example, let's download and install one, which we will use at the end of this chapter. I've chosen a font called **Poppins**, which is, at the time of writing, available to download from `https://fonts.google.com/specimen/Poppins` (click the **Download family** button in the top right-hand corner of the screen to download a ZIP file).

Generally, fonts can be installed by opening the ZIP file (with a double-click), then double-clicking the font file you want to install. These font files (usually ending in `.ttf`) contain different styles for the same font. For example, the Poppins ZIP file contains `Poppins-Bold.ttf`, `Poppins-Thin.ttf`, `Poppins-Black.ttf`, and so on.

Double-clicking one of these files will usually bring up a font viewer showing you a preview of what the font looks like, accompanied by an **Install** button, which you can then click to install the font. After installing, you'll be able to choose these styles from a dropdown next to the font name, as you will see later in this chapter.

You can also extract font files from the ZIP file directly to your system's font folders to install them. If this isn't working for you, then a quick search on the web for *How to install fonts on Windows*, for example, would give you the most up-to-date information on the process for installing fonts in your version of Windows. The same goes for macOS and Linux.

To use your new fonts in Inkscape, you must restart Inkscape to see the newly installed fonts in your font list, since the font list is only refreshed when Inkscape starts up.

The Text tool

Like all the tools we've discussed so far, the **Text** tool is located in the Tool control bar on the left. By activating it, we get a little cross cursor with the letter *A* beside it, and we can now click anywhere on the canvas to start writing text. Inkscape gives us a little blinking cursor you're probably familiar with in other writing tools. When this blinking cursor is showing, all we need do is type out our text on the keyboard, as shown in *Figure 7.1*:

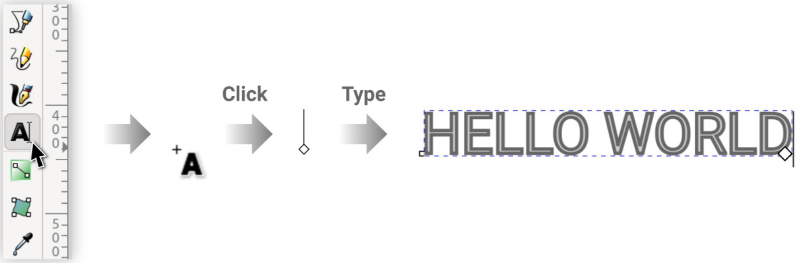


Figure 7.1 – Using the Text tool to type our first text

Note that, by default, Inkscape has made our text the same color for **Fill** and **Stroke** that we were using before in our other shapes. We'd like just black text for the moment, so we can simply click on the black swatch at the bottom to make the fill black, press *Shift*, and click the red X swatch to unset the stroke color, as shown in *Figure 7.2* (you can also do this in the **Fill and Stroke** dialog if you have it up):

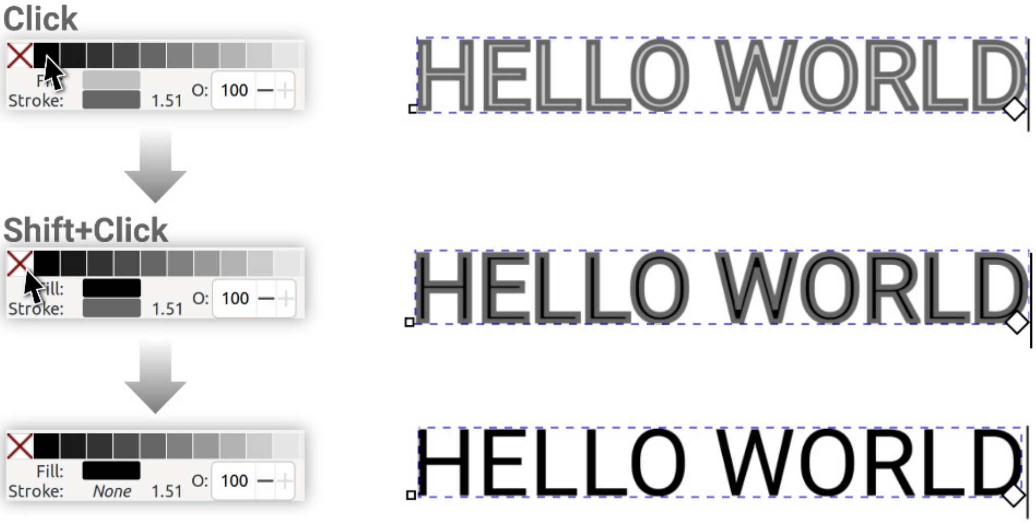


Figure 7.2 – Changing the Fill and Stroke properties of our text

As in other text editing programs, we can hit the *Enter* key to write more text on a line below. Sometimes, we'd like the text to wrap around automatically to the next line at a certain point. You may have noticed the little diamond handle in the lower right corner of our text. That handle controls the wrapping, as shown in *Figure 7.3*; simply drag that handle to set the wrapping limit:

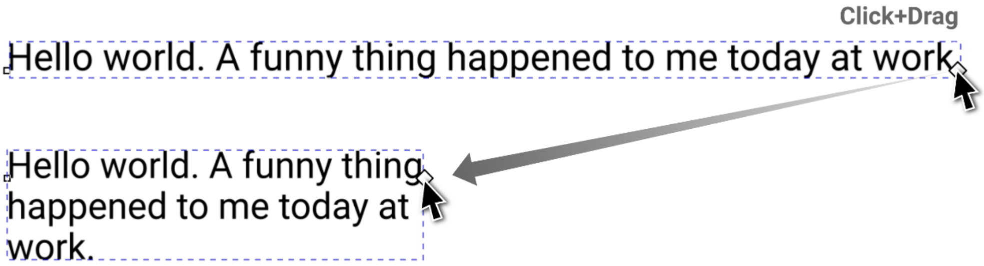
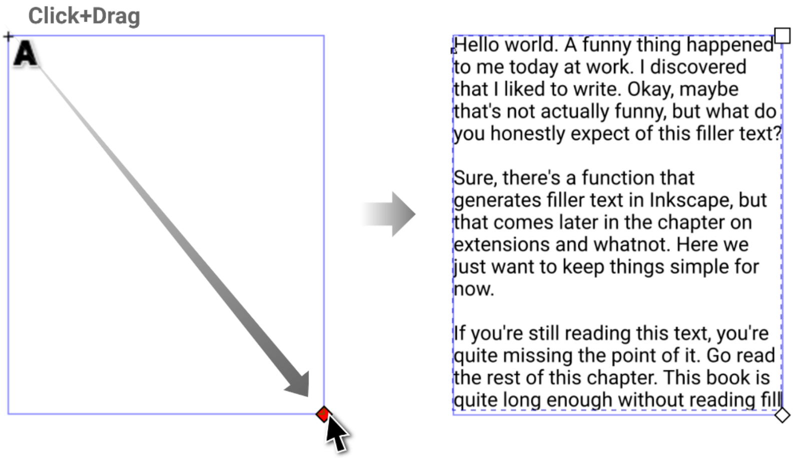

Click+Drag
Hello world. A funny thing happened to me today at work.
Hello world. A funny thing
happened to me today at
work.

Figure 7.3 – Moving the text wrapping handle to auto-wrap the text

This works pretty well, but maybe we'd like to define the wrapping area and bounds of a paragraph before we start typing. To do this, we can simply click and drag the **Text** tool on the canvas to make a rectangle for our text to flow into. Then, when we type, it will wrap the sentences to that rectangle, as shown in *Figure 7.4*:



Hello world. A funny thing happened to me today at work. I discovered that I liked to write. Okay, maybe that's not actually funny, but what do you honestly expect of this filler text?

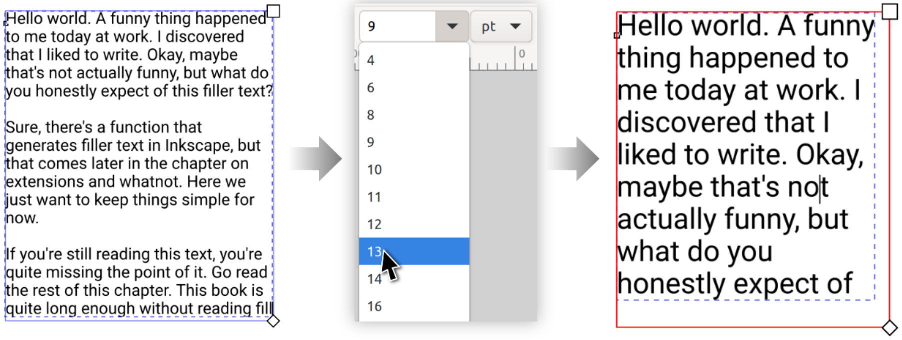
Sure, there's a function that generates filler text in Inkscape, but that comes later in the chapter on extensions and whatnot. Here we just want to keep things simple for now.

If you're still reading this text, you're quite missing the point of it. Go read the rest of this chapter. This book is quite long enough without reading fill

Figure 7.4 – Dragging out a text box with the Text tool

Note that you can resize the text box at any time by dragging that bottom-right handle. But what does the top-right handle do? If you click and drag it, you will see that it adds a margin to the text. This margin is applied evenly on all sides of the text box and is quite useful for adding some padding around your text.

Depending on the size of your text box, the text may be too big to fit inside it. You can change the size of the text using the **Font Size** dropdown in the Tool control bar. You can either use the values in the dropdown or simply click in the box and type a custom value, as shown in *Figure 7.5*:



Hello world. A funny thing happened to me today at work. I discovered that I liked to write. Okay, maybe that's not actually funny, but what do you honestly expect of this filler text?

Sure, there's a function that generates filler text in Inkscape, but that comes later in the chapter on extensions and whatnot. Here we just want to keep things simple for now.

If you're still reading this text, you're quite missing the point of it. Go read the rest of this chapter. This book is quite long enough without reading fill

Hello world. A funny thing happened to me today at work. I discovered that I liked to write. Okay, maybe that's not actually funny, but what do you honestly expect of

Figure 7.5 – Changing Font Size from 9 to 13

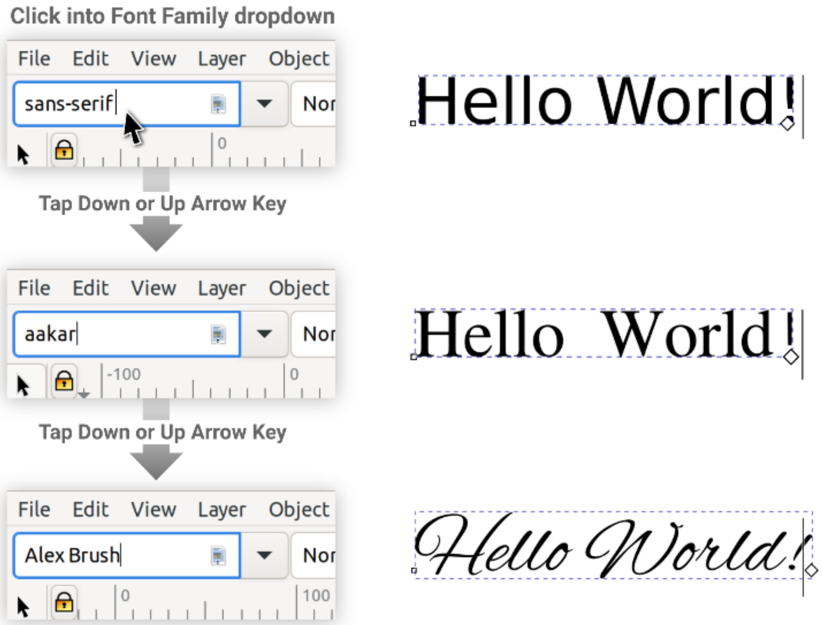
Notice that when we increased the font size from **9** to **13** all of a sudden, our text got cut off at the bottom and our text box turned red. This is because Inkscape is warning us that some text is being cut off at the bottom. This cut-off text is called **overflow**.

To get it back, we only need to resize our text box so that it fits with the diamond handle in the lower right corner. Once there is no more text overflow, Inkscape will turn the box blue once more.

Also, note that you can change the units of measurement in the drop-down directly right of the **Font Size** field. Typically, I just use the default, which is `pt` or *points*. This is the default for a document measured in millimeters or inches and assumes you want your font size to behave like in a word processor such as Microsoft Word or LibreOffice Writer. You can change these units to `mm` or `in`, however, if you need more precision size control over your fonts.

Font Size is only one of the many options in the Tool control bar for styling our text. Starting at the far left of the bar, we have our **Font Family** dropdown, which will show previews of all the fonts you have installed on your system. This makes it quite easy to choose a new font at a glance.

However, sometimes, we'd like to be able to preview the font in our document with text we've already typed. We can do this by clicking into the **Font Family** box and using the up and down arrow keys on the keyboard to change the font of the current text, as shown in *Figure 7.6*:


Click into Font Family dropdown

sans-serif

Tap Down or Up Arrow Key

aakar

Tap Down or Up Arrow Key

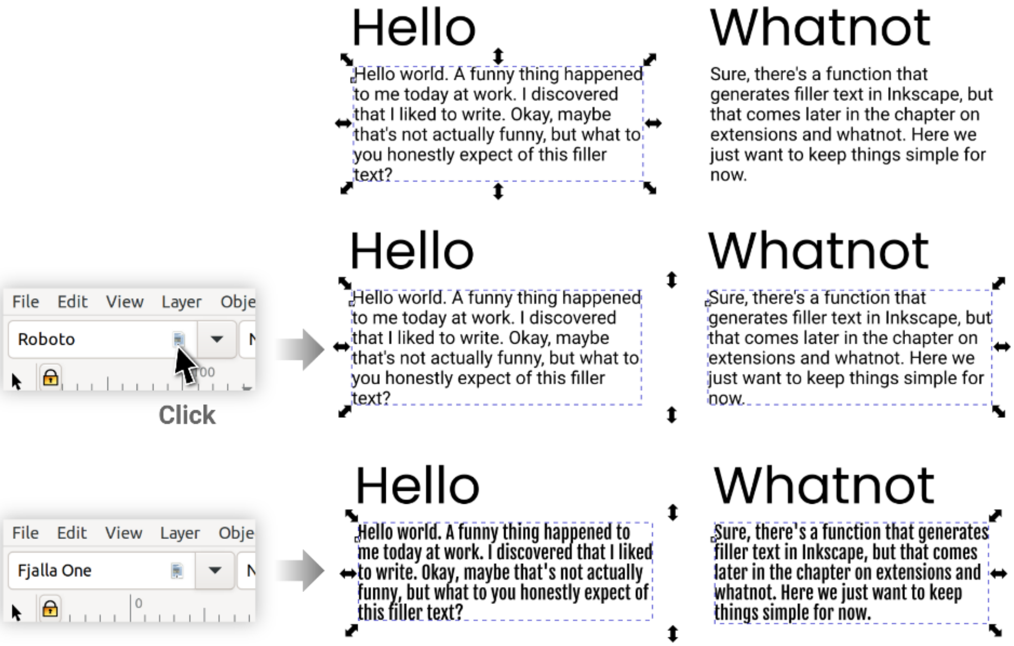
Alex Brush

File Edit View Layer Object
Nor
0
Hello World!
File Edit View Layer Object
Nor
-100
0
Hello World!
File Edit View Layer Object
Nor
0
100
Hello World!

Figure 7.6 – Changing the font on the canvas

You can also use the dropdown as a search bar if you know the name of the font you want to use. Just erase the contents and start typing, and Inkscape will narrow down the font options as you type.

There's one more neat feature of the **Font Family** dropdown, and that's the little button on the right-hand side of the field that looks like a page. Clicking this tiny button will select all text on the canvas that has the same font as what's listed in the field. This feature makes it very easy to replace fonts across all text boxes in your document. *Figure 7.7* shows this process, starting from a single selected text box:



Hello

Hello world. A funny thing happened to me today at work. I discovered that I liked to write. Okay, maybe that's not actually funny, but what to you honestly expect of this filler text?

Whatnot

Sure, there's a function that generates filler text in Inkscape, but that comes later in the chapter on extensions and whatnot. Here we just want to keep things simple for now.

Roboto

Fjalla One

Figure 7.7 – Using the Select all text with this font family button to select and then change the font

Notice in the preceding example that the title and paragraphs are different text objects. I generally keep the titles separate because it makes it easier to change the font on different objects.

Beside the **Font Family** dropdown in the Tool control bar is the **Font Style** dropdown. Depending on the font you have chosen it may have one, two, or many styles, including **Bold**, **Italic**, **Italic Bold**, and so on. *Figure 7.8* shows all the styles for the **Roboto** font, for example:

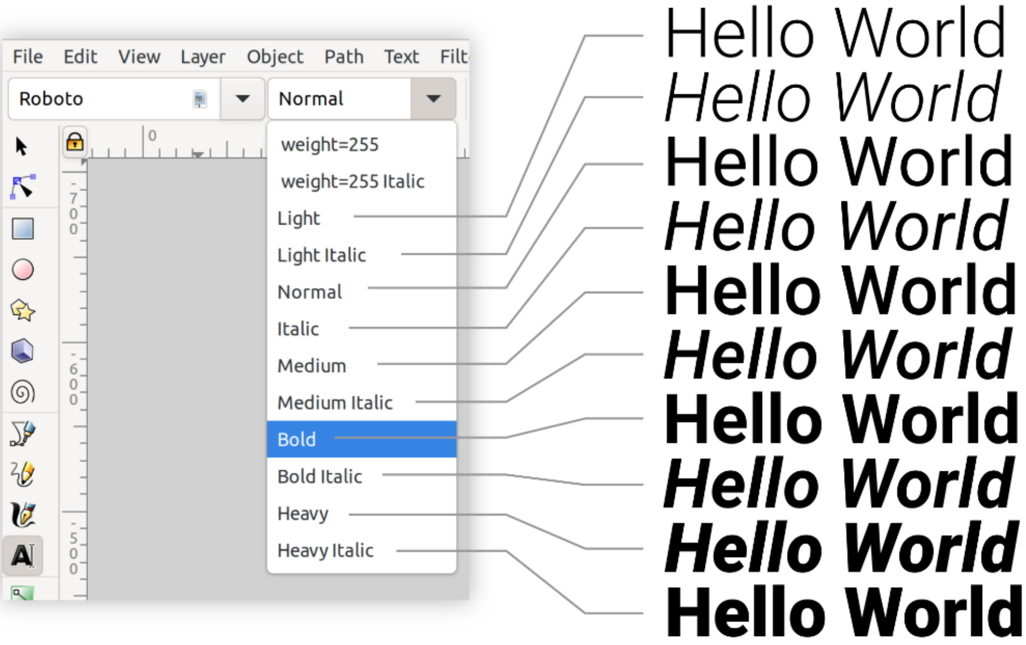
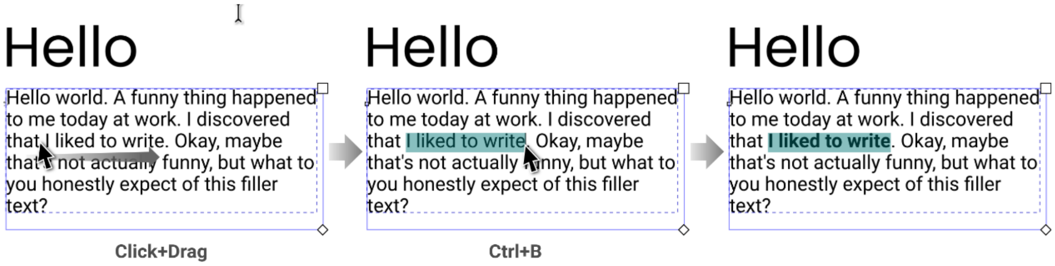


Figure 7.8 – Drop-down list of font styles for the Roboto font family

This is great for titles, but also good for making certain words in our paragraphs stand out. Thus far, we have changed the font for the entire text object, but if we highlight certain words while editing, we can change their style without affecting the rest of the text. We even have two handy hotkeys we can use to bold or italicize words.

Those are *Ctrl* + *B* and *Ctrl* + *I*, respectively. To select text, we can either click and drag the mouse cursor over the word while editing the text or hold the *Shift* key while using the forward or back arrow keys to highlight.

Then, we can either choose a new **Font Style** from the drop-down and hit *Ctrl* + *B* to bold it or *Ctrl*+ *I* to make it italic, as shown in *Figure 7.9*:



Hello

Hello world. A funny thing happened to me today at work. I discovered that I liked to write. Okay, maybe that's not actually funny, but what to you honestly expect of this filler text?

Hello

Hello world. A funny thing happened to me today at work. I discovered that I liked to write. Okay, maybe that's not actually funny, but what to you honestly expect of this filler text?

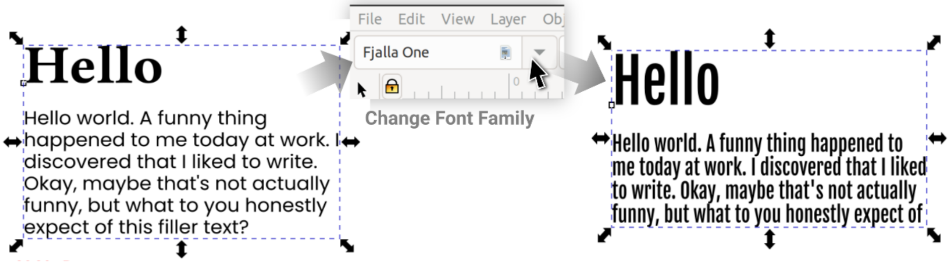
Hello

Hello world. A funny thing happened to me today at work. I discovered that **I liked to write**. Okay, maybe that's not actually funny, but what to you honestly expect of this filler text?

Filter 7.9 – Selecting text with the mouse and using Ctrl + B to bold the selected text

We can also change our **Font Family** and the color of the selected text in the same way – just select the text you want to change and click a color swatch, just as you would any other object.

This, of course, makes it possible to include your titles inside the same text object as your paragraphs. However, I recommend that you don't do this as it makes it harder to change the fonts later. This is because changing the font for the text object changes the font style for *everything* unless you have a specific bit of text selected. Changing the font for an entire text block can be seen in *Figure 7.10*:



Hello

Hello world. A funny thing happened to me today at work. I discovered that I liked to write. Okay, maybe that's not actually funny, but what to you honestly expect of this filler text?

Hello

Hello world. A funny thing happened to me today at work. I discovered that I liked to write. Okay, maybe that's not actually funny, but what to you honestly expect of

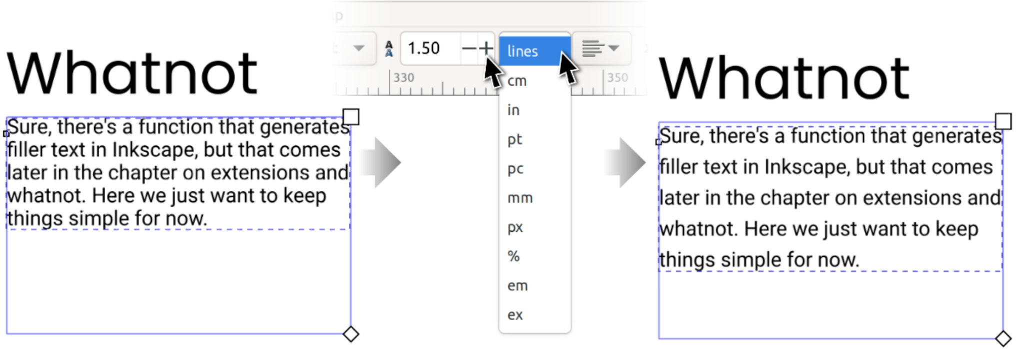
Figure 7.10 – Changing the font family for a text object with two different fonts and styles makes them all one style

So, now we know how to style bits of our fonts, but our text may be looking a little cramped. This is fine for text-heavy articles, but sometimes, we'd like to give the lines of our text a little more breathing room.

This brings us to the next item in the Tool control bar for the **Text** tool: the **Baseline Spacing** field. By default, the units for this are set to the same units as the document, but what we'd like in this case is to change those units to **lines** instead of physical units.

This is because some fonts are taller or shorter than others, and if we have `mm` for example, when we change our **Font Family**, we will have to manually change the spacing to fit the height of the new font.

What we'd rather do is have Inkscape take the height of our font and then space out the baselines using that measurement. In most cases, this little trick will save us from having to redo our spacing if we change fonts. *Figure 7.11* shows the effect if we change our spacing to **lines**, with a `1.50` line spacing:



Whatnot

Sure, there's a function that generates filler text in Inkscape, but that comes later in the chapter on extensions and whatnot. Here we just want to keep things simple for now.

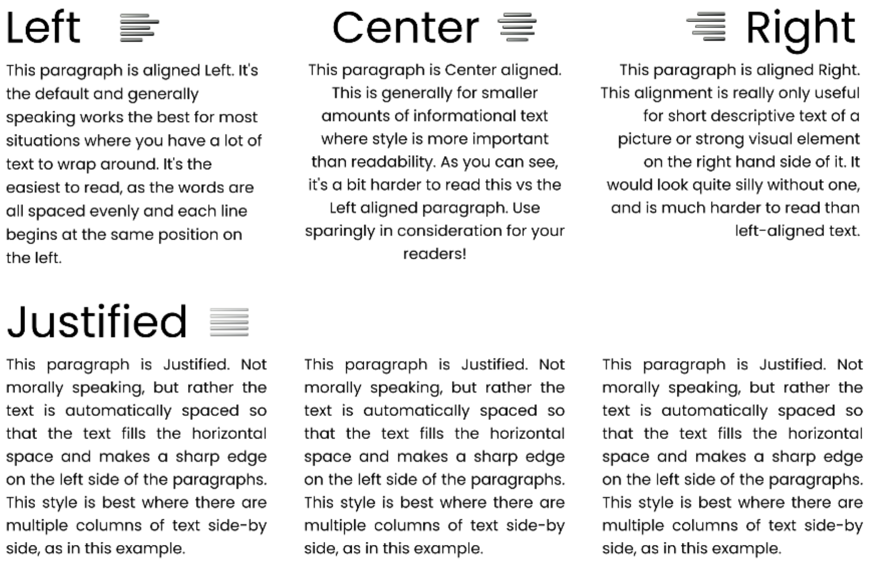
Whatnot

Sure, there's a function that generates filler text in Inkscape, but that comes later in the chapter on extensions and whatnot. Here we just want to keep things simple for now.

Figure 7.11 – Using lines to space our text 1.50 lines apart

Many of us may remember the scholastic research paper requirement for text to be in **12pt** font and double-spaced for essays. That's essentially what setting the units as **lines** allows us to do: we can set the spacing to **2.0 lines**, and we have our double spacing.

So, now, we can uniformly space our lines as we like, but what if we don't want default left-aligned text? Maybe we'd like center alignment or even right alignment for our paragraphs. That brings us to the next dropdown for **Text Alignment**. *Figure 7.12* shows the results of changing **Text Alignment** to the different options:



Left

This paragraph is aligned Left. It's the default and generally speaking works the best for most situations where you have a lot of text to wrap around. It's the easiest to read, as the words are all spaced evenly and each line begins at the same position on the left.

Center

This paragraph is Center aligned. This is generally for smaller amounts of informational text where style is more important than readability. As you can see, it's a bit harder to read this vs the Left aligned paragraph. Use sparingly in consideration for your readers!

Right

This paragraph is aligned Right. This alignment is really only useful for short descriptive text of a picture or strong visual element on the right hand side of it. It would look quite silly without one, and is much harder to read than left-aligned text.

Justified

This paragraph is Justified. Not morally speaking, but rather the text is automatically spaced so that the text fills the horizontal space and makes a sharp edge on the left side of the paragraphs. This style is best where there are multiple columns of text side-by side, as in this example.

This paragraph is Justified. Not morally speaking, but rather the text is automatically spaced so that the text fills the horizontal space and makes a sharp edge on the left side of the paragraphs. This style is best where there are multiple columns of text side-by side, as in this example.

This paragraph is Justified. Not morally speaking, but rather the text is automatically spaced so that the text fills the horizontal space and makes a sharp edge on the left side of the paragraphs. This style is best where there are multiple columns of text side-by side, as in this example.

Figure 7.12 – Different text alignments available in the Text Alignment dropdown in the Tool control bar

It's worth noting that at the time of writing, the **Justified** alignment is only available when using flowed text (text in a box). So, if you find it grayed out, just click-drag yourself a box with the **Text** tool and copy/paste your paragraph contents into it. You can then choose the **Justified** option. That's all well and good for horizontal text alignment, but what if we want to take a part of some text, and make it superscript or subscript – that is, to appear smaller and just above or below the end of the previous text?

For example, we may want to take the number **2910** and make it 29 to the 10th power. We could edit the baseline and font size for just the 10, but Inkscape has these nifty buttons that do it for you. Simply select **10** and click either the **Superscript** or **Subscript** button in the Tool control bar, as shown in *Figure 7.13*:

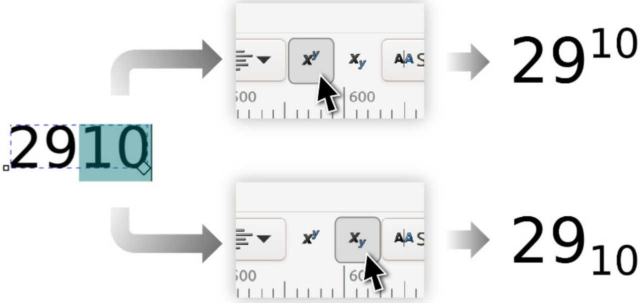


Figure 7.13 – Making the 10 superscript or subscript in the number 2910

If you're not into math, the superscript option is still useful for making the *th* in the *5th* superscript as an example.

This is nice, but sometimes, you may want way more control over the spacing and position of individual characters. This is why Inkscape 1.2 has the rest of these options sorted into a **Spacing** popover. This is shown in *Figure 7.14*:

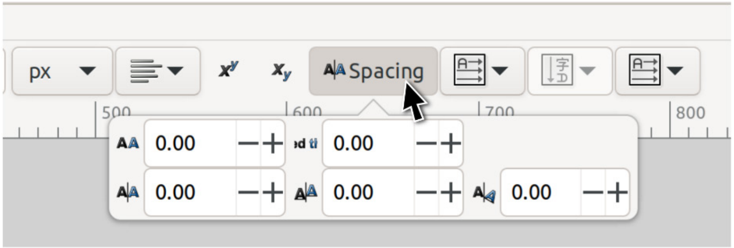
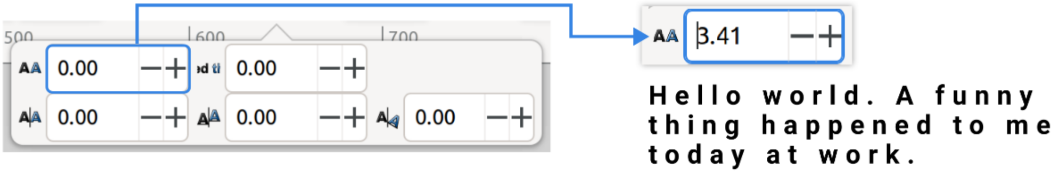


Figure 7.14 – The Spacing popover in the Tool control bar while the Text tool is active

The first item in this box is character spacing. As shown in *Figure 7.15*, changing this value will change the spacing between all letters in the text:



3.41

Hello world. A funny thing happened to me today at work.

Figure 7.15 – Adding more letter spacing

Likewise, the box directly to the right of character spacing controls the spacing between words, as shown in *Figure 7.16*:

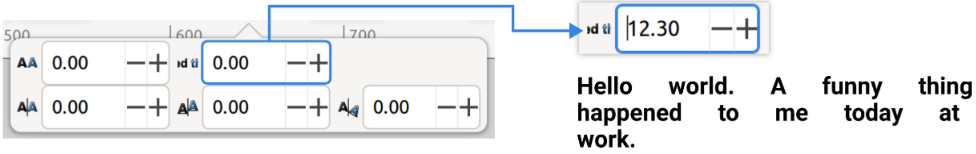


Figure 7.16 – Increasing spacing between words

At the far-left side of the next row of options in the **Spacing** popover, we have **Horizontal Kerning**, which increases or decreases the gap between two letters. This is shown in *Figure 7.17*:

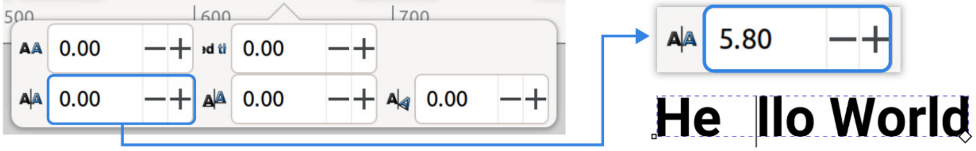


Figure 7.17 – Increasing the letter spacing between the "e" and "l" in "Hello"

Note that you can undo/remove this manual letter spacing by choosing **Path** > **Remove Manual Kerns**. One field to the right, we can control **Vertical Kerning** between the same two characters, as shown in *Figure 7.18*:

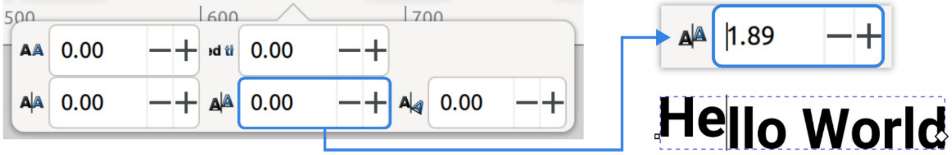


Figure 7.18 – Increasing vertical letter kerning

At this point, you might be thinking, *Why on Earth would I use these options?* In typography and graphic design, things such as line spacing and letter kerning are quite important to achieve a professional-looking headline or an airier paragraph feeling.

The fact of the matter is some letters just don't look that great next to each other. Take letter kerning for example; the gaps between certain letter combinations may make them seem too far apart or too close together. Take, for example, the headline in *Figure 7.19*:

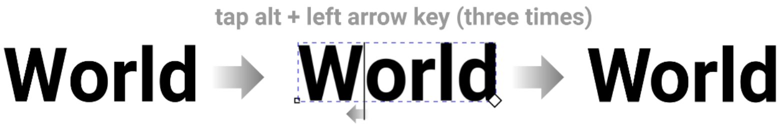


Figure 7.19 – Changing the kerning between the "W" and "o" for more even spacing

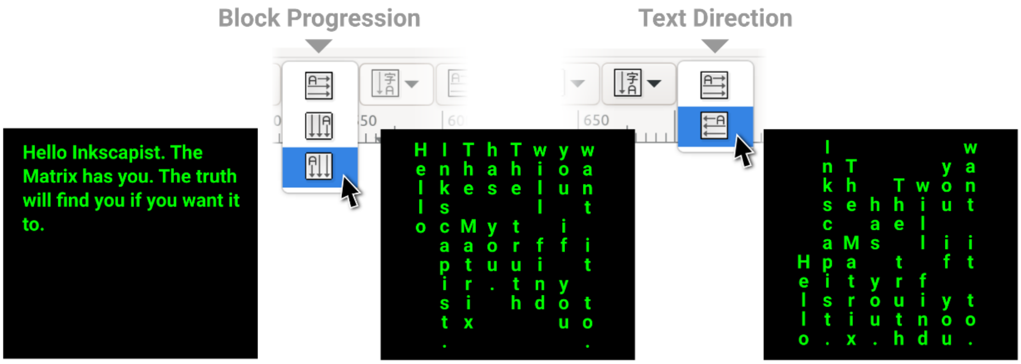
If you look closely at the *W* and *o*, the spacing seems just slightly too wide than with the other letters. It's almost as if the *W* were a bit scared of the *o* and is just standing slightly farther away as a result.

This may not matter so much in a web article headline, but if you're designing a logo for a business, these small details are important, not only to the perceived quality of the brand but also to the quality of your skill as a designer.

In this case, we simply move the blinking text cursor between the *W* and the *o*, hold the *Alt* key, and tap the left arrow key a few times (in this case, three times works well) to pull that *o* a little closer to the *W*. Notice how that makes the visual separation of the letters in *World* look more even.

Lastly, we will touch on the last three options in the Tool control bar for the **Text** tool, which are the **Block Progression**, **Glyph Orientation**, and **Text Direction** dropdowns. These are most useful when using alternate character sets for different languages that do not write or read from left to right, such as Latin and Germanic-based languages (such as English).

However, that doesn't mean we can't have a bit of fun with them anyway. As an example, in *Figure 7.20*, we can use a different combination of these three options to get a text effect that resembles an effect found in the movie *The Matrix*:



Hello Inkscapist. The Matrix has you. The truth will find you if you want it to.

Figure 7.20 – Changing the Block Progression and Text Direction to get a Matrix-like text effect

This concludes our exploration of the **Text** tool options in the Tool control bar. It's worth noting that at the time of writing, character kerning in single-line text is only available if it *is not wrapped* (as per *Figure 7.3*). You will have to avoid using the single-line text wrapping handle and just use the *Enter* key for new lines if you need kerning in your titles.

The Tool control bar offers a convenient way to get to the most used options while editing text, but there's also a whole dialog for this, with more options, worth exploring. So, let's move on and see what other wonderful options Inkscape has for us to play with in the **Text and Font** dialog box.

Font collections

With all the many hundreds of fonts on your system (thousands in my case), you may just want to select a subset of your favorites to categorize based on what you're doing. Thus, the Inkscape developers have once again blessed us with a brand new feature in Inkscape 1.3: the ability to make Font collections. If you click the `font` folder icon in the far-right of the Tool control bar (with the **Text** tool selected, of course), you get a popover that shows each of your Font collections and a checkbox beside each entry. Checking these boxes will hide all other fonts from the **Font Family** dropdown, as shown in the before and after diagram in *Figure 7.21*.

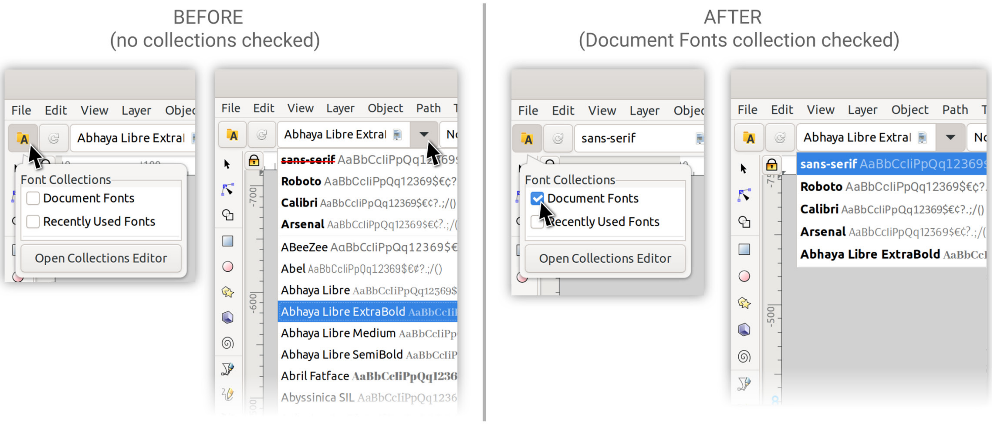


Figure 7.21 – No collections checked versus Document Fonts collection checked

Thus, when there are no collections checked, Inkscape shows us all our fonts. If we check one or more collections, however, Inkscape will hide all fonts but those that are contained in the selected collections.

Notice that by default, Inkscape gives us a few collections to start with:

- **Document Fonts**: This collection is automatically filled with the fonts that you are currently using in your document
- **Recently Used Fonts**: Inkscape fills this collection automatically with fonts that you have recently used across your Inkscape projects (not just the ones in the current document)

The last option in the popover is the **Open Collection Editor** button, which brings up the **Font Collections** dialog, as shown *Figure 7.22*.

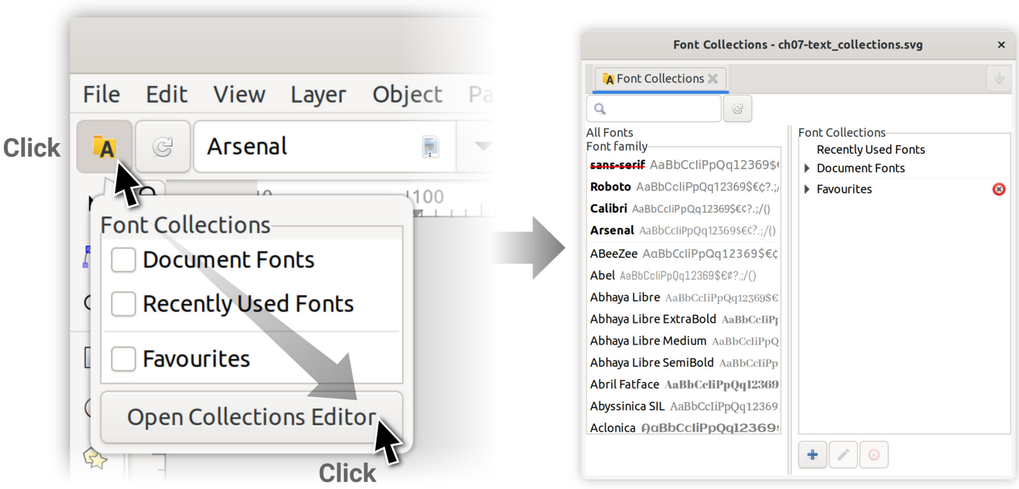

Click
File Edit View Layer Object Pa
Arsenal
Font Collections
Document Fonts
Recently Used Fonts
Favourites
Open Collections Editor
Click
Font Collections - ch07-text_collections.svg
Font Collections
All Fonts
Font family
~~sans-serif~~ AaBbCcIiPpQq12369$
Roboto AaBbCcIiPpQq12369$€¢?.;
Calibri AaBbCcIiPpQq12369$€¢?.;/()
Arsenal AaBbCcIiPpQq12369$€¢?.;/()
ABeeZee AaBbCcIiPpQq12369$€¢
Abel AaBbCcIiPpQq12369$€¢?.;/()
Abhaya Libre AaBbCcIiPpQq12369$€
Abhaya Libre ExtraBold AaBbCcIiPp
Abhaya Libre Medium AaBbCcIiPpQ
Abhaya Libre SemiBold AaBbCcIiPp
Abril Fatface AaBbCcIiPpQq12369
Abyssinica SIL AaBbCcIiPpQq12369
Aclonica AaBbCcIiPpQq12369
Font Collections
Recently Used Fonts
Document Fonts
Favourites

Figure 7.22 – Calling up the Font Collections dialog from the Tool control bar

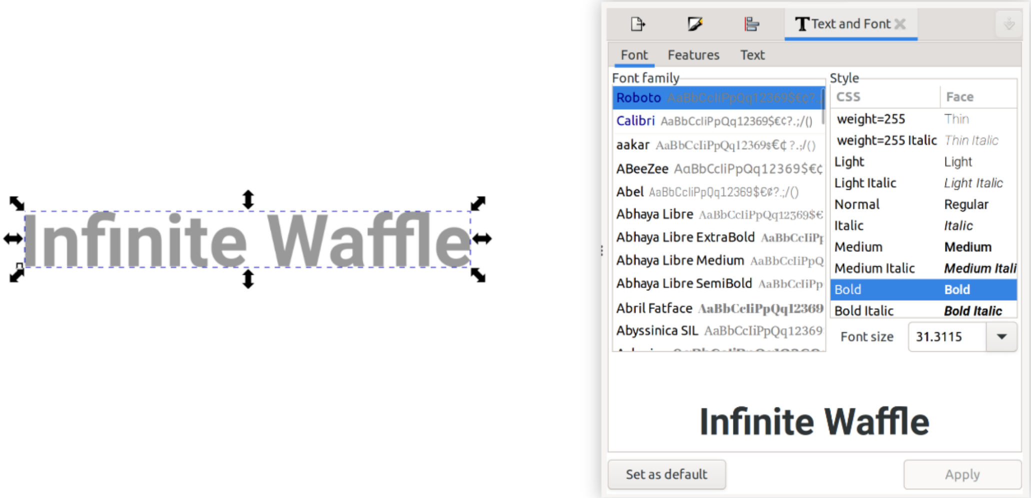
Here, we can add our own collections using the button with the + on it and providing a name for the collection. We can then add fonts to our new collection by just dragging and dropping fonts from the left side under **All Fonts – Font Family**, right into our new collection.

You can see in *Figure 7.22* that I've added a folder called `Favourites`, which I've filled with some of my favorite fonts. Inkscape stores these font collections automatically for you, so your collections will be accessible in all projects you make in Inkscape. To remove a collection, click the red **X** icon after the name of your collection.

The Text and Font dialog

To open the **Text and Font** dialog, simply choose **Text and Font...**, which is the first entry in the **Text** menu at the top of the screen. Once it's open, you will see lots of familiar items that we saw previously in the Tool control bar while using the **Text** tool.

However, notice that we have a preview at the bottom, and not only are we able to see various samples of different fonts, but we also get nice visual examples of each of the font weights. *Figure 7.23* shows what this dialog looks like when we call it up and have a text object selected:



CSS	Face
weight=255	Thin
weight=255 Italic	Thin Italic
Light	Light
Light Italic	Light Italic
Normal	Regular
Italic	Italic
Medium	Medium
Medium Italic	Medium Itali
Bold	Bold
Bold Italic	Bold Italic

Font size 31.3115

Figure 7.23 – The Text and Font dialog with a text object selected

Unlike the Tool control bar, however, changing the **Font family**, **Font size**, or **Font style** option does not automatically change our text object on the canvas. Instead, it only changes the preview. This can be handy in certain situations where you just want to see what's available without affecting your text object.

When you're happy with the preview, simply click the **Apply** button at the bottom. You can also click the **Set as Default** button next to the **Apply** button, and Inkscape will automatically use that font and all the nice options you chose as the default for creating new text.

There are three tabs at the top of the **Text and Font** dialog: **Font**, **Features**, and **Text**. Since we've had a look at the **Font** tab already, let's explore the **Features** tab next. This tab, shown in *Figure 7.24*, shows expanding lists of checkboxes for font options sorted into categories:

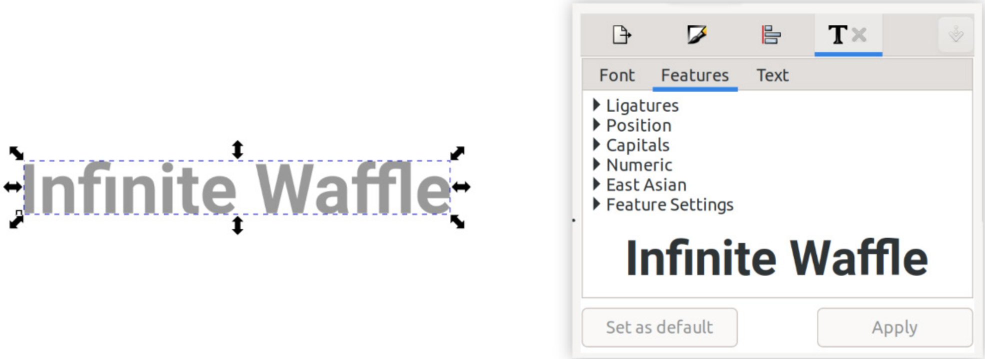
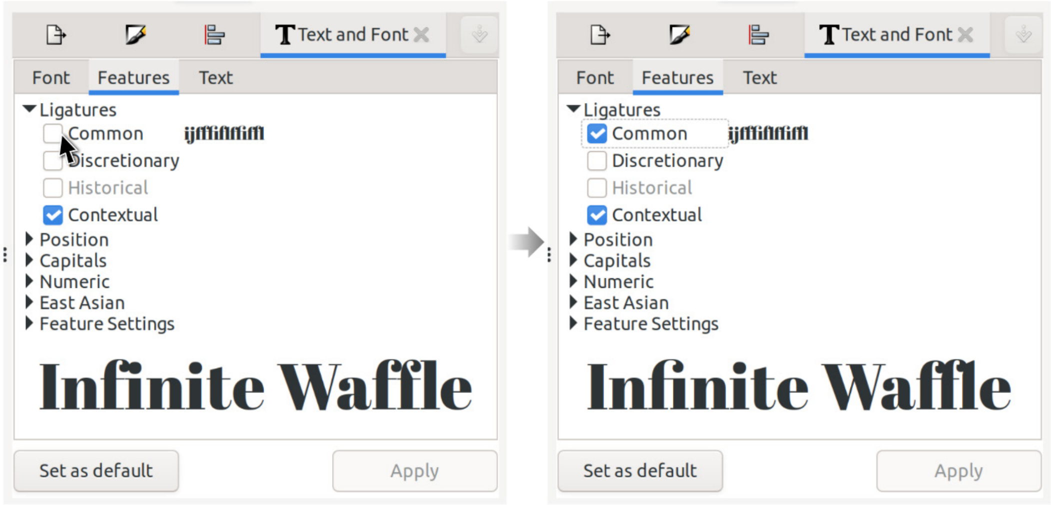


Figure 7.24 – The Features tab in the Text and Font dialog

You will notice when looking through these that some are grayed out. This is because not all fonts have the same options built into them. Ligatures, for example, are special glyphs (letter symbols) that take the place of two letters that would otherwise overlap or be too close together. *Figure 7.25* shows a font called **Abril Fatface** with ligatures turned on and off:



Ligatures
- ☐ Common ijfffiflffiffl
- ☐ Discretionary
- ☐ Historical
- ☑ Contextual

Position
Capitals
Numeric
East Asian
Feature Settings

Infinite Waffle

Ligatures
- ☑ Common ijfffiflffiffl
- ☐ Discretionary
- ☐ Historical
- ☑ Contextual

Position
Capitals
Numeric
East Asian
Feature Settings

Infinite Waffle

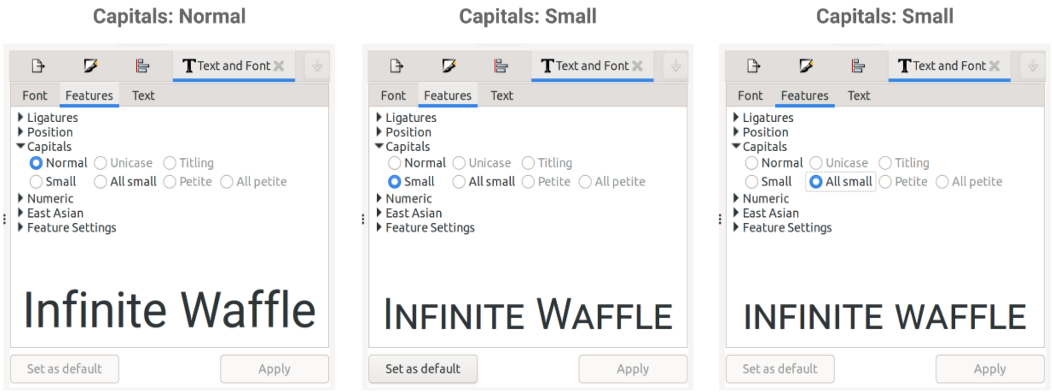
Figure 7.25 – The difference between turning on and off "Common" ligatures

Look closely at how the *f* overlaps the dot of the *i* when those letters are next to each other. With ligatures turned on, Inkscape replaces that character collision with a very nice *fi* character. This is the same with the *ffl* in *Waffle*. Three characters are replaced with one connected character, which beautifies what would otherwise be characters too far apart, or too close together.

These special ligatures are hand-made letters designed by the maker of the font. It's often the hallmark of a professionally designed font to have these ligatures included in the characters. Common ligatures are turned on by default, but Inkscape allows you to turn them off by simply unchecking the **Common** checkbox. Some fonts also have **Discretionary**, **Historical**, and **Contextual** ligature glyph sets, which can be turned on and off here as well, if available.

The next **Feature** category is **Position**. Remember the **Subscript** and **Superscript** toggles mentioned earlier in this chapter? Well, some fonts have special characters for these. You can toggle them on and off here by toggling either the **Subscript** or **Superscript** radio button under the **Position** category. Very few fonts support this, however, so it's nice that Inkscape will make them for you via the aforementioned toggles in the Tool control bar.

Capitals is the next category and this one is quite useful for those fonts that offer it. It turns every character you type into a specially made, small-caps character. The different options are shown in *Figure 7.26*:



Capitals: Normal

Text and Font

Font | Features | Text

- ▸ Ligatures
- ▸ Position
- ▾ Capitals
  - ● Normal ○ Unicase ○ Titling
  - ○ Small ○ All small ○ Petite ○ All petite
- ▸ Numeric
- ▸ East Asian
- ▸ Feature Settings

Infinite Waffle

Set as default | Apply

Capitals: Small

Text and Font

Font | Features | Text

- ▸ Ligatures
- ▸ Position
- ▾ Capitals
  - ○ Normal ○ Unicase ○ Titling
  - ● Small ○ All small ○ Petite ○ All petite
- ▸ Numeric
- ▸ East Asian
- ▸ Feature Settings

INFINITE WAFFLE

Set as default | Apply

Capitals: Small

Text and Font

Font | Features | Text

- ▸ Ligatures
- ▸ Position
- ▾ Capitals
  - ○ Normal ○ Unicase ○ Titling
  - ○ Small ● All small ○ Petite ○ All petite
- ▸ Numeric
- ▸ East Asian
- ▸ Feature Settings

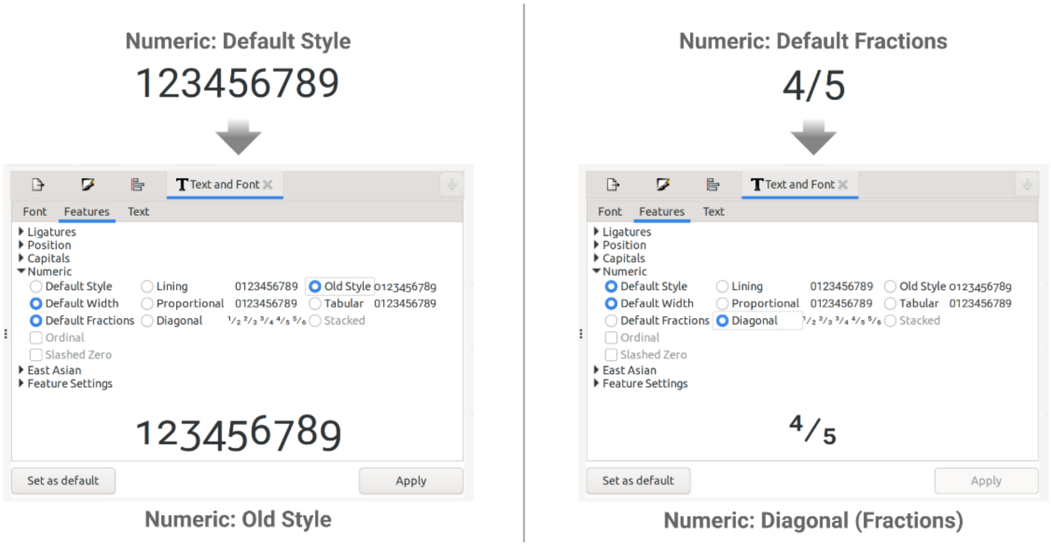
INFINITE WAFFLE

Set as default | Apply

Figure 7.26 – Different capitals examples for the Roboto Regular font

Notice how it makes the **Roboto Regular** font look more official. It's almost like a different font hidden within it!

Some fonts have special styling available for numbers, which brings us to the **Numerical** category. As you can see from *Figure 7.27* (left), we can switch from the default number style to old-style number glyphs that resemble fancy numbers on houses, or part of an address:



Numeric: Default Style

123456789

Text and Font

Font | Features | Text

- ▸ Ligatures
- ▸ Position
- ▸ Capitals
- ▾ Numeric
  - ○ Default Style ○ Lining 0123456789 ● Old Style 0123456789
  - ● Default Width ○ Proportional 0123456789 ○ Tabular 0123456789
  - ● Default Fractions ○ Diagonal ½ ⅔ ¾ ⅘ ⅚ ○ Stacked
  - □ Ordinal
  - □ Slashed Zero
- ▸ East Asian
- ▸ Feature Settings

123456789

Set as default | Apply

Numeric: Old Style

Numeric: Default Fractions

4/5

Text and Font

Font | Features | Text

- ▸ Ligatures
- ▸ Position
- ▸ Capitals
- ▾ Numeric
  - ● Default Style ○ Lining 0123456789 ○ Old Style 0123456789
  - ● Default Width ○ Proportional 0123456789 ○ Tabular 0123456789
  - ○ Default Fractions ● Diagonal ½ ⅔ ¾ ⅘ ⅚ ○ Stacked
  - □ Ordinal
  - □ Slashed Zero
- ▸ East Asian
- ▸ Feature Settings

⁴⁄₅

Set as default | Apply

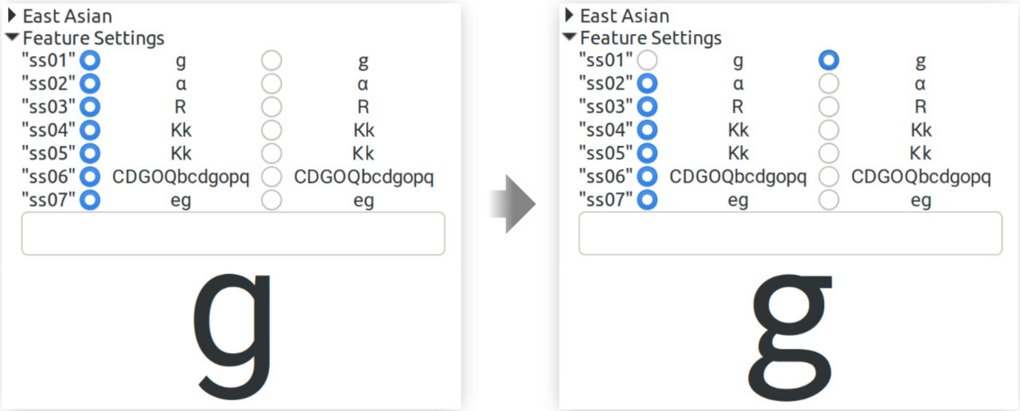
Numeric: Diagonal (Fractions)

Figure 7.27 – Old-style numeric glyphs (left) and diagonal numerical fractions (right)

We could also choose a different style for fractions that we type out as 4/5. Changing the **Default** fraction automatically adds superscript and subscript to the left and right sides of our slash, as shown in *Figure 7.27* (right).

The **East Asian** category has features relevant to East Asian language font glyphs, but at the time of writing, I am not at all familiar with East Asian language font glyphs. We will give them a pass and go straight to what I feel is the most fun category, **Feature Settings**. This is a catch-all for anything special that the font maker decided to put into the font.

Maybe it bugged the designer that they couldn't include a curly style (also known as a **looptail**) *g* character in the default font, so they hid it in **Features**, just waiting for knowledgeable people like you and me to come along and discover it. *Figure 7.28* shows a bunch of these features for the same **Roboto Regular** font we've been using, including our coveted looptail *g*:



▸ East Asian
▾ Feature Settings
"ss01" ● g ○ g
"ss02" ● ɑ ○ ɑ
"ss03" ● R ○ R
"ss04" ● Kk ○ Kk
"ss05" ● Kk ○ Kk
"ss06" ● CDGOQbcdgopq ○ CDGOQbcdgopq
"ss07" ● eg ○ eg

g

▸ East Asian
▾ Feature Settings
"ss01" ○ g ● g
"ss02" ● ɑ ○ ɑ
"ss03" ● R ○ R
"ss04" ● Kk ○ Kk
"ss05" ● Kk ○ Kk
"ss06" ● CDGOQbcdgopq ○ CDGOQbcdgopq
"ss07" ● eg ○ eg

g

Figure 7.28 – Feature Settings for Roboto Regular, and changing the "g" style

You may have noticed that box at the bottom of **Feature Settings**. This is the place to type in the **Cascading Style Sheets** (**CSS**) values for these features. CSS is beyond the scope of this book, however, so let's move along to the next roadside attraction on our text and font tour: variable fonts!

Understanding variable fonts

Up to this point, we've seen a lot of hidden font options that replaced certain glyphs in the font with alternate ones, but there is a class of font that lets you change many more aspects of the letter geometry.

Let's look at one of the more interesting ones: **Decovar24** (see `https://v-fonts.com/fonts/decovar` for more information and download links for this font). *Figure 7.29* shows what tweaking some of the many variable font sliders, which appear at the bottom of the **Font** tab of the **Text and Font** dialog, can do:

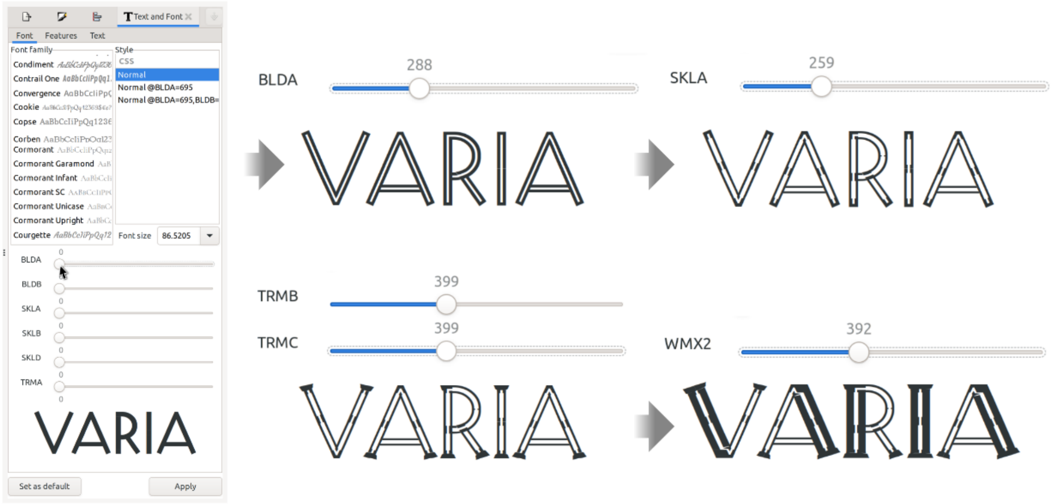


Figure 7.29 – Changing some of the variable font sliders dramatically changes the look

At the time of writing, there are still only a hundred or so good variable fonts. As adoption increases for variable fonts in other graphics applications, you can expect to see more options. Many have a basic set of options such as sliders for tilt, character width, boldness, or thinness, and I encourage you to go find your favorites and have a play with them in Inkscape.

Text on a curve

We have certainly explored a wealth of options for font styling on a straight line. Occasionally, though, we'd like to bend our text around the periphery of an object, such as a circle, or even just a curve. For example, we'd like to put some text in a wavey line so that it flows with the curve of the line rather than the standard, strictly straight baseline.

We can do that by making a wavy line path, selecting the text and the path that we've made, and choosing **Path** > **Put on Path**. As you can see from *Figure 7.30*, this process results in text with a curvy baseline:

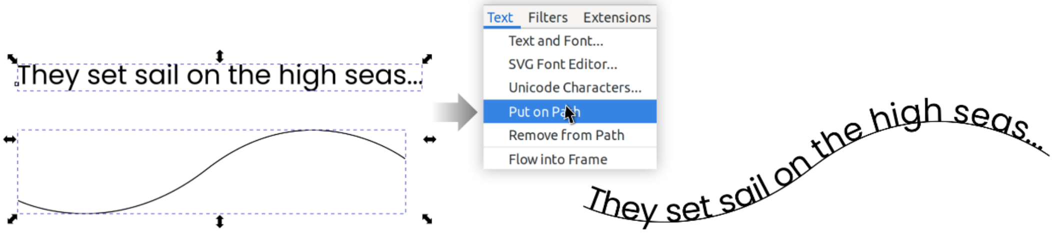


Figure 7.30 – Using Put on Path to flow text along a curved line

Note that moving the text moves the text away from the curve. To prevent this from happening, I generally either move the line instead of the text (the text follows it) or group them and move the group instead of the text or line.

Chances are, you probably don't want that curvy line to be visible. Removing the stroke color is my general go-to for hiding it. The problem then arises of how to select it to modify it later, since it's completely invisible. There are two ways to do this.

One is to drag a selection box around the invisible line to select it. The other way is to use the **Outline Overlay View** mode by selecting **View** > **Display Mode** > **Outline Overlay**. As you can see, this adds graphical overlays to all the hidden stuff so that you can select and edit the invisible curvy line. *Figure 7.31* shows the effect of this overlay view:

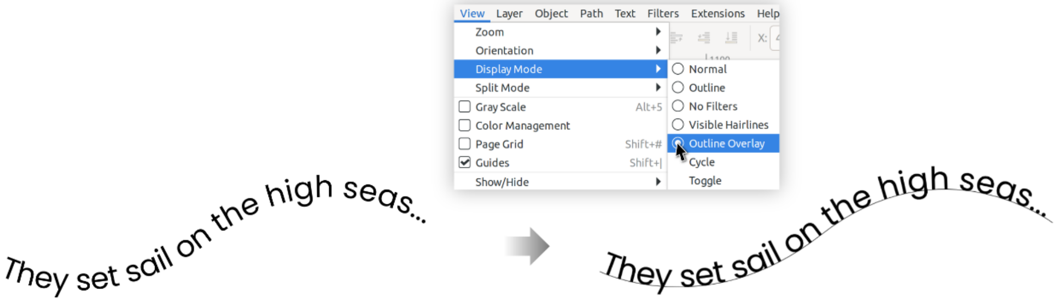


Figure 7.31 – Setting Outline Overlay Display mode to make our invisible line visible again

To switch back to **Normal Display Mode**, it's **View** > **Display Mode** > **Normal**:

This technique works well for putting text on a path, and indeed, we can put text on shape objects as well as use the same method. *Figure 7.32* shows some examples of putting text on shapes:

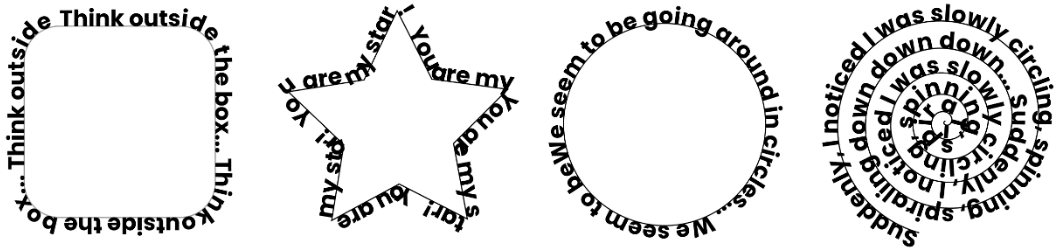

Think outside the box... Think outside the box... Think outside the box...
You are my star! You are my star! You are my star!
We seem to be going around in circles...
I was slowly circling down down... I was slowly spinning... Suddenly I noticed I was spinning... Suddenly circling down

Figure 7.32 – Text put on different shapes

You can also control which side of the shape (interior or exterior) the text flows by either flipping the shape horizontally or vertically or converting the shape into a path by selecting **Path** > **Object to Path**, and then selecting **Path** > **Reverse**, as we did in the previous chapter.

When you do that, you will notice that the letters are a bit more cramped together as a result of being on the inside of the circle, so you'll want to adjust the letter spacing as we did earlier in this chapter. *Figure 7.33* shows this process and the results:

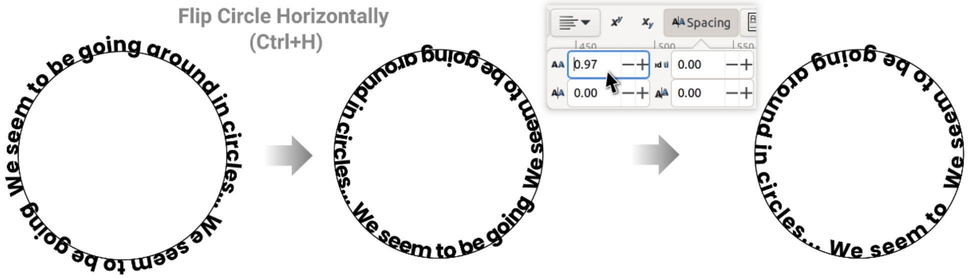

Flip Circle Horizontally
(Ctrl+H)
We seem to be going around in circles... We seem to be going
Spacing
0.97
0.00
0.00
0.00

Figure 7.33 – Flipping the circle and adjusting text spacing to fit text to the inside of the circle

We'll do a bit more circle text in the practice lesson because there are some tricks to doing it right. For now, we have enough information about wrapping text around shapes and can move on to flowing text through shapes!

Flowing paragraphs into shapes

Sometimes, we'd like to be able to flow text through shapes other than just the standard rectangle shape. Fortunately, Inkscape can do this easily! Simply make a block of text and one or more shapes, select them all, and choose **Text** > **Flow Into Frame**.

As an example, we will start with a block of text and four hexagons. All we need to do is click to select the text object, then press *Shift* and click on each shape in the order we want our text to flow through them. *Figure 7.34* shows this process, adding the hexagons sequentially to the selection, and then selecting **Text** > **Flow Into Frame** to flow the text through the shapes in the order we clicked:

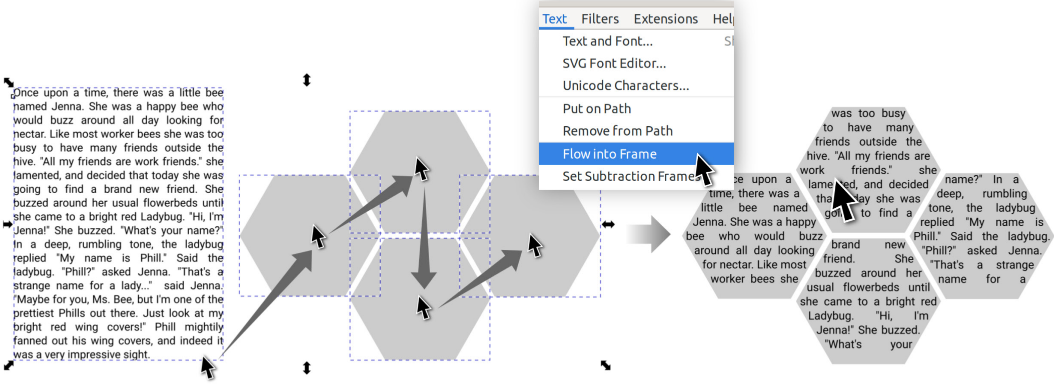

Once upon a time, there was a little bee named Jenna. She was a happy bee who would buzz around all day looking for nectar. Like most worker bees she was too busy to have many friends outside the hive. "All my friends are work friends." she lamented, and decided that today she was going to find a brand new friend. She buzzed around her usual flowerbeds until she came to a bright red Ladybug. "Hi, I'm Jenna!" She buzzed. "What's your name?" In a deep, rumbling tone, the ladybug replied "My name is Phill." Said the ladybug. "Phill?" asked Jenna. "That's a strange name for a lady..." said Jenna. "Maybe for you, Ms. Bee, but I'm one of the prettiest Phills out there. Just look at my bright red wing covers!" Phill mightily fanned out his wing covers, and indeed it was a very impressive sight.

Figure 7.34 – Flowing text through four hexagons

Just like the text boxes we drew earlier, if we select our text and double-click (or switch to the **Text** tool) to edit, we get a handle to control the margins.

Sometimes, we may want that flowing text to flow around a shape instead of through it. Inkscape has an easy way to do this as well. Simply select your text object, as well as the shape you want it to flow around, and choose **Text** > **Set Subtraction Frames**. As you can see in *Figure 7.35*, the text will then flow around the shape:

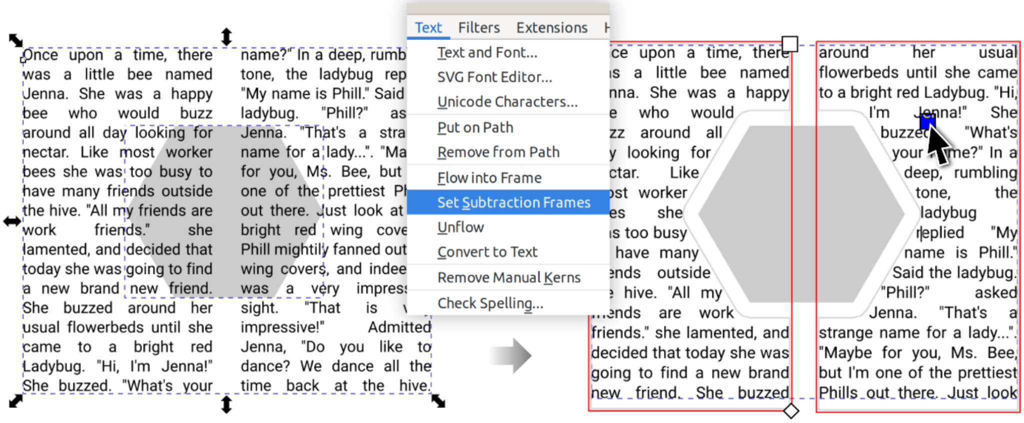


Figure 7.35 – Using Set Subtraction Frames to flow text around a hexagon shape

Note that to get this to work properly, you need to have flowed text through two rectangle shapes, as we did with the hexagons in *Figure 7.34*.

Notice that now, we get two handles when we edit the text – one for the text margins and one for the **Set Subtraction Frames** margin.

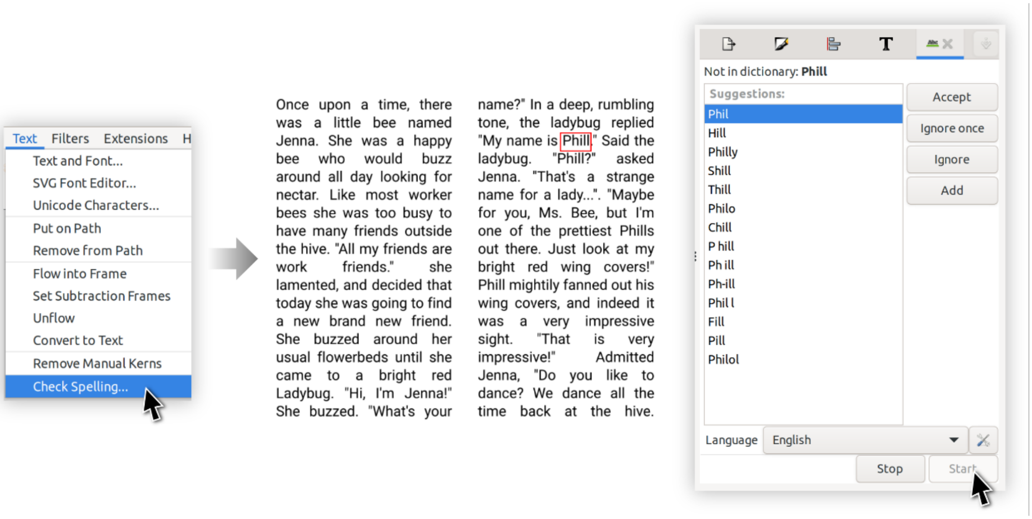
It's also worth noting that you can unflow your text at any time by selecting the text object and choosing **Text** > **Unflow**. This will convert the flowed text back into a single line of text.

So, that's most of the text options, and we've certainly learned quite a bit about how to get your text looking exactly the way you want it. Before we move on to the practice lesson, however, we'll quickly go over a few text-related tools that are the proverbial icing on the cake in working with text in Inkscape.

Check spelling and other handy tools

We've almost exhausted all the entries in the **Text** menu, but since editing SVG fonts and Unicode characters is beyond the scope of this book, we will cover something you'll probably want to make use of every single time you use text in Inkscape: **Text** > **Check Spelling**!

Predictably, this opens up the **Check Spelling** dialog. To begin the spellcheck, press the **Start** button at the bottom right of the dialog. If Inkscape finds something, it will highlight the word with a red box, as shown in *Figure 7.36*:



Once upon a time, there was a little bee named Jenna. She was a happy bee who would buzz around all day looking for nectar. Like most worker bees she was too busy to have many friends outside the hive. "All my friends are work friends." she lamented, and decided that today she was going to find a new brand new friend. She buzzed around her usual flowerbeds until she came to a bright red Ladybug. "Hi, I'm Jenna!" She buzzed. "What's your name?" In a deep, rumbling tone, the ladybug replied "My name is Phill." Said the ladybug. "Phill?" asked Jenna. "That's a strange name for a lady...". "Maybe for you, Ms. Bee, but I'm one of the prettiest Phills out there. Just look at my bright red wing covers!" Phill mightily fanned out his wing covers, and indeed it was a very impressive sight. "That is very impressive!" Admitted Jenna, "Do you like to dance? We dance all the time back at the hive.

Not in dictionary: **Phill**

Suggestions:
Phil
Hill
Philly
Shill
Thill
Philo
Chill
P hill
Ph ill
Ph-ill
Phil l
Fill
Pill
Philol

Language English

Figure 7.36 – Summoning the Check Spelling dialog and starting the spellcheck

At this point, you can either click on the word and edit it with the **Text** tool or choose one of the words in the **Suggestions:** box, and then click the **Accept** button to change the word and move on.

You could also decide that the name **Phill** is the correct spelling of the ladybug's name, and stubbornly hit the **Ignore** button, which will take you to the next word without bothering you about it again during the rest of the spell-checking process.

Likewise, **Ignore Once** will ignore the spelling of the word one time, but keep reminding you that it's really *Phil* the next time it finds your quirky spelling of the name. Depending on how stubborn you are, you may want to add that double *l* spelling of *Phil* to the dictionary, so Inkscape will never bother you again by saying it's incorrect. If this is the case, you can press the **Add** button instead, and Inkscape will not bother you again about *Phill* during spelling checks.

You can also stop the spell-check process at any time with the **Stop** button. Finally, the **Language** dropdown shows all the language dictionaries you have installed on your system. Thus, if you have more than one, you can switch languages there.

Before we move on and get into some practice, it's worth mentioning there are more **Text** tools located in the **Extensions** menu (**Extensions** > **Text**). Here is a short list of the most useful ones to be aware of:

- **Change Case**: Maybe you want to replace all your lowercase letters with capital letters. Maybe you want no capital letters, or maybe you want some combination of them. You'd rather not re-type everything manually, so these tools change the letter case (capital or lowercase) automatically. Simply select the piece of text you want to convert into the chosen case, and choose one of the options in the **Extensions** > **Text** > **Change Case** menu.

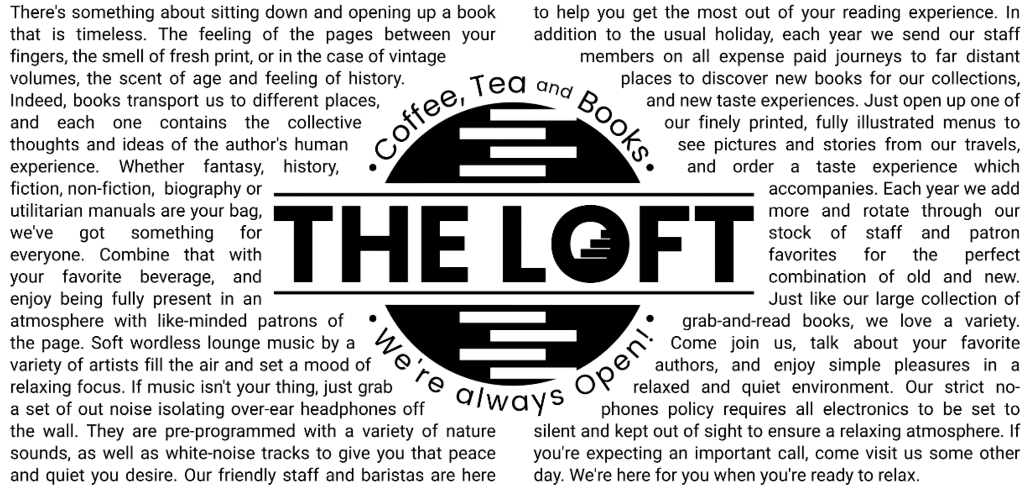
- **Convert to Braille**: This converts your text into Braille dots, to be raised and read by the visually impaired using only touch.
- **Hershey Text**: If you're using a router to cut out your text, this function will replace the current font with one composed of router-path lines. If none of that makes any sense to you, don't worry about it. It's a function for specialist text engraving.
- **Lorem Ipsum**: Also called **Lipsum** text, this function just fills your textbox with endless nonsense text that looks as if it were typed words. This is useful if you want to see how your paragraphs look and don't want to make up little stories as I have done for the graphics in this chapter.
- **Merge**: This tool combines one or more selected text objects to make a new text object with the text of both. If you can't see the result right away, look around the page and desk for the new text object.
- **Replace Font**: This tool goes through all your text objects (or just the one you have selected, if you tick the box for that option), and replaces one font with another. This is useful if you have used more than one font inside of a text object and want to replace one of them without changing the rest of the text.
- **Split Text**: This tool splits up your text by lines, words, and so on. It makes a lot of new text objects. It's useful for making word clouds out of a paragraph without you having to type each word.

Well, that wraps up text! We've learned a lot about how to make it, style it, flow it, and modify it. So, now that we have a good grasp of how text works in Inkscape, let's get some hands-on experience with some practice!

Try it yourself

To get some practice with all of Inkscape's great text features, we're going to make a logo with an article wrapped around it. You don't have to make exactly what I make. Let your imagination run free since you've acquired the skills in previous chapters to make whatever shapes you desire.

Imagine for a moment that you're writing an article that describes your perfect place. That could be a pub, a restaurant, or in my case an imaginary coffee shop/bookseller, as shown in *Figure 7.37*:

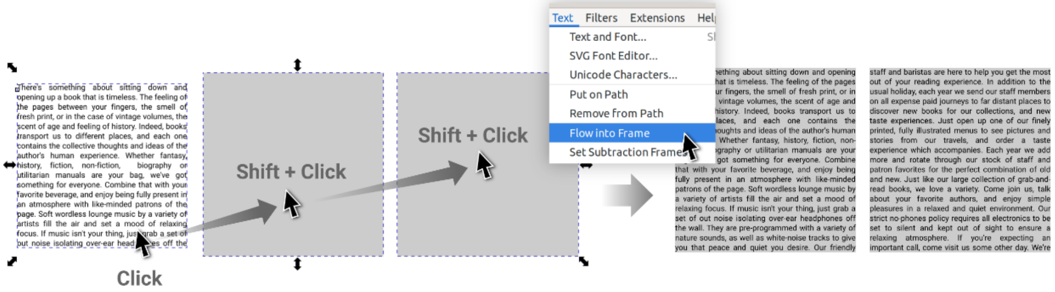


There's something about sitting down and opening up a book that is timeless. The feeling of the pages between your fingers, the smell of fresh print, or in the case of vintage volumes, the scent of age and feeling of history. Indeed, books transport us to different places, and each one contains the collective thoughts and ideas of the author's human experience. Whether fantasy, history, fiction, non-fiction, biography or utilitarian manuals are your bag, we've got something for everyone. Combine that with your favorite beverage, and enjoy being fully present in an atmosphere with like-minded patrons of the page. Soft wordless lounge music by a variety of artists fill the air and set a mood of relaxing focus. If music isn't your thing, just grab a set of out noise isolating over-ear headphones off the wall. They are pre-programmed with a variety of nature sounds, as well as white-noise tracks to give you that peace and quiet you desire. Our friendly staff and baristas are here to help you get the most out of your reading experience. In addition to the usual holiday, each year we send our staff members on all expense paid journeys to far distant places to discover new books for our collections, and new taste experiences. Just open up one of our finely printed, fully illustrated menus to see pictures and stories from our travels, and order a taste experience which accompanies. Each year we add more and rotate through our stock of staff and patron favorites for the perfect combination of old and new. Just like our large collection of grab-and-read books, we love a variety. Come join us, talk about your favorite authors, and enjoy simple pleasures in a relaxed and quiet environment. Our strict no-phones policy requires all electronics to be set to silent and kept out of sight to ensure a relaxing atmosphere. If you're expecting an important call, come visit us some other day. We're here for you when you're ready to relax.

Figure 7.37 – An article wrapped around a logo for my fantasy bookstore and coffee shop called "The Loft"

First, let's make some text and flow it into two rectangles:

1. Use the **Text** tool to make a box for our text by click-dragging a box. Write our article in this box or use the **Extensions** > **Text** > **Lorem Ipsum…** generator to fill it.
2. Draw two boxes with the **Rectangle** tool that will contain our article. Then, we must select our text, and the two rectangles in the order we want the text to flow into them and choose **Text** > **Flow into Frame** (make sure that the text is on top of the object in the stack). This process is shown in *Figure 7.38*:



There's something about sitting down and opening up a book that is timeless. The feeling of the pages between your fingers, the smell of fresh print, or in the case of vintage volumes, the scent of age and feeling of history. Indeed, books transport us to different places, and each one contains the collective thoughts and ideas of the author's human experience. Whether fantasy, history, fiction, non-fiction, biography or utilitarian manuals are your bag, we've got something for everyone. Combine that with your favorite beverage, and enjoy being fully present in an atmosphere with like-minded patrons of the page. Soft wordless lounge music by a variety of artists fill the air and set a mood of relaxing focus. If music isn't your thing, just grab a set of out noise isolating over-ear headphones off the

ething about sitting down and opening hat is timeless. The feeling of the pages ur fingers, the smell of fresh print, or in vintage volumes, the scent of age and history. Indeed, books transport us to laces, and each one contains the oughts and ideas of the author's human Whether fantasy, history, fiction, non-graphy or utilitarian manuals are your got something for everyone. Combine that with your favorite beverage, and enjoy being fully present in an atmosphere with like-minded patrons of the page. Soft wordless lounge music by a variety of artists fill the air and set a mood of relaxing focus. If music isn't your thing, just grab a set of out noise isolating over-ear headphones off the wall. They are pre-programmed with a variety of nature sounds, as well as white-noise tracks to give you that peace and quiet you desire. Our friendly

staff and baristas are here to help you get the most out of your reading experience. In addition to the usual holiday, each year we send our staff members on all expense paid journeys to far distant places to discover new books for our collections, and new taste experiences. Just open up one of our finely printed, fully illustrated menus to see pictures and stories from our travels, and order a taste experience which accompanies. Each year we add more and rotate through our stock of staff and patron favorites for the perfect combination of old and new. Just like our large collection of grab-and-read books, we love a variety. Come join us, talk about your favorite authors, and enjoy simple pleasures in a relaxed and quiet environment. Our strict no-phones policy requires all electronics to be set to silent and kept out of sight to ensure a relaxing atmosphere. If you're expecting an important call, come visit us some other day. We're

Figure 7.38 – Using Flow into Frame to flow article text into two rectangles

Let's put aside this article text for the moment and draw out our logo using some of our new text tricks.

3. Using the **Text** tool, click anywhere on the canvas to start a single line of text. I've chosen the **Poppins** font for the logo, typing in all caps. To add the stairs in the *O*, I've made a series of small rectangles, but before we can **Union** them with the letters, we need to convert our text into letter shapes.
4. To do this, we will choose **Path** > **Object to Path** and **Path** > **Ungroup** the resulting letters. Then, we will move and scale the stairs into place inside the *O*. With all our letter and stair shapes selected, we can now choose **Path** > **Union** to get a single logo shape. This process is shown in *Figure 7.39*:

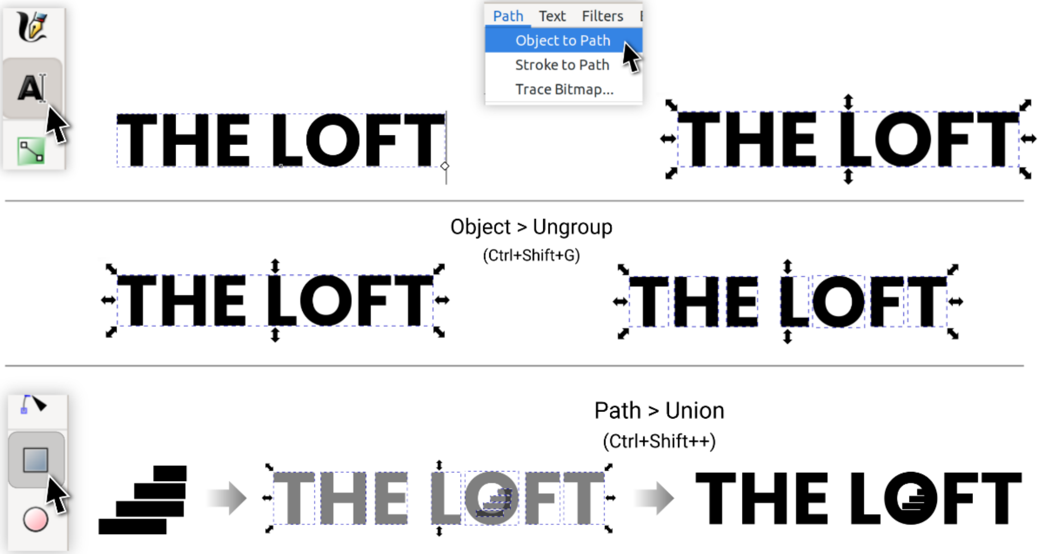


Figure 7.39 – Making the main logo for "The Loft"

Now that we have the basis for our logo, let's place the text on a circle around it. We'll start with the top part, **Coffee, Tea, and Books**, and a circle, as we saw previously in this chapter.

5. Select both the text and the circle and choose **Text** > **Put on Path**, then rotate the circle until the text is in the correct place above the main logo, as shown in *Figure 7.40*:

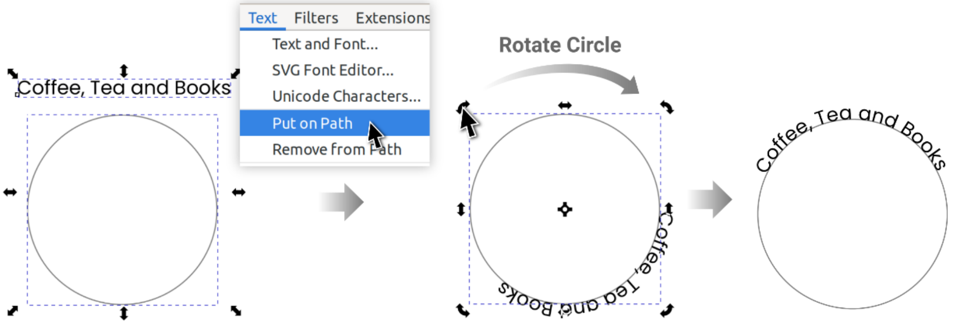


Figure 7.40 – Creating the top tagline, wrapped around a circle

Now, we need to add the bottom text. We'll start by copying our original circle and scaling it larger from the center so that the top of it touches the tops of the letters we curved in *step 5*.

6. Then, we must select both the new circle and the text for the bottom tagline – choose **Text** > **Put on Path**, as before. This time, however, we will flip our circle horizontally (by selecting it and tapping the *H* key once) to flip the shape and thus reverse the path.

   This makes the text flow along the inside of the circle. We must then rotate the circle to position the text and use the **Spacing** popover in the Tool control bar of the **Text** tool to give our letters some more breathing room. This process is shown in *Figure 7.41*:

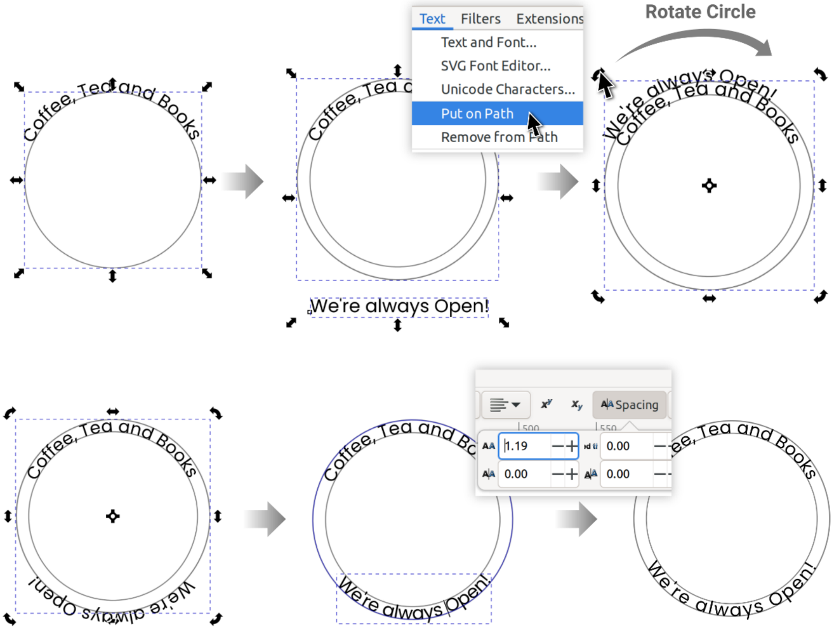


Figure 7.41 – Making the bottom part of the circular tagline

7. Now that we have the general shape of our text, we can build out some of the other parts of the logo using the shape tools we are already familiar with. This process is shown in *Figure 7.42*:

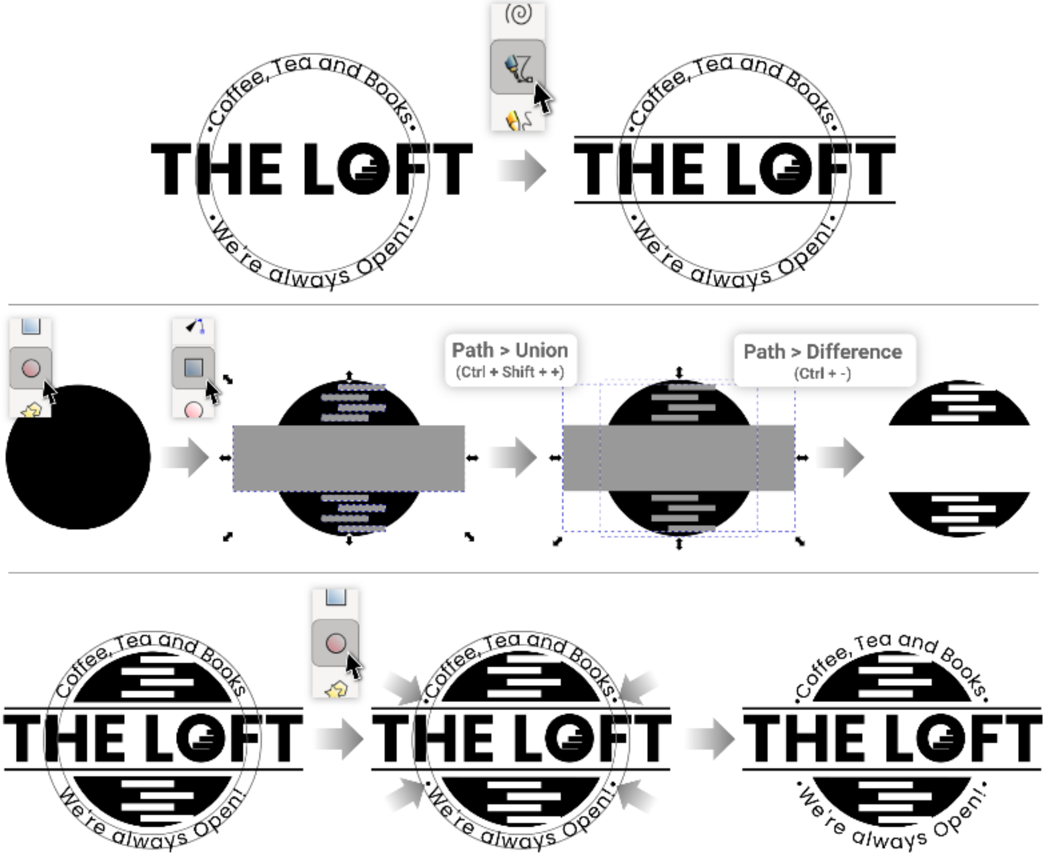


Figure 7.42 – Making the non-text elements of the logo

We now have the main text components of our logo, but we need a shape to flow the paragraph text around so that it appears to flow around our logo.

8. To do this, we will simply make a rectangle and a circle behind our logo, **Union** them together, and select both the text and shape by choosing **Text** > **Set Subtraction Frame**, as shown in *Figure 7.43*:

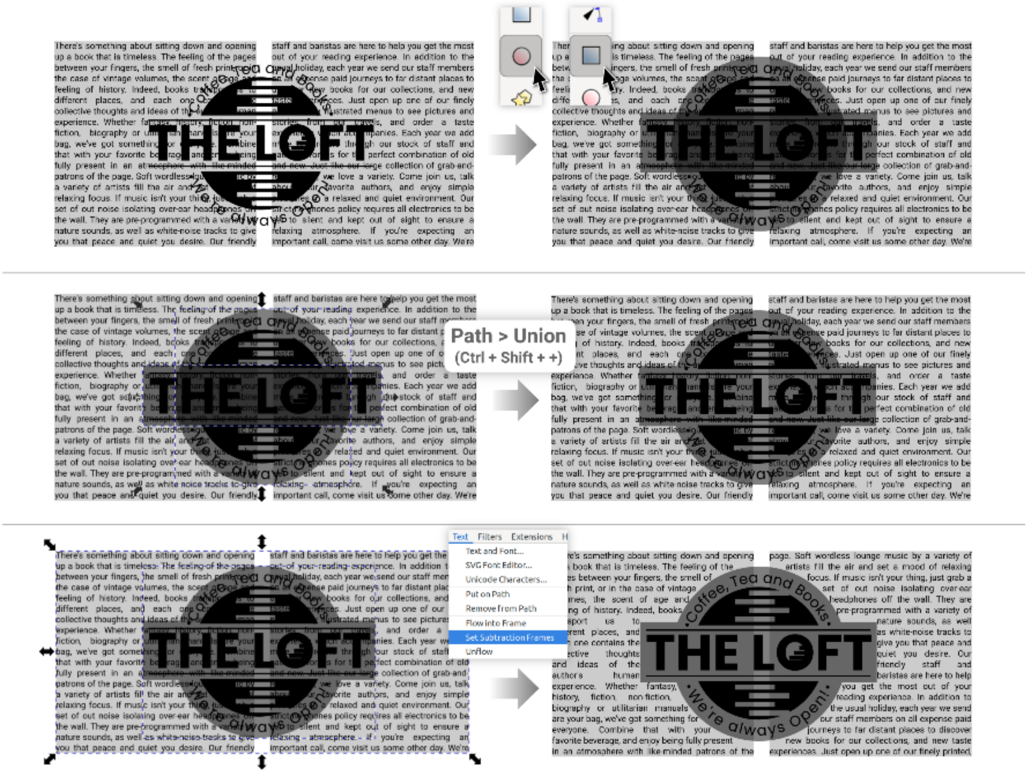


Figure 7.43 – Making the Set Subtraction Frame shape flow the text around the logo

9. Now, we just need to grab the bottom handles of our two paragraph rectangles (using the **Node** tool of course) and drag them down until our text fits. Then, we must simply move the logo and **Subtraction Frame** shape down in the center and remove the fills on our **Subtraction Frame** shape and the paragraph flow shapes, as shown in *Figure 7.44*:

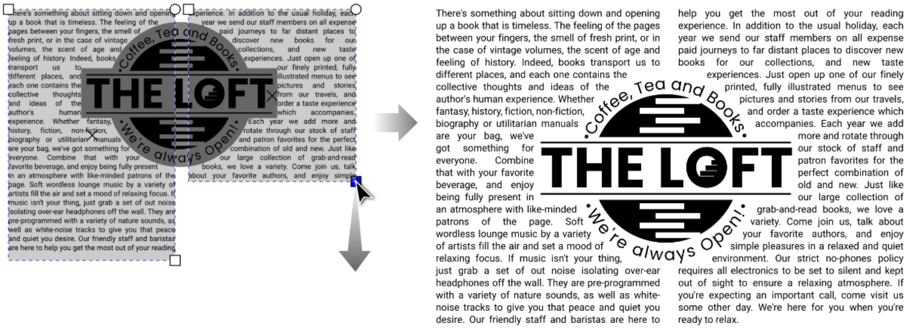


Figure 7.44 – Resizing the paragraph flow rectangles (left) and the result after removing the fills on the rectangles and Subtraction Frame shape (right)

That completes our logo and article combination!

Summary

We've covered a lot in this chapter! We learned how to install fonts, create and edit text, and flow it along paths. We also learned how to flow paragraphs through some shapes and around others. We even got to learn some typographical terms such as **kerning** and **ligature**, as well as play with some customizable variable fonts. All this information is quite useful for adding text to charts, diagrams, logos, and fliers, or making the latest issue of your very own e-zine.

Since we now have command of the **Text** tool and its accompanying functions, we'll move on to some advanced coloring techniques in the next chapter.

8

Advanced Shading and Coloring

So far, we've been using simple flat colors and a bit of opacity here and there to get the colors and effects we want. Inkscape allows us to do much more than that, fortunately. In this chapter, we will discover many ways to blend colors together in our shapes for even more advanced, realistic coloring.

We will cover the following topics in this chapter:

- The **Gradient** tool
- **Gradient** options in the **Fill and Stroke** dialog
- Mesh gradients
- Patterns
- Using blur for complex shading
- Color blending with blend modes

Technical requirements

You can download the practice files for this chapter from GitHub at

`https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape/tree/main/Chapter08`

The Gradient tool

Gradients are color transitions. For example, we may want our shape to be black on the right, transparent on the left, and have a smooth blending of both between. Let's explore this scenario by first drawing a black rectangle. We will then select the **Gradient** tool, click and hold where we want our black to start, then drag to where we want it to be fully transparent. *Figure 8.1* shows this process and the resulting gradient.

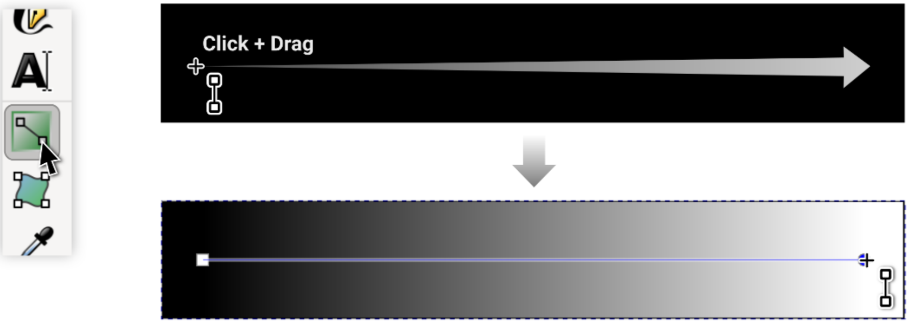


Figure 8.1 – Using the Gradient tool to make a black-to-transparent gradient

Note that after we've drawn that gradient, we can click and drag the handles to move the gradient around. Not only that, the gradient doesn't need to be completely inside the confines of our shape. For example, *Figure 8.2* shows what happens when we move the gradient end nodes outside the shape.

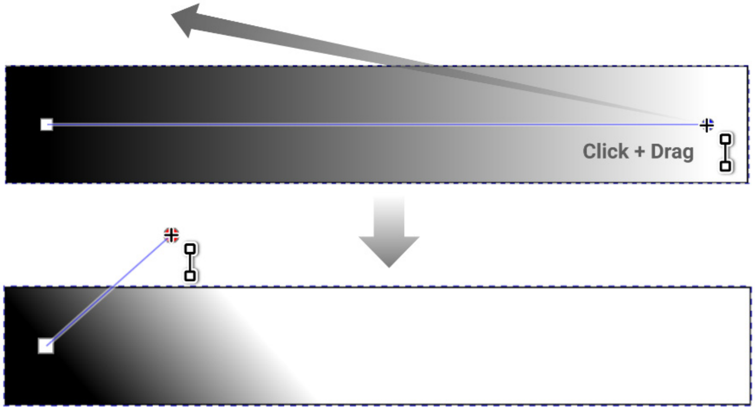


Figure 8.2 – Moving the transparent end node outside the shape

You can see that no matter where we move the gradient end nodes, Inkscape always makes a nice smooth gradient between the two points.

We can also make this gradient span multiple shapes at once by selecting them and clicking and dragging our gradient across the selected shapes, as shown in *Figure 8.3*.

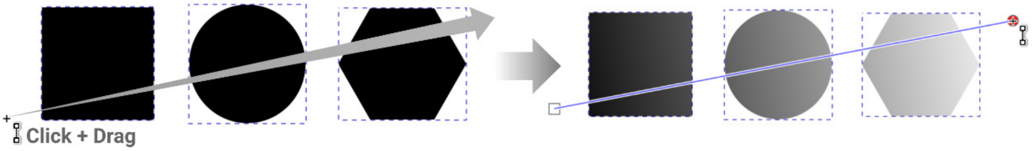


Figure 8.3 – Dragging a gradient across selected shapes

That's pretty neat, but what if we don't want our black to fade into transparency? Note that when you click a node in the gradient, it selects it. You can change the color and opacity of this selected node by clicking a swatch from the color bar at the bottom of the Inkscape interface, just as you would any regular object.

Not only that, you can add more nodes to the gradient line by double-clicking it. *Figure 8.4* shows the process of adding three more gradient nodes (also called **gradient stops**) and assigning the last node a different color.

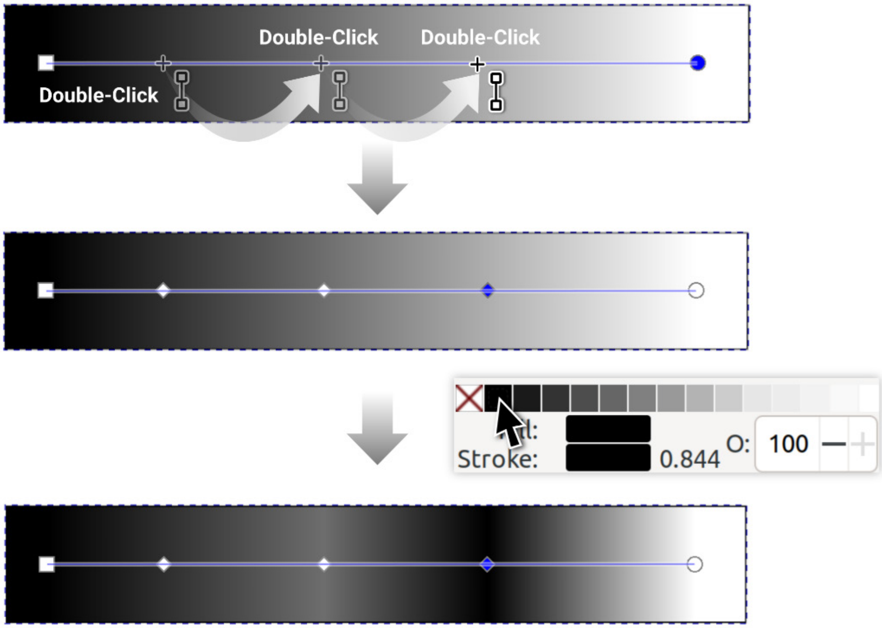


Figure 8.4 – Adding stops to the gradient and coloring the selected one with the black swatch

Pretty neat, eh? You might be able to see how we can easily make metallic-looking fills or rainbows using this method. We are not stuck with just a line gradient, though. We can indeed switch from a linear gradient like this one to a radial gradient. Using a circle with our linear gradient as an example, we can simply click **Fill Color Indicator** to bring up the **Fill and Stroke** dialog, and you can choose the **Radial gradient** option, as shown in *Figure 8.5*.

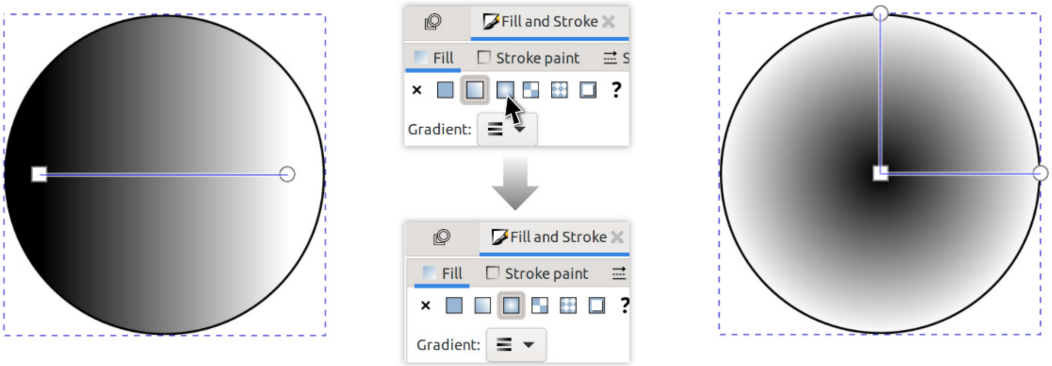


Figure 8.5 – Switching from linear gradient to radial inside a circle

You can see that we now have two gradient lines with handles instead of just one. Clicking and dragging the handle at the center moves the whole gradient. Clicking and dragging the handles horizontally or vertically squishes or stretches the oval shape, and moving those same end handles can also rotate the gradient, as shown in *Figure 8.6*.

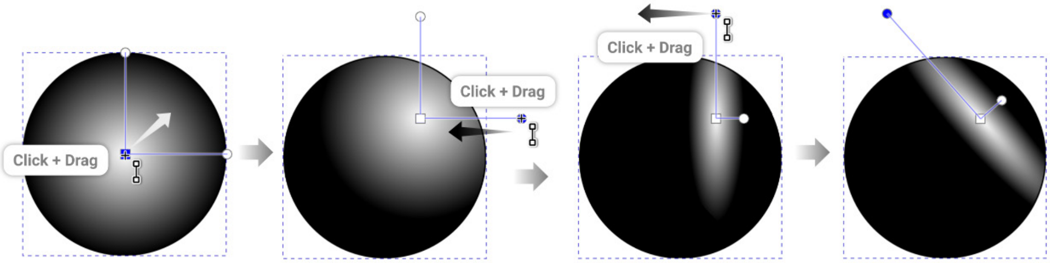


Figure 8.6 – Moving and adjusting gradient handles to resize and rotate the radial gradient

We also have control over what kind of gradients we make in the Tool control bar for the **Gradient** tool. Let's use another circle, but this time, we will change some settings in the Tool control bar to influence how our new gradient is applied to the flat-filled shape. *Figure 8.7* shows the Tool control bar options for the **Gradient** tool and what they influence when we click and drag a gradient.

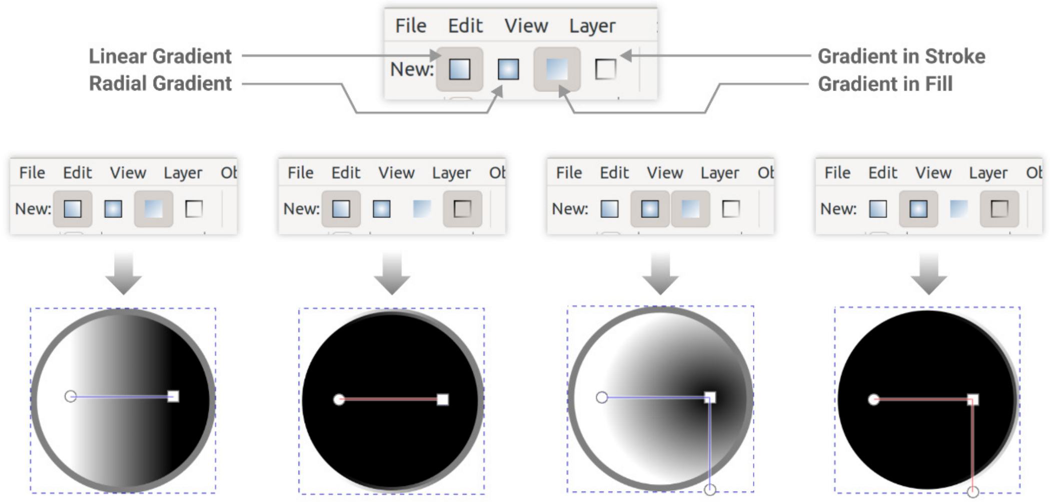


Figure 8.7 – Making new gradients with settings (left to right) –
Linear fill, linear stroke, radial fill, radial stroke

Note that it doesn't have to be just one or the other; we can have an object with a fill and stroke that have different gradients for each. We will use this later to get some neat shading effects in the *Try it yourself* practice section at the end of the chapter. Note that the gradient line for the stroke is in red, and the fill gradient line is blue. Inkscape will automatically snap the beginning and end nodes of these together if you get them too close to each other.

The reason it does this is that, in some cases, you want the gradients to flow in the same direction. This offers an easy way to snap them together to save you from having to manually move each. However, this can be annoying if you want to move them separately. The solution is to hold the *Shift* key while clicking and dragging your gradient start or end stop.

This will prevent snapping and also release the snapping behavior if they are already snapped together. *Figure 8.8* demonstrates this behavior.

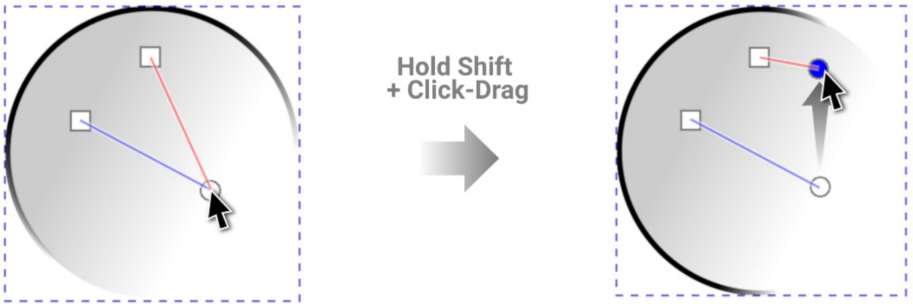


Figure 8.8 – Holding Shift and dragging handles to separate the gradient line end stops

To the right of the **New** section of the Tool control bar, we have a **Select** drop-down menu, which contains a handy list of all the gradients on shapes in our document. This allows us to choose a gradient we've previously made and use it in place of the current gradient. Note that we need to have made a gradient on the shape first for this list to show anything. *Figure 8.9* shows the result of choosing different gradients out of the list on a circle shape with a gradient.

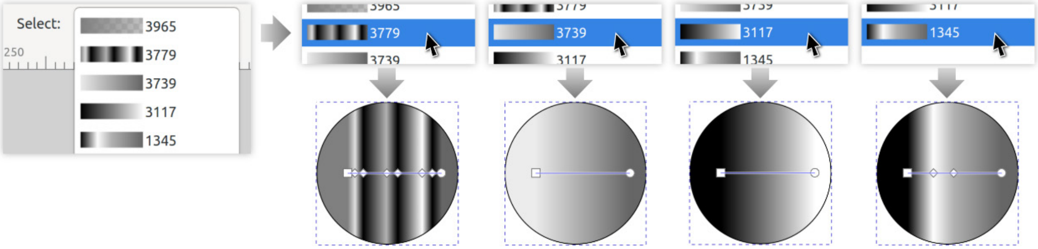


Figure 8.9 – Changing gradients using the Select drop-down menu in the Tool control bar

You may have noticed the little lock icon next to the **Select** drop-down menu. This toggle allows you to edit all the gradients assigned to the same gradient. By default, Inkscape will make a new gradient when you make changes, such as adding a white color stop to the middle of your gradient. That is to say, Inkscape will not change the same gradient assigned to the other shapes if the lock is toggled off. *Figure 8.10* shows the effect of adding a white color stop to the same gradient assigned to three shapes.

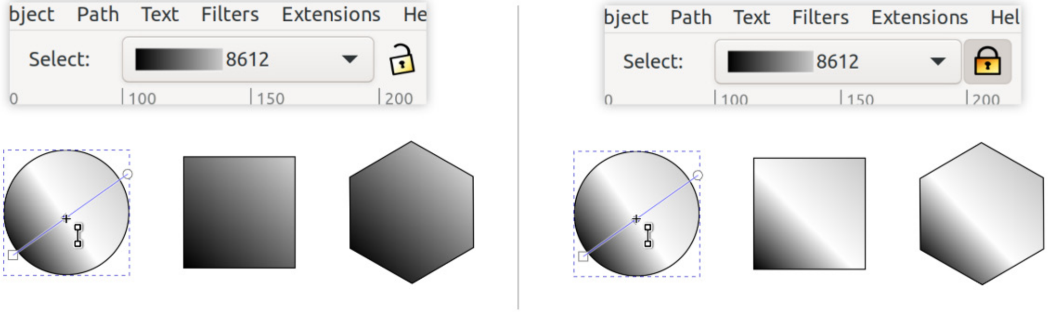

bject Path Text Filters Extensions He
Select:
8612
0
100
150
200
bject Path Text Filters Extensions Hel
Select:
8612
0
100
150
200

Figure 8.10 – Adding a color stop to a gradient without locking (left) and with locking (right)

Directly to the right of the lock toggle is an even smaller double-arrow icon. This is the **Reverse Gradient** button, and as the name suggests, it reverses the current gradient making the first stop into the last stop and vice versa. All the stops in the middle are reversed as well. *Figure 8.11* shows this effect on a radial gradient, which we'd like to reverse to give it that 3D-shaded look.

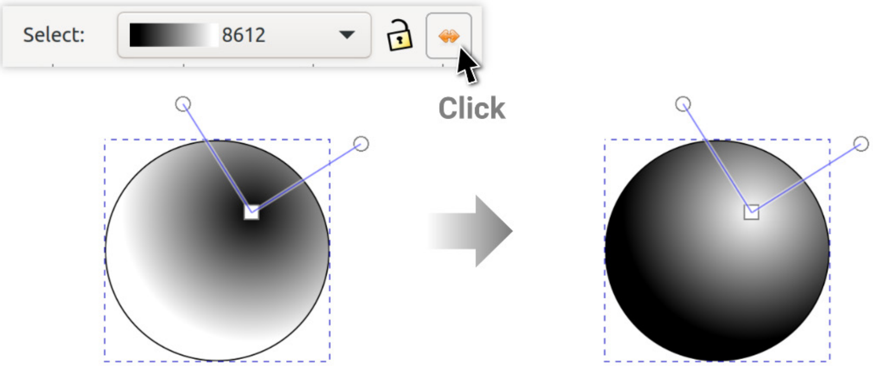


Figure 8.11 – Reversing the gradient so the white is at the inner gradient stop and black is on the outer gradient stops

Directly to the right of the **Reverse Gradient** button is the **Repeat** drop-down menu, which controls what happens before the first stop and after the last stop in the gradient. By default, the stop color is just extended infinitely at the start and end of the gradient. Setting this drop-down menu to **Reflected** makes the gradient mirror itself at the ends, which creates a smooth transition from one end to the other, repeating over and over into infinity.

Setting **Repeat** to **Direct** just restarts the gradient from the first stop at the end, making the pattern repeat without smooth transitions. These options are demonstrated in *Figure 8.12*.

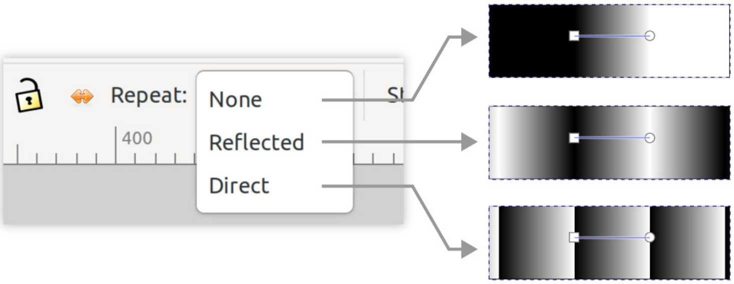


Figure 8.12 – The different Repeat options for the current gradient

Next up, we have a **Stops** drop-down menu for the current gradient. Choosing one from this menu selects it on the canvas. This is just another way of selecting the stops in a gradient, as opposed to just clicking one. The **Offset** field lets you manually adjust how far the current stop is from the beginning of the gradient, as you can see in *Figure 8.13*.

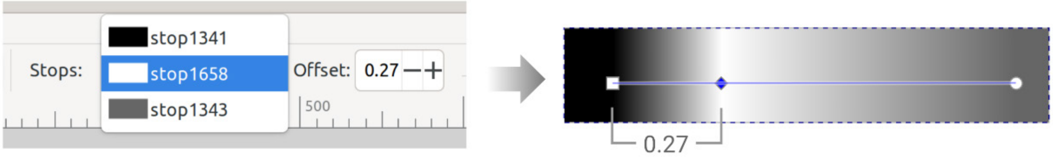


Figure 8.13 – Choosing a stop in the Stops drop-down menu to select the stop on the canvas

This is nice to have if there's a particular value you want to use to space your stops out evenly.

Selecting more than one stop on the gradient (by holding shift and clicking stops on the gradient stops), you can activate the **Add stop** and **Delete stop** buttons in the Tool control bar. As we learned previously, this works the same way as adding and removing nodes by double-clicking with the **Node** tool. Select two gradient stops and click the **Add stop** button, as shown in *Figure 8.14*.

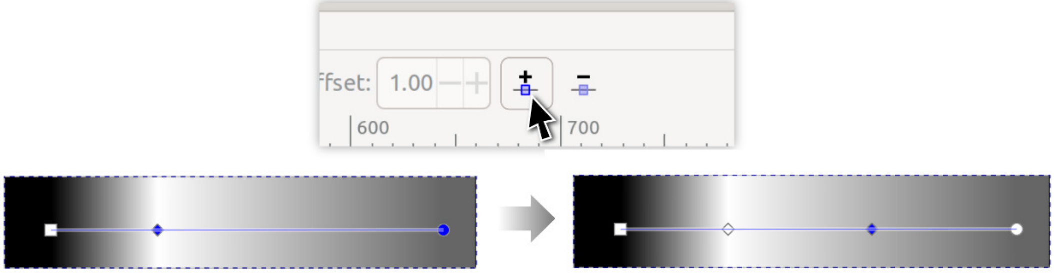


Figure 8.14 – Adding a gradient stop with the Add stop button

Predictably, clicking the **Delete stop** button will naturally remove any stops you have highlighted as well.

There are more hidden options in the **Fill and Stroke** dialog to accompany these great options in the Tool control bar. Let's explore some of those now.

The Gradient Editor

We've already gone through the finer points of coloring flat-shaded objects in previous chapters; however, when we add a gradient, the options in the **Fill and Stroke** dialog change to add a useful gradient editor. This can be seen as an alternative or addition to the Tool control bar when the **Gradient** tool is active. As you can see from *Figure 8.15*, this combines the usual color-mixing tools we've seen before in the **Fill** tab with the options in the Tool control bar for gradients.

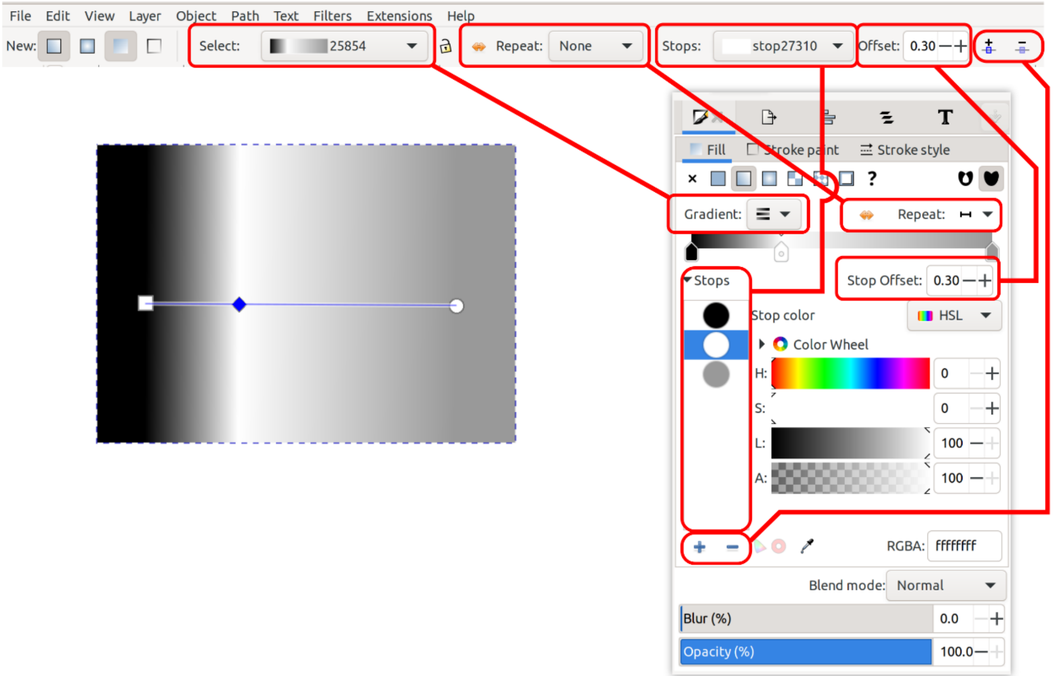


Figure 8.15 – The Gradient Editor in the Fill tab with equivalent features shown in the Tool control bar for the Gradient tool

So, why have all this duplicate functionality packed into the **Fill and Stroke** dialog? Remember that the Tool control bar options for the **Gradient** tool are only visible when the **Gradient** tool is active. By placing this functionality in the **Fill and Stroke** dialog, we can edit gradients on selected shapes without activating the gradient tool every time.

The on-canvas gradient editing features are also brought into the dialog. As you can see in *Figure 8.16*, you can edit the stops with the slider in the **Gradient Editor** to get the results you want.

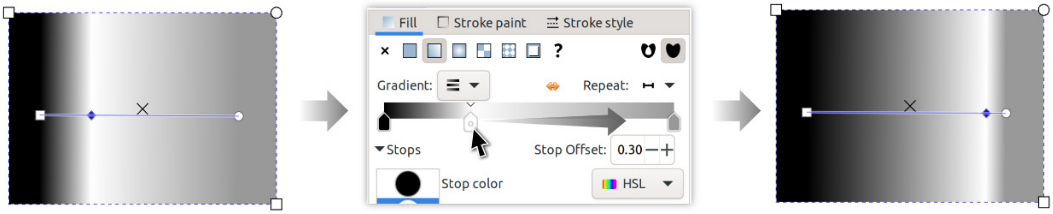


Figure 8.16 – Dragging stop handle in the Gradient Editor dialog moves the stop node on the canvas

It's nice having all the tools in one place, isn't it? You get fine control over the color using the color sliders (and even the color wheel if you expand it) for each stop without the need to move the mouse back and forth from the canvas to the **Fill and Stroke** dialog. This may not seem like much time, but when dealing with an illustration with hundreds of shapes, it soon becomes tedious to move the mouse back and forth so much.

So far, we've looked at linear and radial gradients, but what if you want a more complex gradient shape? Inkscape allows you to make custom grid-based gradient shapes, which we will explore in the next section.

The Mesh Gradient tool

Like the **Gradient** tool, this tool allows us to assign and edit gradients on the canvas. Instead of clicking and dragging a gradient, we simply select the **Mesh Gradient** tool and double-click the flat-color shape we want to assign a mesh gradient to, as shown in *Figure 8.17*.

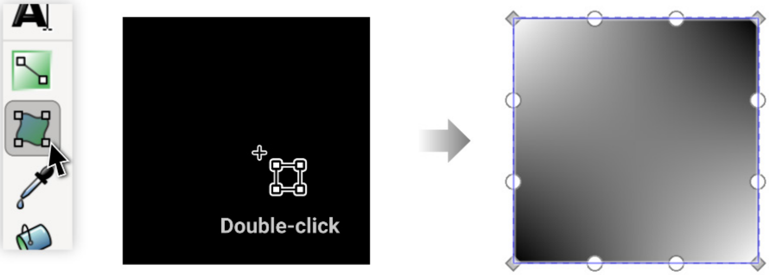


Figure 8.17 – Double-clicking on a black square with the Mesh Gradient tool active

By default, this gives us a 1x1 grid of color stops to work with. As before, we can select one of the diamond-shaped stops and click a swatch to assign it another color. Additionally, Inkscape gives us a set of little circular handles to change the curvature of the gradient between stops, so we can make whatever custom gradient shapes we like.

Figure 8.18 shows the default 1x1 mesh gradient before and after we move around the stops and handles to better fit the shape.

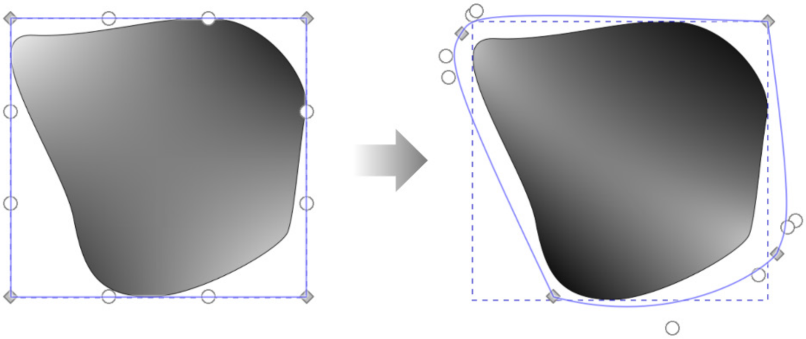


Figure 8.18 – A default mesh gradient edited to fit a non-uniform shape

In this way, we can sculpt the gradients to make shadows and light areas, which help convey a complex 3D shape for our illustration work. We aren't limited to a 1x1 grid of four nodes, however. Let's say we want more grid points to work with. All we need to do is double-click on one of the mesh grid lines to add another line to the grid, as shown in *Figure 8.19*.

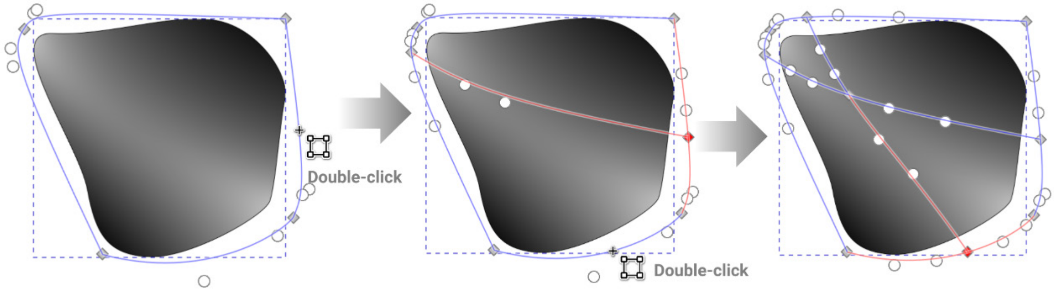


Figure 8.19 – Adding two more color-stop nodes to the grid, making a 2x2 mesh grid

You can see that this gives us more edge stops to use and a new stop inside the shape where the grid lines cross. *Figure 8.20* shows the result of selecting and changing this stop node to white.

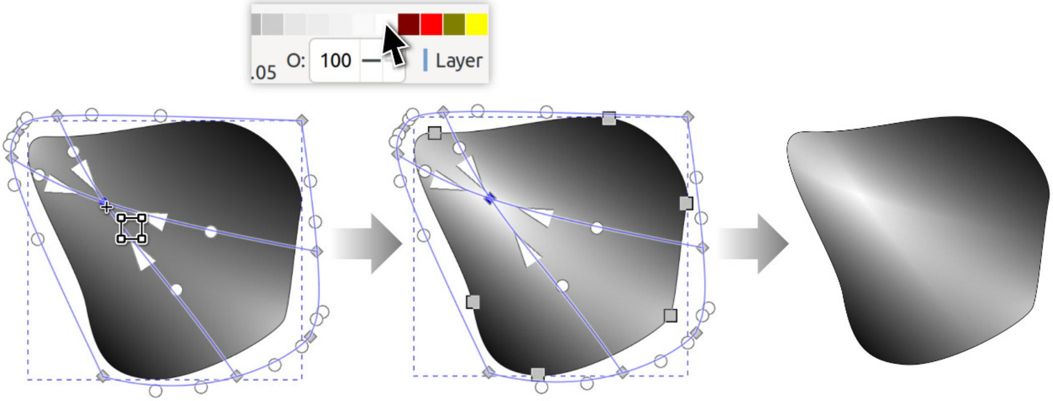
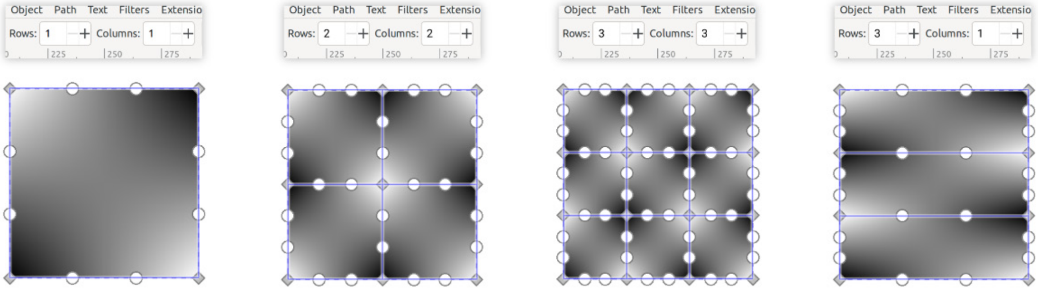


Figure 8.20 – Turning the new grid stop white

We don't have to add stops along the way, though. A quick glance at the Tool control bar shows that we can select the number of **Rows** and **Columns** Inkscape gives us when we first assign a mesh grid to a shape's fill or stroke. *Figure 8.21* shows the result of changing these values before double-clicking a solid color-filled shape.



Rows: 1 Columns: 1

Rows: 2 Columns: 2

Rows: 3 Columns: 3

Rows: 3 Columns: 1

Figure 8.21 – Different row and column mesh gradients added with the Mesh Gradient tool

Notice how we don't need to set an equal number of rows and columns to edit. Instead, we can just choose a number of gradient rows and/or columns that suit what we are doing. I recommend starting with a small number, however, as editing lots of grid color stops can get quite tedious.

On the far left of the Tool control bar, we see toggles similar to the ones we explored in the **Gradient** tool, which let us make gradients on the fill or stroke. Instead of the linear and radial gradient options, we have the **Mesh Gradient** and **Conical Gradient** options. The conical gradient, however, is just another sneaky mesh gradient.

This becomes obvious if you make a circle, add a mesh gradient, and click the **Conical Gradient button**. If you then try to click and drag to move the center stop, you will find there are more under it! If you keep doing that, you keep un-picking the overlapping gradient stops this way; you will see that a conical gradient is really just a 1x4 mesh gradient, which is distorted around to produce the conical gradient effect. This isn't too much of a problem though, as you can simply drag and select (rubber-band select) a selection box around the overlapping center stops and move them all at once. The difference is shown in *Figure 8.22*.

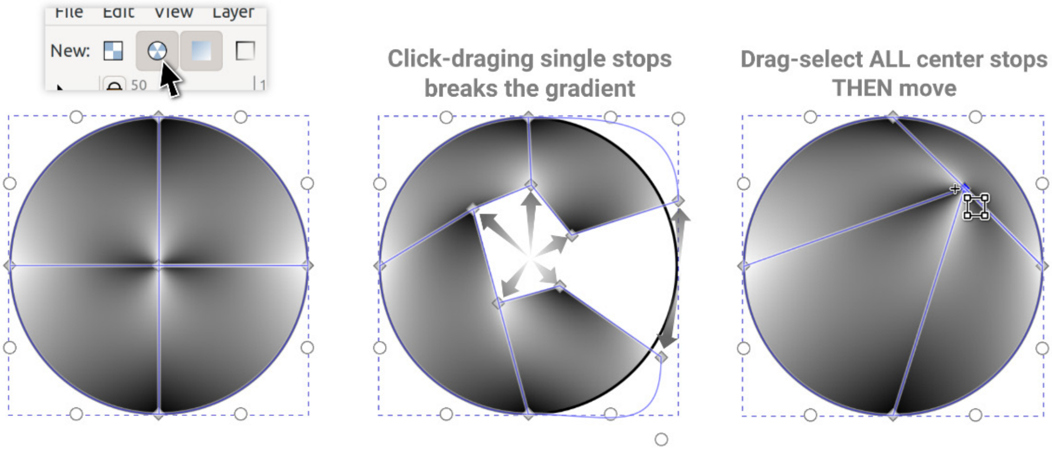

File Edit View Layer
New:
50
Click-draging single stops breaks the gradient
Drag-select ALL center stops THEN move

Figure 8.22 – Click and dragging versus drag and selecting the overlapping nodes in a conical gradient

Returning to our regular non-conical mesh gradients, there are a few more nice-to-have tools available in the Tool control bar for our stop-editing pleasure. For example, we can straighten out curved gradient segments and get rid of those pesky handles if we don't want them by clicking the **Toggle between Beziers and Lines** button. Then, if later we decide we'd really like the handles back, we can select the stops we want to add them back to and click the same button again, as shown in *Figure 8.23*.

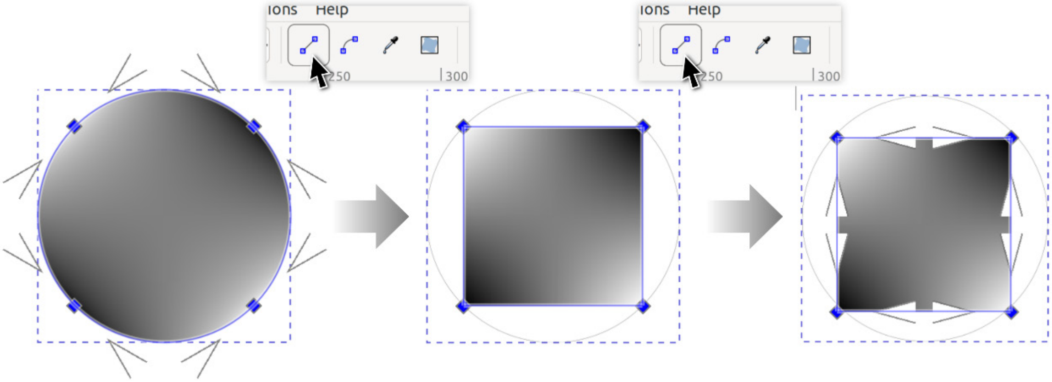


Figure 8.23 – Using the Toggle between Beziers and Lines button to straighten out grid lines and remove Bezier curve handles, clicking again to add them back

The other line icon which features a curve will try to smooth out your color stop handles to approximate an ellipse; however, at the time of writing, I could not get this to work as well as it does in the **Node Editing** tool with smooth nodes. Have a play with this yourself and see if you find it useful.

We have two more options in this section of the Tool control bar. The first looks like an eye dropper and will suck up the color of objects underneath the mesh gradient you're working on. This is handy if you're constructing a mesh gradient based on a photo. Let's say, for example, we have a 20x20 mesh gradient placed over a photo.

As you can see in *Figure 8.24*, by selecting all the stops in the mesh gradient and clicking the eye dropper icon, we can suck up the color into the mesh gradient.

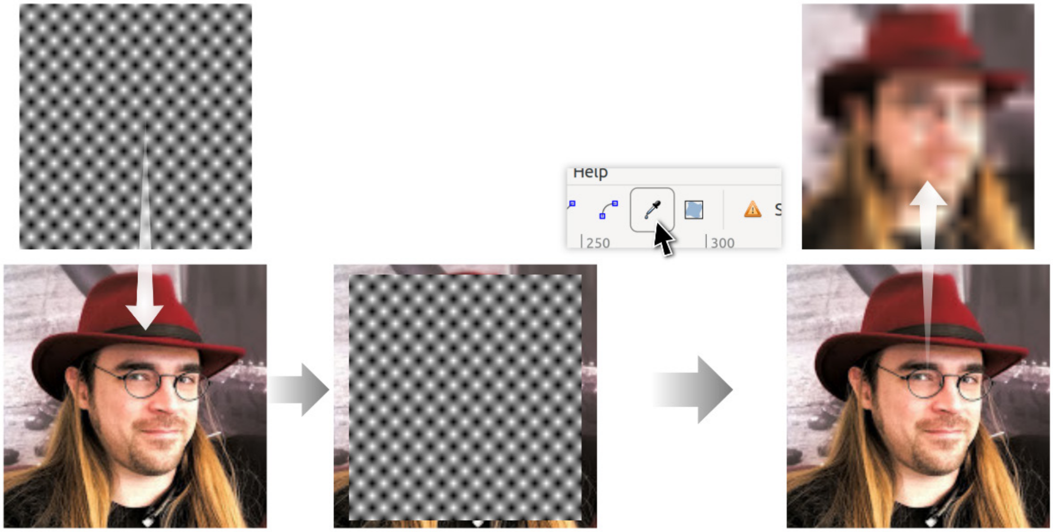


Figure 8.24 – Transferring color to a mesh gradient

The last feature in the Tool control bar for the **Mesh Gradient** tool is the **Smoothing** option. **Coons** is the default option, which means there is no smoothing across the grid boundaries. This creates sort of a four-pointed star gradient around each stop. We can change this to **Bicubic**, which adds smoothing across the grid boundaries. *Figure 8.25* shows a 2x2 mesh grid with both smoothing types for comparison.

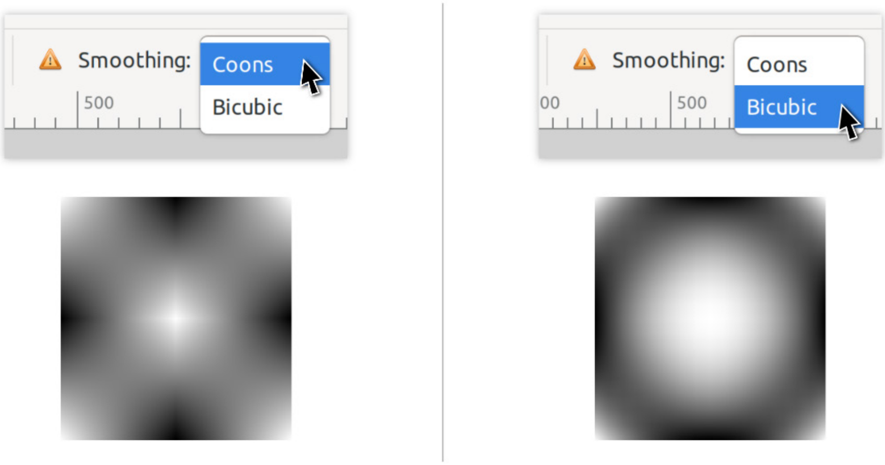


Figure 8.25 – Coons (no smoothing) versus Bicubic (smoothing)

Before we move on, you may have noticed the little orange warning icon to the left of the **Smoothing** drop-down menu. This is a warning that mesh gradients are not officially supported by the **Scalable Vector Graphics** (**SVG**) spec. This only matters if you are saving your SVG to display directly in web browsers, in which case you should convert your mesh gradient to a bitmap (**Edit** > **Make Bitmap Copy**) before saving it as an SVG for the web.

This concludes our basic run-through of gradients. In the next section, we'll see another option for filling our shapes: patterns!

Pattern Fills

We've covered quite a bit about gradients of various types up to this point, but what if we'd rather fill our shape with a repeating pattern instead? Inkscape allows us to fill our shapes with its built-in patterns through the **Fill and Stroke** dialog, by clicking the **Pattern Fill** button directly to the right of the **Mesh Gradient** button, as shown in *Figure 8.26*.

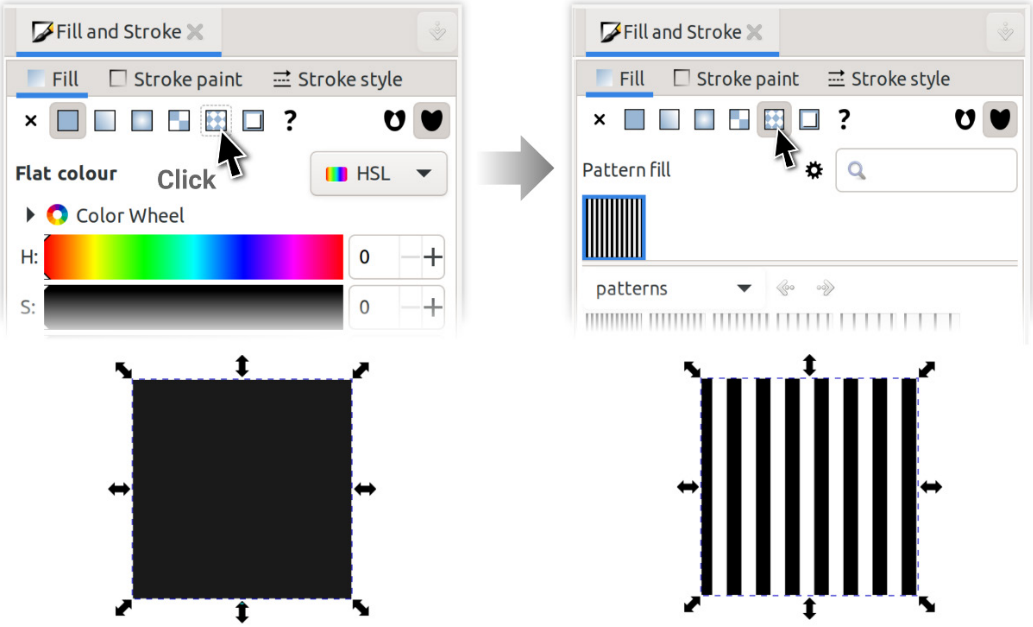


Figure 8.26 – Selecting the Pattern Fill option in the Fill tab of the Fill and Stroke dialog

At the time of writing, this gives us the default **Stripes 1:1** pattern fill (though it may show some other default as more patterns are added in the future), which you can see in the new (as of Inkscape 1.3) pattern menu along with other options. Switching to the **Node** tool, we can then see the pattern handles inside the pattern-filled object.

Note that if you're using an older version of Inkscape, the handles may be somewhere else on the page (you may have to look around). You can see this in *Figure 8.27*, where I've labeled the handles and then shown the pattern handles moved and rotated so you can see the effect of doing so (I've reduced the opacity of the object so you can see the handles better).

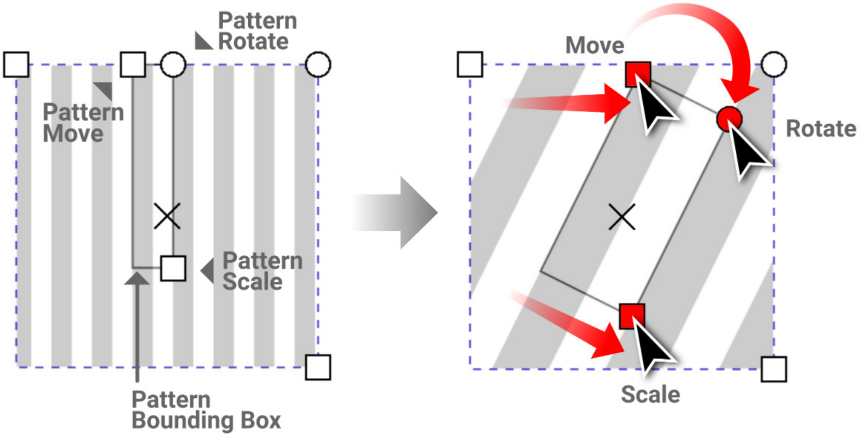


Figure 8.27 – Moving the handles from off-canvas onto the rectangle and rotating and scaling to get diagonal stripes

In this way, you can move, rescale, and rotate your pattern without moving your shape. You can even hold the *Ctrl* key to scale uniformly, move perpendicularly, or rotate incrementally). The **Pattern** menu in the **Fill and Stroke** dialog gives you a variety of black and white stripe patterns and a bunch of other options sorted into categories.

There's even a search bar integrated into Inkscape 1.3. and a little cog/gear button that lets you toggle on and off the names of all the patterns with the **Show names** checkbox, as shown in *Figure 8.28*.

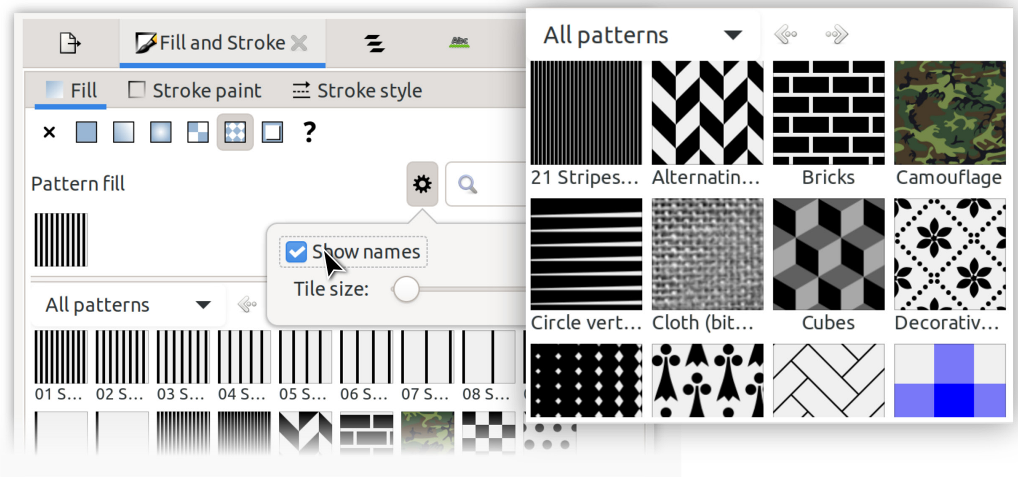


Figure 8.28 – Toggling on pattern names

Maybe you would prefer to use your own custom pattern, however. Inkscape makes this rather easy, and you can make a pattern out of almost anything simply by selecting your object or objects and choosing **Object** > **Pattern** > **Objects to Pattern**. Inkscape then uses the rectangular bounding box of your objects and repeats it infinitely in both directions, clipped to the original rectangle.

At first, nothing looks like it's changed, but if you use the **Node** tool to move the lower-right handle of the rectangle, you will see we now have control over the pattern within the rectangle, as shown in *Figure 8.29*.

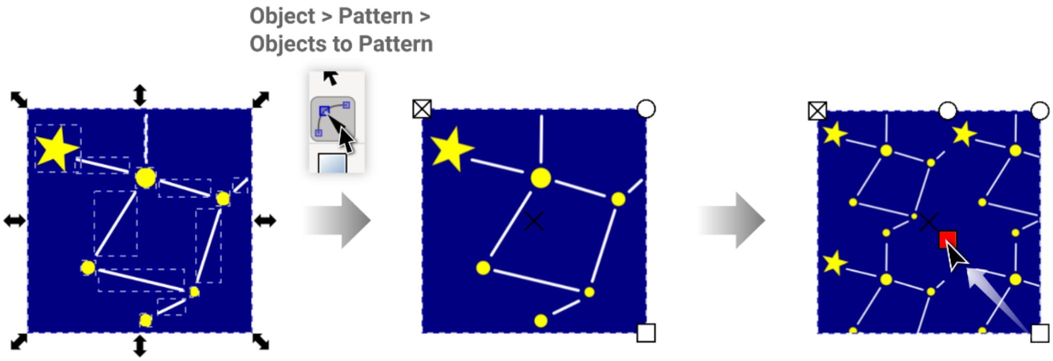


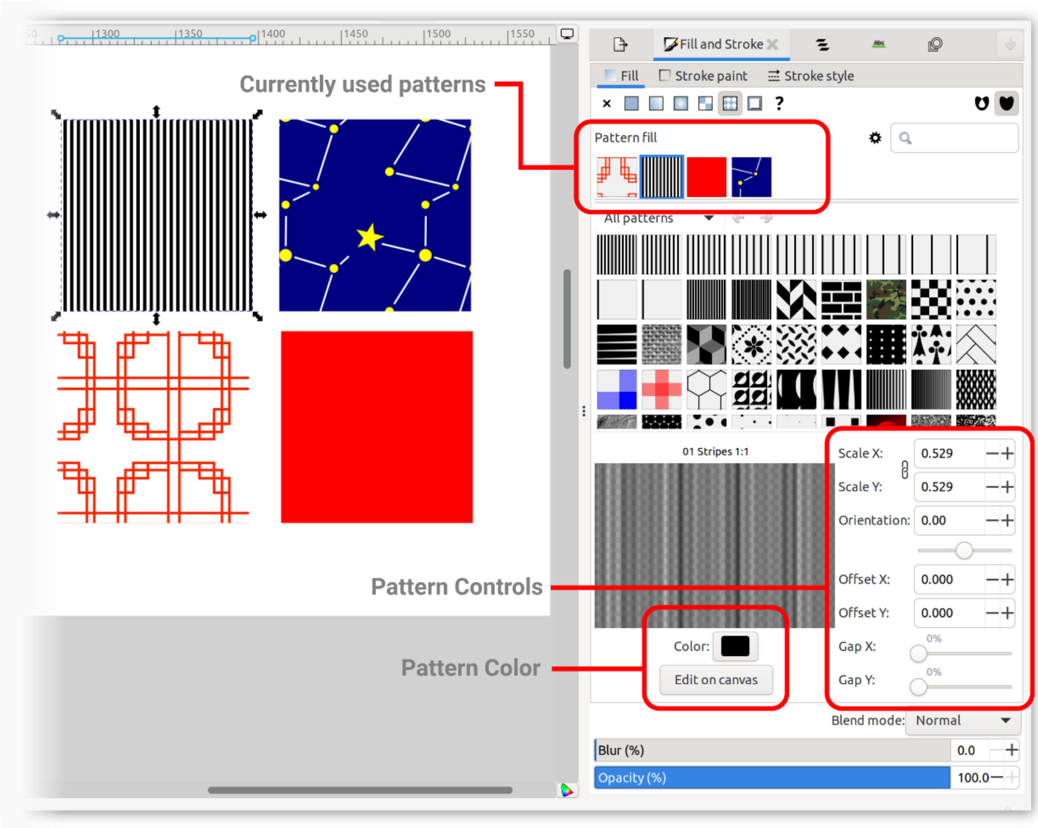
Figure 8.29 – Making a pattern from a selection of shapes

When you're making patterns, it's important to note that Inkscape will always make it rectangular based on the bounding box of the selected shapes. Therefore, I recommend adding a rectangular background to the shapes before converting them to a pattern, as it helps visualize the result.

Also note that Inkscape assigns the move handle to the top-left of the pattern, and the scale handle to the lower right, with the rotate handle at the top-right. As such, you may have to move them slightly out of the way to edit your shape.

You can get your original shapes back from your pattern by choosing **Object** > **Pattern** > **Pattern to Objects**. Note that any scaling, moving, or rotating you did in the pattern will be retained by your shape when you perform this operation, so your shape vanishes, have a look around the canvas for it. It's there somewhere!

Before we move on, let's explore the brand-new **Pattern** options, as shown in *Figure 8.30*.

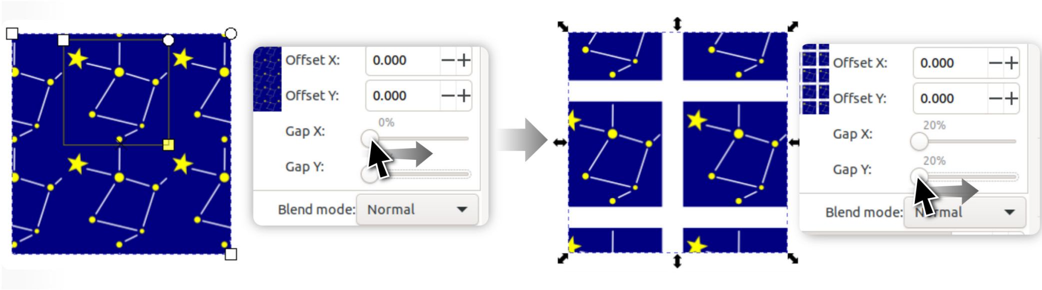


Scale X: 0.529
Scale Y: 0.529
Orientation: 0.00
Offset X: 0.000
Offset Y: 0.000
Gap X: 0%
Gap Y: 0%
Color: ■
Blend mode: Normal
Blur (%) 0.0
Opacity (%) 100.0

Figure 8.30 – New Inkscape 1.3 options for patterns

As you can see, there's now an area for patterns that are currently in use, even the one we made earlier with stars. Clicking one of these pattern blocks will change the pattern in all currently selected objects. Likewise, the **Pattern Controls** area show you the pattern values for your currently selected shape.

These values are set automatically when we use the pattern handles to scale (the **Scale X** and **Scale Y** values), rotate (the **Orientation** value), and move (the **Offset X** and **Offset Y** values), as we did earlier. There's one extra set of values, **Gap X** and **Gap Y**, which controls the spacing between each repetition of your pattern, in case you want to add gaps, as demonstrated in *Figure 8.31*.



Offset X: 0.000

Offset Y: 0.000

Gap X: 0%

Gap Y:

Blend mode: Normal

Offset X: 0.000

Offset Y: 0.000

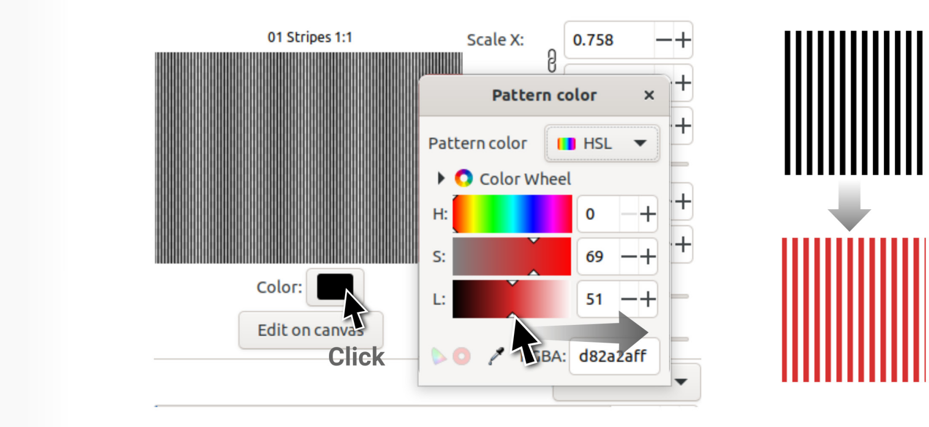
Gap X: 20%

Gap Y: 20%

Blend mode: Normal

Figure 8.31 – Increasing the gap size in the pattern

Lastly, for some patterns, such as the stripes, we can control the pattern's color. If, for example, we want red stripes instead of black stripes, we can simply click the **Color** button and choose a nice red color, for example, as in *Figure 8.32*.



01 Stripes 1:1

Scale X: 0.758

Color:

Edit on canvas

Click

Pattern color

Pattern color: HSL

Color Wheel

H: 0

S: 69

L: 51

RGBA: d82a2aff

Figure 8.32 – Changing the color of the stripe pattern

Now that we've covered the fill options, it should be noted that gradients and patterns are not the only way to shade and color your illustrations. In fact, mesh gradients can be a bit cumbersome when I'm using complex objects. I prefer to use **Blur** for this, as we will explore in the next section.

Using blur for complex shading

Mesh gradients are a great tool for advanced control of your gradients, but the more complex your shape is, the more complex your mesh gradient may have to be to get the effect you want. Also, the more color stops you're going to have to edit to get the gradient to conform to the shape.

Take, for example, a splotch shape made with the **Star and Polygon** tool, which we covered in *Chapter 3*, *Drawing Shapes with the Shape Tools*. It's nearly impossible to get a good shadow on this sort of shape with a mesh gradient. Rather than undertake hours of tedious stop-editing, we can start using our knowledge of Booleans to create an area for our shadow.

Simply copy the splotch shape, then select **Edit** > **Paste in Place**, and then move it up and over a bit to the left. We can then select both shapes and perform a difference path operation (**Path** > **Difference**) to get a shadow shape, then select **Paste in Place** again to get back our original splotch and move it down below our shadow shape. This process is shown in *Figure 8.33*.

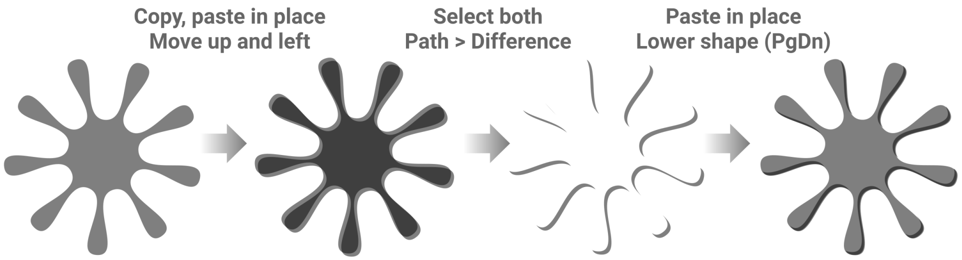
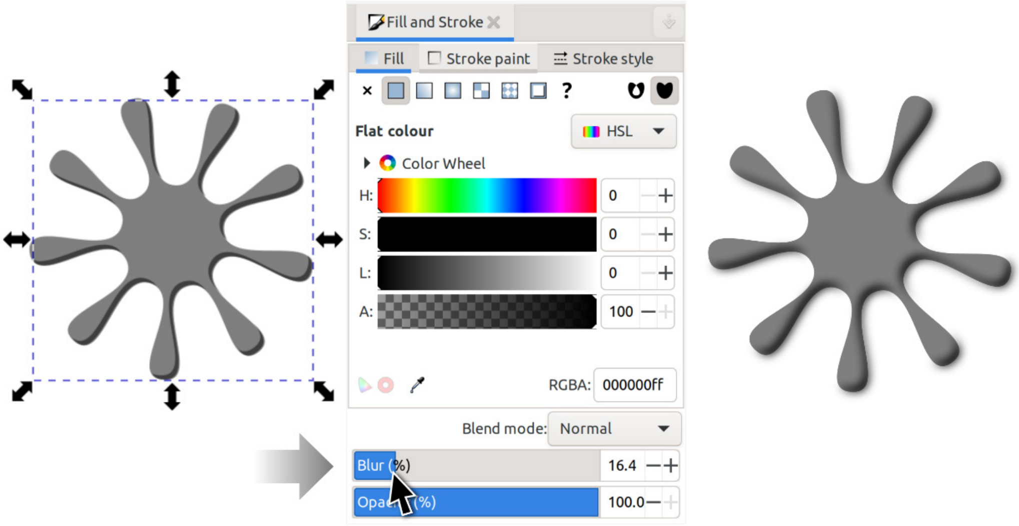


Figure 8.33 – The process of making our splotch shadow

This produces a hard edge shadow, though, and we'd really like a bit of a gradient transition to make it look rounded instead. To do that, we will use the **Blur** slider bar of the shape, located at the bottom of the **Fill and Stroke** dialog right above the **Opacity** slider bar. Dragging this value up a little (to the right) gives our shadow object a nice blur, as you can see in *Figure 8.34*.



H: 0
S: 0
L: 0
A: 100
RGBA: 000000ff
Blend mode: Normal
Blur (%) 16.4
Opacity (%) 100.0

Figure 8.34 – Setting blur on the shadow shape

It really gives it a nice, rounded effect doesn't it? This also gives us a nice drop-shadow effect, making the splotch look like it's sitting on a white surface. Sometimes, however, we'd like to exclude that drop-shadow effect and limit the blur to only the inside of the shape. For this, I use Inkscape's **Clip** feature, which we will discuss in much greater detail in later chapters, but for now, we can use it to clip our shadow to the inside of the splotch.

We do this by selecting **Paste in Place** to place our original splotch shape over the shadow. We then select both the splotch shape and shadow and choose **Object** > **Clip** > **Set Clip**. This will clip (or cut off) the parts of the shadow outside the shape, eliminating the drop-shadow effect, as shown in *Figure 8.35*.

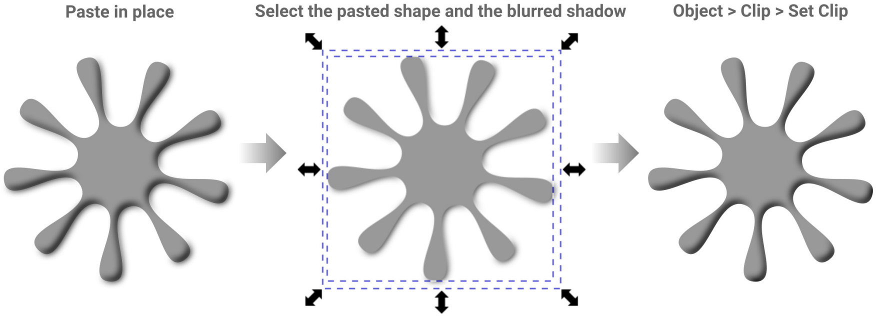
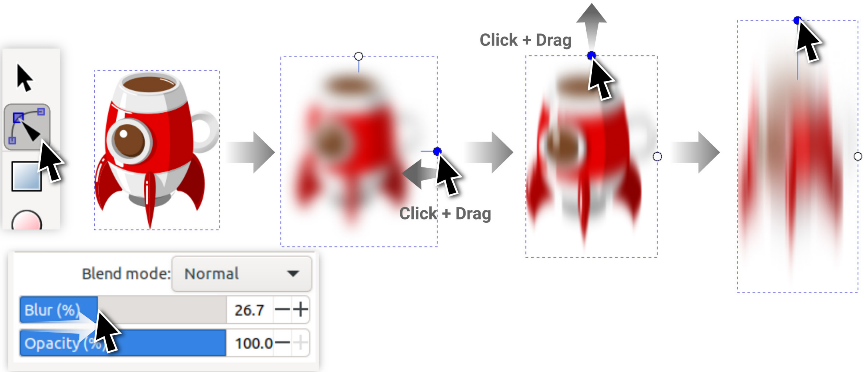


Figure 8.35 – Using Clip to clip off the extra bits of the blurred shadow

This is the method I use most for shading complex shapes as it's quite fast and easy and produces good results.

You may have noticed that we have extra handles on our shape all of a sudden when we add some blur with the slider and switch to the **Node** tool. These are the new **Blur Control** handles, and they allow us to increase or decrease horizontal and vertical blur separately. This is useful for effects such as speed blur, which blurs in one direction more than another, as demonstrated in *Figure 8.36*.



Blend mode: Normal
Blur (%) 26.7
Opacity (%) 100.0

Figure 8.36 – Simulating speed blur with the new Blur Control handles

As you can see, we can send my rocket coffee mug blasting off by dragging the horizontal blur handle inward and dragging the vertical blur handle outward. Remember that, like most other handles in Inkscape, these handles only appear when the **Node** tool is active.

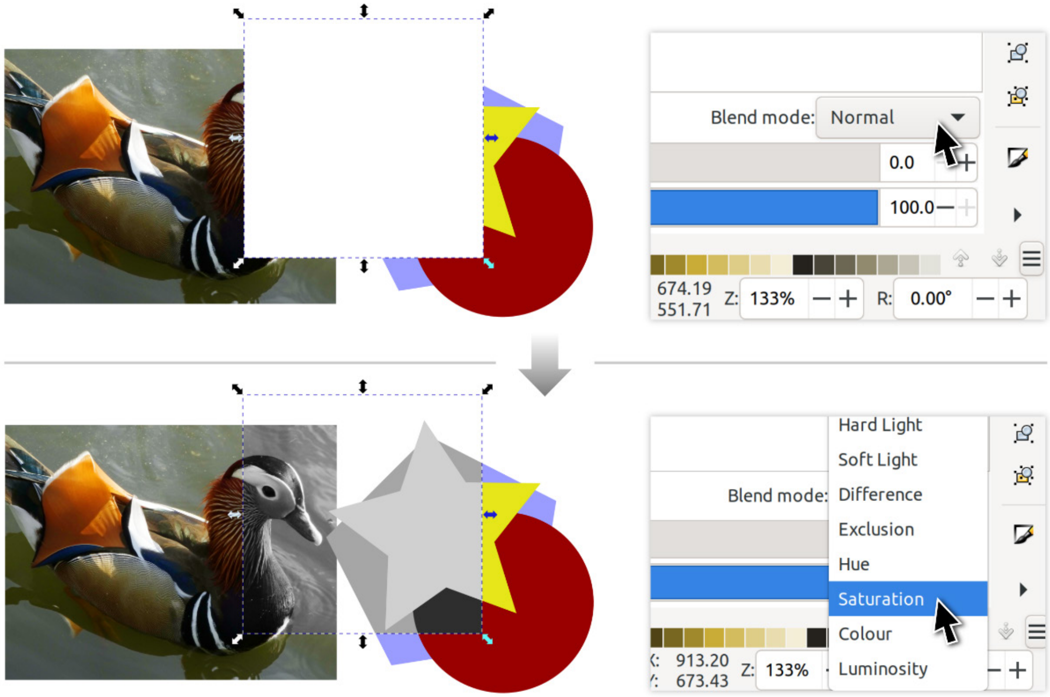
They also vanish if **Blur** is set to `0.0`, so make sure you add a little starting blur first, and then you can tweak these new blur handles to your satisfaction.

Before we move on to our practice lesson, let's take a quick dive into the last area of the **Fill and Stroke** dialog, which allows us to use color and shapes to perform some color modification tasks.

Color blending with Blend modes

The last stop on our coloring journey gives us tools that normally one would use photo-editing software such as GIMP or Photoshop to do. Say, for example, we have a group of shapes or a picture that we'd really like to make grayscale. For the shapes, we could just re-color them but this may prove quite tedious across many shapes with gradients, and for the photo, we have just a bunch of pixels to deal with.

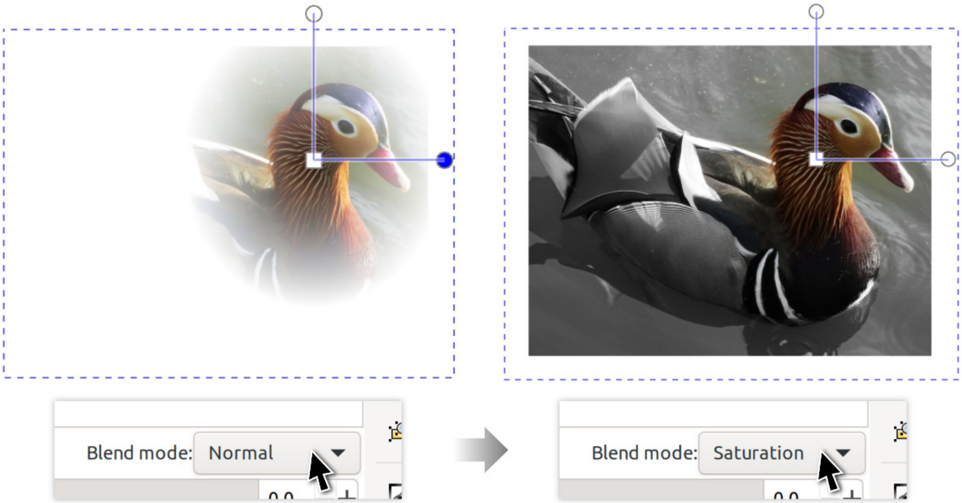
Fortunately, we can do as we desire using a white rectangle and changing the blending mode to **Saturation**. If we put this rectangle over our picture and shapes, you can see where it sucks all the color out of those areas in *Figure 8.37*.



Blend mode: Normal

Figure 8.37 – Changing the Blend mode on a rectangle from Normal to Saturation

The **Saturation** blending mode takes the color of the square and makes the objects under it the same saturation level. It doesn't have to be a flat color, however. For example, we can use a radial gradient fill to blend from grayscale to color, as shown in *Figure 8.38*.

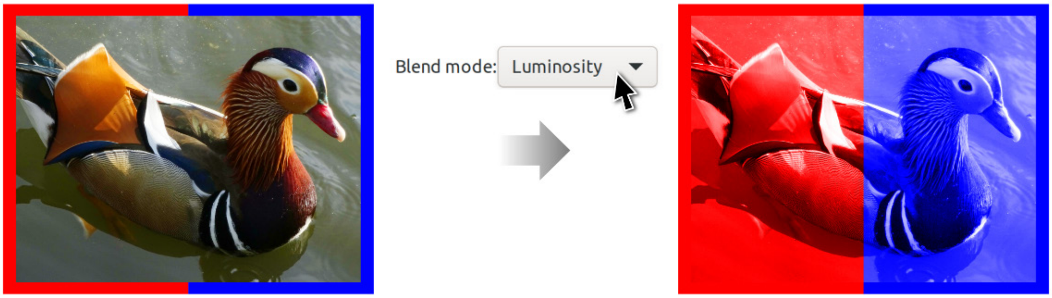


Blend mode: Normal

Blend mode: Saturation

Figure 8.38 – Using a gradient to control the color blend effect

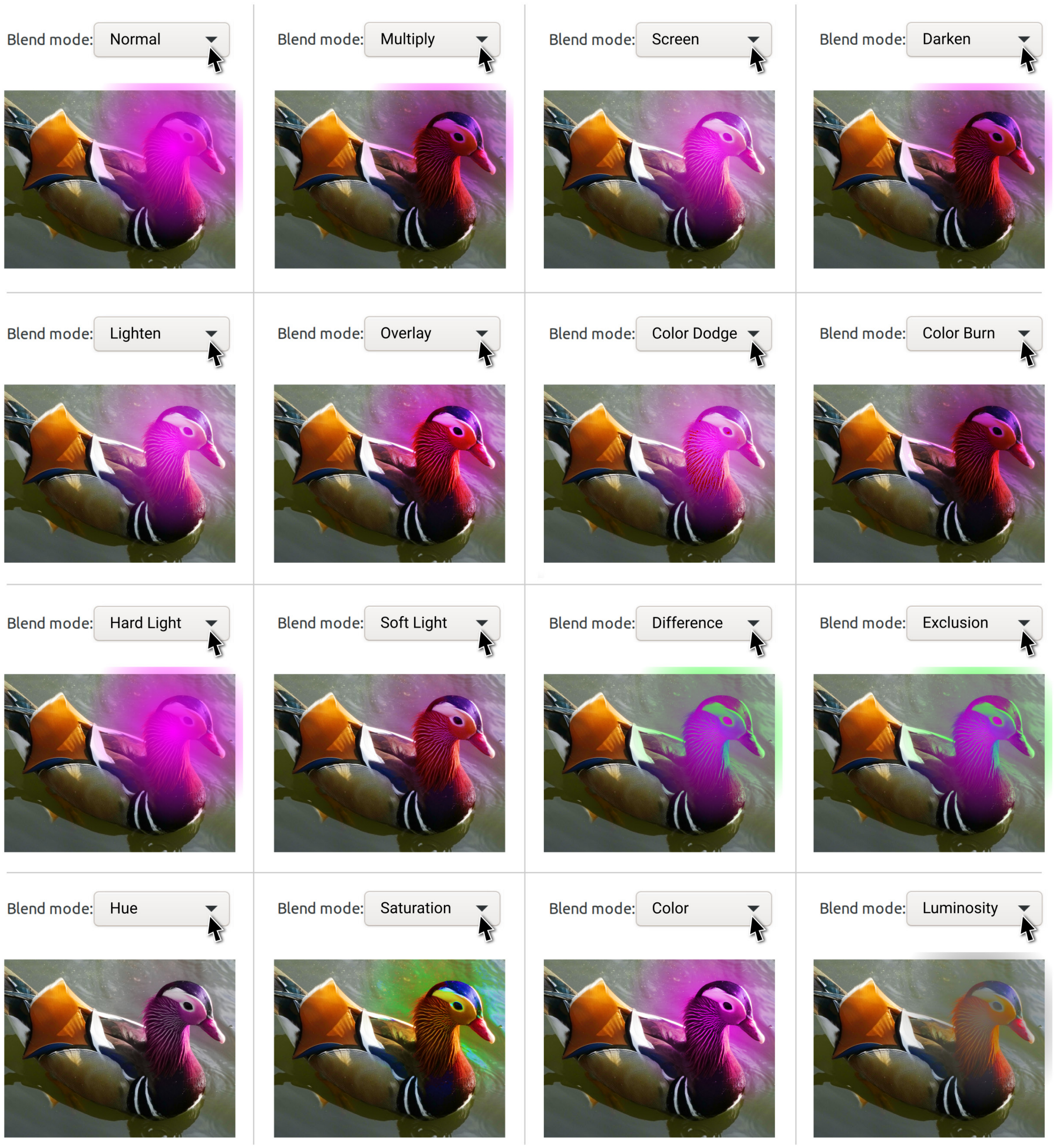
Blending modes also work for images; for example, we can use our duck picture to grayscale parts of a flat-shaded object, as shown in *Figure 8.39*.



Blend mode: Luminosity

Figure 8.39 – Setting the Blend mode on an image to change its color to the red and blue rectangle underneath

The preceding example uses the **Luminosity** blending mode, taking the lightness of each value in the image and projecting it onto the colored rectangles beneath. *Figure 8.40* shows a radial gradient (pink) in a rectangle with various blending modes over our duck picture to demonstrate the effects.



Blend mode: Normal

Blend mode: Multiply

Blend mode: Screen

Blend mode: Darken

Blend mode: Lighten

Blend mode: Overlay

Blend mode: Color Dodge

Blend mode: Color Burn

Blend mode: Hard Light

Blend mode: Soft Light

Blend mode: Difference

Blend mode: Exclusion

Blend mode: Hue

Blend mode: Saturation

Blend mode: Color

Blend mode: Luminosity

Figure 8.40 – Demonstration of various blending modes in Inkscape

In general, the **Blend** mode describes the operation the color of the shape has on objects underneath it. Thus, **Multiply**, **Darken**, and **Color Burn** take the pink gradient and use it to darken the photo underneath it in various ways, while **Screen**, **Lighten**, and **Color Dodge** lighten the colors with that same pink.

Overlay, by contrast, lightens the light parts and darkens the dark parts based on that same pink gradient. **Soft Light** takes the lightness of the pink and treats it as if it were a soft pink light, shining over that area of the image. The **Hue**, **Saturation**, and **Color** blending modes use pink to change the color of the image underneath. With **Saturation** making the colors of the duck's head much more vivid to match the saturation of the hot pink (recall earlier we used it to suck the color out with a desaturated white), and **Hue** shifts the hue of the colors towards the pink range without changing the lightness or darkness of the colors.

I recommend playing with these blending modes in your projects to discover their hidden utility for color correction. They can be a real time-saver and add that extra bit of realism to your work. Armed with our new advanced knowledge of gradients, patterns, and coloring. Let's get in some practice!

Try it yourself

Let's take a picture and do something fun with it! Feel free to use whatever image you like for this. After all, it is *try it yourself*. If you'd like to use my duck photo, it's available in the materials for this chapter. We'll be turning whatever picture you choose into the attractive, glossy postage stamp picture in *Figure 8.41*.

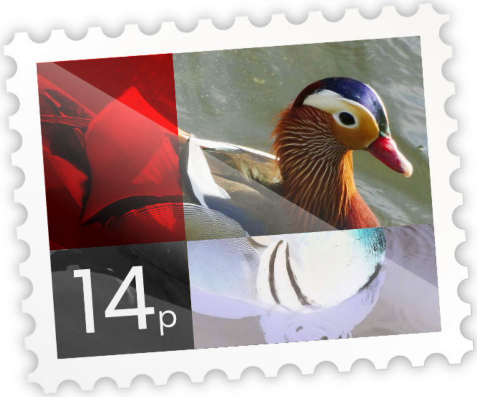


Figure 8.41 – The postage stamp illustration we're going to make in this practice lesson

1. We'll start by importing our image, scaling it down, drawing a white rectangle, and moving it behind the image. We will then rescale the rectangle to make a nice even border around the image, as shown in *Figure 8.42*.

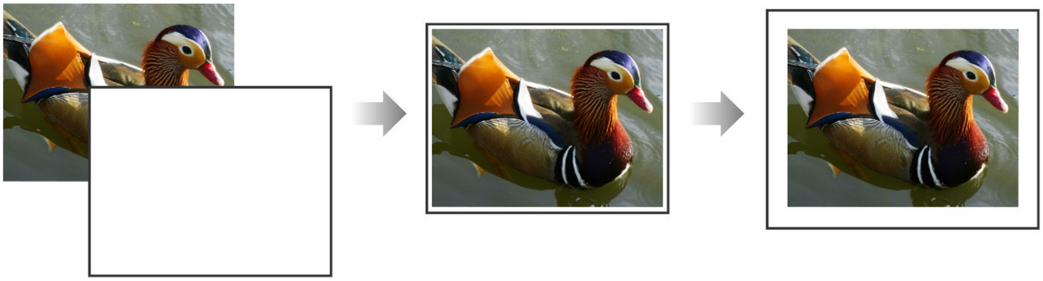
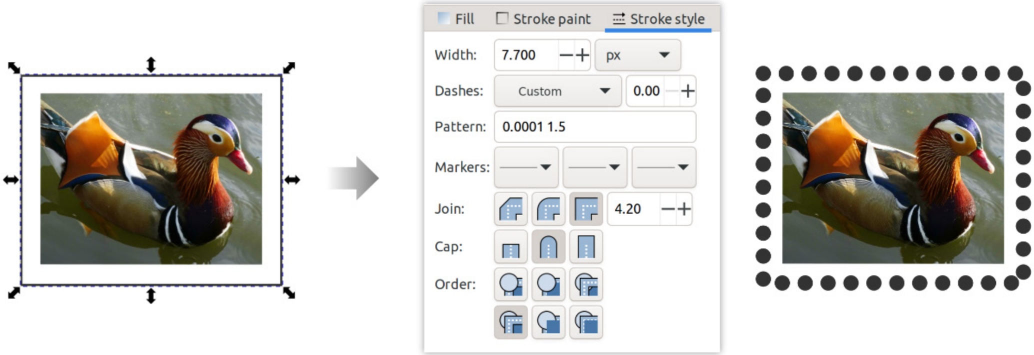


Figure 8.42 – Draw and move a rectangle behind the image, resizing it to add a thick border to cut out stamp perforations

So how do we make those neat stamp perforations along the edges? Well, we could just make a bunch of circles and use the alignment tools and our Boolean operations, but we can save ourselves the hassle by using a dashed line.

2. We simply select our rectangle and change the stroke to a thicker value and the **Dashes** value field to `0` to turn those rounded stroke ends into circles to get a nice, dotted line. Settings for this are shown in *Figure 8.43*.



Width: 7.700 px

Dashes: Custom 0.00

Pattern: 0.0001 1.5

Markers:

Join: 4.20

Cap:

Order:

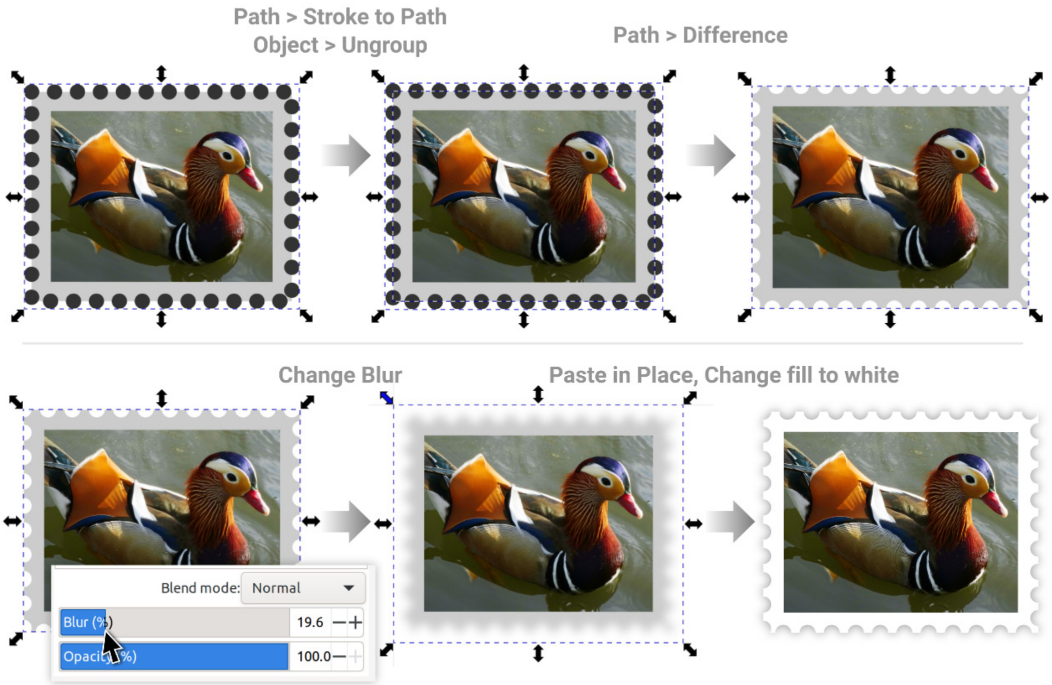
Figure 8.43 – Using stroke properties to make dots around the border of our postage stamp

Note that if your dots turn out really small with these values, you can turn up the stroke width to make them larger.

3. Now, let's change the fill of the rectangle to gray so we can see it better. We need to convert those custom dashes to filled shapes now, so we perform a **Path** > **Stroke to Path** operation.

4. This will give us a group of the dots and the rectangle. We can ungroup those (*Ctrl* + *Shift* + *G*) and, with both selected, select **Path** > **Difference** to cut the dots from the rectangle, completing our perforated stamp edge shape.

5. To make the shadow beneath, we just copy our gray rectangle and crank up the blur value a bit (at the bottom of the **Fill and Stroke** dialog) to complete our shadow. We can then use **Paste in Place** on our original blurred stamp shape and change the fill to white. This process is shown in *Figure 8.44*.

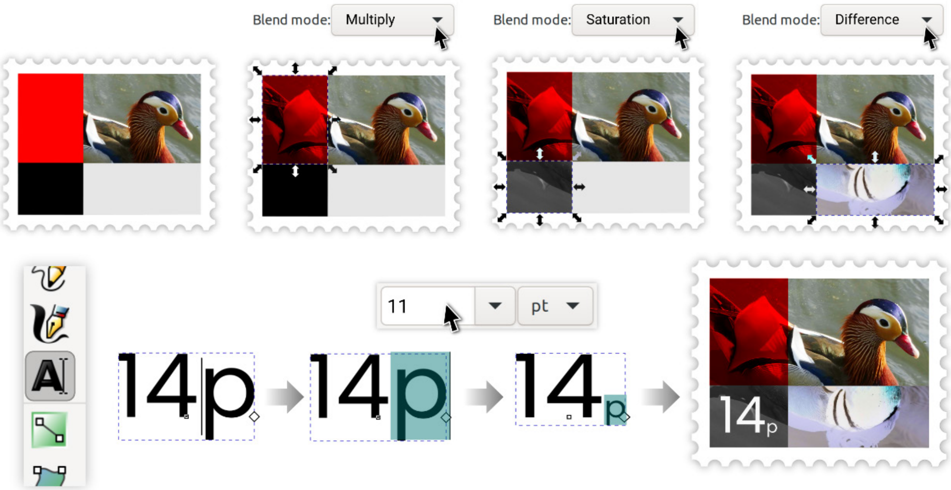


Blend mode: Normal
Blur (%) 19.6
Opacity (%) 100.0

Figure 8.44 – Making the perforated stamp shape and shadow underneath

Let's make the color effects of the stamp by drawing a series of rectangles (or other shapes if you want) over the picture. We will then assign a color to each and set their blending modes to things we find attractive.

6. This is a great opportunity to play with the various effects to get something you like. Add some text for the monetary value of your stamp, changing the text and color to your liking. My results are shown in *Figure 8.45*.



Blend mode: Multiply

Blend mode: Saturation

Blend mode: Difference

Figure 8.45 – Adding rectangles with blend modes to make our stamp more interesting

7. We can now add a 1x1 mesh gradient to the stamp with the **Gradient** tool by double-clicking on our white stamp backing. This adds a little variation to the white of our stamp, making it look slightly more 3D. Adjust the opacity values for the gray color creates a more subtle effect, as shown in *Figure 8.46*.

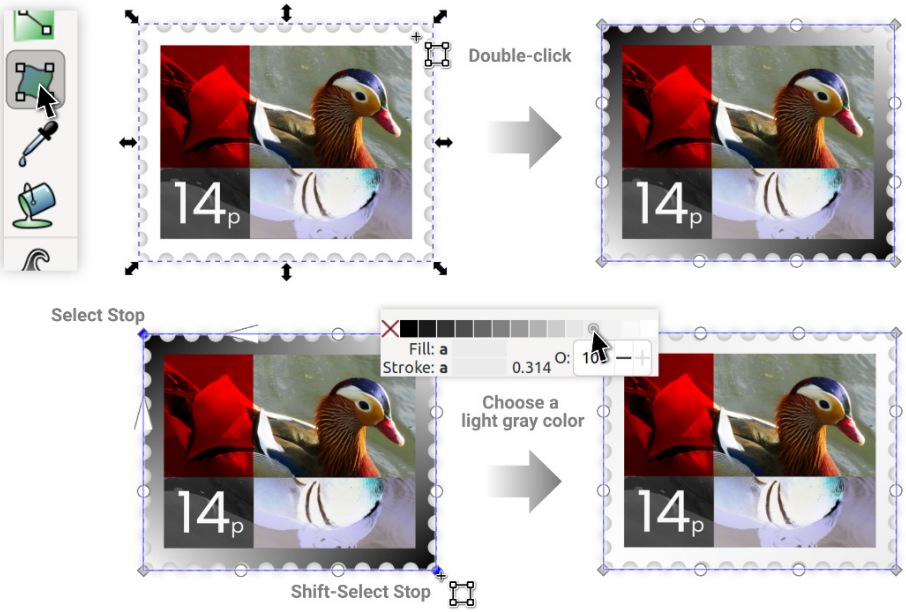
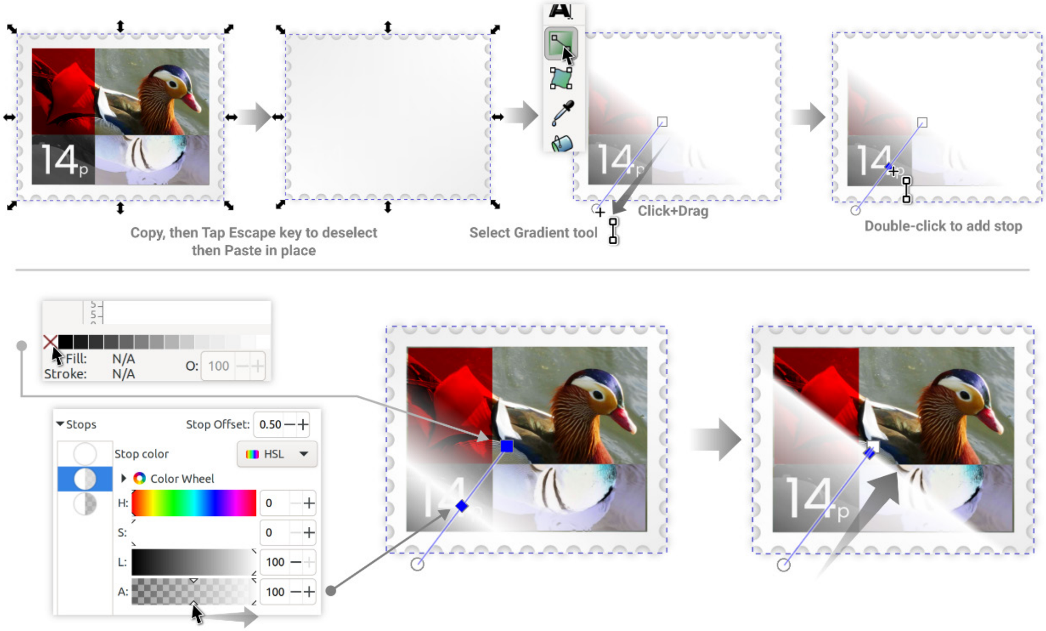


Figure 8.46 – Adding a mesh gradient to the white stamp for more visual interest

8. Lastly, we'll copy our white stamp backing again, tap the *Esc* key to deselect all, and then **Edit** > **Paste in Place** to paste a new copy of the backing on top of all our shapes. We can then use the **Gradient** tool set to the **Fill** mode and click and drag a new white-to-transparent gradient across the pasted shape from the middle to the bottom-left corner of the image.

9. This almost works as is, but let's modify it a bit by adding a stop by double-clicking the gradient line in the middle. Then change it to opaque white and the end stops of the gradient to transparent. Lastly, drag the middle stop up towards the end stop in the middle of the picture, as shown in *Figure 8.47*.

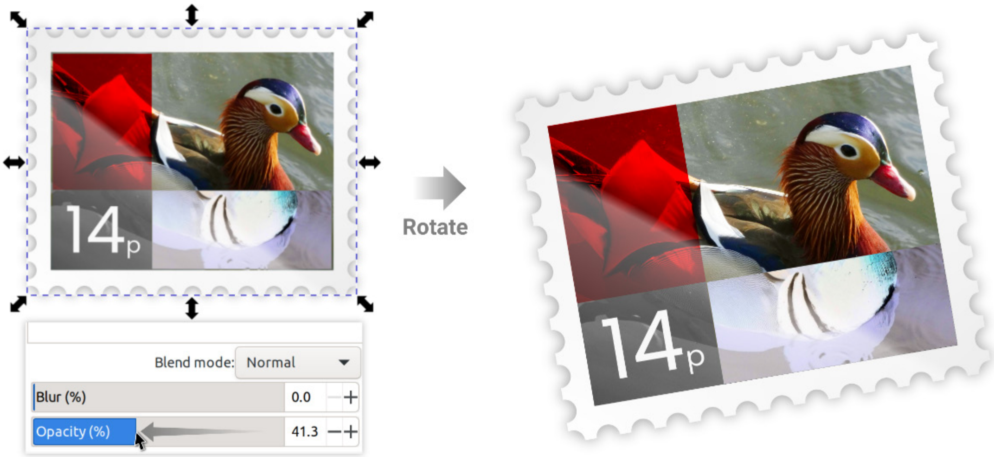


Fill: N/A
Stroke: N/A
O: 100

Stops
Stop Offset: 0.50
Stop color HSL
Color Wheel
H: 0
S: 0
L: 100
A: 100

Figure 8.47 – Adding the gloss effect with a linear gradient fill

10. The reflection gradient is still a bit much, so we can just use the **Opacity** slider bar for the object (at the bottom of the **Fill and Stroke** dialog) to make it more transparent. We can then select all our objects, group them together and then add some rotation for extra stamp-like visual interest, as in *Figure 8.48*. And that's it! We're done!



Blend mode: Normal

Blur (%) 0.0

Opacity (%) 41.3

Figure 8.48 – Adjusting stamp reflection opacity and rotating to complete our stamp

Summary

Wow! We learned a lot about using gradients, patterns, and blurs for various illustration tasks. All this great information really improves our graphic design game. In the next chapter, we will go even further with masks and clipping to achieve a variety of advanced effects using combinations of vector and bitmap image objects.

9

Clips and Masks

In the previous chapters, we've used Boolean operations to control where our shapes cut off – that is, we've cut them ourselves and deleted the bits we don't want. Sometimes, we want to temporarily hide part of our shapes without all the extra work. This is where clips and masks come in handy!

In this relatively short chapter, we will cover the following topics:

- Clips and clipping groups
- Clips inside clips
- Masking with vector shapes and images

Technical requirements

You can download the practice files for this chapter from GitHub at

`https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape/tree/main/Chapter09`

Clips and clipping groups

We got to play with clips a little in the last chapter when we used them to clip our blur, making a hard edge at the bottom of our shape to get rid of what looked like a shadow. Let's take another example. Say we want to subtract the background from an image. This is a common photo editing task that we can do right in Inkscape with vector precision!

As you can see in *Figure 9.1*, we can use our **Bezier Curve** tool to trace the periphery of this fetching mannequin and select both the shape and the picture, but instead of **Object** > **Clip** > **Set Clip**, as we did before, we're going to right-click on our selected objects and just choose **Set Clip** from the context menu that pops up. Easy, right?

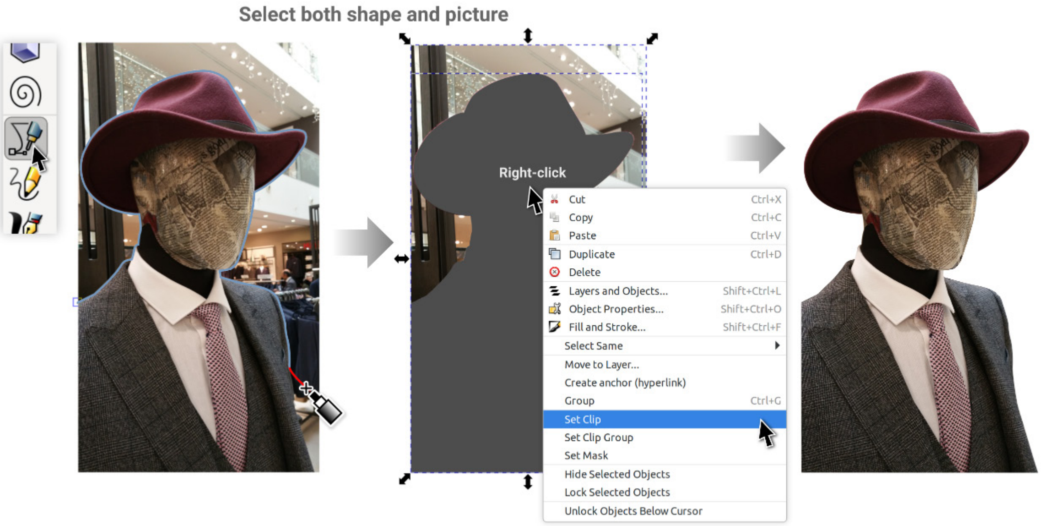

Select both shape and picture
Right-click
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Duplicate Ctrl+D
Delete
Layers and Objects... Shift+Ctrl+L
Object Properties... Shift+Ctrl+O
Fill and Stroke... Shift+Ctrl+F
Select Same
Move to Layer...
Create anchor (hyperlink)
Group Ctrl+G
Set Clip
Set Clip Group
Set Mask
Hide Selected Objects
Lock Selected Objects
Unlock Objects Below Cursor

Figure 9.1 – Using the context menu (also known as the right-click menu) to set a clip

Note that there are other clipping options in that context menu, but before we dive too deep, let's explore some neat tricks we can do with **Set Clip**. Note that when you have your clipped object selected and the **Node** tool is active, there is an extra option in the Tool control bar.

Yep, it's the one with the green line (it may be a different color, depending on your theme). That's the **Show Clipping Paths** toggle, and it does exactly that – it shows you the original path you used to clip the picture with. Even better, it gives you nodes in which to edit that shape, as shown in *Figure 9.2*.

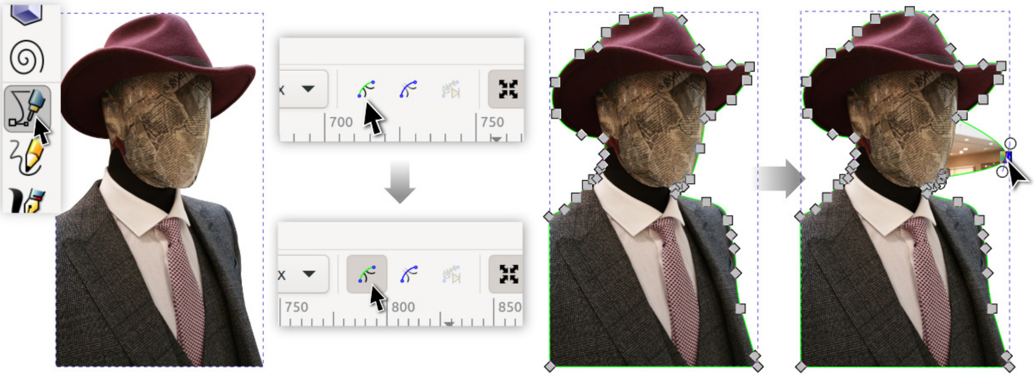


Figure 9.2 – Showing and editing the clipping path in the Tool control bar of the Node tool

This is pretty cool, but what if we want to edit the object *inside* the clip? Well, we could just right-click and choose **Release Clip**, and then edit our picture and use **Set Clip** again; however, this gets a bit tedious with more than one object, and it doesn't let us see the result until after we've clipped again.

There's a trick, however – Inkscape can apply clips to groups! So, if we select our picture and choose **Object** > **Group** (or press *Ctrl* + *G*), Inkscape will make it a one-object group. We can then select the clipping shape and the group and use **Set Clip**.

This will clip the group just like it clipped our original picture. Nothing much is different at this point, except that now we can double-click *into* our group and move and resize our image around inside the clip. Inkscape updates it in real time so that we can see the results right away, as shown in *Figure 9.3*.

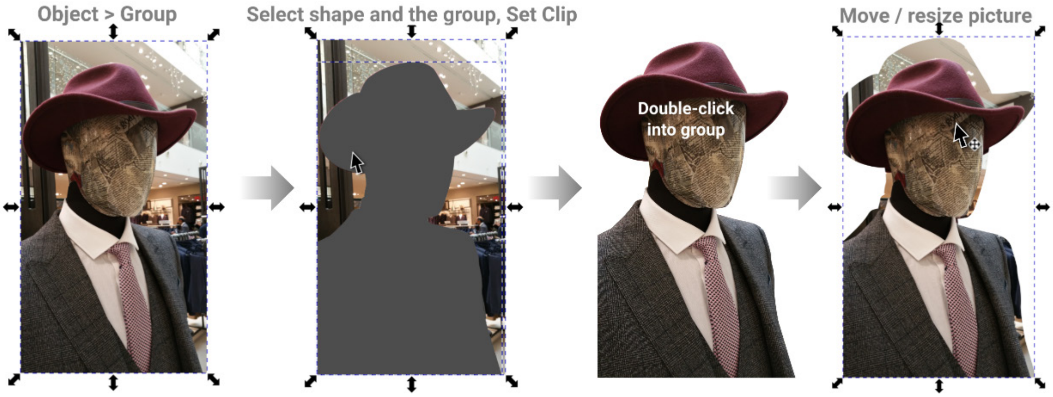
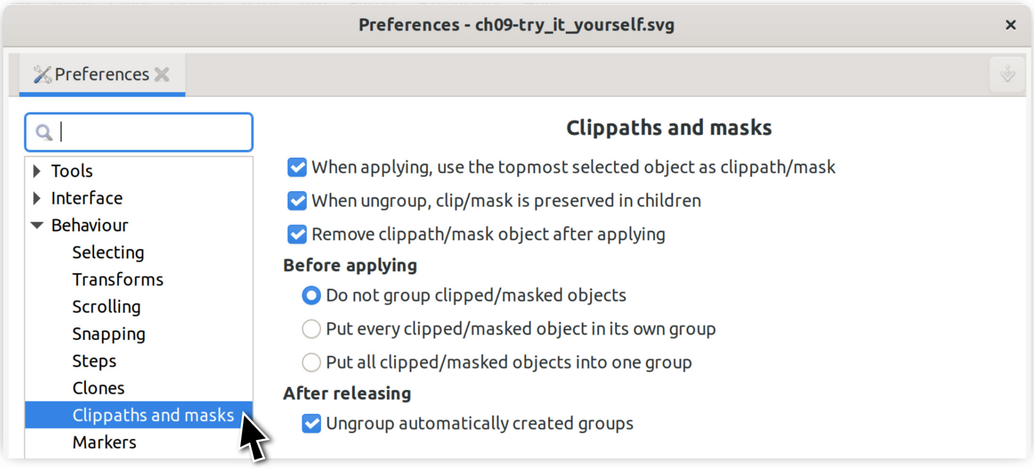


Figure 9.3 – Grouping the image to itself first, then clipping that group with the outline, and clicking into the image group to edit the shape inside the clip

While we are inside this clipped group, we can also add shapes as normal, and they will be inside the clip as well. We can exit the group by tapping the *Esc* key a few times, or by double-clicking outside the group on an area of blank canvas. When clipping more than one shape, Inkscape will take the top shape and use it to clip all the shapes you have selected individually.

This is not generally what you want though, as it creates many different shapes, each clipped with a copy of its own clip shape. Thus, I recommend grouping first before you use **Set Clip**.

It's just easier all around to work with the results if you do. In fact, you can choose **Edit** > **Preferences** > **Behavior** > **Clippaths and masks** and set the **Put all clipped/masked objects into one group** option to always make a group for your clipped objects each time you set a mask or clip, among some other useful options, as shown in *Figure 9.4*.



Clippaths and masks

☑ When applying, use the topmost selected object as clippath/mask

☑ When ungroup, clip/mask is preserved in children

☑ Remove clippath/mask object after applying

Before applying

● Do not group clipped/masked objects

○ Put every clipped/masked object in its own group

○ Put all clipped/masked objects into one group

After releasing

☑ Ungroup automatically created groups

Figure 9.4 – The Clippaths and masks section of Inkscape's preferences

These options are quite useful for changing what Inkscape does when you set a clip or mask:

- **When applying, use the topmost selected object as clippath/mask**: When we set a clip or mask, we've seen how Inkscape hides the original clipping shape to make it a clip or mask for underlying objects. If you want a visible copy of this shape hanging around instead, tick this box.
- **When ungroup, clip/mask is preserved in children**: Normally, if you ungroup a clipped group, the clip will be removed. This option lets you ungroup things without the objects being unclipped.
- **Remove clippath/mask object after applying**: When we release a clip or release a mask, Inkscape usually gives us back our original clipping shape. Maybe you don't want that, and if so, tick this box.
- **Before applying**: This is a series of options for what Inkscape does before setting the clip or mask. There are currently three options.
- **Do not group clipped/masked objects**: This is the default. Inkscape does *not* automatically create a group while clipping or masking.
  - **Put every clipped/masked object in its own group**
  - **Put all clipped/masked objects into one group**

Occasionally, we may want to keep the shapes we are using to clip the picture or objects instead of having them disappear when we use **Set Clip**. Inkscape has a function called **Set Clip Group** that does precisely that. Say, for example, I want to create a clip made of three circles.

Rather than clipping them to a different shape, I'd like to clip them to themselves so that I can add more circles and other shapes to the clip. Normally, we'd need to duplicate those three circles, group them together, and then use them as a clip shape over the original three (also grouped) circles.

Instead, we can just select our three circles, right-click, and choose **Set Clip Group**. As you can see in *Figure 9.5*, this lets us then click into the group and add more shapes (in this case, three smaller circles) inside the clip. Super convenient!

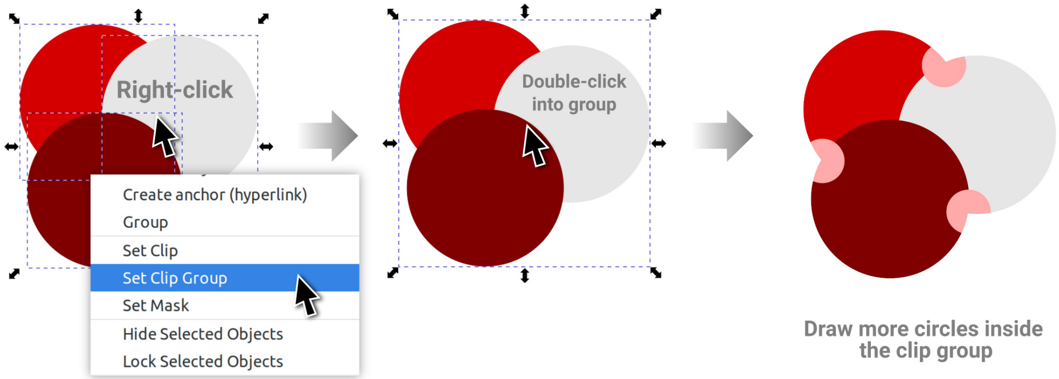


Figure 9.5 – Using Set Clip Group to make a clipping group out of selected shapes

...

Clips inside clips

At this point, you may be thinking, *well, this is pretty cool... but can I make a clip inside of a clip?* The answer is *yes*, you can definitely clip inside of another clip. But when you do, you'll need to make sure that the clipped object is *grouped to itself* first. If you don't do this, only the topmost clipped shape will be used for clipping/masking, and any clipping shapes inside will be erased.

Take, for example, a circle clipped with a star, and then clipped with a rectangle. In *Figure 9.6*, you can see that if we do not group the newly clipped star circle, Inkscape will completely toss away the star clip when we try to clip it to the rectangle. Grouping the star circle to itself after clipping fixes this issue, giving us the result we want.

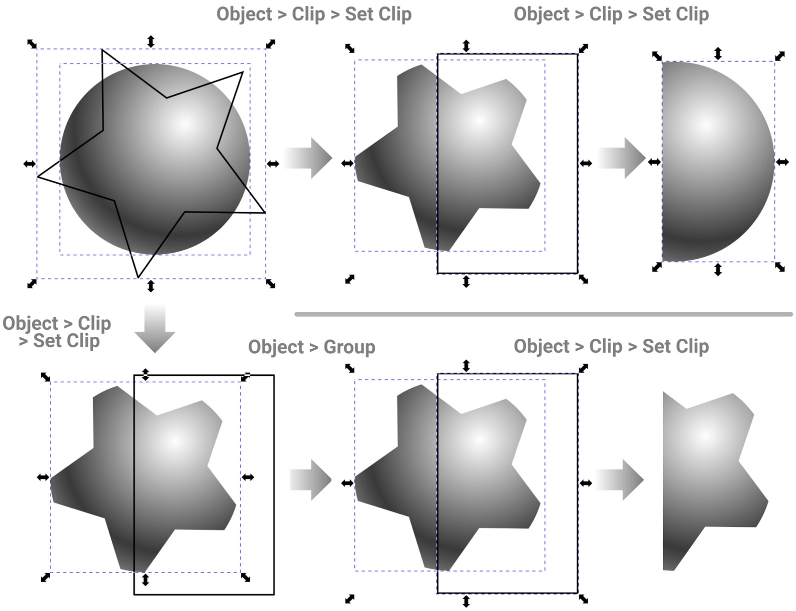


Figure 9.6 – The results of clipping an already clipped shape without grouping (top) and with grouping (bottom)

Of course, if you've told Inkscape to automatically add your group to what's clipped via the preferences we talked about previously, you don't need to perform this extra grouping action yourself. Inkscape will automatically group the clipped shape when you clip again.

That's pretty much the entire story for clips in Inkscape. Basically, whatever vector shape you use, filled or unfilled, will simply clip the other shapes to the geometric boundary of that shape. However, maybe we'd like to do something a bit fancier and use a color gradient, or even a picture to control the transparency of objects rather than just clip them. This brings us to the next topic in this chapter – masks!

Masking with vector shapes and images

Vector shapes with **Set Clip** allow you to set the boundary cut-off of shapes. Masks let you control the transparency of those same shapes in a much more flexible way. Say, for example, we have an illustration of a fish, and we'd like to make it half… I don't know… squirrel! Sure, why not?

We'd like to gradually fade out the squirrel half into the fish half. That would normally take a lot of complex gradients, but fortunately, we can use a simple black-to-white gradient shape as a mask to fade the squirrel into the fish, as shown in *Figure 9.7*.

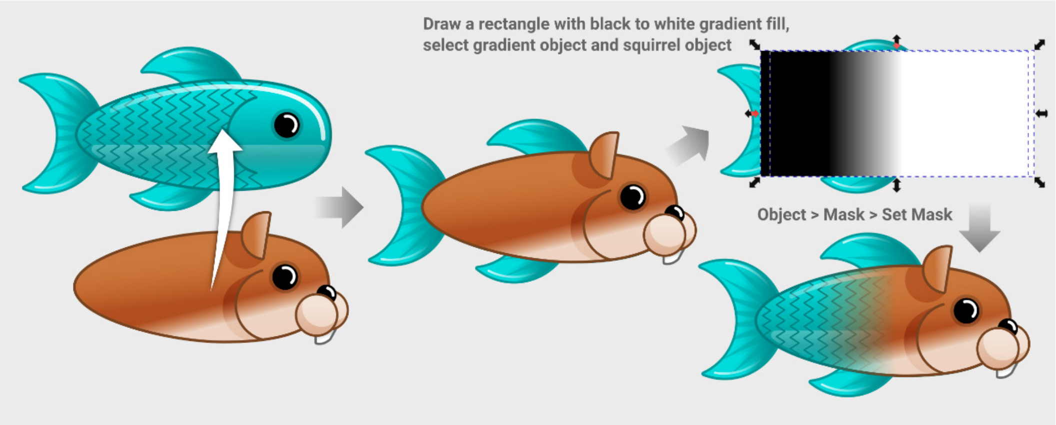

Draw a rectangle with black to white gradient fill,
select gradient object and squirrel object
Object > Mask > Set Mask

Figure 9.7 – Fading squirrel to fish with a black-to-white gradient set to mask

Note that where the gradient is white, the squirrel head is more opaque and the darker the gradient gets, the more transparency we get. This is the essence of how masks work in Inkscape and many other programs. We can, of course, use radial and even mesh gradients to control the color of our masks and, thus, the areas of transparency.

We are not limited just by gradients. We can, in fact, use *any* object with light and dark colors, including patterns or even images, to do our masking. *Figure 9.8* shows some different results we can achieve by using different objects as masks.

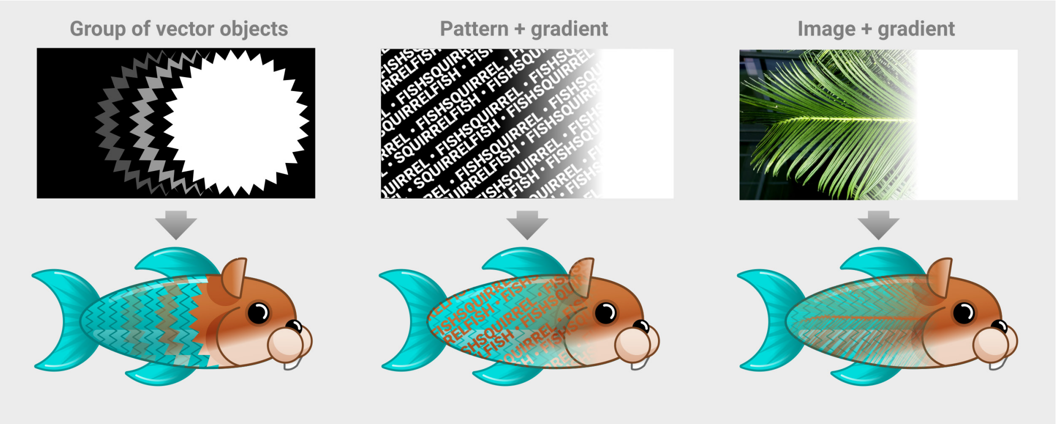


Figure 9.8 – Using different grouped objects as a mask over our fish-squirrel

Pretty neat, eh? You can see how much variation and detail we can get by mixing something like that leaf image into our mask while preserving our vector shapes. Doing such things by hand with vector shapes would take far too long. Thus, masking gives us a great alternative to get some really great results with minimal work.

It's worth noting that Inkscape treats transparent areas as if they were a dark color in the mask – that is, the transparent parts in the mask will be the transparent parts in the result. Therefore, you can simply use a white-to-transparent gradient mask and get the same results as if you were going to use a white-to-black gradient mask.

This is handy especially when we want to blur-mask something. We can simply make a white object, blur it, and use it as a mask, as shown in *Figure 9.9*.

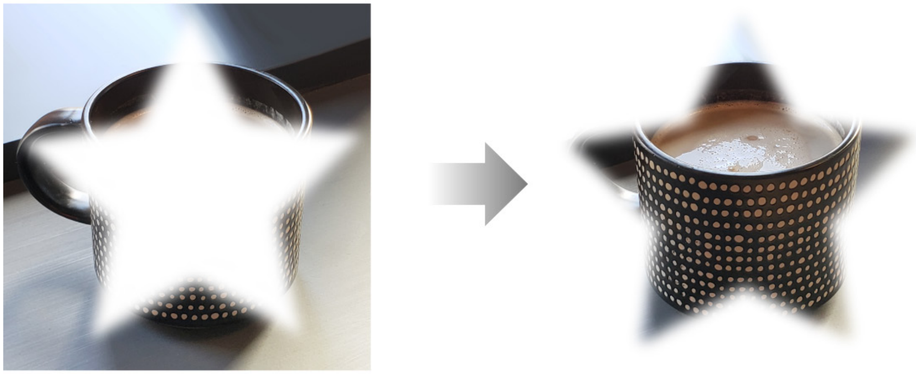


Figure 9.9 – Using a blurred white star shape to achieve our masking result

Now that we know the basics of clipping and masking, let's try our hand at making something with our newfound knowledge.

Try it yourself

For this exercise, we'll be taking a photo and making a travel advertisement poster for a particular destination (London, for example). It doesn't have to be the one I picked. Feel free to choose your own destination, either a place you've been to or would like to go. I've used this excellent photo from the free stock photo website `https://www.pexels.com/` and have included it in the book materials for our practice.

I found this by typing `London` into the search engine on the Pexels website. Basically, we're after any image that has foreground and background elements we can separate. We're going to be using some text and gradient masks and vector clips to get the result shown in *Figure 9.10*:

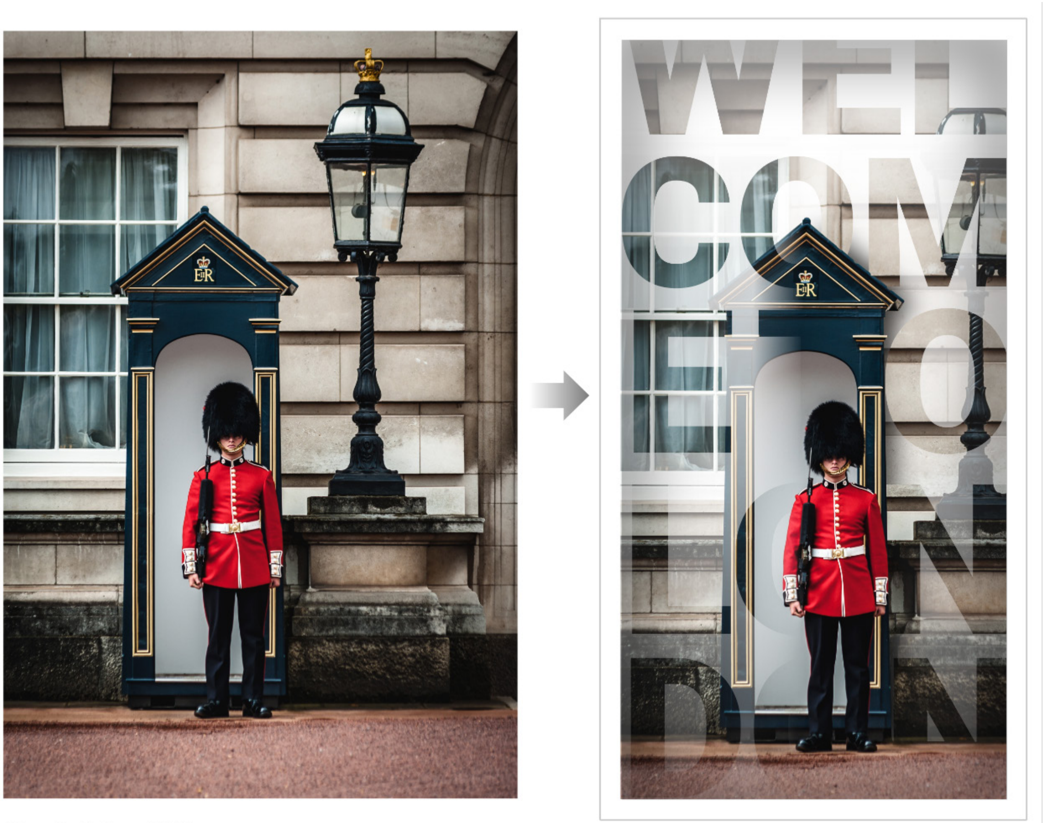


Photo Credit: Samuel Wölfl
Source: pexels.com/photo/royal-guard-standing-near-lamp-post-1427581/

Figure 9.10 – The base photo and our end result

So, let's get started! We want to separate the foreground part from the background – in this case, the guard and the ground from the booth behind hi:.

1. To do that, we trace around the ground, booth, and guard with the **Bezier Curve** tool, keeping our line just a little inside the parts we want to clip so that there's no slight halo of the background when we select **Set Clip**. This tracing is shown in *Figure 9.11*.

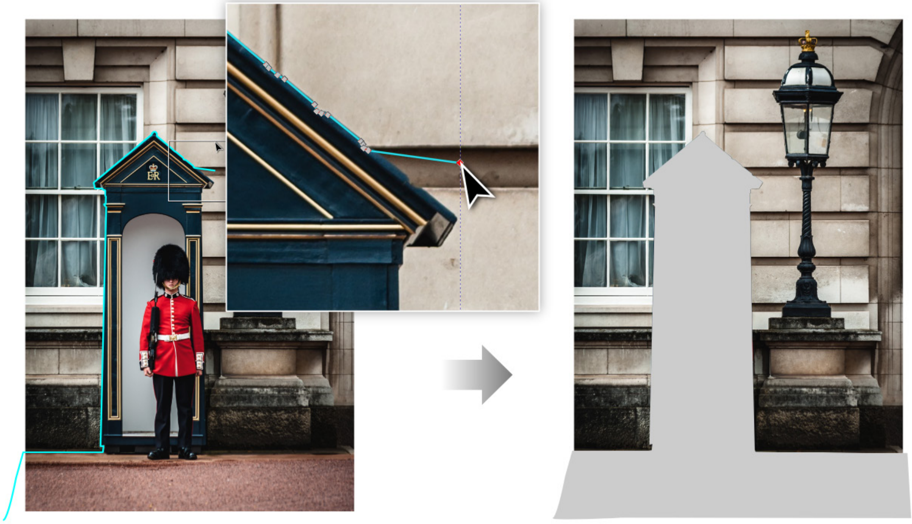


Figure 9.11 – Drawing our clipping shape

Once we have our mask drawn, we'll copy the image and paste it in place.

2. Select both the clip shape and the pasted image and then select **Object** > **Clip** > **Set Clip**. This will give us a separate ground, booth, and guard and a copy of our original behind, as shown in *Figure 9.12*.

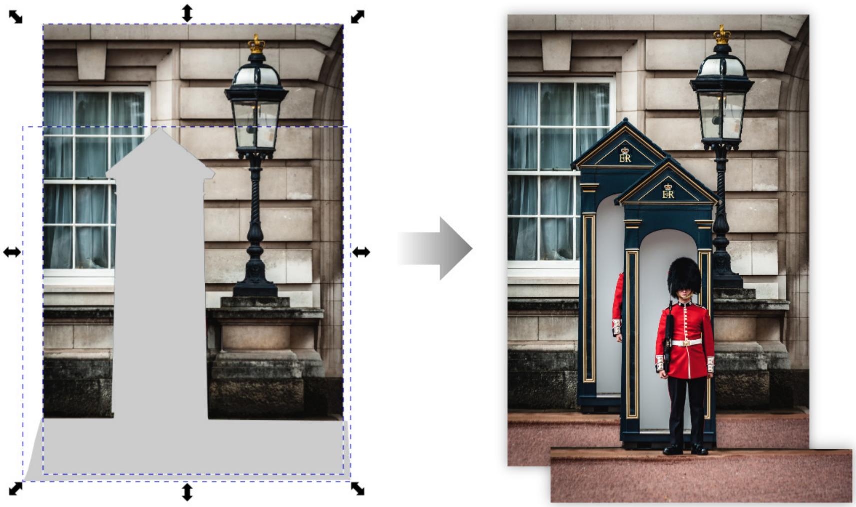


Figure 9.12 – Setting the clip on our copy/pasted image to give us a separate ground, guard, and booth

Now, we can make big blocky bits of text. I'm using the **Roboto** font set to **Heavy** rather than just **Bold**. Once the text is typed out, I like to separate the letters so they are easier to resize.

3. Recall that we can do that by first choosing **Path** > **Object to Path** and ungrouping a few times by pressing *Ctrl* + *Shift* + *G*. Arrange the letters in three-letter chunks and resize them to fill in the space of our background. Make a rectangle with a black-to-white gradient and place the text over it, turning it white, as shown in *Figure 9.13*.

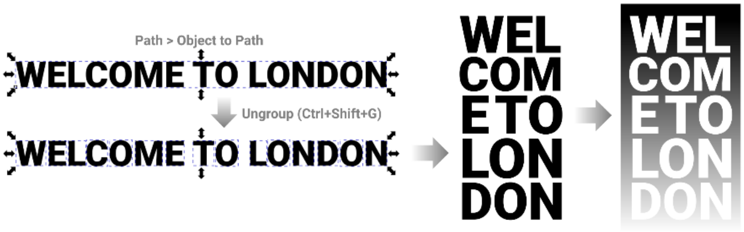


Figure 9.13 – Making our background mask

4. Before we can use our new mask, we need to group the letters and gradient-filled rectangle together. Then, move this group between the background and our cutout portion of the image (foreground). Then, we select the mask group and the background and select **Object** > **Mask** > **Set Mask**. This will give us our masked shape, as shown in *Figure 9.14*.

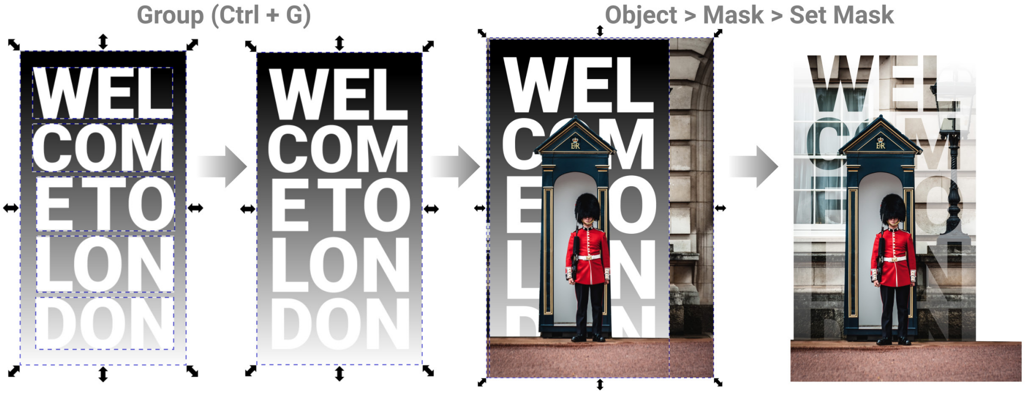


Figure 9.14 – Grouping the mask parts together and then using them to mask the background

Looks pretty good! However, it's a bit hard to read the lower part of **LON DON**, as the booth is obscuring too much of it.

5. Fix this by releasing the clip on our ground, booth, and guard, and applying a gradient to the clip shape to show through more of the background text when we select **Set Mask** instead. We can clip off all the extra bits by clipping it all into a rectangle. *Figure 9.15* shows this process.

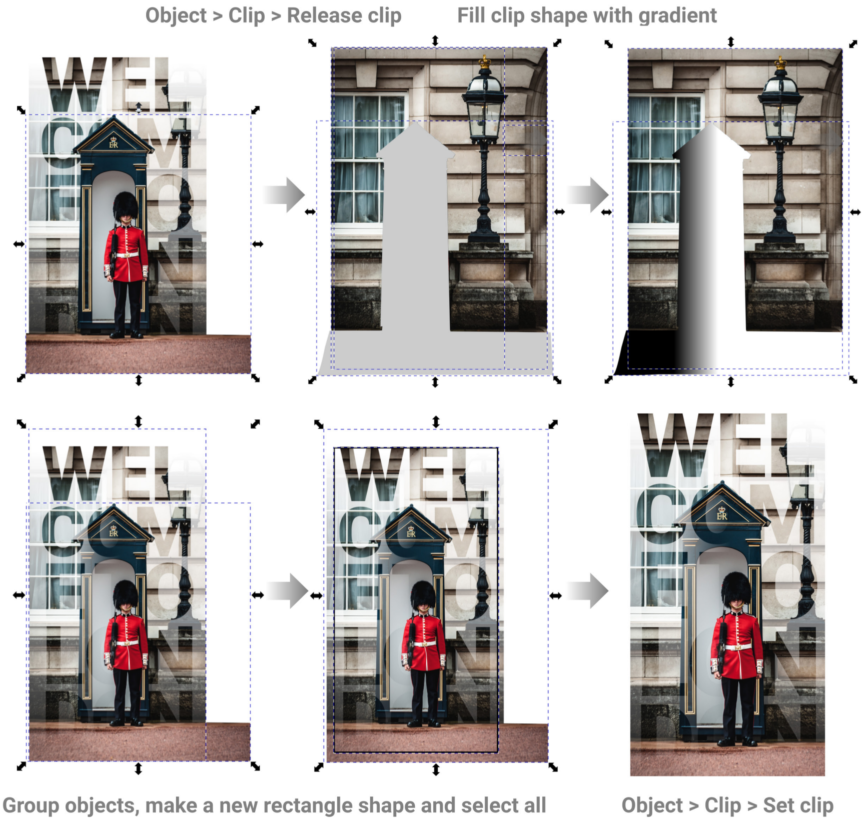

Object > Clip > Release clip
Fill clip shape with gradient
Group objects, make a new rectangle shape and select all
Object > Clip > Set clip

Figure 9.15 – Changing our ground booth and guard clipping shape to a gradient-filled mask and then clipping to a rectangle

We'd like to make our booth pop out just a little more by adding a fake shadow behind it. This will complete our optical illusion and create that last bit of visual tension that will grab eyeballs for our advertisement.

6. To do this, we will just make a black shape that is generally the same as the booth, give it some blur, and send it to the very back, behind all of our masked shapes, as shown in *Figure 9.16*.

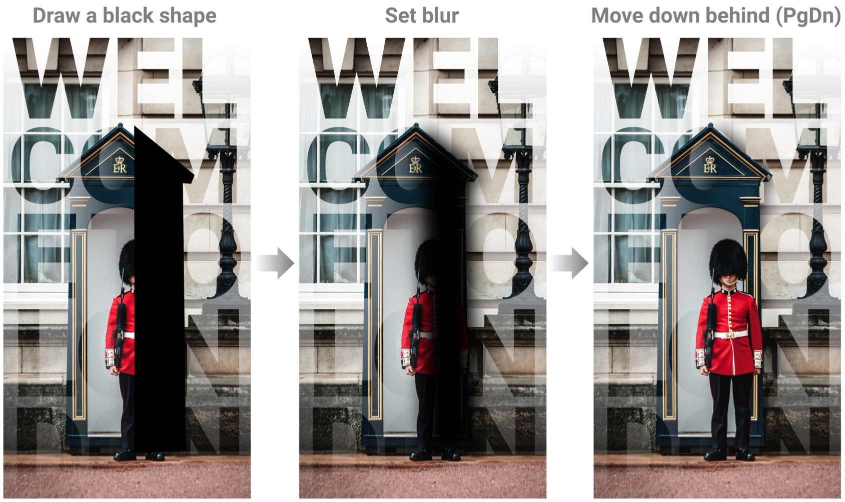


Figure 9.16 – Adding a fake shadow behind our masked objects

As a last step, we will add a blurred border to make the edges of the photo darker (called a **vignette** in graphic design and photography). We'll use a clip group, as we saw earlier in this chapter, for this to reduce the number of steps.

7. Just draw a rectangle around the group, set the fill to none, and the line thickness to something thicker than usual. We can then add some blur to create a shadow effect, and then right-click and choose **Set Clip Group**. This will clip the shadow to the interior of the shape, play with the opacity a bit, and complete our vignette, as shown in *Figure 9.17*.

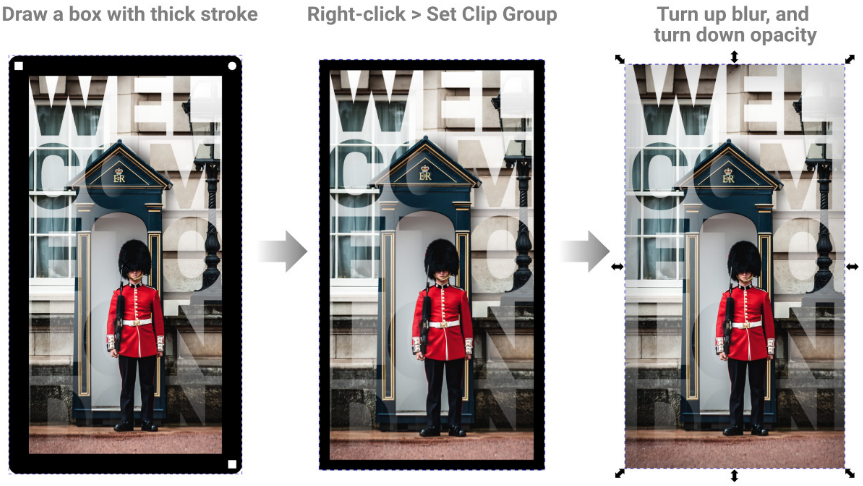
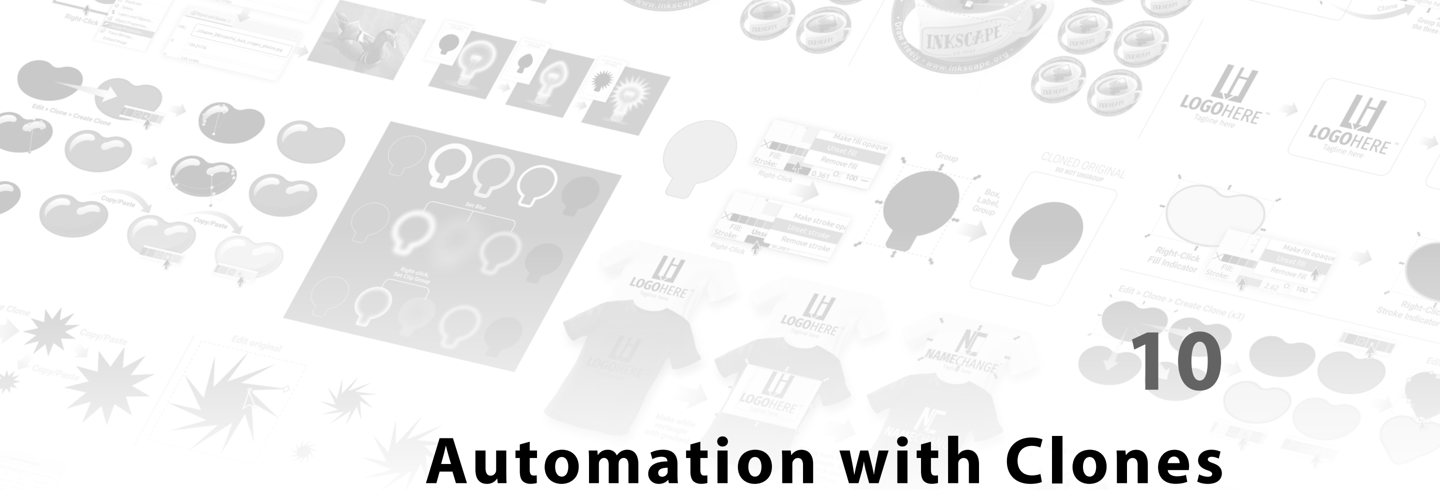

Draw a box with thick stroke
Right-click > Set Clip Group
Turn up blur, and turn down opacity

Figure 9.17 – Adding a shadow border (vignette) to the image

Summary

How cool is that? We've done a lot in this chapter to improve our skill set. We learned about masking and clipping, and a variety of ways in which we can use both to achieve some visually dazzling results without much work.

In the next chapter, we'll have a look at a few other neat features of Inkscape to save us all sorts of time – using clones and linked objects!



10

Automation with Clones and Linked Files

You've finished the logo! Well, you thought you had anyway...but after meticulously pouring over every detail of the logo or illustration and accompanying swag mock-ups (that is, t-shirts, mugs, business cards, and so on), you get that dreaded last-minute *Can you just change one thing in the logo for us?* email from the client. You could go back and re-do all those carefully designed mock-ups or copy/paste in the new logo revisions, or you can hop in my patented *Should Have Done it Differently* time machine and make all your mock-ups with clones instead!

That would allow us to change just the original logo and have the changes appear automagically everywhere we used clones, but what are clones really, and how do we use them?

In this chapter, we will cover the following topics:

- What is a clone?
- Cloning groups
- Clones as clips and masks
- Changing **Clone Fill** and **Stroke Colors**
- Clone management using clones of clones
- Leveraging linked files

Technical requirements

You can download the practice files for this chapter from GitHub at

`https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape/tree/main/Chapter10`

What is a clone?

Put simply; a **clone** is a copy of an object that will change if the original clone's geometry (that is, nodes, curves, fills, strokes, and so on) is changed. Clones are scalable, rotatable, and movable but are *not* editable in the same way as the original, taking all their geometry and most styling from the original object.

As an example, let's make a star shape and then clone it by selecting **Edit** > **Clone** > **Create Clone**. This will give us a new star exactly the same as the first star, and the Status Bar tells us this new copy is a clone of the object. Now when we edit the original to give it more points, the clone automatically changes to match, as shown in *Figure 10.1*.

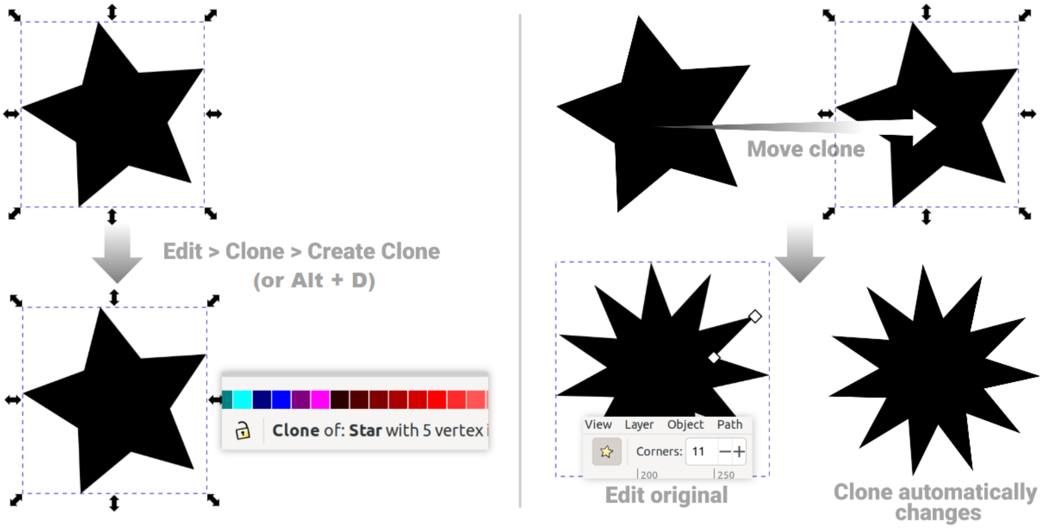

Edit > Clone > Create Clone
(or Alt + D)
Clone of: Star with 5 vertex i
Move clone
View Layer Object Path
Corners: 11
200
250
Edit original
Clone automatically changes

Figure 10.1 – Creating a clone and changing the original geometry

So, what if we want more than one clone? We can create a new clone by selecting **Edit** > **Clone** > **Create Clone** for each new clone, but we can also just copy/paste or use the spacebar to stamp/drop copies of the clone we already made. Since these copies are all clones of the original, they will all be affected by editing the original.

We can even rotate and scale the clone copies, and they will still be modified when the original geometry is edited, as shown in *Figure 10.2*.

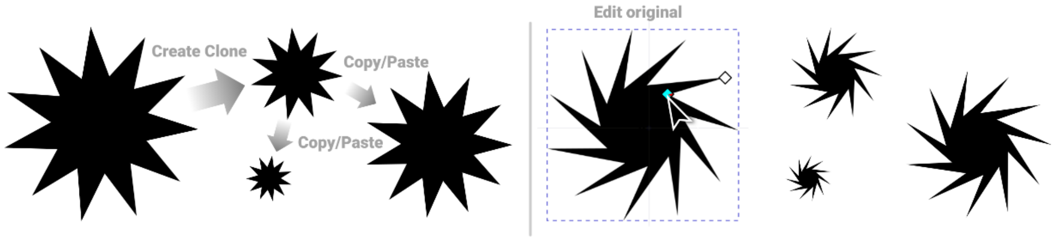


Figure 10.2 – Copying and resizing the clone to make more clones of different sizes

Note that when we move or resize the original object, all the clones also move and resize. This can pose a problem if we want to move the original off the page for easy location and *safekeeping*. We can do this by encasing the original in a group (grouping the original to itself), then moving the group.

This allows the original to keep its location and transformations relative to the group while using the group to relocate it elsewhere. We can then double-click into the group and edit the original with the same effect as before. *Figure 10.3* shows the difference between scaling an original versus moving a grouped original on the clones.

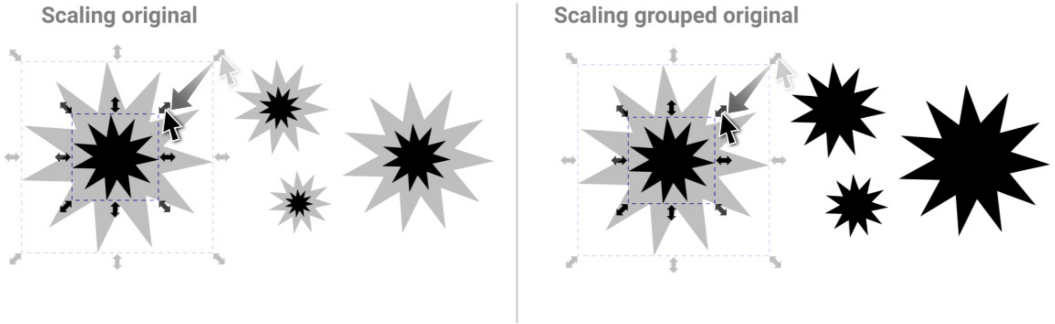


Figure 10.3 – Scaling the original scales all clones, whereas scaling a grouped original does not affect the clones of the original

Note that this only works because the clones are of the *original* and not of the grouped original. If you create clones of the original after you group it to itself, then moving the group will still move the clones of the group as well.

Therefore, you should make your clones before grouping, and if you want to make more clones of the original, you will need to click into the group, select the original, and then select **Edit** > **Clones** > **Create Clone** of the original. Then cut/paste the new clone outside the group. Since it's the group that is cloned, ungrouping it will break the link to all the clones, making them regular objects instead.

Or you could just continue to make copies of the clones you made before you grouped the original to itself. If this seems confusing, give it time. Through working with clones, you will quickly understand how they work and be able to use all these little grouping tricks to get your desired results.

Of course, many things we might want to clone will have *multiple objects* we will want to edit after the fact and have those changes reflected in all our clones. Let's move on to using groups to act as the original – let's clone some groups!

Cloning groups

Let's take, for example, a sheet of stickers. We could just copy/paste it into a bunch of new stickers and arrange them evenly throughout the page; however, we don't want to re-do all this arrangement just to edit the sticker. Thus, we can use clones to do the work for us.

Our stickers are probably made of more than one object, however. Fortunately, Inkscape supports cloning groups, so all we need to do is group all of our sticker parts together, then clone that group to clone the whole sticker and all its parts.

Figure 10.4 shows a sticker I've made of many different vector parts, which are grouped, cloned, then arranged on a sheet.

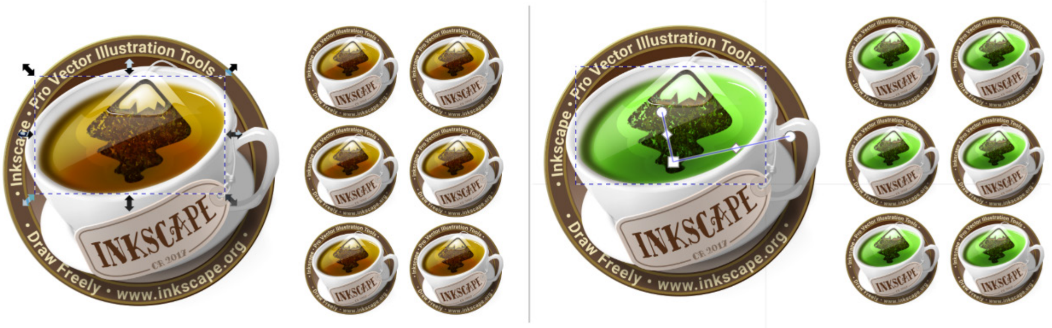


Figure 10.4 – Changing part of the grouped original changes all the clones of the group

As you can see, clicking into the group allows us to edit parts of the original sticker, such as the gradient I used for the tea color. You can see how cloning a group can be quite a time saver!

Over the years, I've used this trick to make hundreds of sticker sheets for commercial product packages. But that's not all clones can do. Let's have a look at some other things they can do, such as functioning as clips and masks.

Clones as clips and masks

Using clones as clips and masks can give us a great deal of control over how our clip shapes look. I use this a lot to make mock-ups of, say, a t-shirt with a logo. We'd like the same logo on both shirts, but one is black, requiring a white logo.

Thus, we can use a clone of the original black logo as a clip shape to clip a white rectangle with some subtle gradients that match the shading of the black shirt. With our clone acting as a clip shape, changing the original will also change the clip shape on the black shirt, saving us from having to re-do all the duplication and clipping work, as shown in *Figure 10.5*.

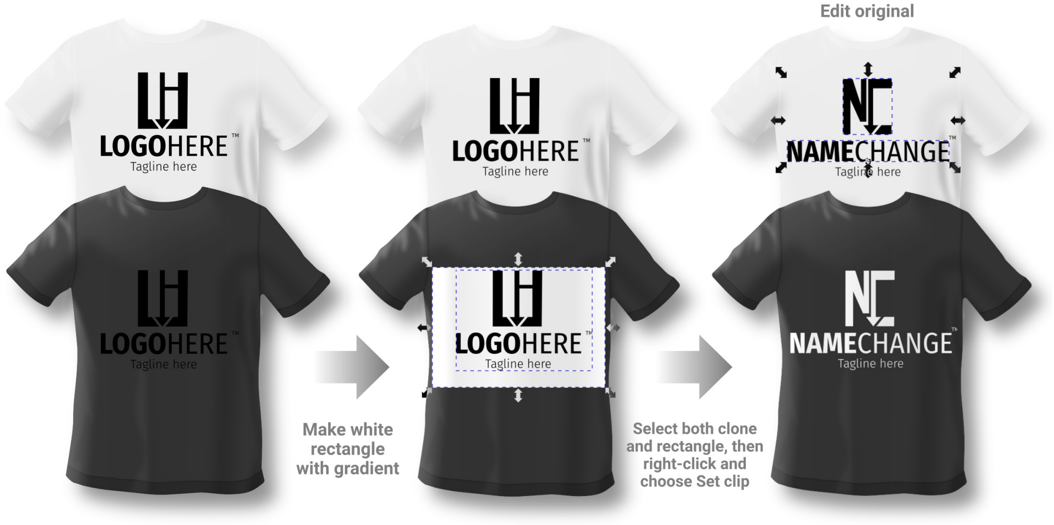

Edit original
LOGOHERE™
Tagline here
NAMECHANGE™
Make white rectangle with gradient
Select both clone and rectangle, then right-click and choose Set clip

Figure 10.5 – Using a cloned logo as a clipping mask for the t-shirt mock-ups

We can also apply blurs and opacity to our clones and layer them to produce interesting effects, such as shadows and glows, which will change according to edits made to the original shape. Sometimes though, we'd like to change the fill and stroke properties of the clones.

Normally, this isn't possible because our original has a fill and stroke already assigned to the shapes. However, if we unset the original fill and stroke, we can then change the fill and stroke of a selected clone to whatever we wish.

Figure 10.6 shows the process of unsetting the fill and stroke by right-clicking the **Fill and Stroke** color indicators and choosing **Unset fill** or **Unset stroke** in each case. The jelly bean will turn black, and we can then clone, color, and edit the original to change the shape of all the jelly beans at the same time.

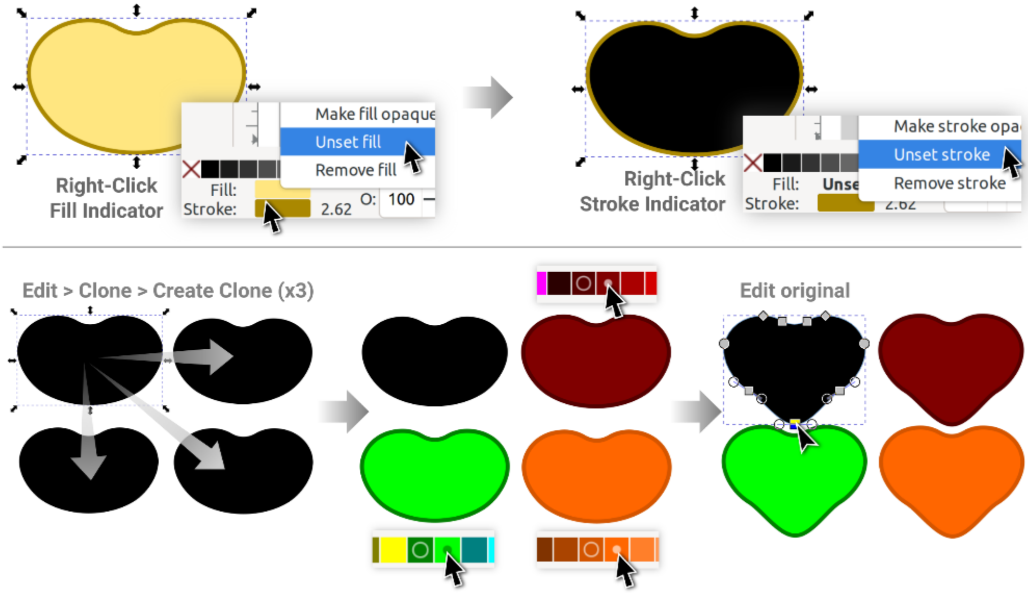


Figure 10.6 – Unsetting fill and stroke to enable us to set the colors on the clones

This is quite useful but maybe a bit boring as an illustration. After all, jelly beans are glossy and have shadows and such. Fortunately, we're not limited to cloning just one object. We can make the glossy reflection and shadow, group them, and clone them as we did with a single object and change the color of the beans.

Moreover, we can set the color of the reflection to a white gradient and the shadow to a black gradient, and changing the bean color will not affect those parts of our cloned group. Let's take the bean shape we've unset the fill and stroke on, clone it, then use the clones to make reflections and shadows.

We can first group our unset bean base to itself and make a clone of the group, turning the clone red as before. This will help us see the effect of what we are doing over the black bean a bit better.

We can then add three shapes to the original unset black bean – two gloss reflections and one shadow shape at the bottom with white and black-to-transparent gradients. We can then copy/paste the cloned red bean and change the colors as before. This process is shown in *Figure 10.7*.

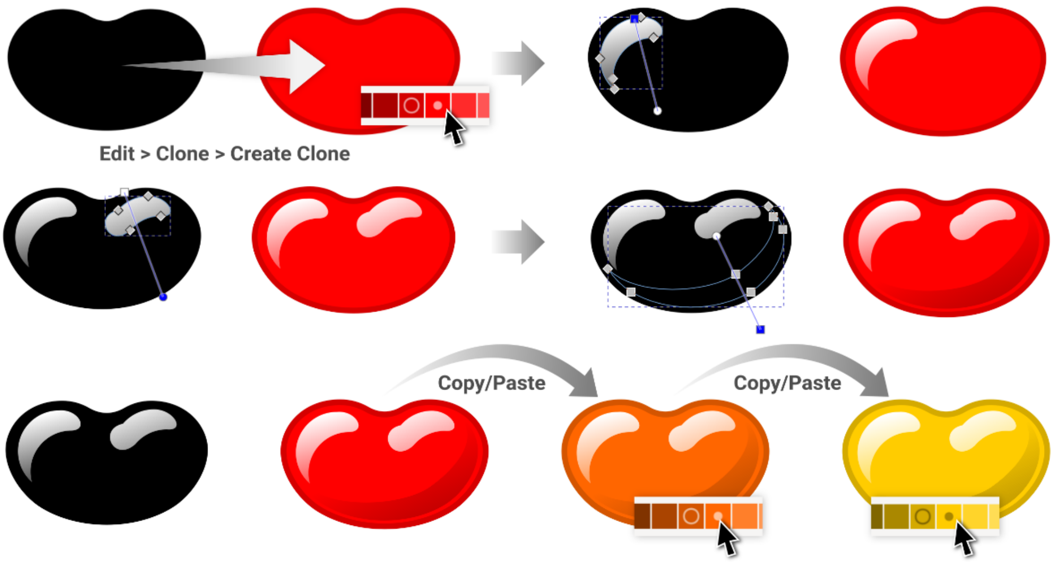


Figure 10.7 – Adding colored shapes to our unset fill and stroked bean base

Notice that when we change the color, the white and black gradients on the new clones stay the same color. That's because only unset fills and strokes are affected by changing the color of a clone. This effect is called **inheritance**. It's named that because the unset shapes in the clone *inherit* the colors you set on the clone.

This is pretty handy, but sometimes we'd like to be able to combine clones and make clones of those shapes.

Clone management using clones of clones

Once we get into more complex combinations of clones, we will want a good way to manage them. After all, if we happen to ungroup or delete our original cloned object accidentally, Inkscape will silently unlink our clones, and we will lose the ability to change them all at the same time.

This is why we will generally group the original, put a rectangle behind that with a label, then move it off the canvas, so it's out of the way of the main design. *Figure 10.8* shows what this looks like.

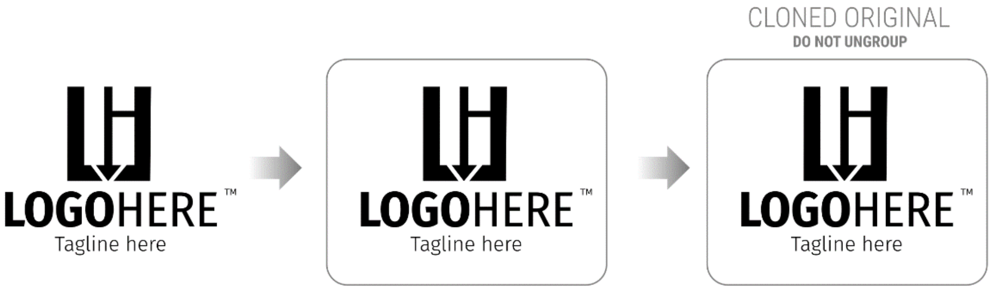


Figure 10.8 – Drawing a rectangle, labeling and grouping the original clone to move it off canvas

As you can see, the warning instructions and box show at a glance what's going on, so there's never any question as to what needs to be done (or not done, in this case).

This is even more useful when we start making clones of clones. Take, for example, our logo symbol, logo name, and tagline; we will not always want them to be arranged like this and may want to exclude parts of them depending on what we are designing. No matter how we have them arranged, we'd like one change to affect all elements in our design. So how do we do this? The answer is with clones of clones!

We start by making each part of our logo a separate group, labeling and boxing them, and moving them off the canvas, as we did before. We can then click into each box group, make a clone of each part of the original, and combine those clones to make two new arrangements: a horizontal and vertical version of our logo, as shown in *Figure 10.9*.

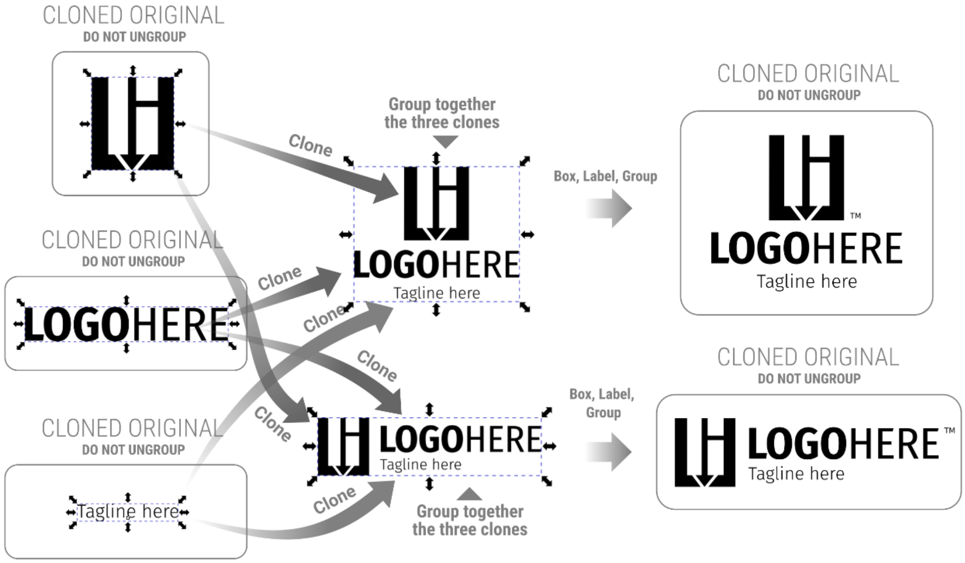


Figure 10.9 – Making two different arrangements of the original logo parts and labeling and boxing them for use as clones

Note that we have added a trademark (™) symbol to each of our arrangements in different locations. If we had added them to the original cloned objects, we would have been stuck with their placement in the clone copies. Thus, it's easy to combine clone and non-clone shapes in this way and retain editability over all the subsequent combinations by editing our original clones.

If we now edit the original logo, the changes will propagate to both the horizontal and vertical arrangements automatically. These changes will then automatically propagate to clones we make of the horizontal and vertical logo orientations as well, allowing us to use these two arrangements throughout our project without having to reassemble them each and every time we make a change to the original logo symbol, logotype, or tagline.

Figure 10.10 shows all of our off-canvas cloned groups and different ways we might use them in different places of our documents.

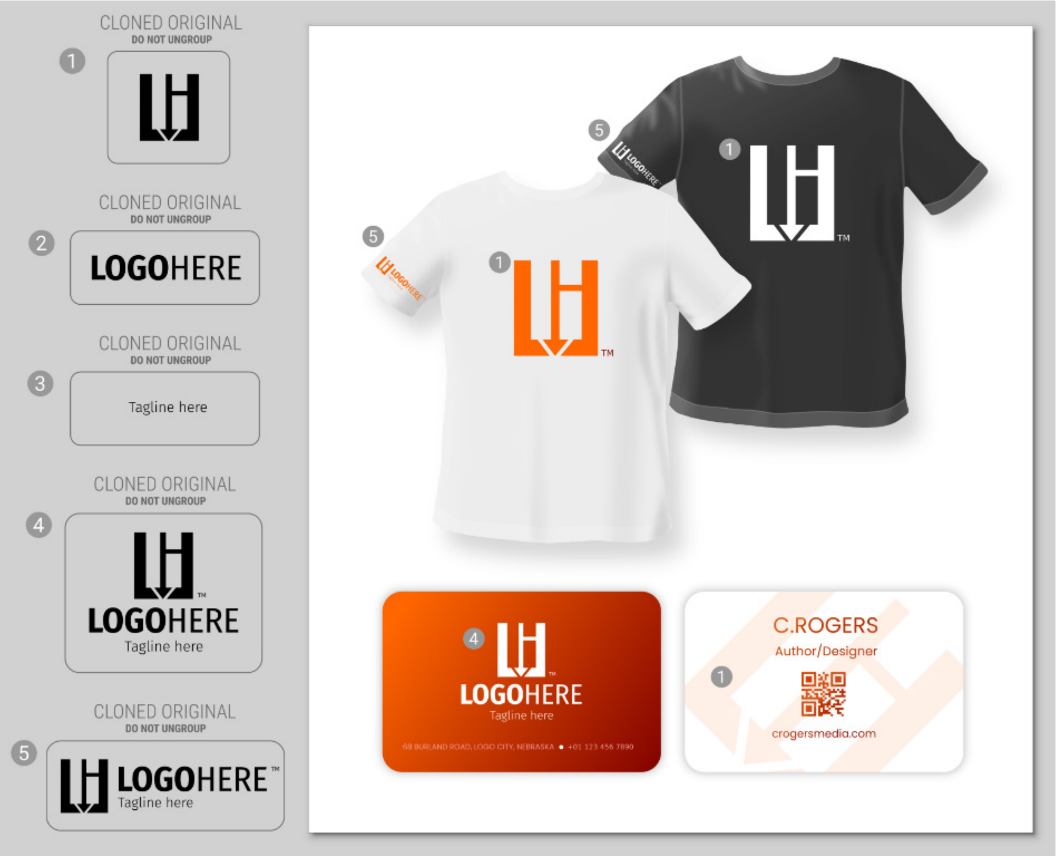


Figure 10.10 – Off-canvas cloned groups, numbered to show where we've used them in these mock-ups

This may seem like a lot of work for only a little gain on just this one project. Fortunately, since we grouped our objects before cloning, we can reuse this same file to automatically create mock-ups for any number of future projects.

We can do this just by copying/pasting the new logo into that original group and deleting the old logo. Obviously, you'll want to tweak the designs a bit to suit each new client's logo and branding, but all the cloning and duplication work is already done.

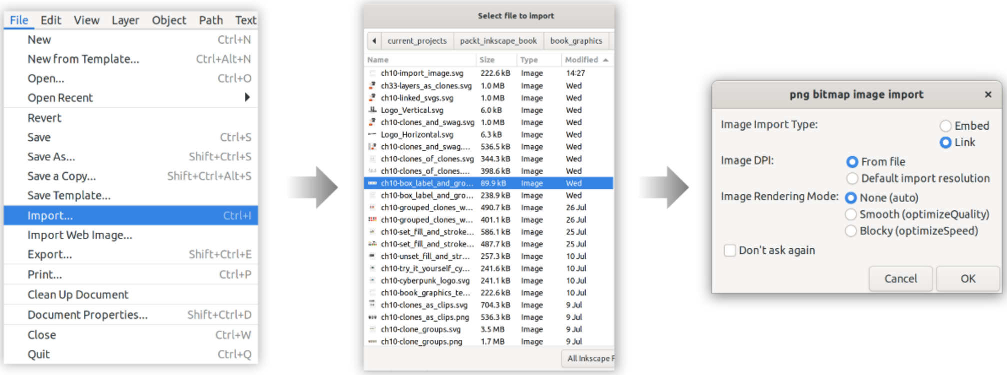
I use this technique for each new client because it saves lots of re-doing the same redundant work over and over. This allows me more time to concentrate on the design work.

This easy-to-use automatic system for making mock-ups is what I like to call a **supertemplate**. Indeed, when we save the file, we will include `supertemplate` in the filename, so I can then find it later, make a copy, and use it for an entirely new project of the same type. We will name this one `teeshirt_and_business_cards_supertemplate.svg`, for example. This will make it super easy to search for using a file browser!

Before we close out this chapter, there's one more useful way to reuse assets in our documents. We can link images into it, and any changes made to the originals will then be reflected in our document!

Leveraging linked image files

When we drag and drop an image file, say a photo, for example, from our filesystem into Inkscape (or choose **File** > **Import**), we're presented with several options for placing that file in our document. *Figure 10.11* shows the importing process and options.



Name	Size	Type	Modified
ch10-import_image.svg	222.6 kB	Image	14:27
ch33-layers_as_clones.svg	1.0 MB	Image	Wed
ch10-linked_svgs.svg	1.0 MB	Image	Wed
Logo_Vertical.svg	6.0 kB	Image	Wed
ch10-clones_and_swag.svg	1.0 MB	Image	Wed
Logo_Horizontal.svg	6.3 kB	Image	Wed
ch10-clones_and_swag....	536.5 kB	Image	Wed
ch10-clones_of_clones.svg	344.3 kB	Image	Wed
ch10-clones_of_clones....	398.6 kB	Image	Wed
ch10-box_label_and_gro...	89.9 kB	Image	Wed
ch10-box_label_and_gro...	238.9 kB	Image	Wed
ch10-grouped_clones_w...	490.7 kB	Image	26 Jul
ch10-grouped_clones_w...	401.1 kB	Image	26 Jul
ch10-set_fill_and_stroke...	586.1 kB	Image	25 Jul
ch10-set_fill_and_stroke...	487.7 kB	Image	25 Jul
ch10-unset_fill_and_str...	257.3 kB	Image	10 Jul
ch10-try_it_yourself_cy...	241.6 kB	Image	10 Jul
ch10-cyberpunk_logo.svg	241.1 kB	Image	10 Jul
ch10-book_graphics_te...	222.6 kB	Image	10 Jul
ch10-clones_as_clips.svg	704.3 kB	Image	9 Jul
ch10-clones_as_clips.png	536.3 kB	Image	9 Jul
ch10-clone_groups.svg	3.5 MB	Image	9 Jul
ch10-clone_groups.png	1.7 MB	Image	9 Jul

Image Import Type: ○ Embed ● Link
Image DPI: ● From file ○ Default import resolution
Image Rendering Mode: ● None (auto) ○ Smooth (optimizeQuality) ○ Blocky (optimizeSpeed)
□ Don't ask again

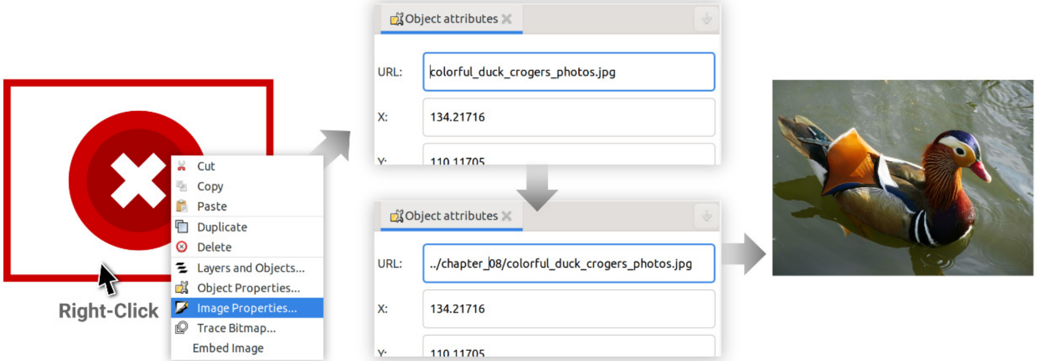
Figure 10.11 – Importing an image onto the canvas

There are lots of options here, but the most important ones for our purposes are the **Embed** and **Link** options. If we choose **Embed**, Inkscape will make a copy of the image and place it in our file. Embedding the image saves it *inside* our document, increasing the file size. However, doing this means we don't need to keep the original (separate) image file around.

This comes with a disadvantage: if we copy/paste this embedded image, each time we paste, it will embed a whole new version of this image, continuously increasing our file size, even though the image is the same! We can get around this in several ways. Instead of copying/pasting the image, we can clone it and copy/paste the clones instead.

The other way (which saves even more space) is to choose **Link** instead of **Embed** when we import. This has the added advantage of being able to change the original image, say in a photo-editing program such as GIMP or Photoshop, and when we save over the original file, the changes will be updated automatically in our Inkscape document.

The disadvantage is that we must keep the image around in the *same place* it was for Inkscape to find and load it. You can also keep the image in the same folder as the `.svg` file you saved, and Inkscape can find it that way too. If Inkscape doesn't find the image, it will instead show a big red circle and an **X** symbol in place of the image, as shown in *Figure 10.12*.



URL: colorful_duck_crogers_photos.jpg
X: 134.21716
Y: 110.11705

URL: ../chapter_08/colorful_duck_crogers_photos.jpg
X: 134.21716
Y: 110.11705

Figure 10.12 – Fixing a broken image link with the Image Properties > Object attributes dialog

If this sounds like too much hassle, you can always just copy the file into the same folder as your saved `.svg` file and re-open it in Inkscape. When Inkscape opens a file and finds a broken file link, it will automatically search in the same folder as the `.svg` file. If it's there, Inkscape will then fix the broken link automatically for you.

For this reason, I usually keep my linked files in the same folder as the `.svg` file. This is generally a good practice, as it makes your images easier to find later and reuse in other parts of your projects as well, making it more likely you'll find them together in the future. We will discuss file management best practices more in *Chapter 15*, *Document Properties, Pages, Exporting, and Printing*.

It should be noted that you can change your mind and **Embed** the file at any time you wish by right-clicking the image and choosing the **Embed Image** option from the **Context** menu. You will also notice the **Edit Externally** option, which will open up an image editor (GIMP by default, if you have it installed). Any changes you make to the original in GIMP will then be reflected in Inkscape when you save over the original linked image. Neat huh?

That brings us to our practice lesson, so let's use our newfound knowledge to make a dynamically adjustable light bulb!

Try it yourself

Clones give us a variety of ways to use an original shape and have it change all the copies we make of it throughout the document. We can use clones on top of each other in a layering effect to make a light bulb whose shape we can control by editing a simple original shape, as shown in *Figure 10.13*.

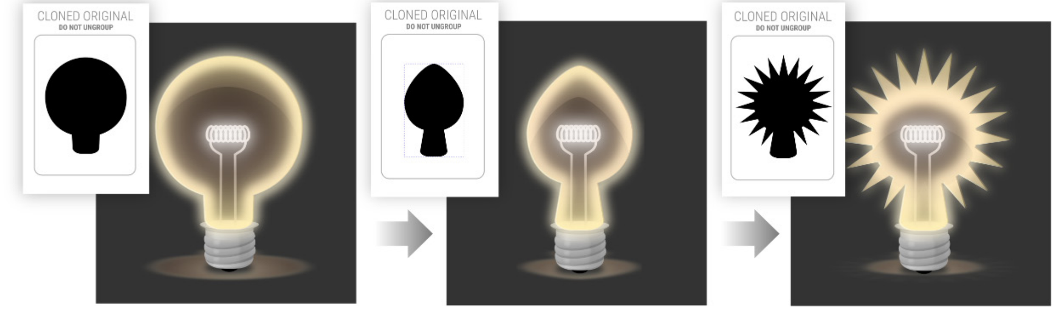


Figure 10.13 – A dynamic light bulb, which changes along with our original shape

Pretty neat, eh? So how do we go about making this? Let's get started!

1. Let's start out by making our basic light bulb parts. These bits will be *static*. That is, they will not be part of the clones we make and thus will be unchanged by our modification of the bulb shape clones. *Figure 10.14* shows the process for making each part of the bulb.

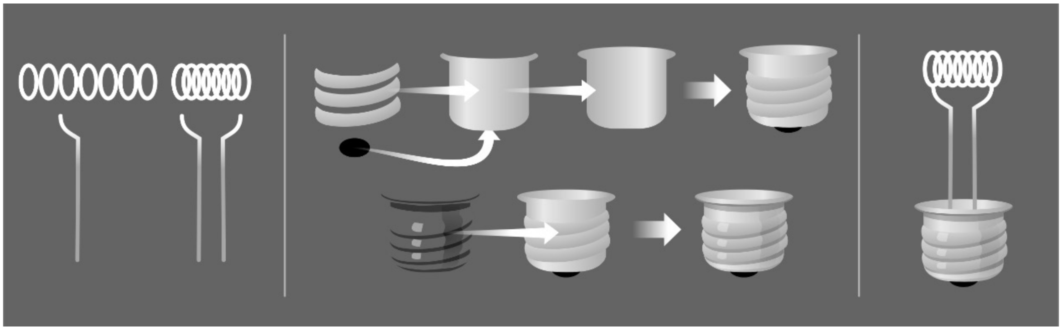


Figure 10.14 – Creating and assembling the static parts of the light bulb

2. Now that we have the base of our light bulb graphic, we can make the basic shape of our light bulb, unset the fill and stroke, group it to itself, box it, label it, group all that together, and move it off canvas somewhere for safe keeping, as shown in *Figure 10.15*.

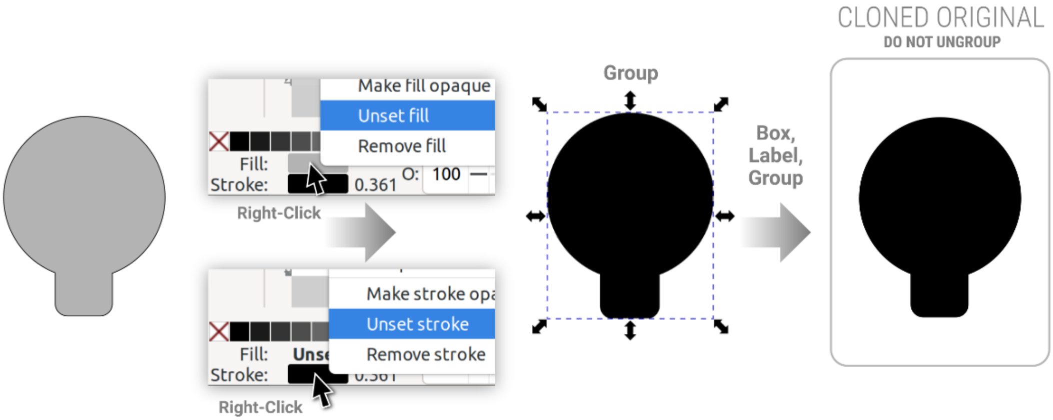


Figure 10.15 – Unsetting the bulb base shape fill and stroke, then group, box, label, and group

3. Now we're going to double-click into the group and select the group with our original in it, and make a clone of it by selecting **Edit** > **Clone** > **Create Clone**. We will then cut/paste this clone outside the group onto our canvas, as shown in *Figure 10.16*.

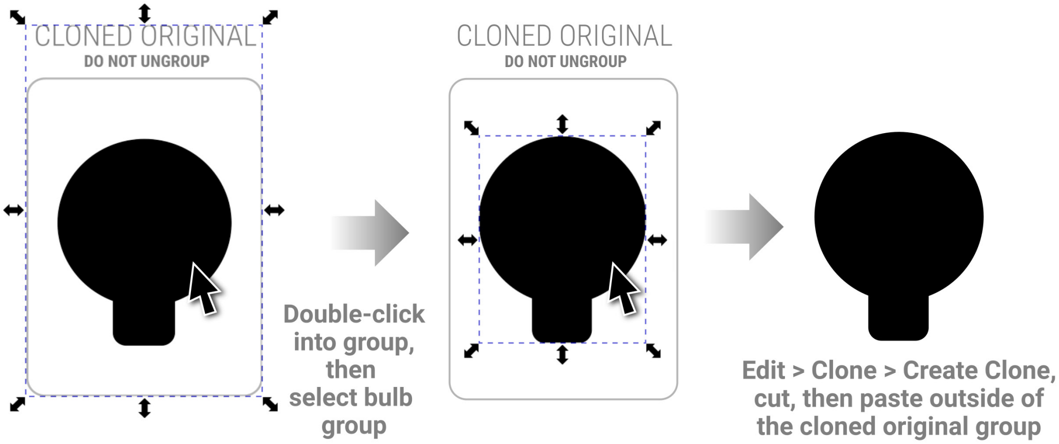

CLONED ORIGINAL
DO NOT UNGROUP
Double-click into group, then select bulb group
CLONED ORIGINAL
DO NOT UNGROUP
Edit > Clone > Create Clone, cut, then paste outside of the cloned original group

Figure 10.16 – Making a clone of the bulb shape group, and cutting/pasting it outside the group

4. Now that we have a clone of our bulb group, we can simply copy/paste it to get five clones of the original. Let's line them up, change the stroke width, change **Fill** to **None**, add some blur to three of them, and set a clip group on two of them individually (by right-clicking and choosing **Set Clip Group**). This process is shown in *Figure 10.17*.

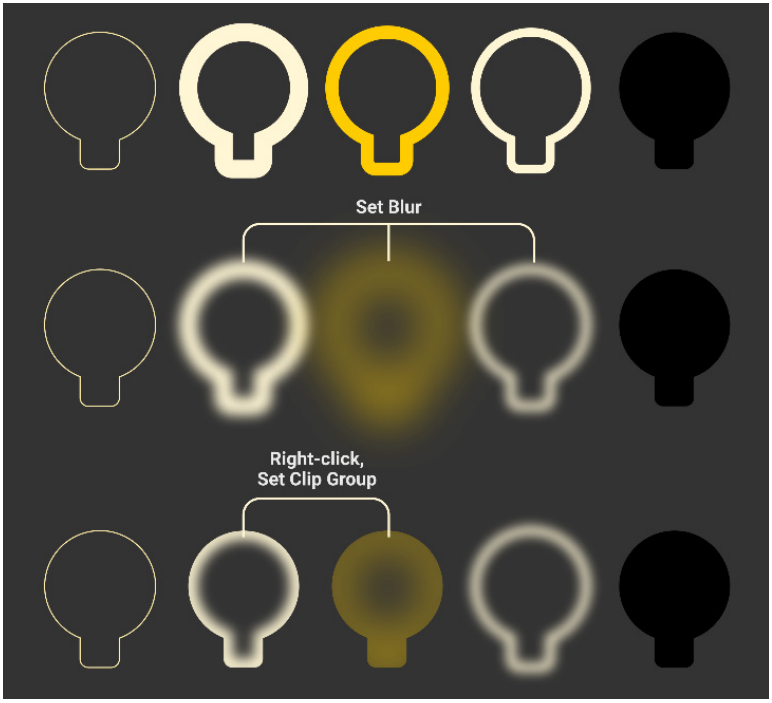


Figure 10.17 – Modifying our clone copies to make parts of our bulb glass

5. When we overlap the first four clone copies, we get most of our bulb. We'd like to make a bit of a shadow to cut out the bottom of the highlight of the bulb. We will do that by giving our fifth clone copy a black-to-transparent gradient and shrinking it a bit. As you can see in *Figure 10.18*, this gives us a nice illusion of a reflection highlight, which will also change according to the original bulb shape.

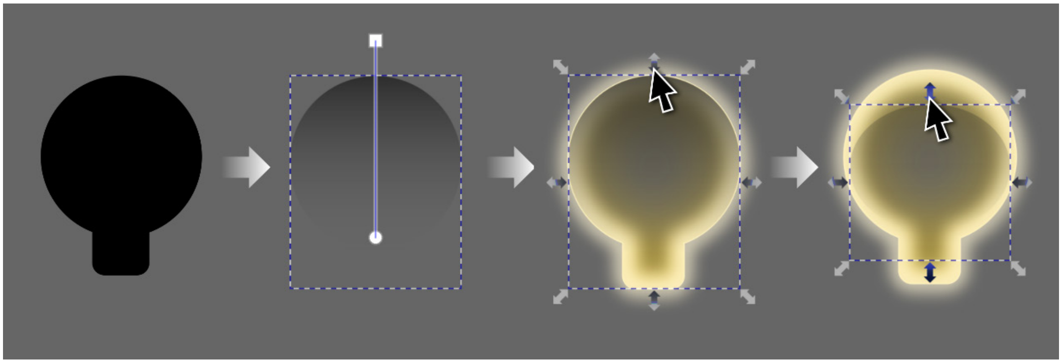


Figure 10.18 – Making the reflection illusion by applying a gradient fill to the clone and squishing it a bit vertically

6. We can now assemble our light bulb parts, copy/paste in place a copy of the top of the filament, and add a blur to it for a nice glow. Before we play with the original, we'll do one sneakier trick to make the glow cast on the floor: we simply make a copy of our bulb shapes, squish it, flip it upside-down, and move it behind the bulb base. We can then give it a little blur, and we have our ground glow. This process is shown in *Figure 10.19*.

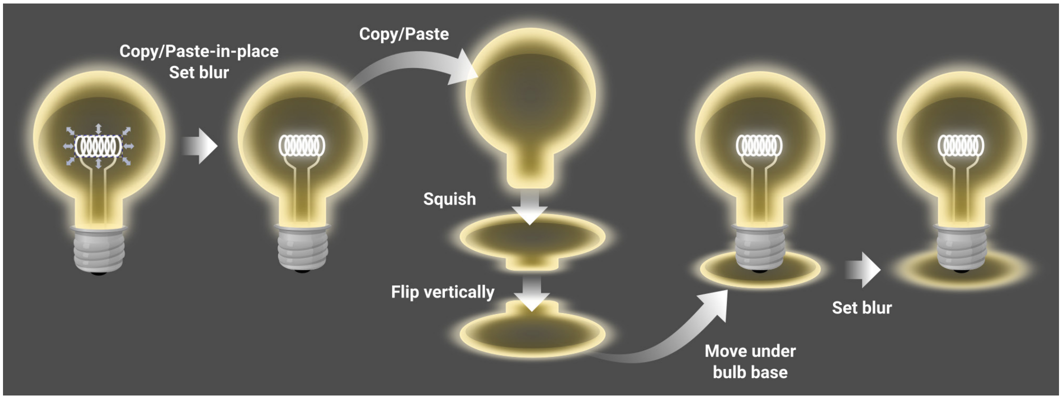


Figure 10.19 – Adding filament and ground glow to our bulb

7. We can now play with the original, clicking inside both groups to edit the original bulb shape. Since we grouped the original bulb shape before cloning it, we can even add shapes to that group, and provided we unset the fill and stroke, the shapes will affect the bulb shape, as shown in *Figure 10.20*. Super!

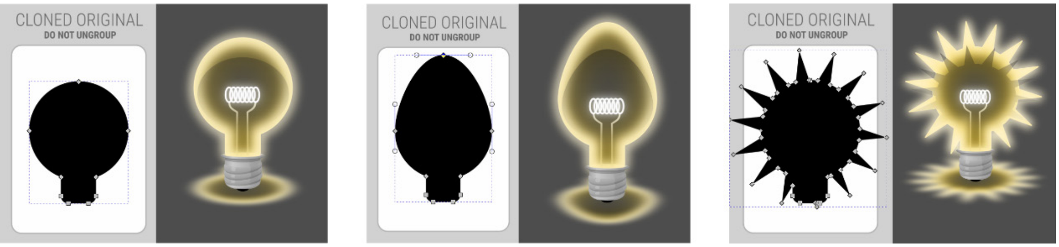


Figure 10.20 – Editing the original changes our bulb shape!

8. Notice, however, that the reflection shadow we made earlier doesn't work well for the star shape, falling outside the bulb. We want it to stay completely inside the original bulb shape, so we simply cut the shadow, pull out one of our clip groups, double-click inside it, and paste our shadow inside it. As you can see from *Figure 10.21*, this ensures the shadow is clipped to the original bulb shape.

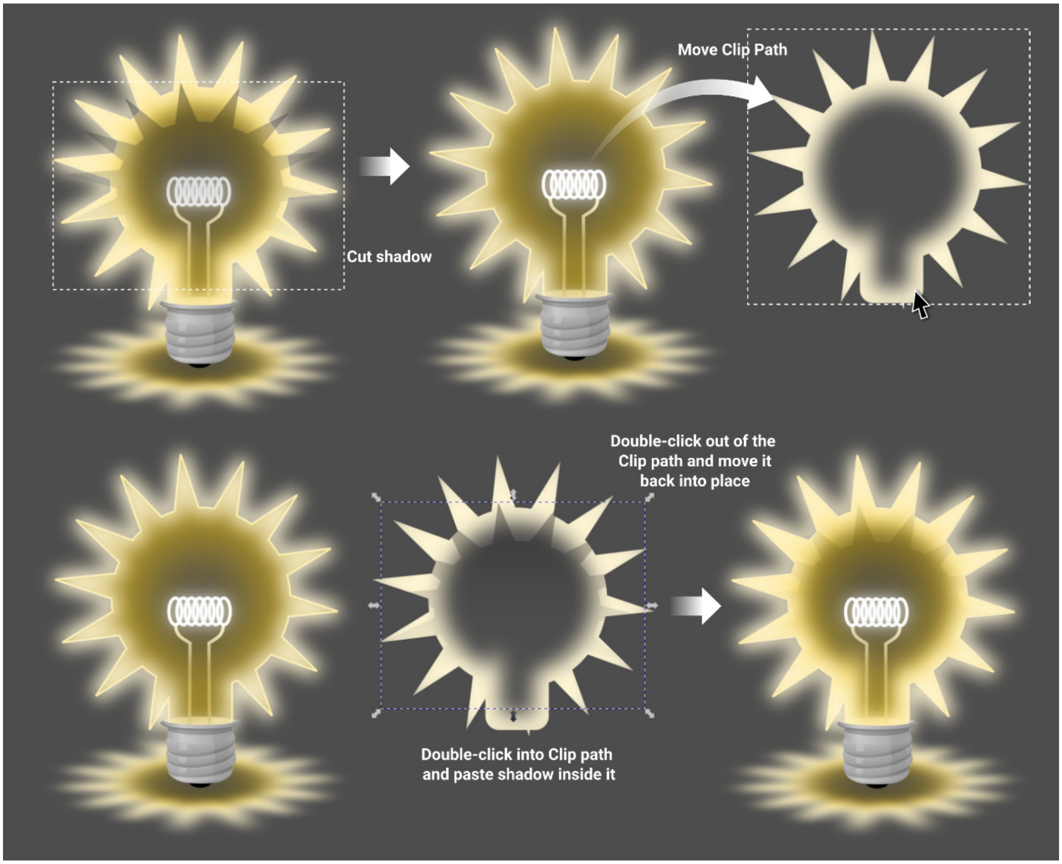

Move Clip Path
Cut shadow
Double-click out of the Clip path and move it back into place
Double-click into Clip path and paste shadow inside it

Figure 10.21 – Fixing the shadow by pasting it into one of our bulb clip groups

And we're done! There are certainly many more things we can do with clones and groups, and I hope you'll have as much fun with them as I do with this experience under your belt.

Summary

In this chapter, we learned a great deal about clones and using them to make supertemplates to save lots of reworking of graphics. In the next chapter, we will learn about how to use groups and layers to organize more complex documents and keep everything nice and tidy!

Part 3: Inkscape's Power Tools

With the basics of Inkscape covered, this part tackles some of Inkscape's power tools, including clipping and masks, Live Path Effects, advanced text effects, layers, and grouping, and ends with details on how to get the most out of files made with Inkscape.

There are the following chapters in this section:

- *Chapter 11, Organization Using Layers*
- *Chapter 12, Live Path Effects*
- *Chapter 13, Filters and Extensions*
- *Chapter 14, Vectorizing with Trace Bitmap*
- *Chapter 15, Document Properties, Pages, Exporting, and Printing*

11

Organization Using Layers

Up to now, we've been using groups to organize things on the fly. However, depending on the complexity of the illustration we're working on, it can get a little hard to use just groups and sub-groups to arrange everything. This is where layers can really help the situation!

In this chapter, we will cover the following topics:

- What is a layer?
- Understanding the **Objects and Layers** dialog
- Organizing projects using groups and layers
- Layer transparency and Blend modes
- Cloning layers and other advanced layer tricks

Technical requirements

You can download the practice files for this chapter from GitHub at

```
https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape/
tree/main/Chapter11
```

What is a layer?

Put simply, a **layer** is just an invisible drawer to store things. Using groups alone is like piling objects together to clean up a room. Sure, it makes the room appear cleaner, but after a while, the groups just become another mess. Wouldn't it be nice if we could put our neat piles of objects in drawers?

This would then enable us to close the drawers and hide those objects until we need them. This is precisely how we use layers in projects. A layer can be seen as a metaphorical drawer, and closing said drawer is equivalent to hiding the layer.

Understanding the Objects and Layers dialog

Inkscape starts you out with one layer called **Layer 1**. If we start a brand new file in Inkscape and call up the **Objects and Layers** dialog by choosing **Object** > **Objects and Layers…** we get a mostly empty dialog containing an item called **Layer 1**, as shown in *Figure 11.1*.

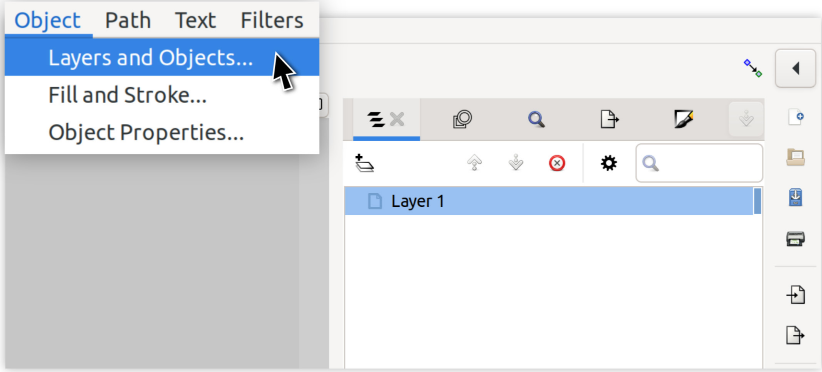


Figure 11.1 – Calling up the Layers and Objects dialog

By default, Inkscape will add objects to **Layer 1**. You can think of it as a group containing all of your objects and object groups. As of Inkscape 1.1, this dialog will also show you each object you create.

We can see this if we draw an object (in our case, a circle) onto our blank canvas; Inkscape will add an entry item indented under **Layer 1** to show that it's been added to that layer, as shown in *Figure 11.2*.

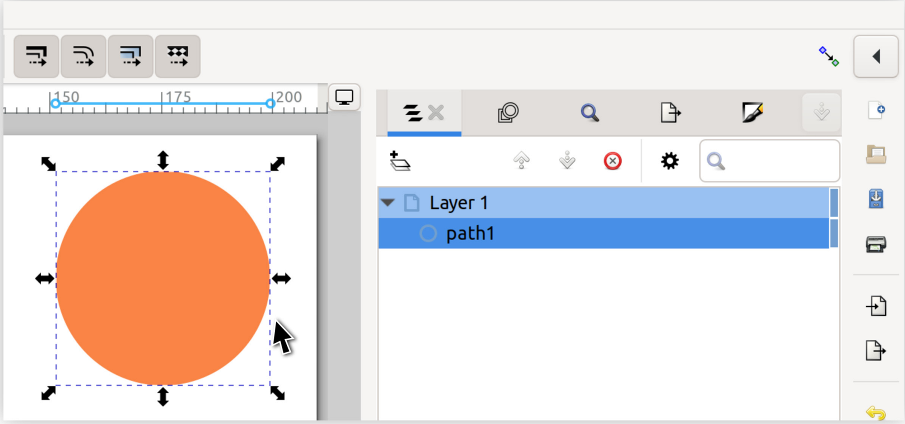


Figure 11.2 – Newly drawn object shown in the Layers and Objects dialog

You can see that the current layer is highlighted in light blue, and the selected object is shown in dark blue. Inkscape has named this circle `path2299` automatically, but we can double-click on the object's label in the dialog and rename it to something more memorable, as shown in *Figure 11.3*.

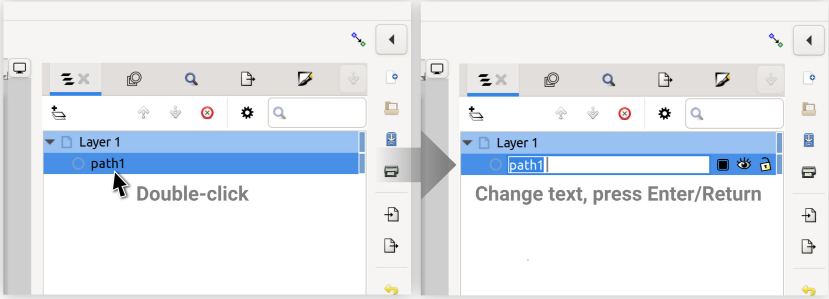
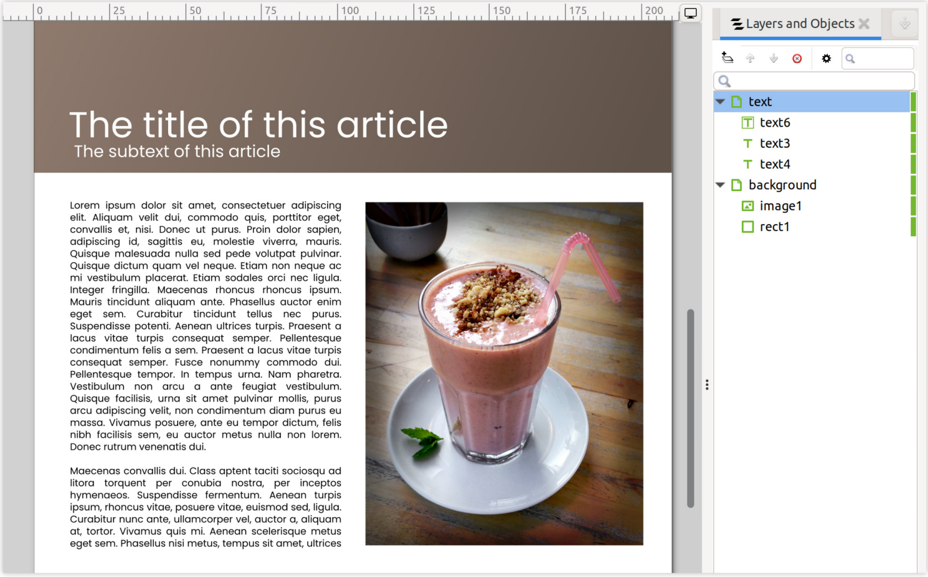


Figure 11.3 – Editing the object label

We can, of course, do the same with the layer label, and indeed, some good candidates for labels might include **background** or **foreground** or **text and titles** if you're designing a flyer.

Organizing projects using groups and layers

Layers provide a nice functional separation between parts of the document you're working on. *Figure 11.4* shows a simple setup with the **background** and **text** layers, each containing the appropriate objects.



The title of this article

The subtext of this article

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aliquam velit dui, commodo quis, porttitor eget, convallis et, nisi. Donec ut purus. Proin dolor sapien, adipiscing id, sagittis eu, molestie viverra, mauris. Quisque malesuada nulla sed pede volutpat pulvinar. Quisque dictum quam vel neque. Etiam non neque ac mi vestibulum placerat. Etiam sodales orci nec ligula. Integer fringilla. Maecenas rhoncus rhoncus ipsum. Mauris tincidunt aliquam ante. Phasellus auctor enim eget sem. Curabitur tincidunt tellus nec purus. Suspendisse potenti. Aenean ultrices turpis. Praesent a lacus vitae turpis consequat semper. Pellentesque condimentum felis a sem. Praesent a lacus vitae turpis consequat semper. Fusce nonummy commodo dui. Pellentesque tempor. In tempus urna. Nam pharetra. Vestibulum non arcu a ante feugiat vestibulum. Quisque facilisis, urna sit amet pulvinar mollis, purus arcu adipiscing velit, non condimentum diam purus eu massa. Vivamus posuere, ante eu tempor dictum, felis nibh facilisis sem, eu auctor metus nulla non lorem. Donec rutrum venenatis dui.

Maecenas convallis dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Suspendisse fermentum. Aenean turpis ipsum, rhoncus vitae, posuere vitae, euismod sed, ligula. Curabitur nunc ante, ullamcorper vel, auctor a, aliquam at, tortor. Vivamus quis mi. Aenean scelerisque metus eget sem. Phasellus nisi metus, tempus sit amet, ultrices

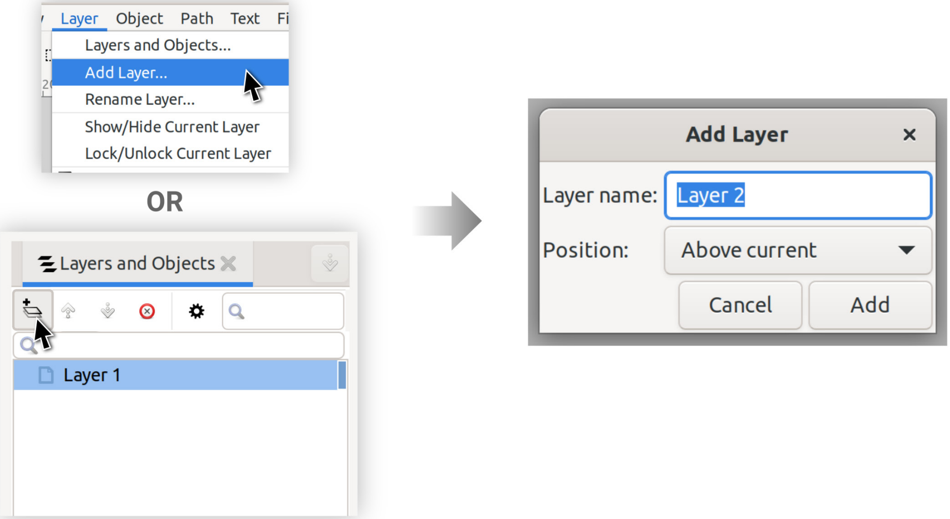
Figure 11.4 – A simple example of using layers for organization

However, the names are solely for your own reference, so if you'd rather name them `Phil` and `Lily`, it's all on you. You can see, in this case, we've sorted three text objects into the text layer and an image and brown rectangle into the background layer.

These appear slightly indented in the **Layers and Objects** dialog. Clicking the small arrow at the side of the layer will expand or collapse the list of objects in the layer.

Inkscape gives us one layer already, but what options do we have to create more? A quick look at the **Layer** menu gives us many options for working with our layers, including **Layer** > **Add Layer…**.

We can also simply add a layer by clicking on the + icon at the top of the **Objects and Layers** dialog. Both of these options pop up the **Add Layer** dialog, which is shown in *Figure 11.5*.



Layer name: Layer 2

Position: Above current

Figure 11.5 – Summoning the Add Layer dialog

By default, Inkscape suggests **Layer 2** as a name for our new layer, but we can enter whatever name we like. We are also given a **Position** drop-down field with various options, such as **Above current**, which places the layer above the currently selected layer. **Below current** adds the new layer below the current layer.

Finally, the **as a sublayer of current** option puts the new layer inside the current layer. You can think of a sublayer as a box inside a box. For example, say you have a layer called **screwdrivers**. You might have several boxes inside it for the **flat-head** and **Phillips-head** screwdriver types.

Inkscape can support any number of nested layers like this, just as you can have groups inside groups. We will explore more ways to use these to organize our documents later in the chapter. For now, we will walk through a simple example of using layers.

Let's create a basic layer structure for the document we saw earlier in the article and the milkshake picture example:

1. We will start by renaming **Layer 1** to `text` and then adding another layer called `background`.
2. Selecting the **background** layer, we can add a color rectangle that will go under our title for a bit of visual interest – we'll name it `color rectangle` in the **Objects and Layers** dialog.
3. Click on our **text** layer to make it the active layer and add a title on-canvas with **Text** tool.

   However, we immediately have a problem. Because of the order in which we added our layers, the **text** layer is beneath the **background** layer.

4. To fix this, we can select our **text** layer and simply click the **Move selection up in the stack order** button at the top of the **Objects and Layers** dialog.

   This will move the layer up the layer stack above our background, and all of a sudden, we can see our text, as shown in *Figure 11.6*.

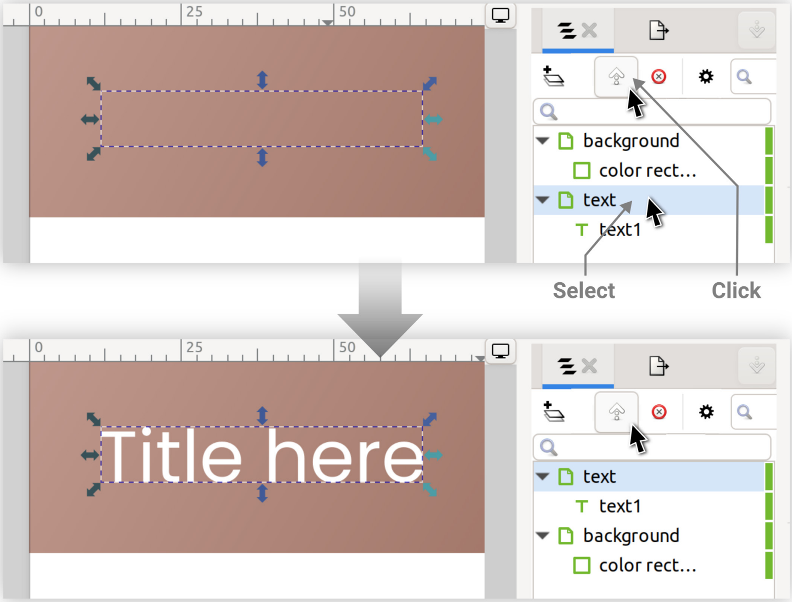


Figure 11.6 – Moving a layer up in the layer stack via a button

5. You can also click and drag objects and layers around in the dialog to rearrange them.

   By clicking and holding anywhere on the layer bar (the blue highlighted part), the cursor turns into a little grabbing-hand icon, and an indicator appears (a blue line with curved edges), showing where the layer (or object) will be dropped when you release the mouse button.

This indicator is shown in *Figure 11.7* in various states to indicate where the layer will end up when you drop it.

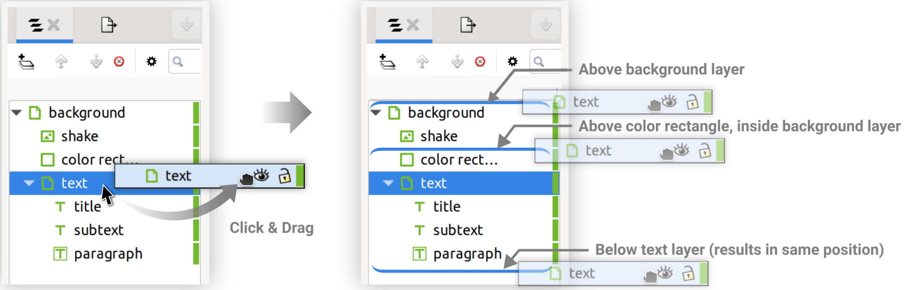

background
shake
color rect…
text
title
subtext
paragraph
text
Click & Drag
Above background layer
Above color rectangle, inside background layer
Below text layer (results in same position)

Figure 11.7 – Dragging a layer and hovering to show the drop indicators (blue)

You may have noticed that when you hover the mouse over a layer or object in the dialog, some extra icons appear on the right side – an *eye* and a *lock*. Respectively, these allow you to hide and lock specific layers and objects.

6. Clicking the **eye** icon hides the object or layer, which removes it (and any of its contents) from the canvas view as if it was not there at all! Clicking the **eye** again unhides the layer/object, restoring it to its former visibility.
7. Likewise, the **lock** icon locks the layer or object so that it cannot be selected or moved on the canvas. *Figure 11.8* shows how this functionality is useful while editing some clouds in situ over a sky background, with some skyscrapers in front.

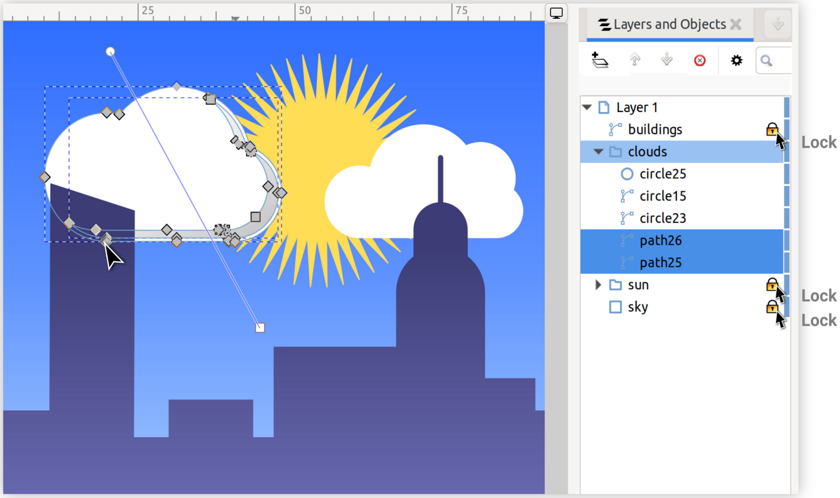


Figure 11.8 – Locking layers on top and behind the working objects

As you can see, locking and hiding layers becomes handy if you are working with objects under other objects, and want to temporarily ensure you don't select the top objects.

These **hide** and **lock** features are what make the **Layers and Objects** dialog great for organization and ease of editing. You don't have to use the dialog directly to lock or hide an object, however.

Simply right-click a selected object and choose **Hide Selected Objects** or **Lock Selected Objects**. Locking or hiding an object this way will show a closed-eye icon to the right of the object name in the **Layers and Objects** dialog.

Likewise, right-clicking anywhere over a hidden object will bring up the **Unhide Objects Below Cursor option**.

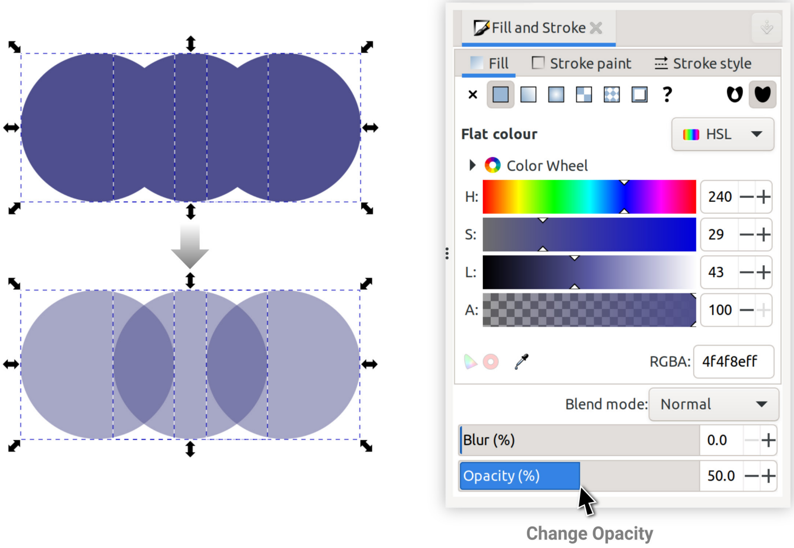
You will notice that the **Layers and Objects** dialog only shows the **hide** and **lock** icons when you hover over an entry or if it's hidden or locked. This convention makes it easy to quickly find which objects and layers are locked and/or hidden.

All this is quite handy for document organization, and the more you get into the habit of naming your groups and layers, the easier it will be to get to work with more complex illustrations and documents. There are more benefits to using groups and layers than just organization, however.

Layer transparency and Blend modes

Yes, that's right! You can indeed change **Blend mode** on a layer (or group). There are several functional advantages to doing this. The most obvious is that you can change the transparency and blend of everything inside the layer all at once. Note that this is functionally different from changing the transparency or Blend mode of each individual object.

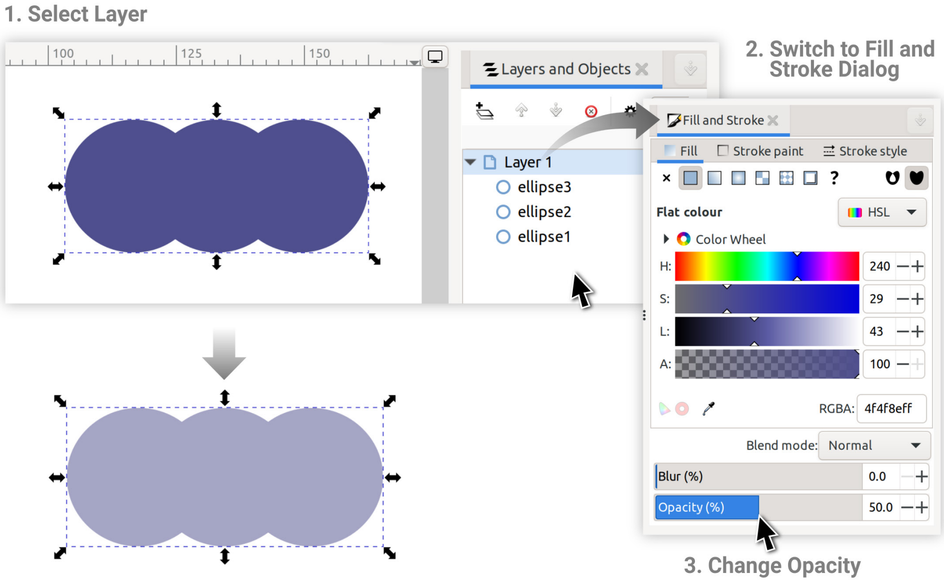
To demonstrate, let's consider a set of overlapping circles. If we select them all and change the transparency, Inkscape will change the transparency of each of the shapes. This means that we see darker areas when the shapes overlap since the transparency of, say, 50% adds up when transparent shapes overlap, as shown in *Figure 11.9*.



H: 240
S: 29
L: 43
A: 100
RGBA: 4f4f8eff
Blend mode: Normal
Blur (%) 0.0
Opacity (%) 50.0

Figure 11.9 – Changing the opacity of selected circles individually via the Fill and Stroke dialog

Sometimes, however, this isn't what we want. If we have a layer containing these three circles, we might want to modify the overall transparency, leaving the containing circles opaque relative to other layers instead of each other. We can do this through the **Layers and Objects** dialog, as shown in *Figure 11.10*.



H: 240
S: 29
L: 43
A: 100
RGBA: 4f4f8eff
Blend mode: Normal
Blur (%) 0.0
Opacity (%) 50.0

Figure 11.10 – Changing Opacity on a layer instead of individual shapes

As you can see, this gives us full control over the transparency of the combined objects in the layer with none of the overlaps. Of course, this can also be done for groups, using the same method.

Indeed, even the **Blend** mode can be changed for the layer in this way, so when you want to change something about your layer, you can select it in the **Layers and Objects** dialog, then switch to the **Fill and Stroke** dialog to make changes.

As of Inkscape 1.3, there are also hidden options in each entry in the **Layers and Objects** dialog. These are in the form of icons, which appear when you hover the mouse over a layer or object entry. The first of these icons is a square and allows you to change layer opacity and other layer view options. Clicking on this square results in the popup shown in *Figure 11.11*.

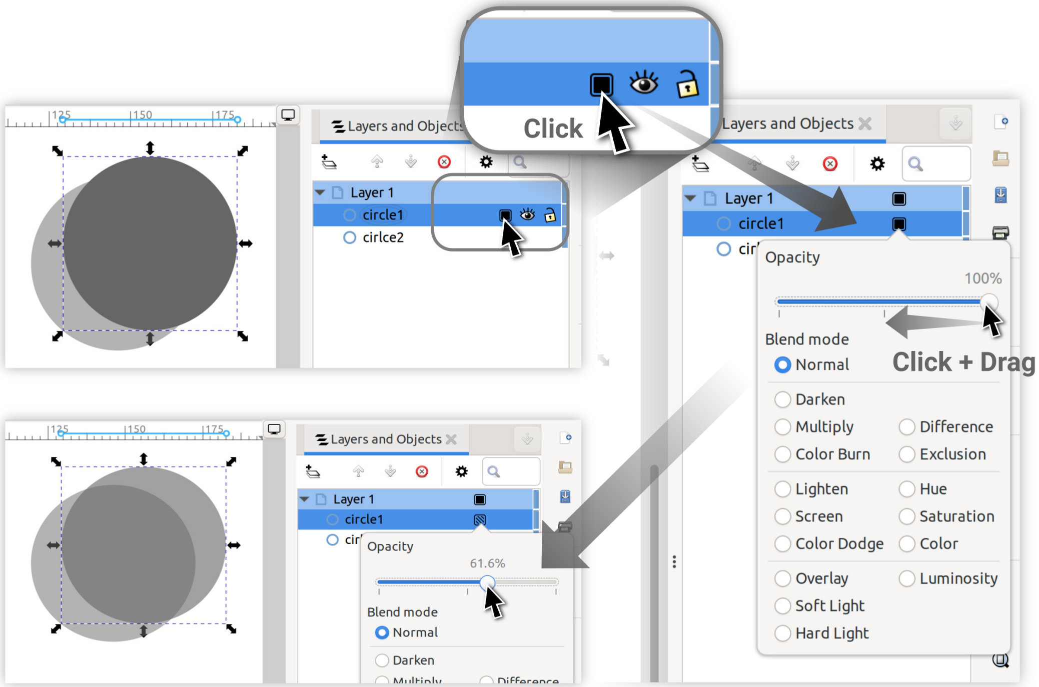


Figure 11.11 – Changing Opacity using the new controls beside each entry in the Layers and Objects dialog

You might notice after changing the opacity that the icon remains in the entry when you take your mouse cursor off it. This rather handy feature lets you know at a glance which layers and objects are not set to 100% opacity.

Also, note the same **Blend mode** options we explored earlier in this book are now accessible in the popup under the **Opacity** settings. These save you from having to rummage around the **Fill and Stroke** dialog to change Blend modes. Handy, isn't it? Let's learn some other cool tricks and explore cloning our layers.

Cloning layers

Remember all the fun we had with clones in the previous chapter? Well, turns out we can have all the convenience of working with clone groups inside a layer by cloning the layer!

This can help prevent us from accidentally destroying our clone hierarchies because Inkscape never automatically selects just a layer – you have to manually select it inside the **Layers and Objects** dialog.

To make a clone of a layer, we need only select the layer and then select **Edit** > **Clone** > **Create Clone**, as before. You can see from *Figure 11.12* that we get a new clone object outside our layer called **use176**, which we can then name as something more memorable, such as **raccoon clone**.

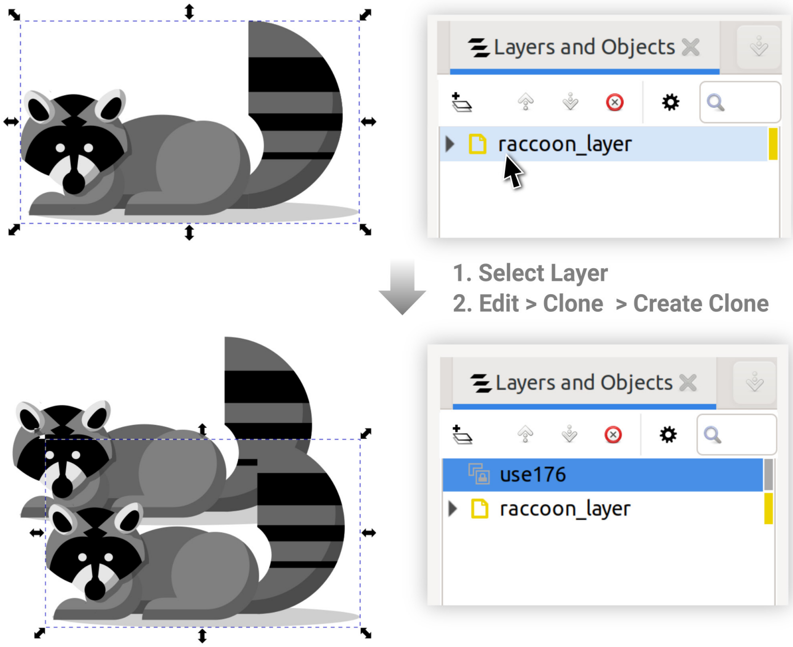


Figure 11.12 – Cloning a layer

There's a catch, though – for that extra protection, we have to sacrifice resizing our original because the moment we group a layer in order to resize the original without resizing the clones, we leave the layer vulnerable to being selected on canvas and ungrouped.

There is a workaround, though – we can *temporarily* group **raccoon_layer**, resize it, then ungroup it once more without affecting the clones, as shown in *Figure 11.13*.

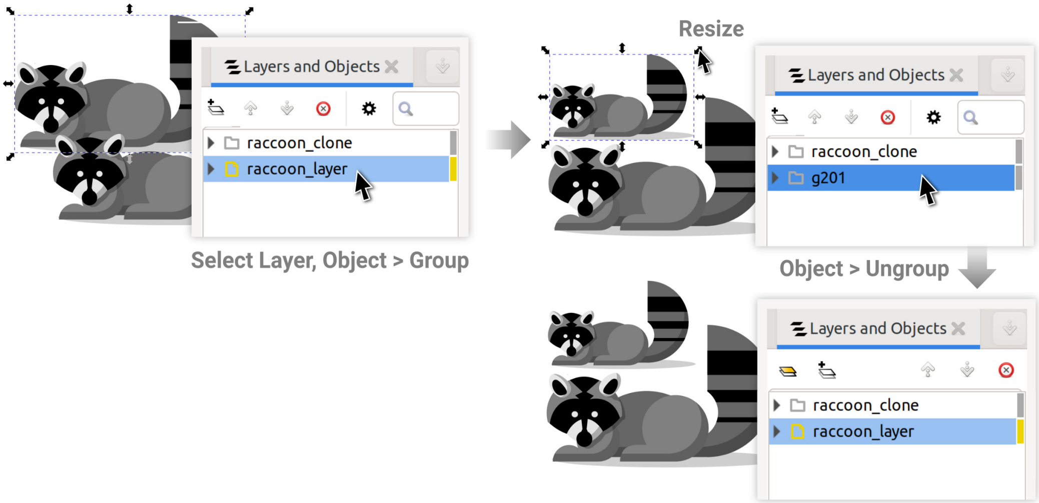


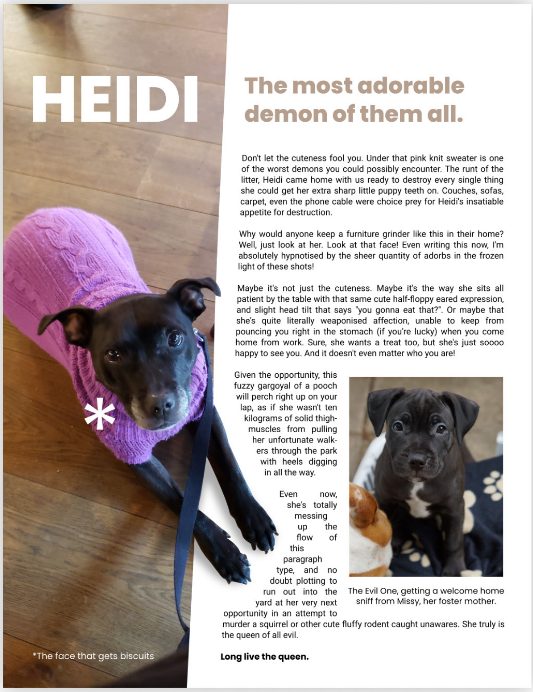
Figure 11.13 – Grouping, resizing, and ungrouping a layer to prevent affecting clones

As a final note, you can also clone, group, and ungroup a layer or object in the **Layer and Objects** dialog directly by right-clicking the entry and choosing the appropriate action out of the context menu, just as you can over the canvas. Pretty handy, eh?

So, there you have it! Your crash course in layers is now complete. Let's get some more practice and construct a full e-flyer, nicely organized with layers!

Try it yourself

We saw a small example of using layers for organization earlier in this chapter with our shake article. In this exercise, you'll do it on your own, on a topic of your choosing. We'll be creating something like *Figure 11.14*.



HEIDI

The most adorable demon of them all.

Don't let the cuteness fool you. Under that pink knit sweater is one of the worst demons you could possibly encounter. The runt of the litter, Heidi came home with us ready to destroy every single thing she could get her extra sharp little puppy teeth on. Couches, sofas, carpet, even the phone cable were choice prey for Heidi's insatiable appetite for destruction.

Why would anyone keep a furniture grinder like this in their home? Well, just look at her. Look at that face! Even writing this now, I'm absolutely hypnotised by the sheer quantity of adorbs in the frozen light of these shots!

Maybe it's not just the cuteness. Maybe it's the way she sits all patient by the table with that same cute half-floppy eared expression, and slight head tilt that says "you gonna eat that?". Or maybe that she's quite literally weaponised affection, unable to keep from pouncing you right in the stomach (if you're lucky) when you come home from work. Sure, she wants a treat too, but she's just soooo happy to see you. And it doesn't even matter who you are!

Given the opportunity, this fuzzy gargoyal of a pooch will perch right up on your lap, as if she wasn't ten kilograms of solid thigh-muscles from pulling her unfortunate walkers through the park with heels digging in all the way.

Even now, she's totally messing up the flow of this paragraph type, and no doubt plotting to run out into the yard at her very next opportunity in an attempt to murder a squirrel or other cute fluffy rodent caught unawares. She truly is the queen of all evil.

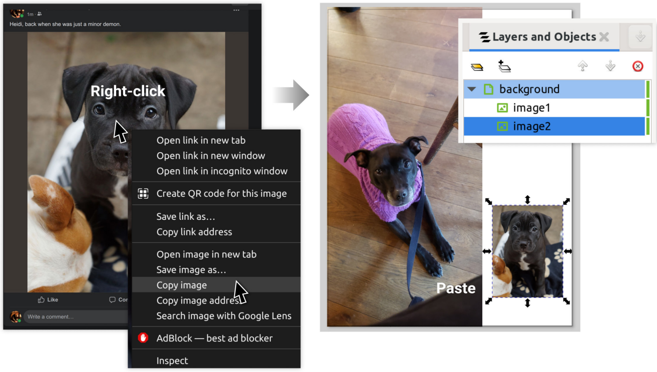
The Evil One, getting a welcome home sniff from Missy, her foster mother.

*The face that gets biscuits

Long live the queen.

Figure 11.14 – Article starring a friend's dog, Heidi

1. Let's begin by renaming **Layer 1** to `background`, then choose an image, either from your computer or one off the internet. You can download the image file and choose **File** > **Import**, or in some cases, you can just right-click the image on the web page and choose **Copy image** (in Google Chrome, for example), as shown in *Figure 11.15*.



Heidi, back when she was just a minor demon.

Figure 11.15 – Copy/paste from Google Chrome right into Inkscape

You can then paste the image directly into Inkscape, resizing it to make room for your text.

We need to make more space for our article, so we'll cut away part of the floor under Heidi (the dog), leaving her paws and part of her ear over the text area. We could use a rectangle for the diagonal slice and union that with a trace around her paws to use as a clip.

2. But since we might want to move things around after the fact (moving Heidi higher or lower, for example), we can instead group the diagonal slice and paw objects, rename that group to `Heidi clip`, and select the image and the **Heidi clip** group using **Object** > **Clip** > **Set Clip** as before.

   Since the clip shapes are grouped, Inkscape treats them as though they were the same shape, as shown in *Figure 11.16*. Neat, huh?

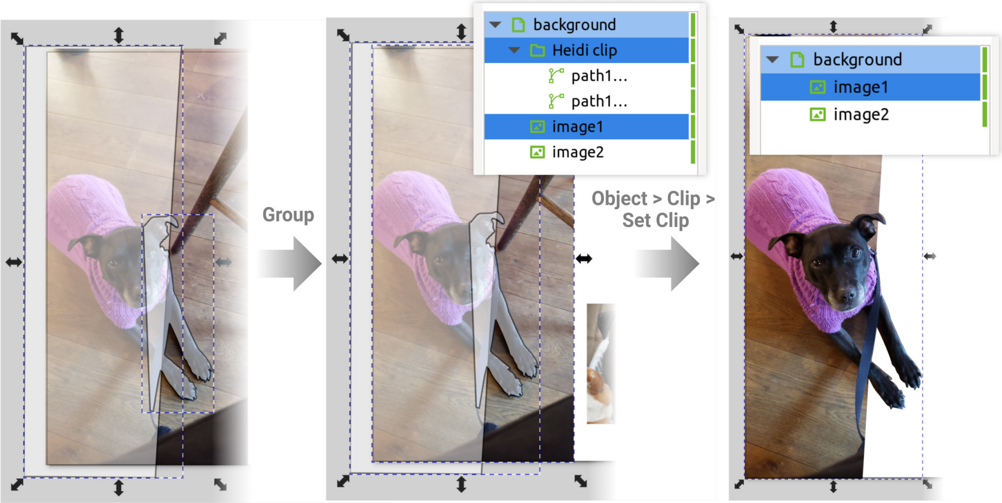
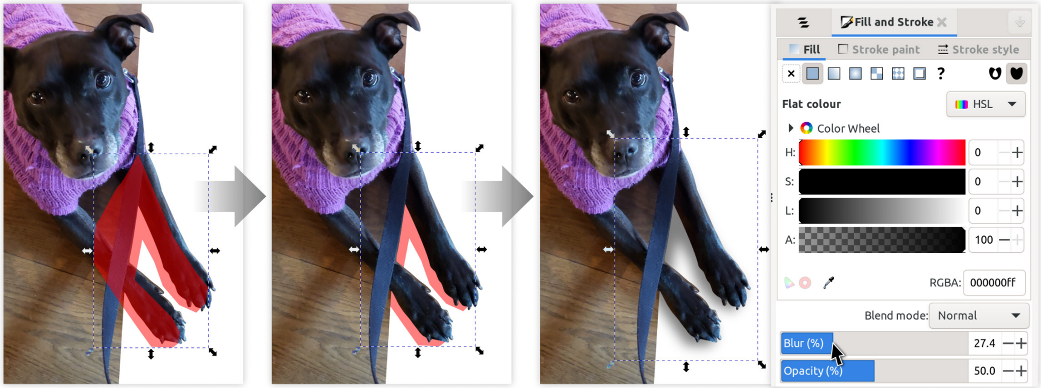


Figure 11.16 – Using a group of objects as a clip

Notice that the **Heidi clip** group vanishes from the **Layers and Objects** dialog when we select **Set Clip**, leaving only the image name. Just note that we can always select **Object** > **Clip** > **Release Clip** to see our **Heidi clip** group in the dialog again.

Before we move on to titles and other text, let's place a shadow under Heidi's paws to make it look like she's sitting on top of the white area.

3. To do this, we can just draw a black shape, put it under her paws, and adjust the blur and opacity on them, as shown in *Figure 11.17*.



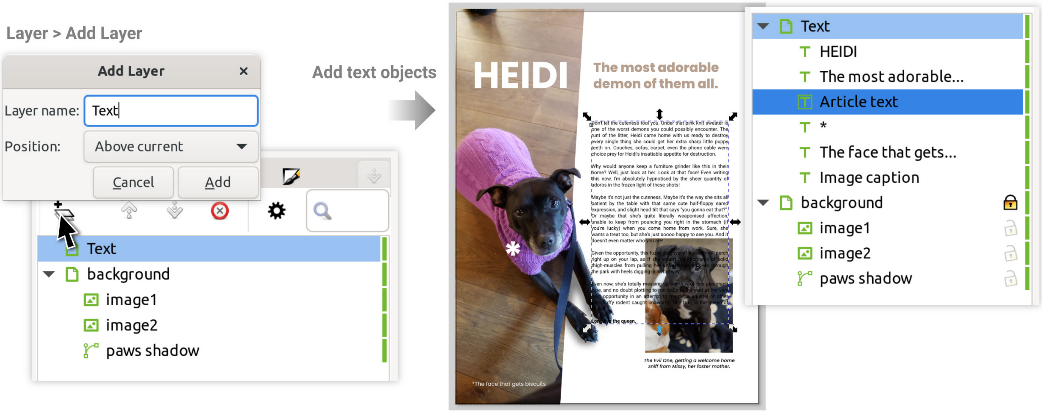
HSL
H: 0
S: 0
L: 0
A: 100
RGBA: 000000ff
Blend mode: Normal
Blur (%) 27.4
Opacity (%) 50.0

Figure 11.17 – Making a shadow under Heidi's paws

We can quickly rename our visible shapes to tidy up and make things easier to find if we edit this file in the future.

4. Then, with our background complete, we can lock it and then select **Layer** > **Add New Layer,** choosing **Text** for the layer name and **Above Current** for **Layer Position**, which will appear over our locked **background** layer.

   Here we can add our titles and text, as shown in *Figure 11.18*. You can see I've renamed the text objects for clarity, but it's not required.



Layer name: Text
Position: Above current

- Text
- background
  - image1
  - image2
  - paws shadow

- Text
  - HEIDI
  - The most adorable...
  - Article text
  - *
  - The face that gets...
  - Image caption
- background
  - image1
  - image2
  - paws shadow

Figure 11.18 – Adding the Text layer and the Text objects

Looks pretty good, but we'd like to use those neat text frame tricks we learned about in *Chapter 7, Using Text in Inkscape*, to flow our article text around the background graphics.

5. To do this, we'll make three shapes: a slanted one to flow the text into, and two shapes to flow the text around Heidi's paws and the image of Heidi as a puppy. I've made these red at a `50%` opacity, and we will set the opacity of each to `0` after performing each flow operation.

6. We flow the text into the rectangle with a slanted side, then select both of the other shapes and choose **Text** > **Set Subtraction Frames**. We can rename all the flow shapes we made for easy location later, as shown in *Figure 11.19*.

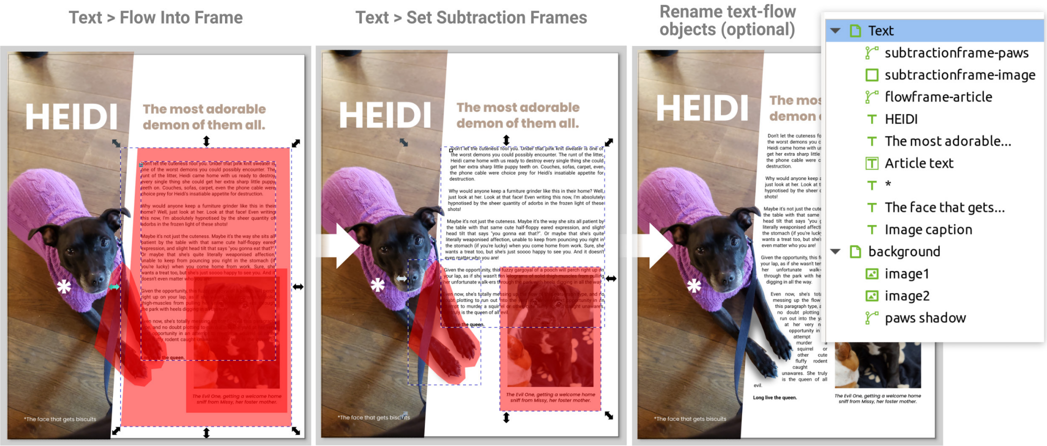


Figure 11.19 – Flowing text into and around shapes, then renaming objects

And that's it! We can now save our SVG file for later use. We'll go over how to save this as an e-flyer PDF in *Chapter 15*, *Document Properties, Pages, Exporting, and Printing*, but for now, we're done with our tour of the **Objects and Layers** dialog. Well done!

Summary

In this chapter, we went over how to organize our layers and objects using the **Layers and Objects** dialog, which has a utility that will serve us well while making more complex documents in Inkscape. The more you get into the habit of naming and organizing your layers and objects, the easier it will be to find, reuse, and edit parts of your documents.

With all this neatness and tidiness out of the way, let's get our hands dirty once more with Inkscape's superpower tools in the next chapter on **Live Path Effects**!

12

Live Path Effects

So far, we have seen some ways to automate certain design tasks with things such as clones, but we've really only just scratched the surface of what Inkscape can do. This chapter is about the power tools of Inkscape – the **path effects**, also known as **Live Path Effects** (**LPEs**), which offer non-destructive effects that can be applied to paths to generate all sorts of effects that would take a long time to do by hand.

In this chapter, we will cover the following topics:

- What is a Path Effect, and why is it *live*?
- The **Path Effects** dialog
- Using Path Effects with single-path objects
- Using Path Effects with groups
- Limitations of Path Effects

Technical requirements

You can download the practice files for this chapter from GitHub at

```
https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape/
tree/main/Chapter12
```

What is a Path Effect, and why is it live?

In graphic design, when we talk about making shapes, edits to photos, and so on, there are two kinds of routes we can take to get what we want – *destructive* and *non-destructive*. Simply put, a **destructive** workflow means that once we do something to a shape, it's done, and cannot be reversed (except for for a short time by the **Undo** function).

For example, when we use our Boolean operations (by selecting **Path** > **Difference**) to subtract a square from a circle, we make a new shape made of nodes, lines, and curves. Thus, the circle and square are destroyed to make the new shape and are no longer editable separately.

If we then wanted to move the square without moving the circle parts, we're pretty much out of luck. We would be better off drawing a new circle and square and then combining them again in the same way.

A **non-destructive** workflow is different; if we take that same circle and square, add a **Boolean path effect**, and set the effect to **Difference**, we can subtract the square and then later move it around separately, which then moves the hole in real time as you drag it on the canvas (which is what makes it *live*), without affecting the geometry of the circle. Even better, since our square is still a rectangle object, we can round the corners, which, in turn, rounds the corners of our hole, as shown in *Figure 12.1*.

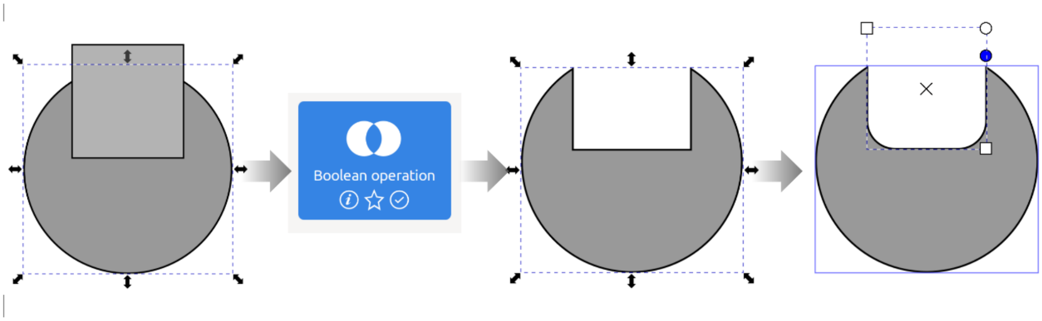


Figure 12.1 – An example of the Boolean path effect and editing the subtracted shape

In addition, we can, at any point, change from a **Difference** operation to any of the other Boolean operations. We can even remove the path effect entirely and preserve our original shapes. This is why it's *non-destructive*; at any point, we can make modifications without permanently changing our original shapes!

So, how do we manage this witchcraft? We start simply by selecting one of our objects and choosing **Path** > **Path Effects…**, which calls up the **Path Effects** dialog.

The Path Effects dialog

Like most of our other advanced functionality, Inkscape provides a dialog for Path Effects. Once opened, we will see a rather minimalist interface that resembles the **Layers** dialog, only completely empty. Without any objects selected, there is nothing to do here yet, and the controls are grayed out.

We need to select one (and only one) object for the **Add path effect** control to be active, which is the little plus sign in the bottom-left corner of the dialog. Once we click that button, we are presented with a myriad of great new tools to non-destructively modify our paths, as shown in *Figure 12.2*.

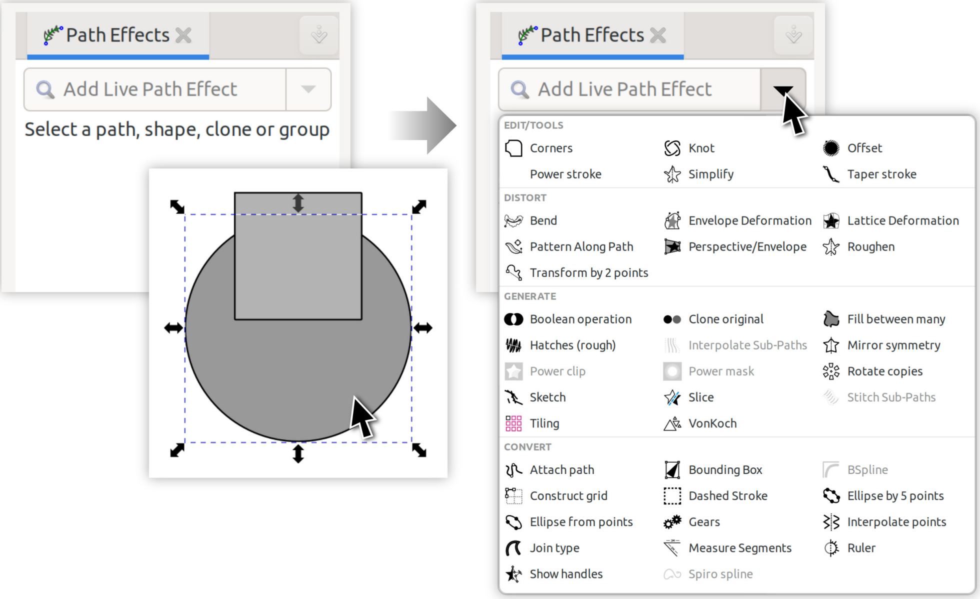
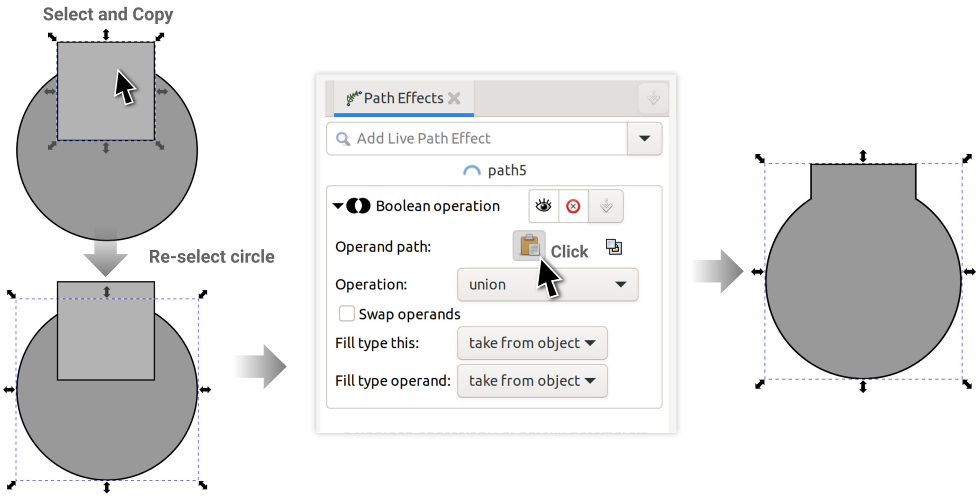


Figure 12.2 – Adding a path effect

Depending on which path effect we choose, different things will happen to the shape we have selected. For example, adding a **Boolean operation** path effect to our circle does nothing right away, except add an entry to the **Path Effects** dialog and show controls for the **Boolean operation** path effect.

This is because we need to link to our square object by selecting and copying it. We then select our circle once more and click the **Link to item** button in the **Path Effects** controls, as shown in *Figure 12.3*.



Operand path:

Operation: union

☐ Swap operands

Fill type this: take from object

Fill type operand: take from object

Figure 12.3 – Linking the square to the Boolean operation on the circle

You can see that this has added the circle and square together. This is because, by default, the **Operation** type is set to **union** in the dropdown in the **Path Effects** dialog. As mentioned before, you can choose any other effect, such as **difference**, by changing the value of this dropdown, as shown in *Figure 12.4*.

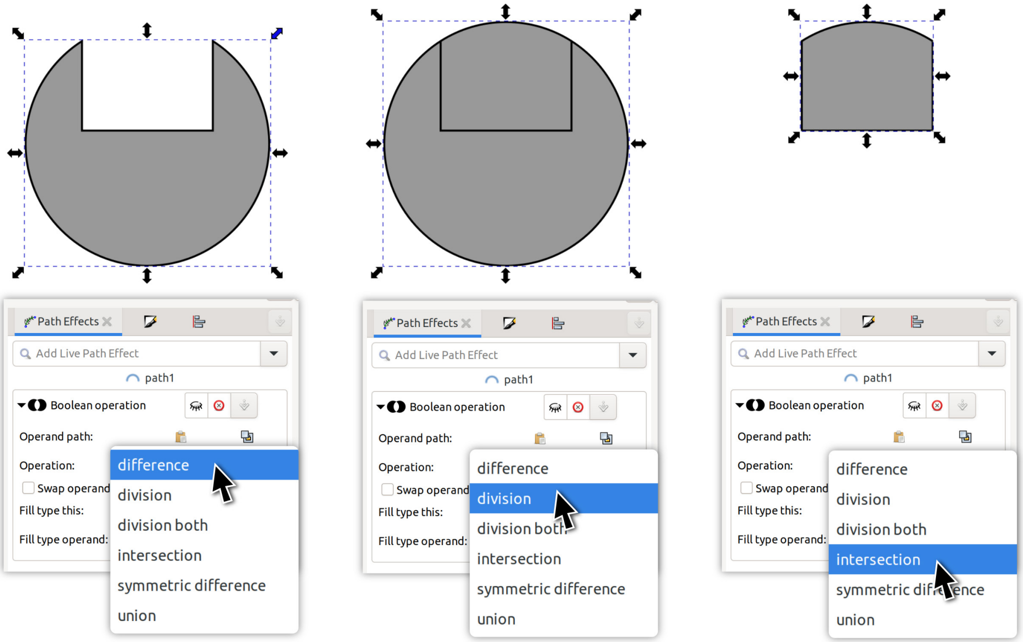


Figure 12.4 – Changing to different Boolean effects

You can see that there are other options given for the **Boolean operation** path effect; however, we will not be covering all of them in this book. Indeed, we could spend another whole book meticulously going through *just* path effects and all their functions.

What I'll do in the next section instead is show you some of my personal favorites – the ones I find most useful – and explore how path effects work in general. Then, you can explore the many options available and pick your own favorites.

Using Path Effects with single-path objects

So, sure, Booleans are great and all, but what if we want to round the corners of a combination shape that we've already made? Fortunately, there's a variety of path effects that work on single-path objects and allow us to do things based on the nodes. Let's say our client wanted us to make a llama out of just rectangles for her Boxy Llama Coffee company.

After doing as requested, she comes back and wants some of those corners rounded. We can use the **Corners** path effect to do this on a corner-by-corner basis. We simply select our boxy llama path and add a **Corners** path effect (**Path** > **Path Effects…**), by clicking the + button in the **Path Effects** dialog as before. This time, however, we will select the **Nodes** tool, which will show us special handles that the path effect has added for us to drag out and round those corners, as shown in *Figure 12.5*.

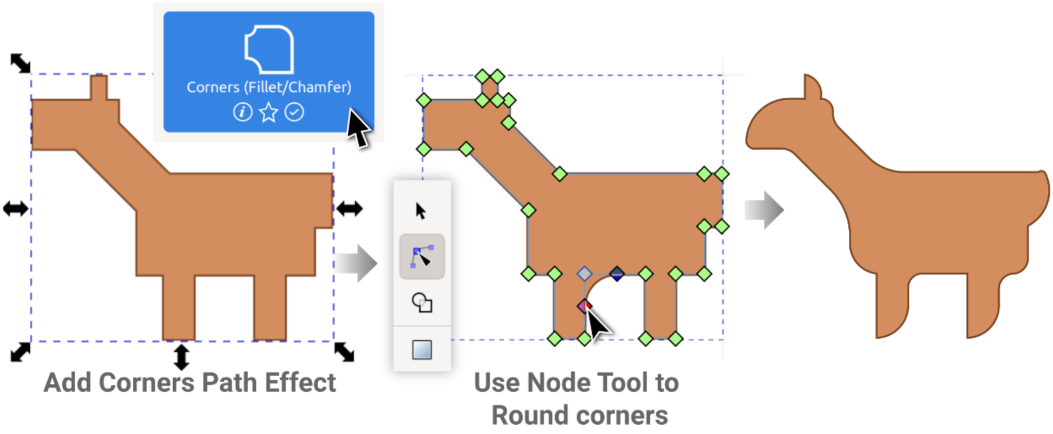


Figure 12.5 – Using the Corners path effect to round the corners of our llama

Since we didn't round all the corners, the llama still looks nice and boxy, but now looks more designed with those nice, rounded edges. Neat, eh?

Of course, to use this effect, we have to convert our object to a path made of nodes. But this isn't the case with every path effect. Some require paths, while others are fine with objects, or even a group of objects. If something's not working as you expect, you might want to try converting the object to a path first.

Take, for example, the **Bend** path effect. We'd love to use it to bend our title text, but at the time of writing, **Bend** only works on paths. We must, therefore, convert our text object to a path by selecting **Path** > **Object to Path** for **Bend** to have an effect, as shown in *Figure 12.6*.

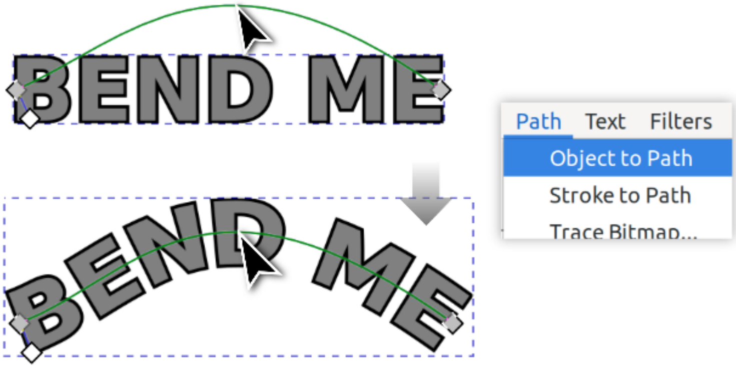


Figure 12.6 – Converting the text object to a path to make the Bend path effect work

We now know a few ways to use path effects on single objects, but what if our object has more than one part to it? As soon as we select more than one object, we cannot add a path effect. However, we can add a path effect to a single group of objects, as we will see in the next section.

Using Path Effects with groups

As mentioned previously, you can only add a path effect on *one* selected object. If you select more than one, the **Add path effect** button in the **Path Effects** dialog will be grayed out. However, some path effects will work on a group of objects, which they treat as a single object.

The **Rotate copies** path effect is a good example. Say, for example, you have a mouse eating some cheese, and you'd really like to turn that mouse into a pinwheel of mice eating the cheese. The mouse parts are all grouped together, so you can simply select the group and add a path effect, scrolling down to find the **Rotate copies** path effect and selecting it. The results of doing this are shown in *Figure 12.7*.

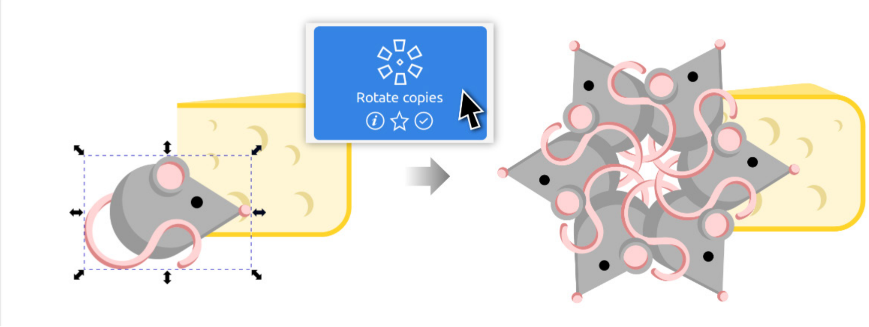


Figure 12.7 – Adding the Rotate copies path effect to the mouse

This indeed does make a pinwheel of mice. However, most of them are facing the wrong direction to be eating our cheese. We need to change the origin point around which they are rotated. We can do this on the canvas by selecting the **Node** tool and moving the origin handle, as shown in *Figure 12.8*.

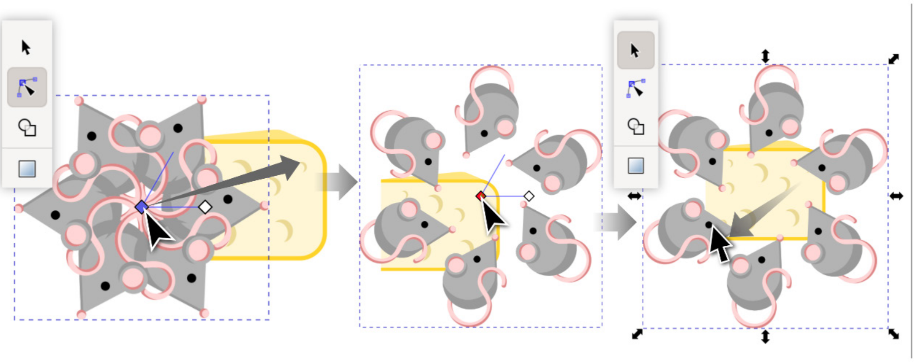


Figure 12.8 – Moving the Rotate copies origin with the Node tool, and then moving the mouse group with the Select tool

Generally speaking, the on-canvas controls for path effects are accessible via the **Node** tool, just like the rectangle and ellipse handles are for those special objects. Since the object is a group, it's also possible to double-click into the mouse group and edit parts of it.

This is particularly useful if we want to, say, change the expression of the mice. We just double-click on the mouse group to get into the group and then use the **Node** tool to change the shape of the eye, for example. As you can see from *Figure 12.9*, we then have a bunch of happy mice, enjoying some particularly good cheese.

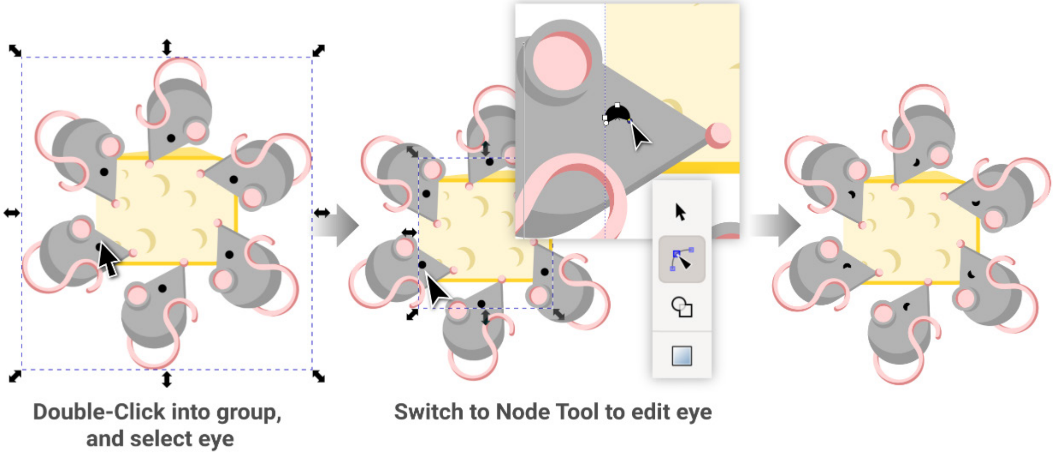

Double-Click into group, and select eye
Switch to Node Tool to edit eye

Figure 12.9 – Double-clicking into the group to select and change shapes inside

It's worth noting that you can change the number of mice, whether the copies are mirrored during rotation, and so on. These and other features make the **Rotate copies** path effect one of my favorites for quick, sunburst-style rotational design work.

This is a really great time-saver, but what if we want to take our ring of mice and bend it like we did our text? Turns out we can do this easily by stacking path effects.

Stacking Path Effects

You may have noticed that the **Path Effects** dialog looks a bit like the **Layers and Objects** dialog that we saw in the previous chapter. As we add effects, we can see they stack up in a similar fashion. Using our mice as an example, we could add a **Bend** path effect to bend our whole ring of munching mice, as shown in *Figure 12.10*.

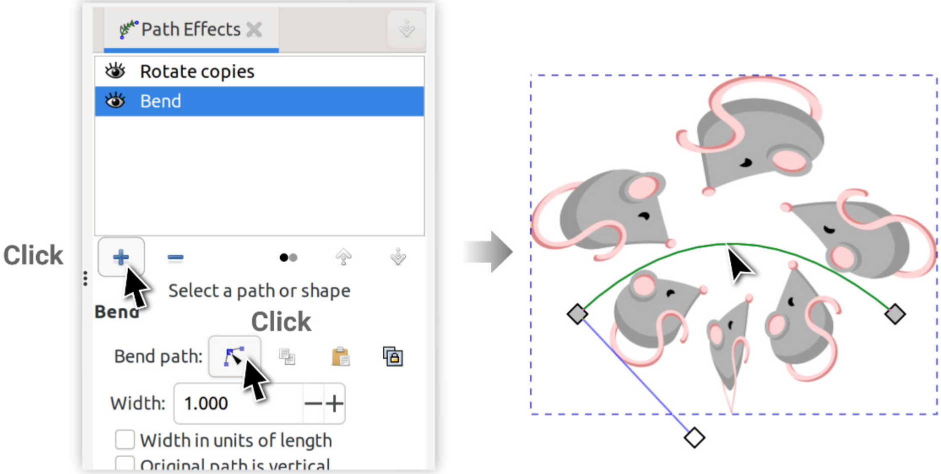


Figure 12.10 – Stacking a Bend path effect with a Rotate copies path effect

It's a bit more likely that we'd want to bend a single mouse rather than the entire rotational group, however. To do this, we simply need to move the **Bend** path effect above the **Rotate copies** path effect.

To do this more easily, collapse **Bend** by clicking on the little arrow on the left side of the bar, and then click-drag the handle on the right side to move the entry to the top of the list. You will see a small colored indicator at the top of the **Rotate copies** entry, which shows you where the **Bend** path effect will be dropped when you release the mouse button. Then, we're able to bend the individual mouse, as shown in *Figure 12.11*.

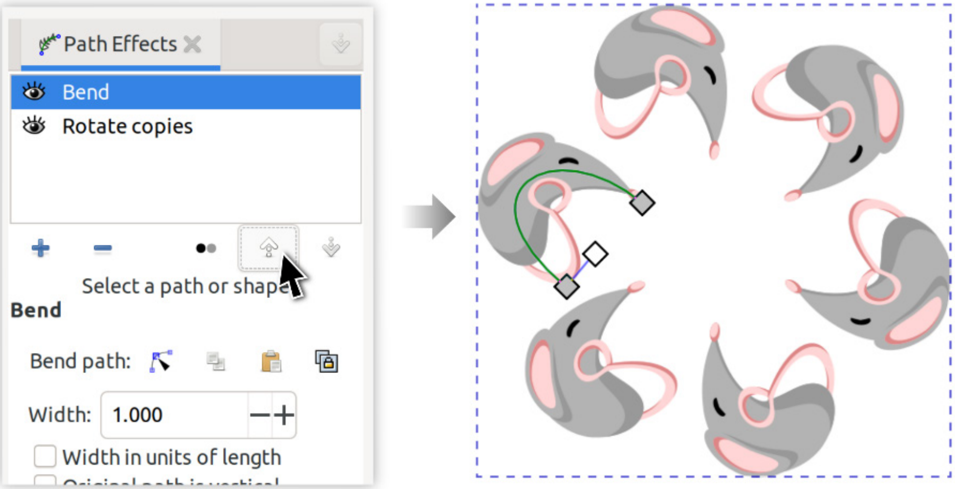


Figure 12.11 – Moving the Bend path effect before the Rotate copies path effect and editing the curve of the mouse

As you can see, Inkscape applies the path effects in top-down order, starting with the first path effect and ending with the last path effect at the bottom of the stack, giving us control over the order of the path effects.

This is a very powerful tool for getting the effects we want. Note how our original geometry remains editable, even though the results are very different than our original. This is the power of non-destructive workflows in Inkscape.

Limitations of Path Effects

Path effects are under constant development to both improve existing effects and add requested features. New path effects are added typically with every release, so the following limitations will likely change in the future:

- Path effects can currently only be applied to one object at a time.
- Path effects that modify geometry (such as bend, perspective, and so on) will not affect pixel/ bitmap images imported into Inkscape.
- Path effects that modify path geometry will not work on text objects. This can be worked around by converting the text object to a path first by selecting **Path** > **Object to Path**.
- Path effects will not show up in SVGs loaded into web browsers.

You can, however, apply path effects to your geometry by choosing **Path** > **Object to Path**, which will apply all your path effects and give you geometry that you can then edit normally. Moreover, this gets around the problems of viewing an SVG in a web browser, since there are no more path effects on the object after the object-to-path conversion.

Try it yourself

In this exercise, we're going to use a few of the many path effects to make a quick logo for an imaginary chainmail jewelry company. We will then stack some more path effects to make the logo appear as if it's on a flag, waving in the wind, as shown in *Figure 12.12*.

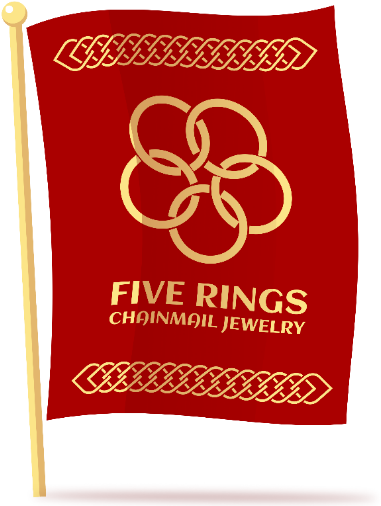
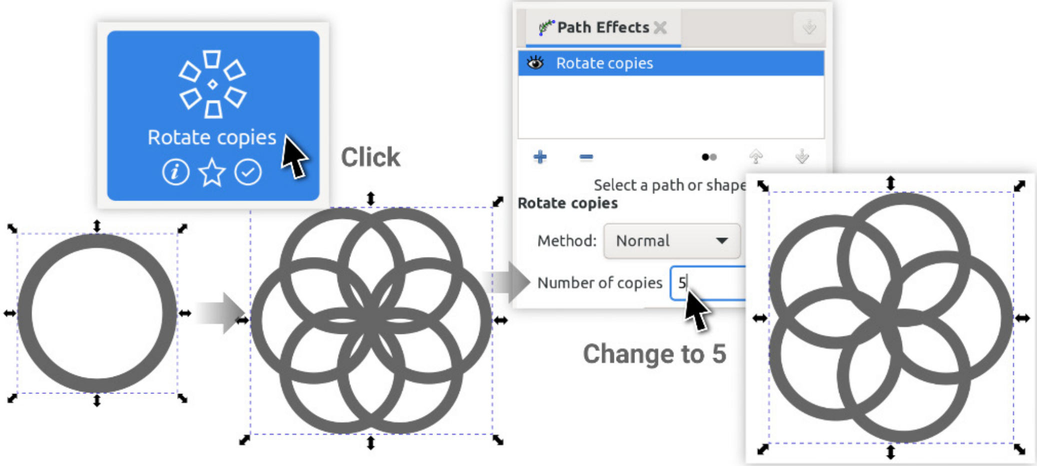


Figure 12.12 – The logo and flag for Five Rings Chainmail Jewelry

Normally, this would be a pretty big task, but with path effects, we can have our logo sorted in no time! Let's start with the rings:

1. We can just make a single circle, with a thick stroke, and then choose **Path** > **Path Effects…** to call up our trusty **Path Effects** dialog. We then hit the + button to add a **Rotate copies** path effect, changing **Number of copies** from the default of 6 to 5, as shown in *Figure 12.13*.



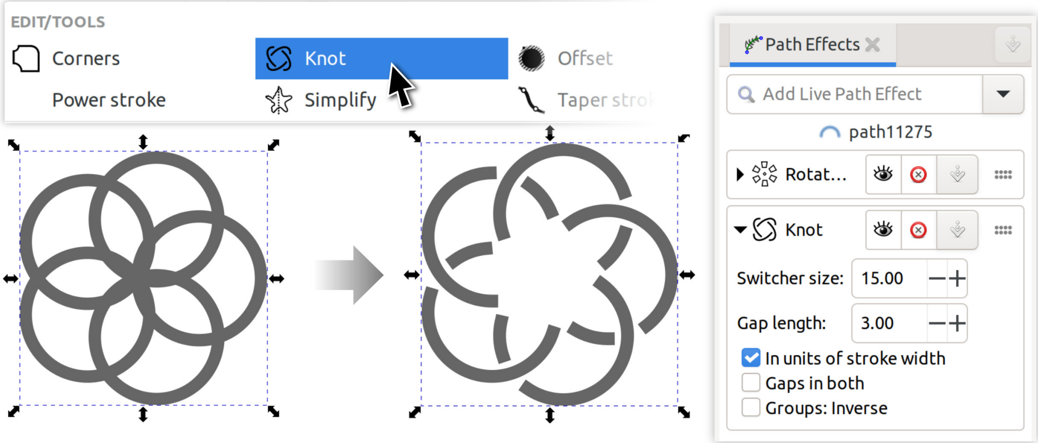
Rotate copies

Method: Normal

Number of copies 5

Figure 12.13 – Adding a Rotate copies path effect and changing the number of copies

2. This gives us our five rings, but how do we style them to look as if they are interlinked? We can simply add the **Knot** path effect after our **Rotate copies** path effect. We do this, again, by hitting the + button, and then we choose **Knot**, as shown in *Figure 12.14*.



Switcher size: 15.00

Gap length: 3.00

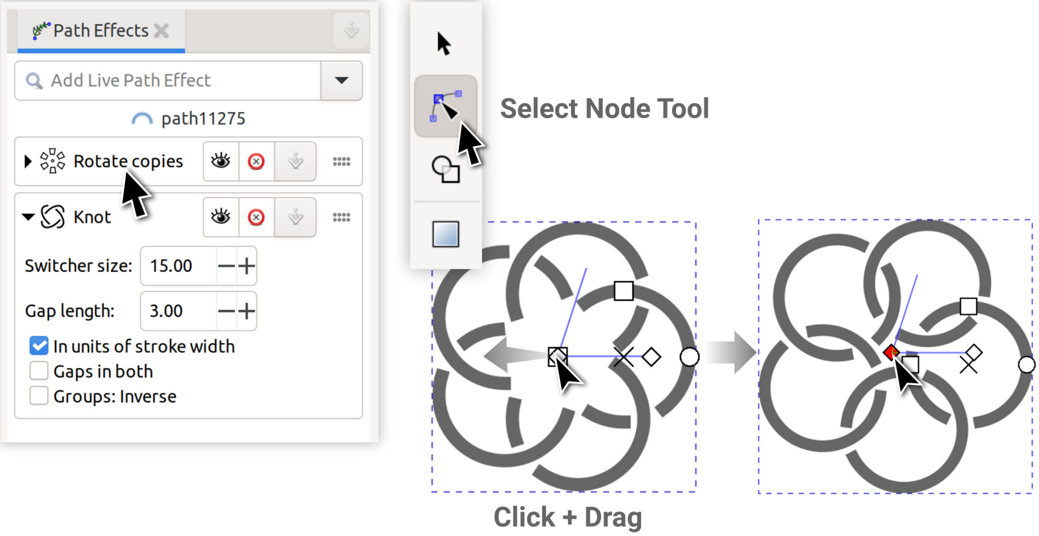
☑ In units of stroke width

□ Gaps in both

□ Groups: Inverse

Figure 12.14 – Adding a Knot path effect after the Rotate copies path effect

3. You can see that Inkscape is trying to figure out a knot pattern based on the five rings, but there's not enough overlap in the middle of our circles to make the center of the knot. We can fix this by selecting the **Rotate copies** path effect entry, selecting the **Node** tool, and using the **Rotate copies** handles to adjust where the rings overlap, as shown in *Figure 12.15*.



Switcher size: 15.00

Gap length: 3.00

☑ In units of stroke width

□ Gaps in both

□ Groups: Inverse

Figure 12.15 – Adjusting the Rotate copies path effect handles, using the Node tool

4. Feel free to play with the different handles to get the result you like best. We'll add the text and flag parts next. I've used a font called **Aclonica**, which you can get for free from Google Fonts, or you can just choose one of your own.

5. Using the **Text** tool, we'll type the name of the company as `Five Rings`, and then add another text object with the subheading `Chainmail Jewelry`. Then, we make the flag parts out of rectangles and circles. Arrange your text, the three-rings logo, and the flag parts as shown in *Figure 12.16*.

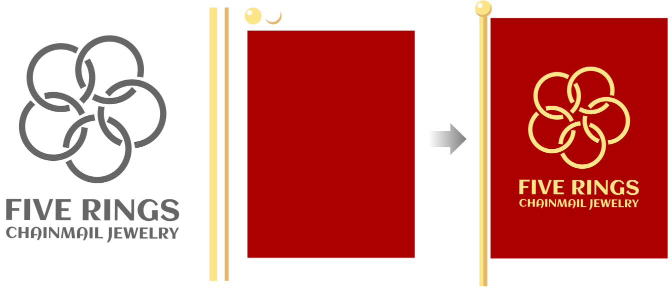


Figure 12.16 – Combining text, logo, and shapes to make a flag

6. We'll now make some more Celtic-style knotwork for the top and bottom of our flag. Starting with a square, we can rotate it 45 degrees so that it's a diamond shape. We'll then choose **Path** > **Object to Path** to convert the square to a path with nodes.

7. Once we've done this, we can add a **Corners** path effect and round the top and bottom corners (again, using the **Node** tool). This process is shown in *Figure 12.17*.

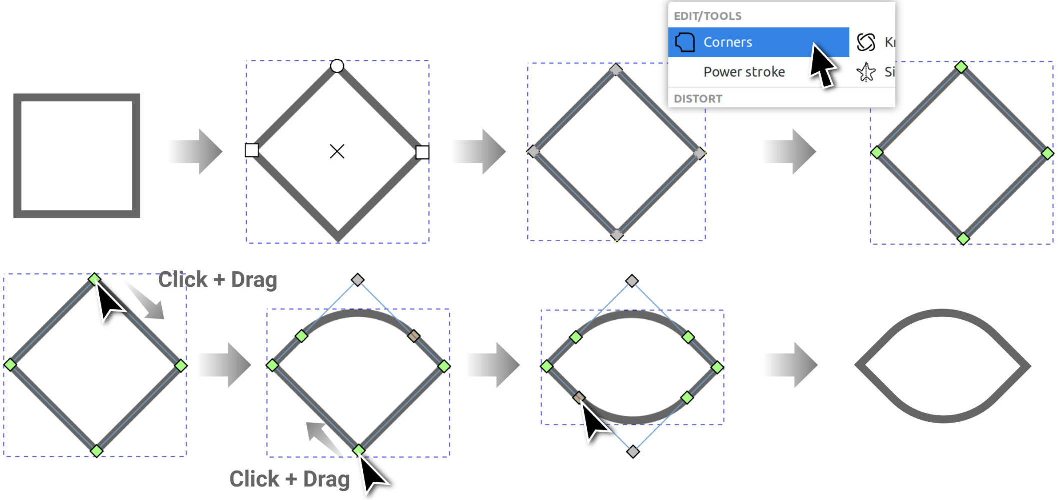
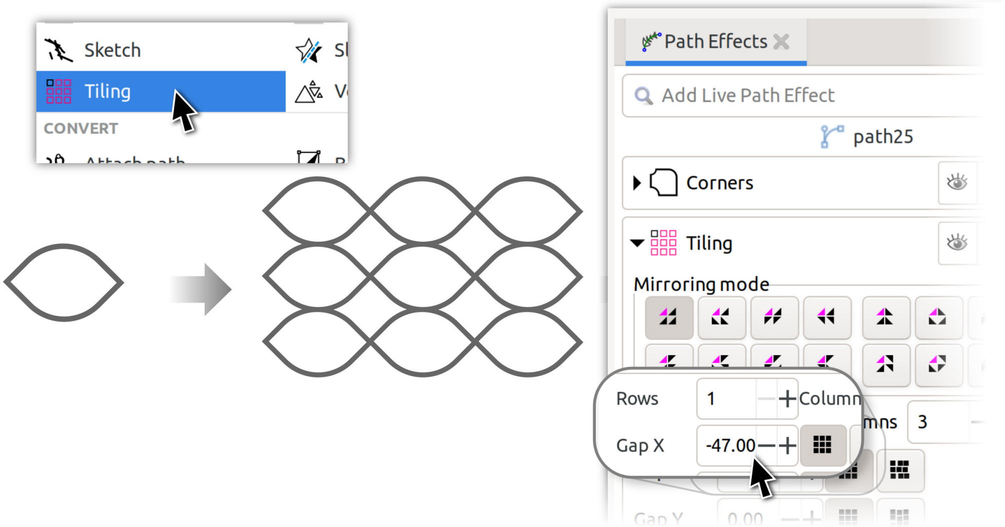


Figure 12.17 – Using the Corners path effect

8. Now that we have our base shape, we need multiple copies of it, like with **Rotate copies** but in a straight line. For this, we can use the **Tiling** path effect, set the rows to `1`, and reduce the **Gap X** value to a negative number so that the parts overlap. This is shown in *Figure 12.18*.



Rows 1

Columns 3

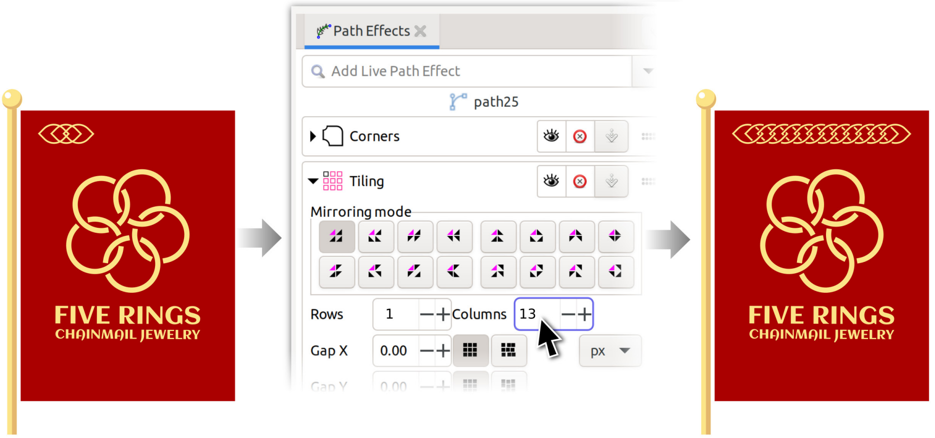
Gap X -47.00

Gap Y 0.00

Figure 12.18 – Adding a Tiling path effect and changing the Rows and Gap X values

In my case, the **Gap X** value was `-47.00`, but depending on how large you made your original loop, the value may be different.

9. We can now increase the **Columns** count in the **Tiling** path effect options to make more overlapping copies and add the same **Knot** path effect we used in *step 2*. The results are shown in *Figure 12.19*.



Rows 1

Columns 13

Gap X 0.00

Gap Y 0.00

Figure 12.19 – Increasing the Columns value on the Tiling path effect, and then adding a Knot path effect

10. From here, we can just copy/paste the line of knots to the bottom. We will also draw a dark-red rectangle shape over the center of the flag and add a gradient to make our shadow. We will also tilt the flagpole a little in preparation for warping our flag to make the graphic a little more interesting. This process is shown in *Figure 12.20*.

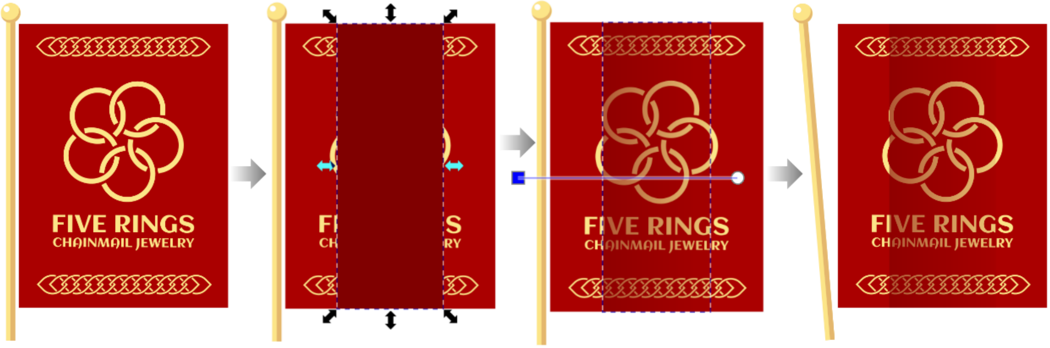


Figure 12.20 – Adding a shadow to the flag and tilting the flagpole

11. Before we can warp the flag, we need to select the red flag rectangle and the shadow we just made and use **Path** > **Object to Path**. This will let Inkscape curve the sides of the rectangle paths – something that is not possible with just rectangle objects. Once we have done this conversion, we can group all the banner parts together and add a **Lattice Deformation 2** path effect, as shown in *Figure 12.21*.

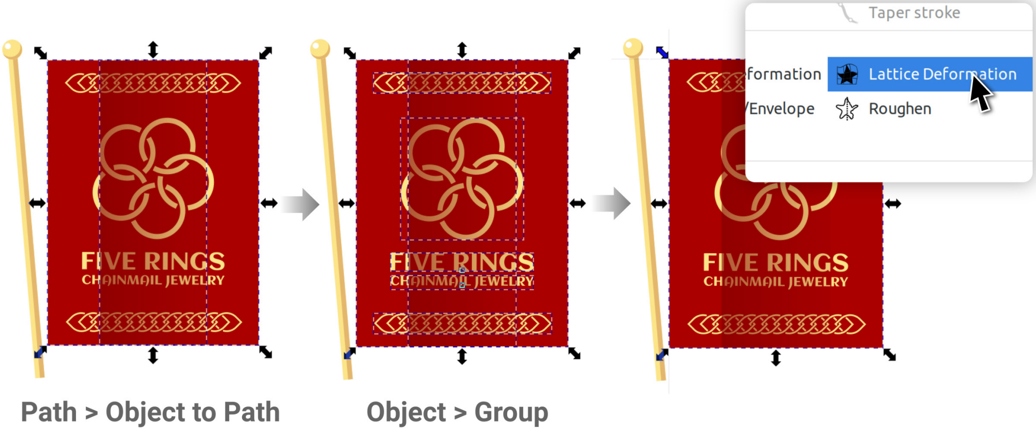


Figure 12.21 – Converting rectangle objects to paths, grouping all banner parts, and adding Lattice Deformation 2

12. From here, we can simply switch to the **Node** tool, and Inkscape gives us points on a grid that we can use to warp our flag around into a pleasing shape. I've done mine as shown in *Figure 12.22*, but feel free to experiment with your own flag shapes.

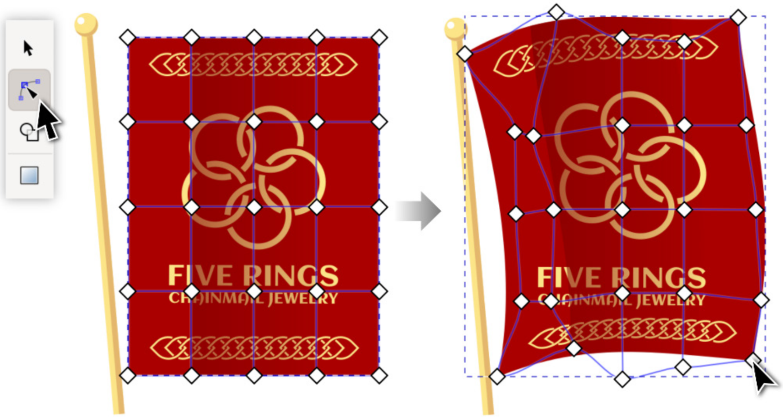


Figure 12.22 – Distorting the image by using the Node tool and the Lattice Deformation 2 grid

Note that since we have not converted the text objects to paths yet, the lattice deformation isn't affecting the shape. There are a few ways to preserve our text for later changes. For example, we could simply tilt both text objects a little to make it look as if the deformation is affecting them, or try our luck with putting the text on a path, as we did before.

13. We could also group both text objects together, make a copy of the group, and hide it for later changes in the **Layers and Objects** dialog. Then, we can use **Path** > **Object to Path** on our original text objects, and the **Lattice Deformation 2** path effect will then properly distort the text, along with all the other objects in our flag banner group.

Regardless of how we handle it, we can then add a thin black ellipse with a blur on it to the base of the flag graphic for a quick shadow to complete our graphic, as shown in *Figure 12.23*.

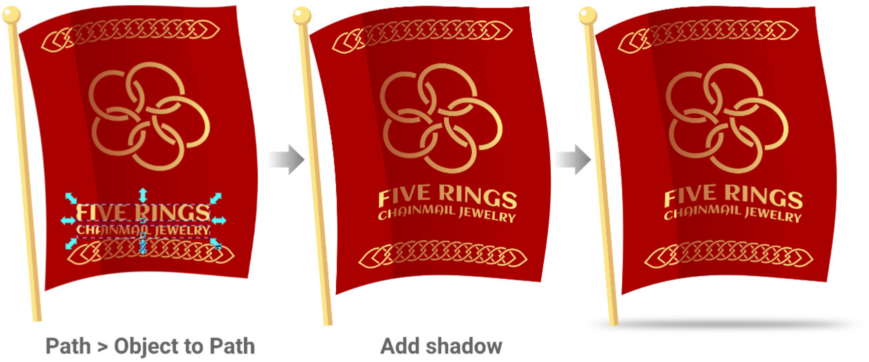


Figure 12.23 – Converting text objects to paths and adding a base shadow under the flag

Summary

Even though we certainly learned a lot about path effects in this chapter, we only touched the tip of the iceberg with all the different things you can do with path effects. We learned that we could add, stack, rearrange, and modify path effects for non-destructive workflows.

Now that you know how they work, I recommend exploring some of the many options in your projects. Certainly, we've seen all kinds of ways path effects can save us time! In the next chapter, we'll explore some other neat features of Inkscape to achieve specific effects, with **Filters** and **Extensions**.

13

Filters and Extensions

Up to this point, we've had a taste of some of Inkscape's time-saving convenience items, such as clones and path effects. In this chapter, we'll showcase a few more types of power tools that you can use in your Inkscape workflow.

In this chapter, we will cover the following topics:

- Using filters
- Making custom filters
- Using extensions

Technical requirements

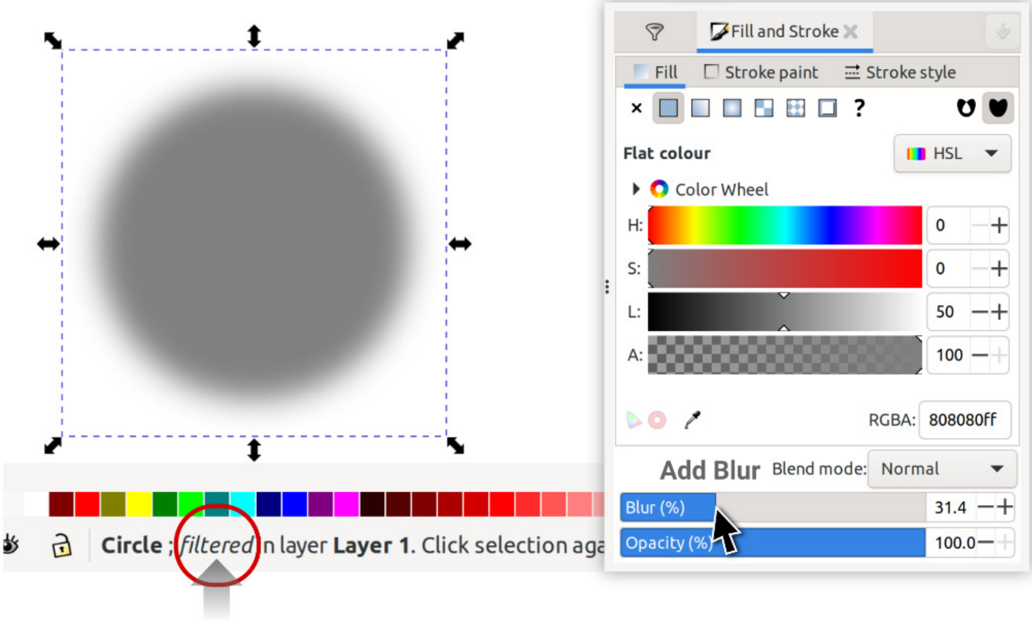
You can download the practice files for this chapter from GitHub at

`https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape/tree/main/Chapter13`

Using filters

In the preceding chapter, we saw how easy it was to apply and stack path effects on our shapes to get specific results. Filters offer a similar sort of advantage but can do much more beyond modifying vectors. Filters can make bitmap-like effects, some of which are hard or impossible with vector shapes alone.

We've actually used filters in a roundabout way already. For example, when we set the **Blur slider** value in the **Fill and Stroke** dialog, we're actually telling Inkscape to apply a **Gaussian Blur** filter to the object. We can confirm this by looking at the status bar to see that the selected object is now filtered, as shown in *Figure 13.1*.



Circle ; filtered n layer Layer 1. Click selection aga

Flat colour: HSL

H: 0

S: 0

L: 50

A: 100

RGBA: 808080ff

Blend mode: Normal

Blur (%) 31.4

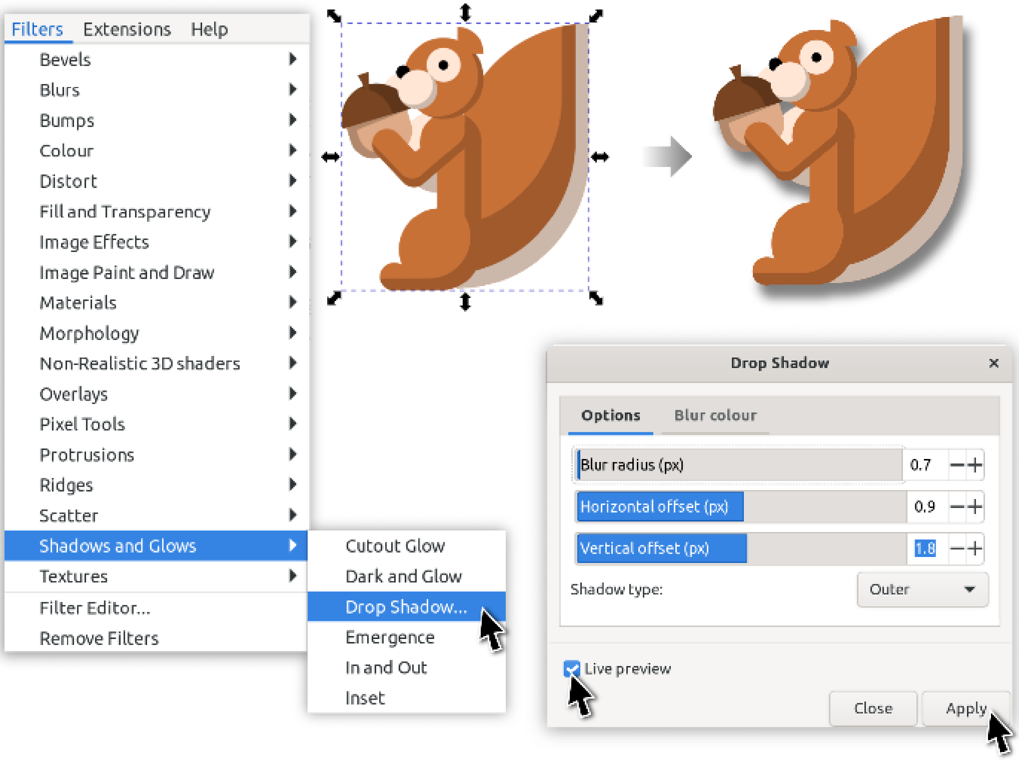
Opacity (%) 100.0

Figure 13.1 – Adding blur in the Fill and Stroke dialog in turn adds a Blur filter to the object

This is also how the **Blend** mode object (also in the **Fill and Stroke** dialog) works. So, what other things can we do with filters? Well, lots of things as it turns out!

A simple look in the **Filter** menu in the menu bar will display many categories of preset filters that can be applied. For example, we can choose **Filters** > **Shadows and Glows** > **Drop Shadow...** and add a drop shadow to our selected object or group.

Figure 13.2 shows us how to add a **Drop Shadow Filter** to the squirrel, using the pop-up dialog to set the specifics for the filter. Make sure you check the **Live preview** checkbox to see what you're doing, and then click the **Apply** button to apply the results.



Blur radius (px) 0.7

Horizontal offset (px) 0.9

Vertical offset (px) 1.8

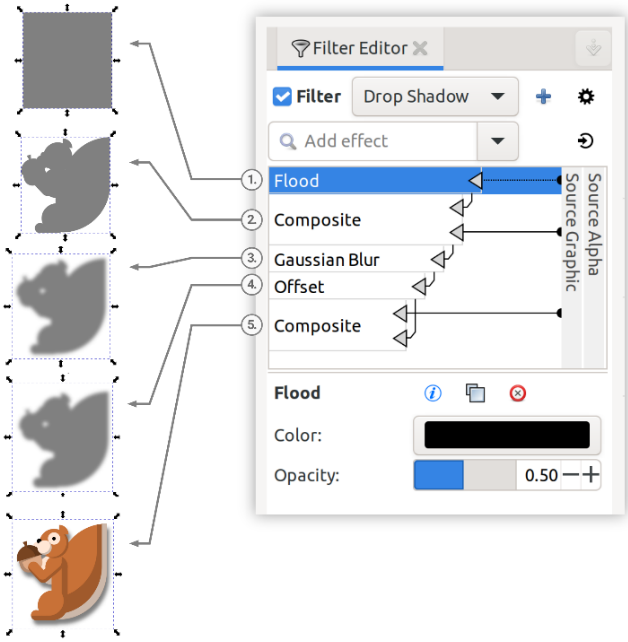
Shadow type: Outer

☑ Live preview

Figure 13.2 – Adding a Drop Shadow filter to a squirrel illustration group

You'll see that if you make changes to the objects in the squirrel graphic, the drop shadow automatically updates when you move things around. This can be quite a time-saver when working with complex images!

Now, you may be asking, *what does the Drop Shadow filter actually do?* To find out, we just need to open up the **Filter Editor** dialog by choosing **Filters** > **Filter Editor....** This will show us what's inside the filter, and even better – it lets us change things! *Figure 13.3* shows the **Filter Editor** dialog, with our filtered squirrel selected. I've labeled the filter recipe from top to bottom to show what the results are at each step in the recipe.



Filter Editor

☑ Filter Drop Shadow

Add effect

1. Flood
2. Composite
3. Gaussian Blur
4. Offset
5. Composite

Source Graphic
Source Alpha

Flood

Color:

Opacity: 0.50

Figure 13.3 – The Filter Editor dialog, and the results at each step of the Drop Shadow filter recipe

At first, this looks a bit complex, with all the arrows going around and connecting parts. However, once you think of this dialog as a top-down recipe of effects to make the drop shadow, it starts to become clearer what's happening.

Let's go through the parts of the **Drop Shadow** filter, starting at the top of the dialog:

1. **Flood**: This effect takes the source graphic (in our case, the squirrel objects) and floods (fills) the rectangular area of the whole graphic with a color. In this case, the color is black, with an **Opacity** value of 50%, as you can see from the bottom of the **Filter Editor** dialog, which shows information for the currently selected effect.
2. **Composite**: Note the small arrow coming from the **Flood** effect, and another one coming from the source graphic. This effect combines the previous effect (**Flood**) with the source graphic, thus cropping the black rectangle to the shape of the squirrel.
3. **Gaussian Blur**: As the name suggests, this effect blurs the results of the first two effects, following the arrow from the previous effect in the recipe.
4. **Offset**: This controls the *x* and *y* positions of our blurred shadow relative to the original, so we can make the shadow go off-center in both directions as we like.
5. **Composite**: This just composites the original squirrel on top of the shadow, completing the effect.

Note that if we decide we don't like that **Drop Shadow** filter anymore, we can choose **Filters** > **Remove Filters**, or uncheck the **Filter** checkbox at the top of the **Filter Editor** dialog.

If it's still not quite clear what's happening, don't worry – we're about to get our hands dirty and make our own filter using some of these same parts.

Making custom filters

Say, instead of a drop shadow, we'd like a filter that automatically shades the interior of one side of our shapes so that they look 3D instead of 2D. We can start with a new file and draw a plain circle, and then choose **Filters** > **Filter Editor** (with the circle still selected).

We are presented with a blank space and a familiar looking + button at the top. Click the + button, and also tick the **Filter** checkbox so that we can see the result of our filter. As you can see from *Figure 13.4*, this makes the circle vanish!

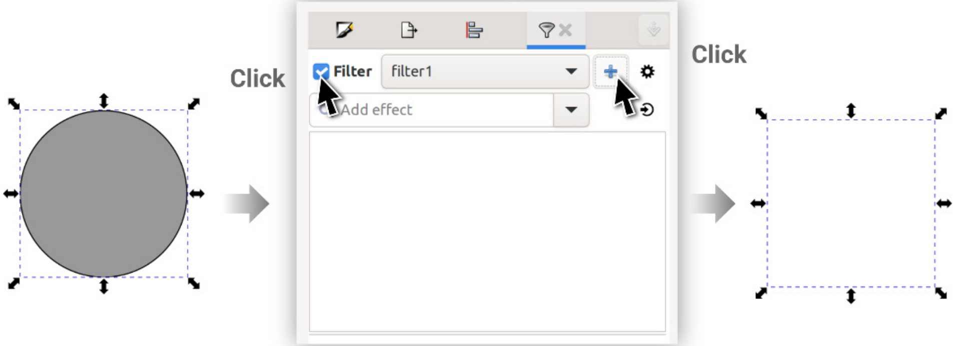


Figure 13.4 – Adding a filter and activating it with the checkbox makes the circle vanish

This makes sense though when you think about it; we've told Inkscape to filter the circle and give it nothing as a recipe, so it gives us nothing back! We could actually use this as a *vanish* effect if we really wanted to, but we already have **Object** > **Hide**.

Let's move along to adding our first effect. We do this by clicking the **Add Effect** dropdown and choosing **Flood** from the options. Immediately, we encounter the same square we saw earlier. We want to composite our original circle over that, so we can add a second effect by clicking once more on the **Add Effect** dropdown, this time choosing **Composite**.

But nothing happens. That's because both of the arrows going into **Composite** are coming from the **Flood** effect, so Inkscape is just compositing two copies of the same square, one over another. We need to connect the first arrow to the source graphic instead. We do that by click-dragging the head of the arrow to pull out a line and drag it over the **Source Graphic** bar to connect it, as shown in *Figure 13.5*.

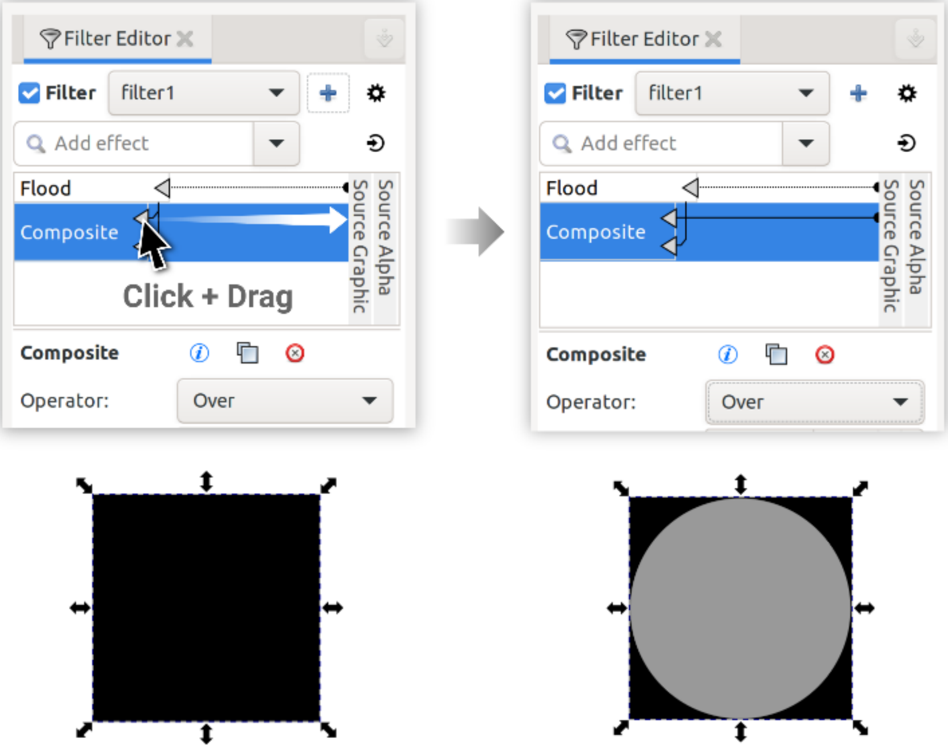
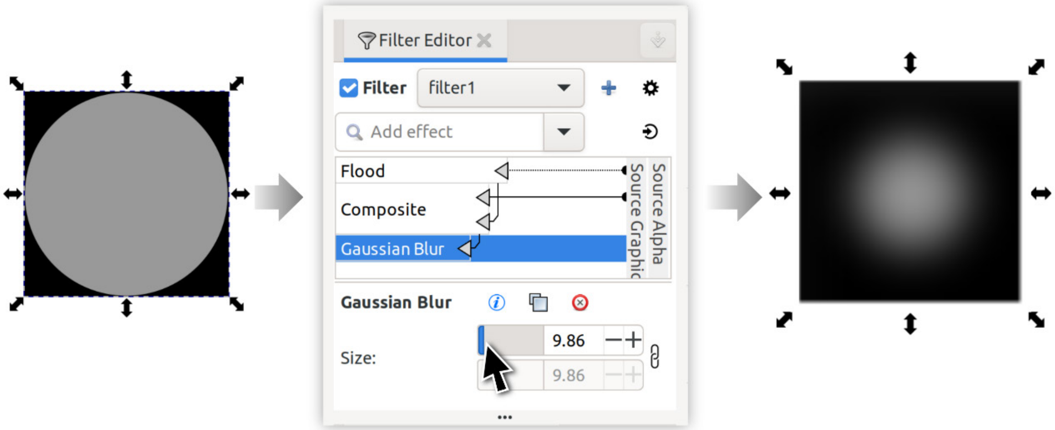


Figure 13.5 – Connecting the first Composite arrow to the Source Graphic bar

As you can see, this has the effect of overlaying the original circle on top of the **Flood** square, which will provide some darkness for our shadow. Next, we add a **Gaussian Blur** effect in the same way as the other two effects and increase the **Size** value to `9.86` (or something similar).

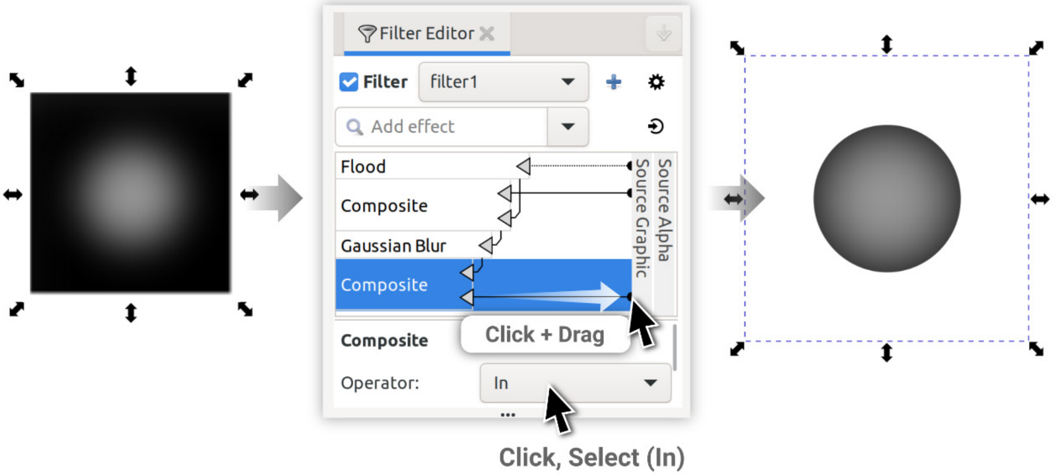
You can see from *Figure 13.6* that Inkscape becomes blacker on the outside to fill the blur area as it blurs.



Size: 9.86 9.86

Figure 13.6 – Adding a Gaussian Blur effect to the filter

Since we now have plenty of black shadow, we need to combine the result into our original circle shape by adding a **Composite** effect as before but, this time, linking the source graphic to the second input by click-dragging the second arrow to the **Source Graphic** bar. This time, we will also change the **Operator** value to `In`. As you can see from *Figure 13.7*, this clips the shadow into the circle shape.



Click + Drag

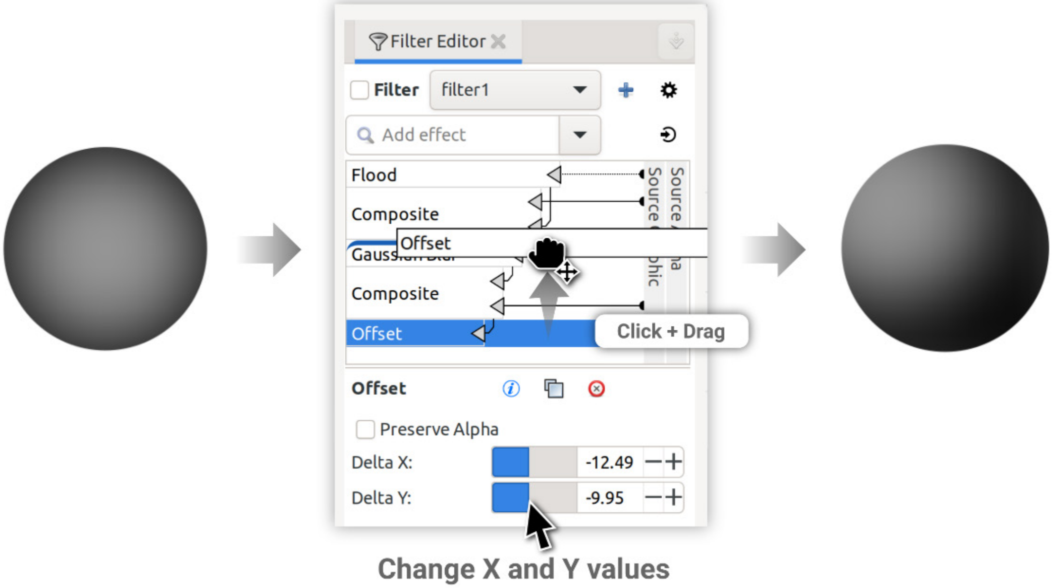
Operator: In

Click, Select (In)

Figure 13.7 – Adding a Composite effect to crop the shadow to the circle

That's given it a nice, even shadow all around, but we'd like our shadow to be offset so that more of it is on the lower-right side. This will make it look more like a ball, with a light source in the upper left. We can do this by adding an **Offset** effect before the **Gaussian Blur** effect.

To do this, we repeat what we did it in the **Layers and Objects** dialog. Simply add it as we did with the previous effects, and then click-drag the bar up so that the drop indicator shows a line between the first **Composite** and **Gaussian Blur** effect bars. Then, we can change the **Delta X** and **Delta Y** values to offset the shadow where we want it. This process is shown in *Figure 13.8*.



□ Preserve Alpha
Delta X: -12.49
Delta Y: -9.95

Figure 13.8 – Moving the Offset effect between the first Composite and Gaussian Blur effects

You may be asking at this point, *why didn't we just do this with a radial blur?* Sure, we could achieve the same effect that way for a circle, but the real magic comes when we modify the shape. As you can see from *Figure 13.9*, when we change the shape of the circle, the filter automatically recalculates and updates our shadow!

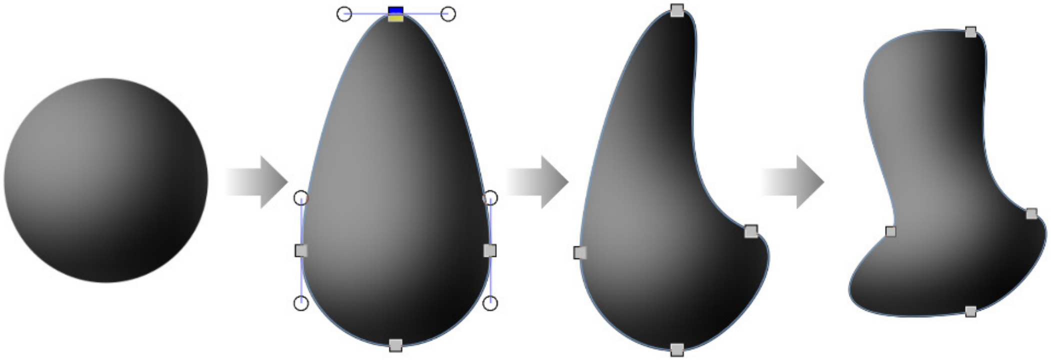


Figure 13.9 – Changing the shape to see the filter updating

Moreover, you can change the color fill of the object and the filter will still apply over it. Remember that before we had to deal with complex clone setups to achieve this same effect. Now, we have a filter we can add to any shape we want, just by bringing up the **Filter Editor** dialog, choosing our filter as **filter1**, and ticking the **Filter** checkbox.

Of course, we want to name our filter something more memorable than *filter1*. We can do that by clicking the **filter1** dropdown and then double-clicking the name, renaming it `3D Shadow`, and pressing the *Enter* key on the keyboard to apply, as shown in *Figure 13.10*.

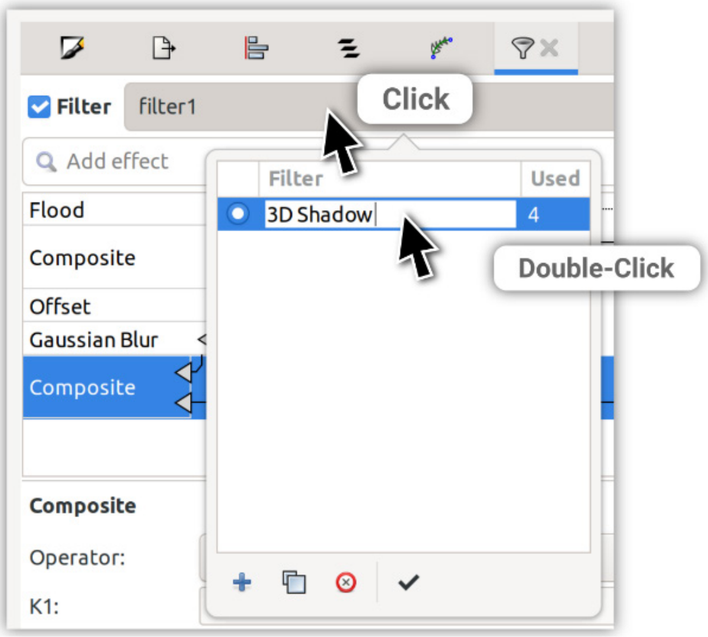


Figure 13.10 – Changing the name of filter1 to 3D Shadow

Of course, like path effects, there are lots of things to explore – so many things that it would indeed take another book like this to cover everything that is possible. However, we now know the basics of how filters work, and even some advanced custom filter stuff.

I'll end this section by encouraging you to explore all the many different effects in the **Add Effects** dropdown, as well as the plethora of preset filters in the **Filters** menu. You can discover a lot of tricks just by looking into these options, as we did with the **Drop Shadow** filter!

Let's move along now to extensions, which provide us with a variety of other time-saving tricks to work quickly in Inkscape.

Using extensions

Extensions are actions that are not included in Inkscape's core code but were written as extras by the Inkscape community. They do all manner of things, which we'll explore in this part of the chapter. As in the previous section, we will not be going over all of them but, rather, give you an overview of what's there at your disposal.

Unlike filters, extensions are single-use actions that either change or add objects to your canvas. These are destructive rather than the previous non-destructive changes – once you run an extension, the only way to get your original shape back the way it was is to use the **Edit** > **Undo** option. For this reason, I also recommend saving your work before using an extension.

Since there are so many different kinds of extensions, they are grouped functionally into categories, which you will see if you expand the **Extensions** menu in the menu bar at the top of Inkscape's interface, as shown in *Figure 13.11*.

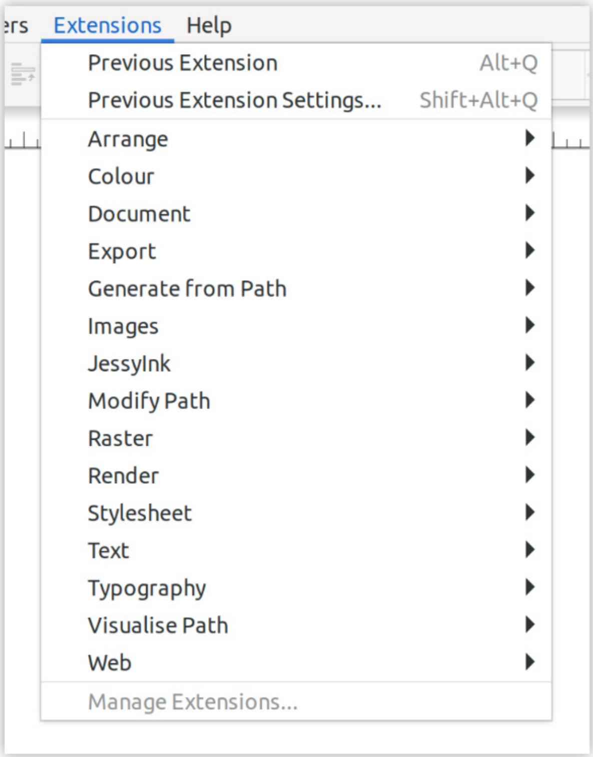


Figure 13.11 – The Extensions menu with categories

The categories are as follows:

- **Arrange**: Contains extensions that reorder the selected items in similar ways to those found in the **Arrange** dialog. Currently, there are only two extensions, **Deep Ungroup** and **Restack**, which, respectively, perform multiple ungroup actions at once and restack items in the **Layers and Objects** dialog, based on their positions on the canvas.
- **Color**: This handy set of extensions works only on vector shapes and adjusts the color fills in ways you might see in photo-editing software. **Greyscale**, **Desaturate**, **Negative**, and **HSL Adjust** (**Hue, Saturation, Lightness**) are some examples of the color transforms you can make with these extensions.
- **Document**: Extensions related to the document – for example, **DOC Info** brings up a text notification box with all the document information, which you can copy/paste as text.
- **Export**: Extensions for exporting to various formats not supported in the regular file menu – for example, **Plot** exports a file meant to be used on industrial cutting machines.
- **Generate From Path**: Extensions that take the selected object geometry and produce more geometry based on it – for example, **Interpolate** takes two selected shapes, such as a circle and square, and adds shapes between them that transform the circle into the square.

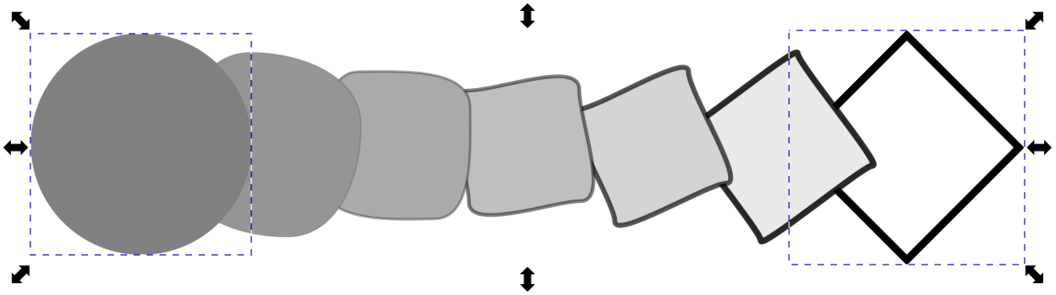


Figure 13.12 – Transformation of a circle into a square

- **Images**: Extensions that work with selected images in your document. For example, if you choose to link in an image instead of embedding it, you can use the **Embed** extension to embed it. Likewise, **Extract** will save a copy of an embedded image to the same folder as your document and convert it to a linked image.
- **JessyInk**: Contains extensions that help you make presentations with Inkscape SVGs, using a JavaScript library called **JessyInk**.
- **Modify Path**: These extensions add modes to your geometry and transform the paths in specific ways – for example, you can add nodes to your path in certain areas with the **Add Nodes** extension. Others, such as **Whirl**, attempt to warp your path around a central point.

- **Raster**: These extensions are for use on bitmap images and provide some photo filters, as you might find in photo-editing software. These are destructive operations, so it's a good idea to save your work before using them. Note that if your image is linked to a document, these filters will automatically embed it before the filter is applied.
- **Render**: This category is full of useful items that generate vector shapes – for example, you can generate a QR code (those blocky codes you scan with your phone camera) with **Extensions** > **Render** > **Barcode** > **QR Code**, standard (**Classic**) barcodes for your inventory, or even calendars you can print out!

  There's a lot of really neat stuff in here. I recommend spending some time just playing with them all to determine your favorites. You never know when they will save you a ton of time, and knowing what the options are is half the battle.
- **Stylesheet**: This will take objects on your canvas and output **Cascading Style Sheets** (**CSS**) code for websites. If you don't know what this is, don't worry – it's a tool for web developers.
- **Text**: You guessed it! This category contains extensions that change/modify text. Some nice options include being able to change your text to all capital letters with **UPPERCASE**, or if you are unlucky enough to get a whole paragraph in uppercase, **Sentence case** will make the start of each sentence a capital letter and the rest lowercase, saving you having to retype everything.

  You can also fill a textbox with example text using **Lorem ipsum**, which simulates how a paragraph would look. Just remember to replace it with your own text before printing! This is another category worth knowing and also exploring after each major release. At the time of writing, there's a new **Increment** extension being added that auto-numbers things. This is a personal favorite of mine, as I number sheets of samples a lot for clients.
- **Typography**: This category contains tools for working with and making your own SVG fonts. Font making is an entire discipline in graphic design, and I personally think it's completely awesome that you can make your own fonts in Inkscape!
- **Visualize Path**: Sometimes, you want more information about the path you're working on. Sure, you can see width and height information, but what if you want to measure the length of a curve or a circle? This category contains just the stuff you're looking for – **Measure Path**, for example, will display a curve measurement for your selected object or the distance around your circle.
- **Web**: Contains a variety of tools for making website stuff. You can even make interactive SVG-based presentations with **Interactive Mockup**, slice up your document to export the bits as PNGs with **Slicer**, or set JavaScript attributes for custom SVG-embedded scripts. If that's all Greek to you, then you can just write this category off as another set of web developer tools.
- **Manage Extensions**: Not finding what you want? This calls up a pop-up browser to search for, download, install, and create your own Inkscape extensions. It draws from those uploaded to `https://inkscape.org/` by the community and curated by Inkscape project members.

It's worth noting that you can write your own extensions in a programming language called Python. You can also download and install more extensions on the Inkscape website and upload your own for others to use. This is outside the scope of this book, however, as the topic demands another lengthy book to cover it properly.

If this overview of **Extensions** seems a bit conceptual, worry not – we'll use one when our journey continues in the *Try it yourself* section that follows.

Try it yourself

For this exercise, we're going to make ourselves a vanity graphic that includes a custom brushed metal filter and the use of the QR code generator to encode a scannable website address. This will look something like *Figure 13.13*.

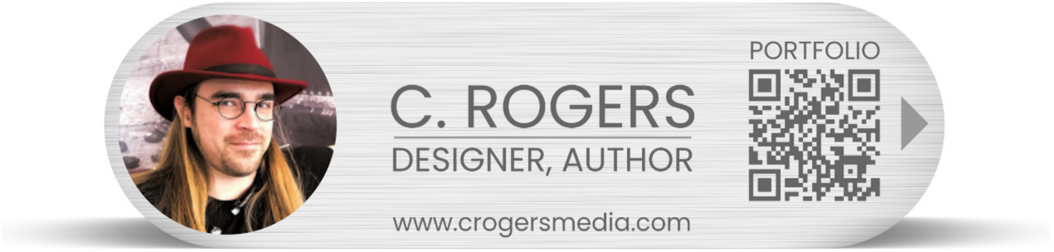


Figure 13.13 – A vanity graphic with a web link QR code

Obviously, you should make it your own – that's half the fun here! You can use any old picture of yourself off your device, and if you don't have an official job title, you can just make one up. Well, let's get into it!

1. The text and base shape should be nothing new. We've created quite a bit of text up to this point, so go ahead and add all the text bits on the canvas, as well as a shape you'd like to use. I've chosen a rectangle with corners rounded to form an elongated pill shape. These parts are shown in *Figure 13.14*.

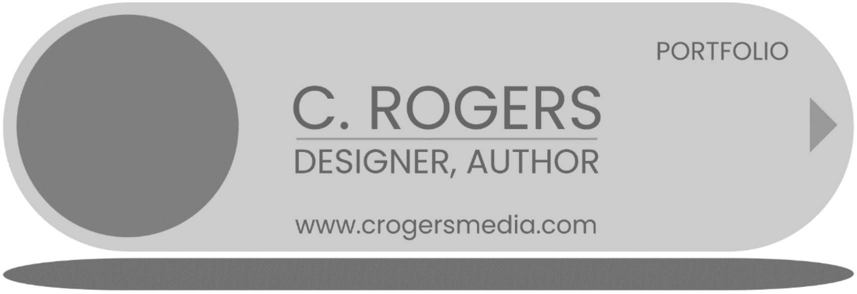


Figure 13.14 – Base shapes of the vanity graphic

2. Now, we get into the hard part – making that brushed metal filter. By default, Inkscape doesn't have one, so we will have to build it ourselves with the **Filter Editor** dialog > **Filters** > **Filter Editor…**. We will select the pill-shaped background and then create a new filter using the + button in the **Filter Editor** dialog, checking the **Filter** checkbox so that we can see the effect as we go along. We will then rename the filter `Brushed Metal` by double-clicking the **filter1** name in the dropdown, as shown in *Figure 13.15*.

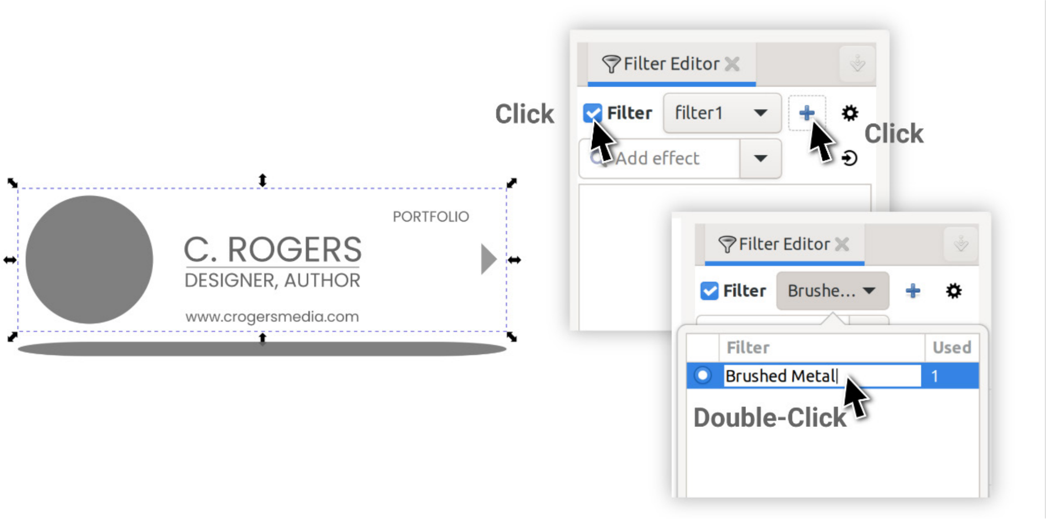


Figure 13.15 – Adding a new filter and renaming it Brushed Metal

3. The first step in our brushed metal filter is to generate a bunch of random spots. We can do this by adding a **Turbulence** effect. Simply select **Turbulence** from the **Add effect** dropdown in the **Filter Editor** dialog. This will immediately add a noise pattern to our shape area, as shown in *Figure 13.16*:

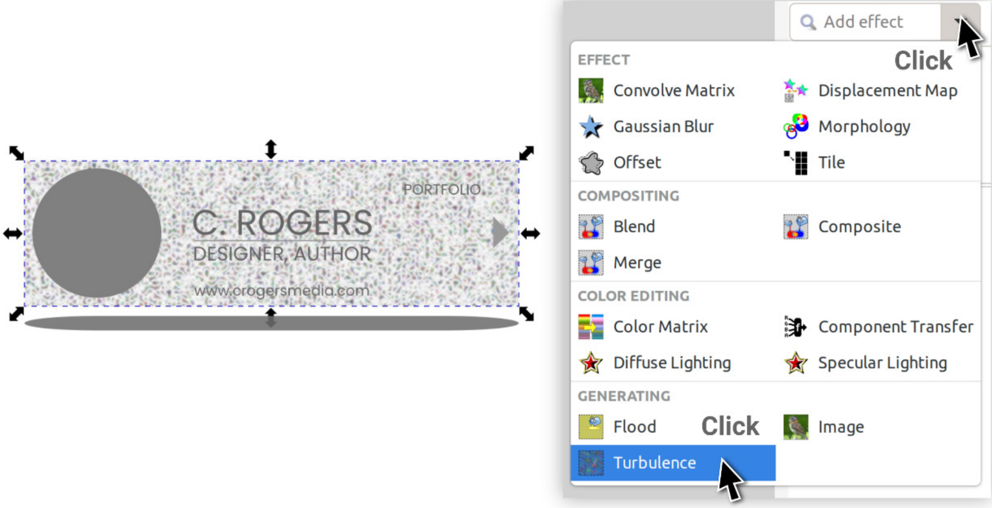
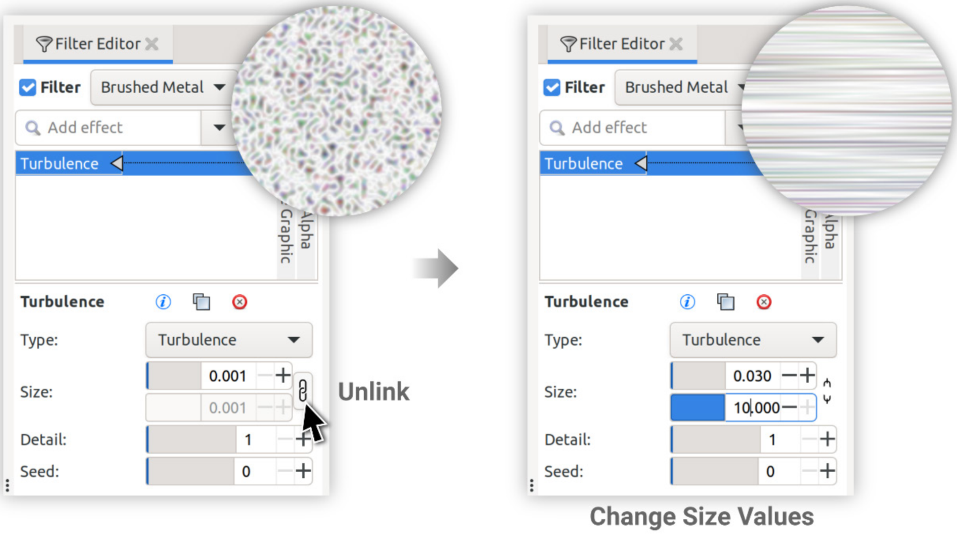


Figure 13.16 – Adding a Turbulence effect

4. The turbulence dots are a bit too big. We'd like something smaller to give our brushed metal a sand-blasted look. We can adjust the **Size** values, but instead of making them just smaller, we can unlink them and set the bottom value larger and the top value smaller to stretch out the dots in one direction. The values in *Figure 13.17* show my results, but you may need to experiment a bit with different values to get them to look the same.


Filter Editor

☑ Filter: Brushed Metal

Turbulence

Type: Turbulence

Size: 0.001 / 0.001

Detail: 1

Seed: 0

Filter Editor

☑ Filter: Brushed Metal

Turbulence

Type: Turbulence

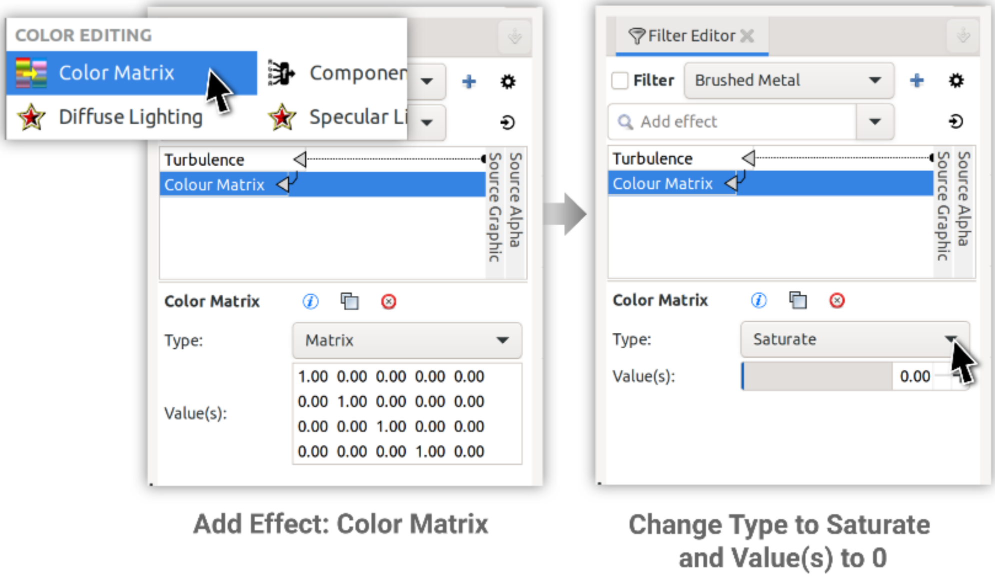
Size: 0.030 / 10.000

Detail: 1

Seed: 0

Figure 13.17 – Unlinking the Size values and adjusting them individually to stretch the Turbulence pattern

5. This looks pretty good already, but if you look closely, there's a bit of color there that we don't want. We will make it grayscale by adding a **Color Matrix** effect and then change **Type** from **Matrix** to **Saturate**, and **Value(s)** to `0.00`. as shown in *Figure 13.18*.



Type: Matrix

Value(s):

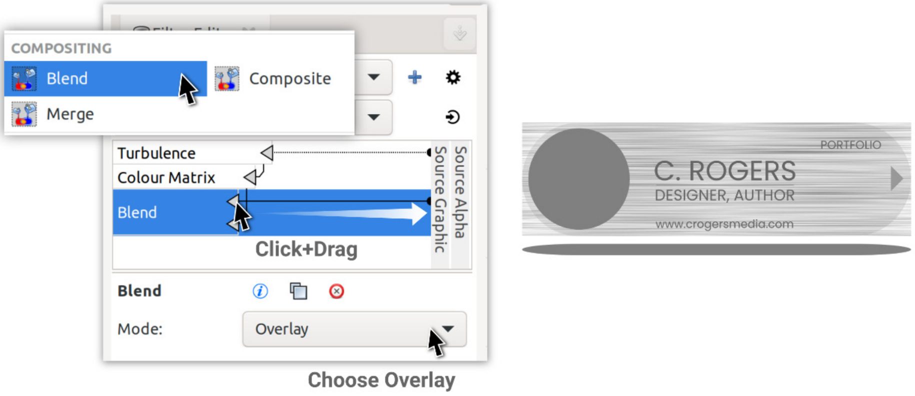
1.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00

Type: Saturate

Value(s): 0.00

Figure 13.18 – Adding the Color Matrix effect and adjusting values to make the texture grayscale

6. Because **Turbulence** replaces our fill color, we'd like to blend the source graphic with it to bring back the influence of our fill. This will let us lighten, darken, or even change the color, just by changing the fill color of the object. We can do this by adding a **Blend** effect, connecting the source graphic to the first node, and setting **Mode** to **Overlay**. As you can see from *Figure 13.19*, this will overlay our texture on top of the original color.



Mode: Overlay

Figure 13.19 – Adding a Blend effect to blend the texture with the source graphic

7. Then, all we have to do to clip the results to the original shape is to add a **Composite** effect, with the second arrow connected to **Source Graphic**, and set **Operator** to **In**, as we did previously in the chapter. This is shown in *Figure 13.20*.

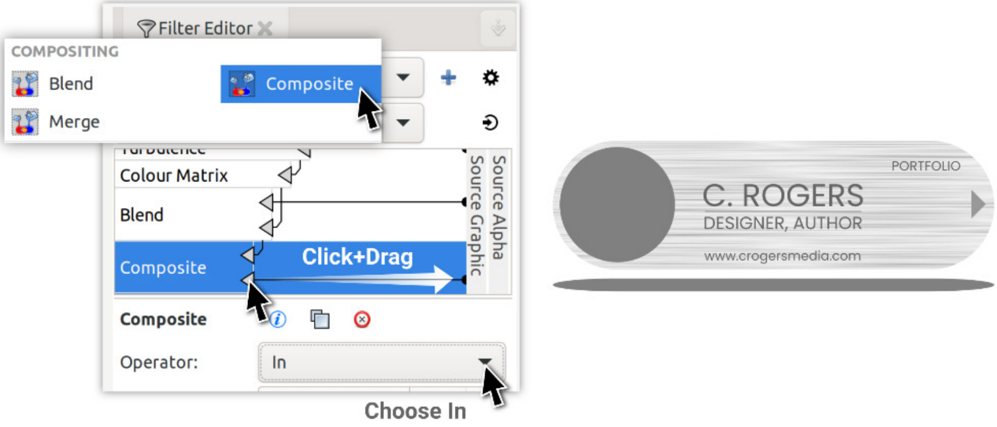
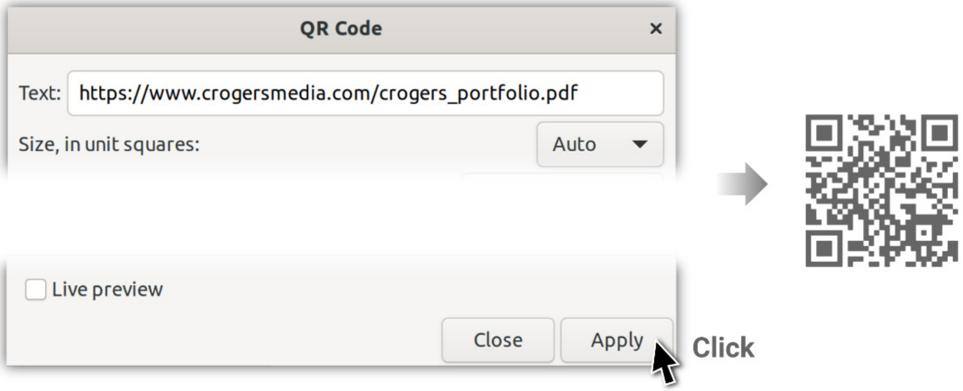


Figure 13.20 – Adding a Composite node to clip the filter to the original shape

8. We can make the QR code in one action by choosing **Extensions** > **Render** > **Barcode** > **QR Code** and entering a website address in the **Text** box pop-up dialog. Then, just hit the **Apply** button at the bottom to generate the QR code, as shown in *Figure 13.21*.


QR Code

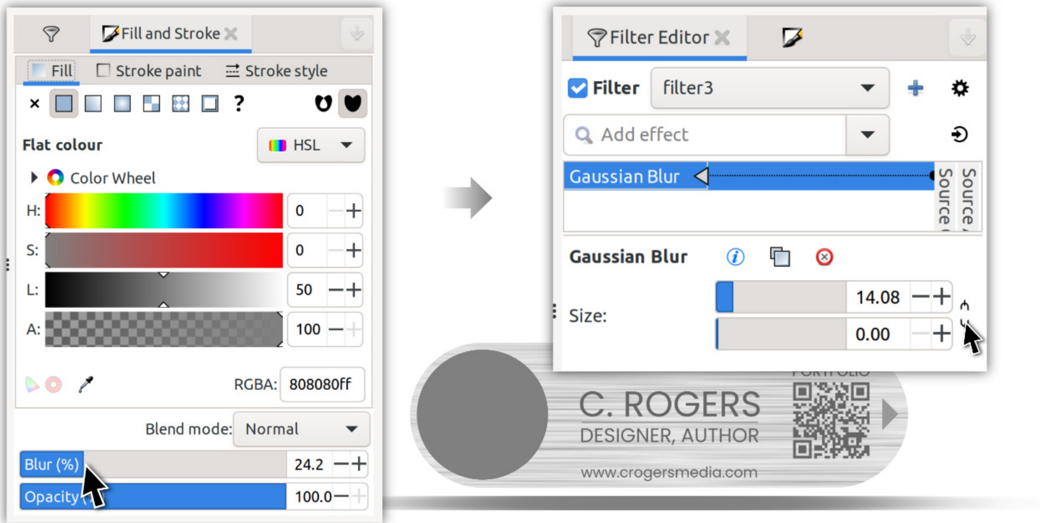
Text: https://www.crogersmedia.com/crogers_portfolio.pdf

Size, in unit squares: Auto

□ Live preview

Figure 13.21 – Generating a QR code

9. We need to blur the shadow at the bottom of our graphic, but we want to blur it in only one direction. We can start by increasing the **Blur** slider in the **Fill and Stroke** dialog, as we've done before. This will automatically add a filter with a **Gaussian Blur** effect, which we can then edit in the **Filter Editor** dialog – simply unlink the **Size** values, and change the values to blur horizontally, as shown in *Figure 13.22*.



H: 0
S: 0
L: 50
A: 100
RGBA: 808080ff
Blend mode: Normal
Blur (%) 24.2
Opacity 100.0

☑ Filter filter3
Add effect
Gaussian Blur

Gaussian Blur
Size: 14.08
0.00

Add Blur

Unlink Size values and change them

Figure 13.22 – Adding and changing Blur values

10. To finish off our graphic, we can import our image and use the circle to crop it, as we did in previous chapters. And that's it – we're done! You can see the end result in *Figure 13.23*.

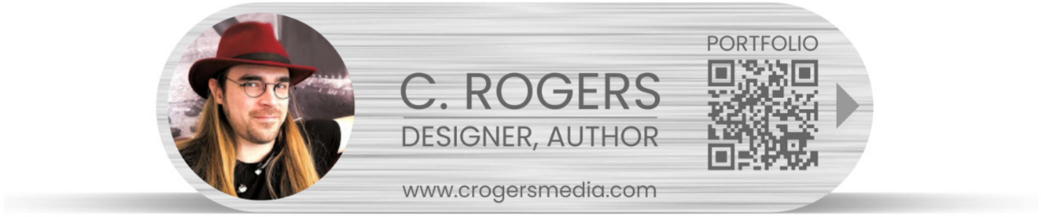


Figure 13.23 – The end result of our practice

Summary

In this chapter, we learned a lot about filters – how to use them and how to create custom ones. We also gave a general overview of how extensions work and how they can save us time on our projects.

We will follow that up in the next chapter with another huge time-saver, as we discover how to use Inkscape to change bitmap images into vectors automatically!

14

Vectorizing with Trace Bitmap

Sometimes, we have images that we'd like to convert to vector shapes. This might take the form of a logo file that a client wants resized, recolored, or edited in various ways, but they only have it in a `.png` or a `.jpg` format (you would be surprised how often this happens).

Fortunately, you have all the tools you need to reproduce the logo in a proper vector `.svg` format, but before you break out the tools that you've learned about for your next project, there's one that may very well save a lot of manual reconstructing. That's the **Trace Bitmap** function!

In this chapter, we will cover the following topics:

- What is Trace Bitmap?
- Vectorizing to black with the **Single scan** tab
- Vectorizing to color shapes
- Tracing parts of a bitmap with the **Fill Bucket** tool

Technical requirements

You can download the practice files for this chapter from GitHub at

`https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape/tree/main/Chapter14`

What is Trace Bitmap?

Put simply, a *bitmap* is any image made of a grid of colored pixels. Photos from your phone camera are bitmaps, for example. We covered this a bit in *Chapter 1*, *The Inkscape Interface*, when we talked about the difference between vector and raster graphics – *raster* being another term for *bitmap*.

Thus, **Trace Bitmap** takes an image of pixels and automatically creates (traces) vector shapes to make a vector representation of the image. This is particularly handy when we have a bitmap that is very simple, such as a black-and-white image.

By selecting the image on a canvas, we can auto-trace it by selecting **Path** > **Trace Bitmap**. *Figure 14.1* shows the original bitmap logo and the result of the trace.

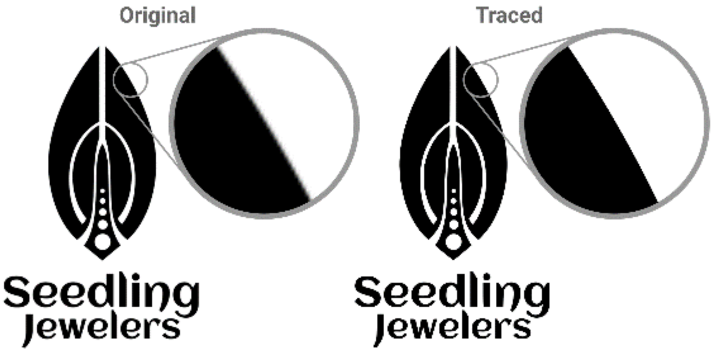
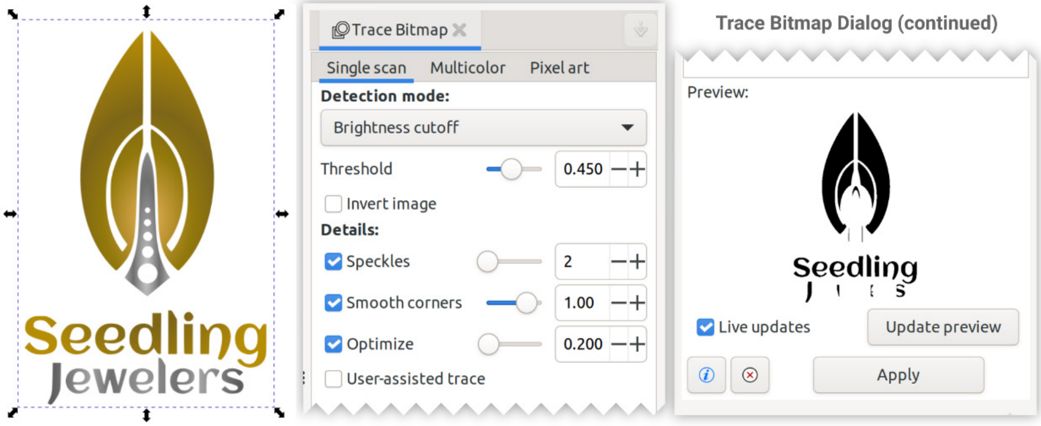


Figure 14.1 – The bitmap original logo (left) and the traced vector result (right)

Both look nearly identical except that when you zoom in to the original, you see pixels on the edge. Inkscape has converted this bitmap to a vector logo, but how? Let's explore the different ways **Trace Bitmap** can do this.

Vectorizing to black with the Single scan tab

The simplest method of vectorizing with **Trace Bitmap** is using a black-and-white or high-contrast image. Once you have your image on a canvas (by dragging the image from your filesystem onto the canvas, or using **File** > **Import**), you simply select the image and then select **Path** > **Trace Bitmap**. This brings up the **Trace Bitmap** dialog, as shown in *Figure 14.2*.



Detection mode: Brightness cutoff

Threshold 0.450

□ Invert image

Details:

☑ Speckles 2

☑ Smooth corners 1.00

☑ Optimize 0.200

□ User-assisted trace

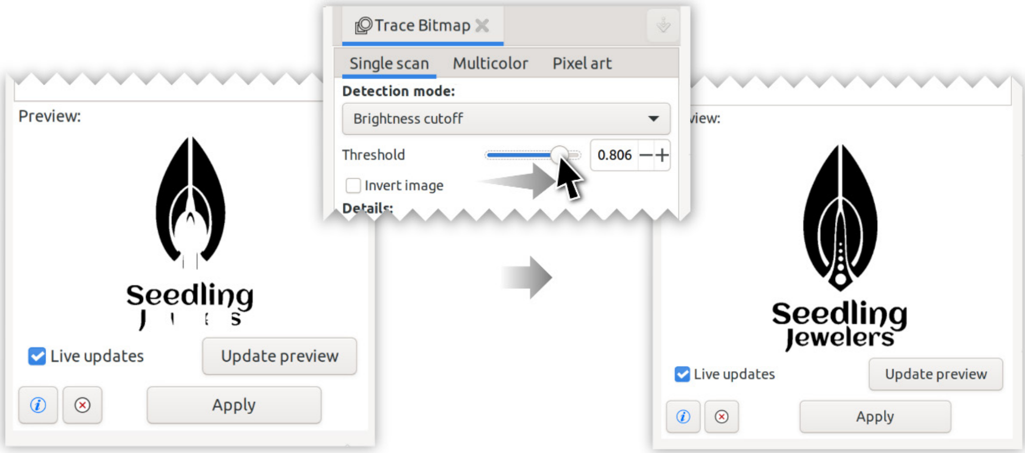
☑ Live updates

Figure 14.2 – The Trace Bitmap dialog

Using the **Single scan** tab, and **Detection mode** set to **Brightness cutoff**, we get a black-and-white preview at the bottom of the dialog. Note that since our logo bitmap is colored with gradients, some of the preview is missing.

This is because **Brightness cutoff** uses the **Threshold** value to determine where the brightness cutoff should be. Smaller values closer to `0.0` will capture only the darkest areas of the image, and values closer to `1.0` will capture lighter and lighter colors. Since some parts of our logo are lighter and do not appear in the preview, we should increase our threshold to a higher value.

Note that if the **Live updates** box is checked, the preview will update automatically when you drag the **Threshold** slider. As you can see in *Figure 14.3*, Increasing the value to about `0.8` fills in the lighter bits of the image that are missing in the preview.



Trace Bitmap

Detection mode: Brightness cutoff

Threshold 0.806

□ Invert image

Details:

Preview:

☑ Live updates

Figure 14.3 – Increasing the Threshold value

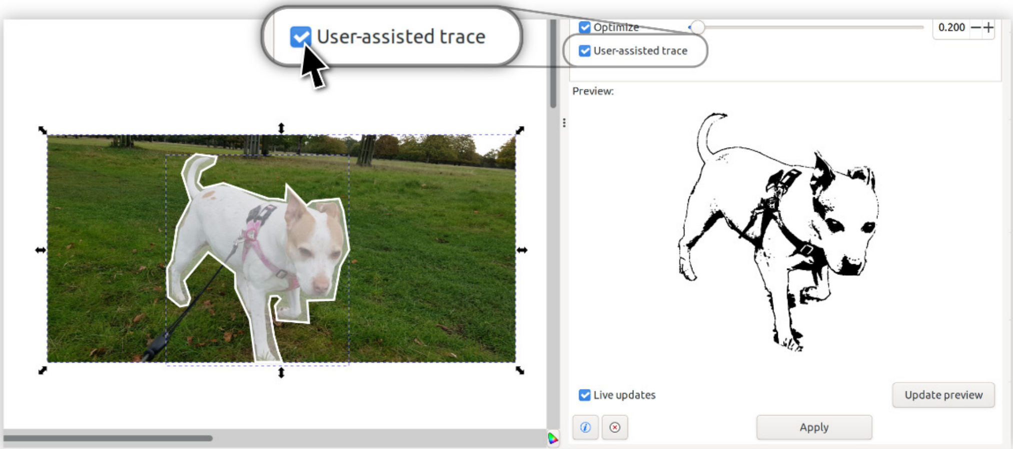
Clicking the **Apply** button will trace the image to vector paths inside a single, black-filled object above the original image. Note that this does not get rid of the original image underneath, which you can choose to keep around for further auto-tracing or to sample colors from with the eyedropper tool when recoloring the logo.

The **Brightness cutoff** detection mode has a few other options under **Threshold**, which are as follows:

- **Invert image**: This checkbox inverts the single scan trace to make black parts white and white parts black. Toggle this on and off a few times to see the results. It's useful when you want to capture the lighter areas of the bitmap rather than the darker ones.
- **Speckles**: This option eliminates speckles (tiny shapes) from the trace. The higher the value of the slider, the larger the shapes that will be removed. Uncheck this option if you're tracing highly detailed grainy texture to capture all the little spots.

- **Smooth corners**: This option controls how smooth the corners will be. If you're getting too much rounding where there should be sharp corners, turn this value down and try again for a more angular result.
- **Optimize**: When tracing higher-resolution images, sometimes Inkscape tries to create too many nodes with too much detail. **Optimize** attempts to reduce the number of nodes that result from the trace. The higher the value, the fewer nodes will be produced to trace the shape.

  The results can get a bit mushy if this value is turned up too high, so I recommend leaving it at the default (close to `0`) unless you need it. If tracing something is crashing Inkscape, or taking too long, increasing this value may help computers with less memory.
- **User-assisted trace**: This option uses a process called **Simple Interactive Object eXtraction** (**SIOX**) in an attempt to separate out a single area of the bitmap. Want to trace a white dog out of an image with a dark background? Just draw a vector shape over the dog, select both the vector shape and the image, and tick the **User-assisted trace** checkbox. Inkscape will update the preview with the new trace, as shown in *Figure 14.4*.



☑ User-assisted trace

☑ Optimize 0.200

☑ User-assisted trace

Preview:

☑ Live updates

Figure 14.4 – Using User-assisted trace to separate the dog from the background

It's worth noting the **Update preview** button to the far right of the **Live updates** checkbox. If generating the preview is taking a long time, you might opt to untick the **Live updates** checkbox and just use the **Update preview** button instead to update the preview only when needed. Similarly, you can press the **x** button to the left of the **Apply** button to cancel the trace operation if the trace takes too long.

There are other options in the **Detection mode** dropdown that are useful for various other kinds of single-scan tracings. **Edge Detection** finds the edge of things and makes a filled vector shape from just the edge. **Color Quantization** picks contrasting colors out of your bitmap and assigns them alternating black-and-white values, based on how many colors you have chosen.

Autotrace averages the colors in the bitmap in an attempt to guess the best shape and color of the resulting vector trace. These three alternatives are shown in *Figure 14.5*.

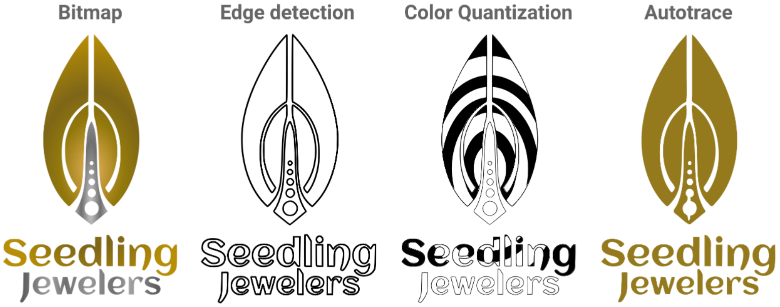


Figure 14.5 – Alternative detection mode options and their effect on the original bitmap

So far, the options have generated closed and filled shapes as a result of the trace. The last option in the dropdown is **Centerline trace**. This is for tracing things such as maps and line drawings into unfilled and unclosed single vector lines.

This is very useful, as it lets you edit the stroke thickness after tracing without all the extra geometry of the other options. *Figure 14.6* shows the results of a **Brightness cutoff** trace versus **Centerline tracing** on a line map of the United States.

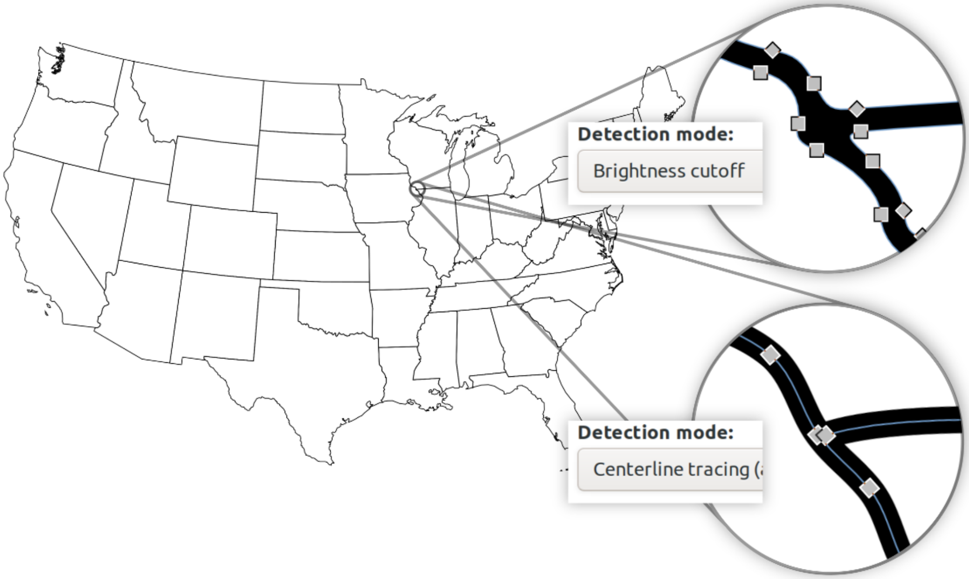


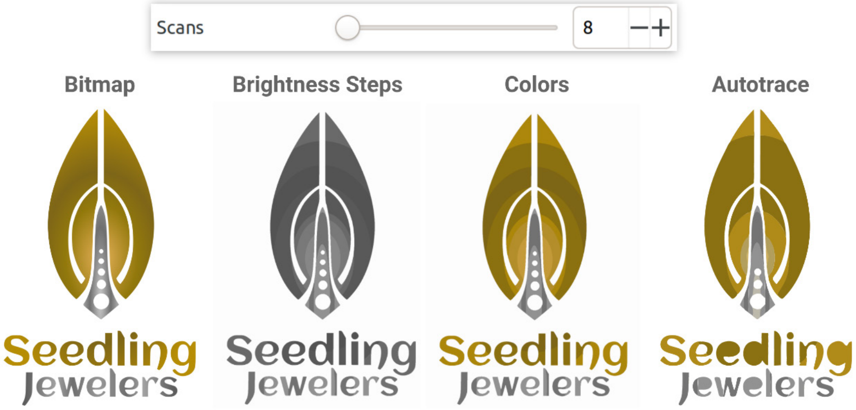
Figure 14.6 – A bitmap of a US map with two different detection modes

That concludes the options in the **Single scan** tab, so let's move on to tracing multi-color vector shapes in the **Multicolor** tab.

Vectorizing to color shapes

So far, we know how to vectorize to single-layer black shapes. Sometimes, we'd like to be able to vectorize an image into a series of stacked, differently colored shapes. This is possible through the **Multicolor** tab in the **Trace Bitmap** dialog. By default, **Detection mode** is set to **Brightness steps**, which produces stacked objects in shades of gray.

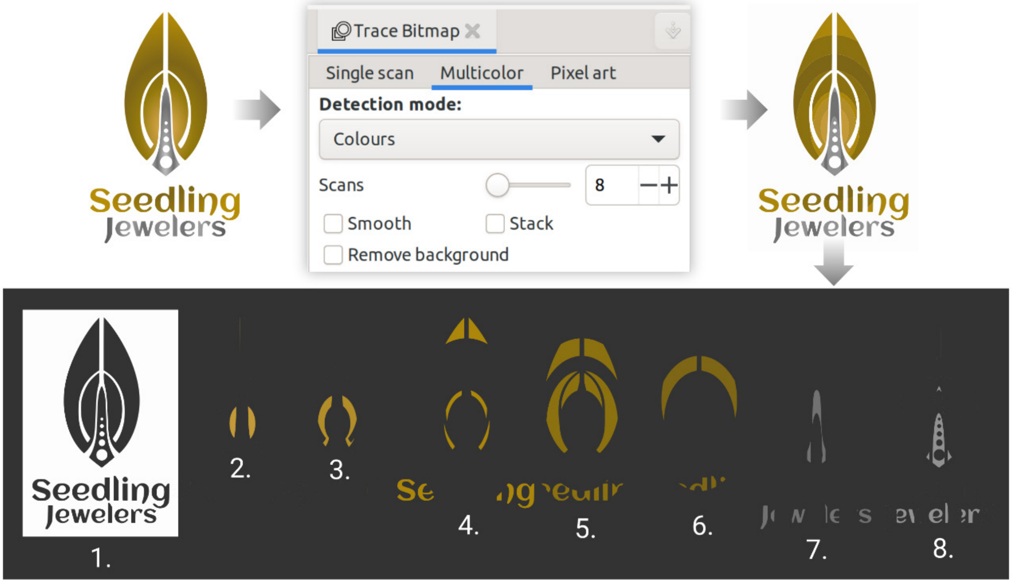
Similar to **Single scan**, **Multicolor** has various options on how to determine the areas of a bitmap that should be made into vector shapes. *Figure 14.7* shows the various results of options available under the **Detection mode** dropdown for our bitmap logo.



Scans 8

Figure 14.7 – The results of each of the Detection mode options in the Multicolor tab

As you can see, options in the **Multicolor** tab try to make separate vector shapes by combining similar colors in the bitmap. The number of scans (the default is `8` at the time of writing) determines how many different color objects Inkscape will create. *Figure 14.8* shows the different objects stacked side by side that result from `8` scans.



Trace Bitmap

Single scan | Multicolor | Pixel art

Detection mode:

Colours

Scans 8

□ Smooth □ Stack

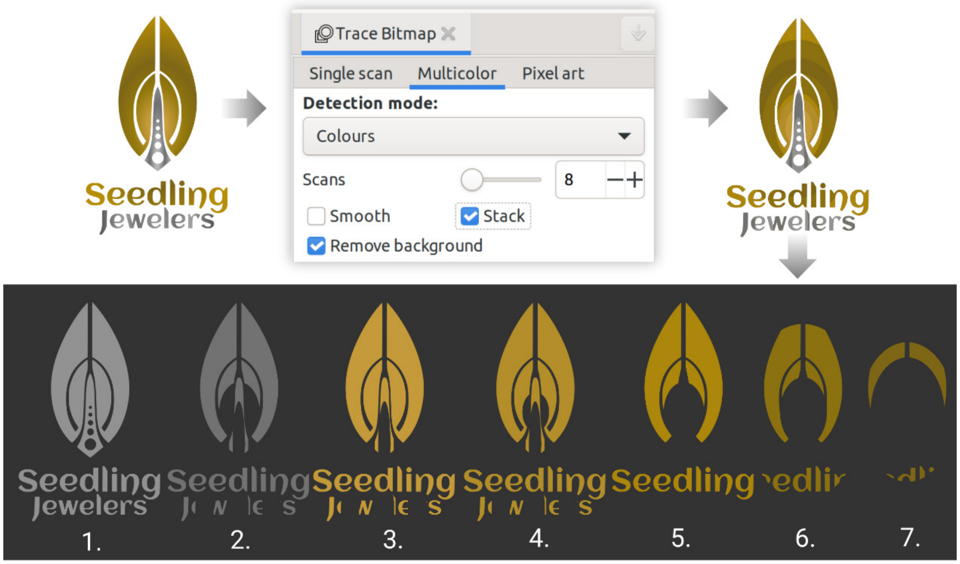
□ Remove background

Figure 14.8 – The resulting shapes of the Colours detection mode breaking apart

The higher this value, the smoother our gradients will look, but more shapes will have to be created. Even worse, when you zoom in on all these flat shapes, you will see edges and small gaps in the resulting vector images. Therefore, generally speaking, we will want to keep the **Scans** value as low as possible.

For tracing a logo with gradients, we will most likely want to do our tracing in the **Single scan** tab, and then add back the gradients manually with the **Gradient** tool, as we learned in *Chapter 8*, *Advanced Shading and Coloring*.

Let's take another look at the options for **Multicolor**, using the **Colour** detection mode, as this will give us the best results and the most control over the colors and shapes. *Figure 14.9* shows that the results are different from those in *Figure 14.8* when we check the **Stack** and **Remove background** checkbox options.



Trace Bitmap

Single scan | Multicolor | Pixel art

Detection mode: Colours

Scans 8

☐ Smooth ☑ Stack

☑ Remove background

Figure 14.9 – Color trace with the Stack and Remove background checkboxes ticked

Notably, if we tick the **Stack** checkbox option, Inkscape will stack the parts one atop another instead of creating cutouts for the overlaying shapes. We can also tick the **Remove background** checkbox, which determines whether the background color (in this case, white) is included as a shape in the resulting group of stacked objects.

The advantage of using **Stack** is that we do not see small gaps in the logo when we zoom in, because the logo is built a bit like a layer cake instead of puzzle pieces. *Figure 14.10* shows the difference between the two results zoomed in.

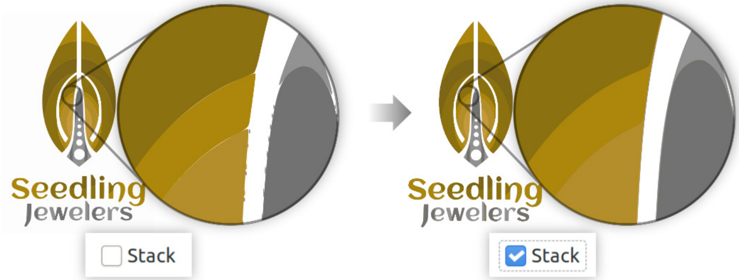


Figure 14.10 – Not stacked versus stacked trace results in Multicolor > Colours

We can also perform this trace on photographs as well. You may want to save regularly while tracing detailed photographs, as this can take a long time and generate hundreds of thousands of nodes in the resulting objects! To reduce the number of nodes, use the **Smooth** checkbox and sliders.

You can also select your traced objects and choose **Path** > **Simplify**, which will attempt to reduce the number of nodes in the shapes as well. Another technique is to cut out the part of a photo you'd like to vectorize using photo-editing software, such as **GIMP** (`https://www.gimp.org/`). *Figure 14.11* shows the results of tracing a duck photo I've cut out of the background to reduce complexity before the trace.

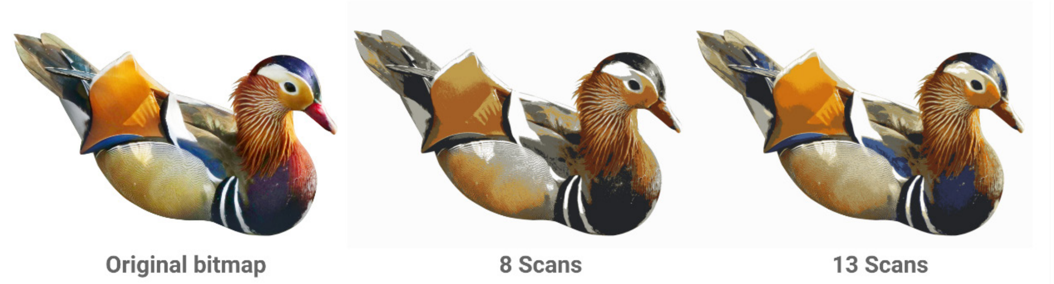


Figure 14.11 – A duck image and the results of 8 and 13 scans

Note that even with 13 scans, I'm still missing that nice splash of red. My node count is also quite high here, due to the complexity of the duck feather patterns. Generally speaking, it's not a great idea to use **Trace Bitmap** to vectorize complex photos.

It may save you from seeing pixels if you zoom in far enough, but pixelation is replaced with rather ragged-looking vector edges, which are not really nice to look at anyway, and would require a lot of manual cleanups. Your mileage may vary, however, so the options and methods are worth knowing about and experimenting with, depending on your project and desired results.

Sometimes, however, we'd like to just grab a shape from part of our vector image without having to use **Trace Bitmap** on the entire thing. We can do just that with the **Fill Bounded Areas** tool (also known as the **Fill Bucket** tool).

Tracing parts of a bitmap with the Fill Bucket tool

Say we'd like to grab parts of our duck head to make a nice logo. We could attempt to auto-trace it with **Trace Bitmap** and cut out the unused portions, but we could also just use the **Fill Bucket** tool to selectively grab shapes. This method also allows us to get shapes from an area of color regardless of how light or dark it is.

For example, we can trace parts of our duck's head by selecting the **Fill Bucket** tool and click-dragging over areas of color we'd like to capture. By doing this multiple times, we can build up a much better illustration than we would get otherwise. This iterative click-dragging process is shown in *Figure 14.12*.

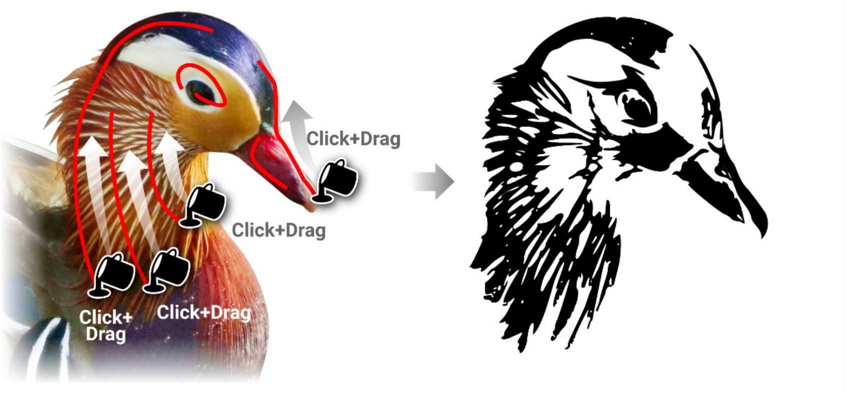
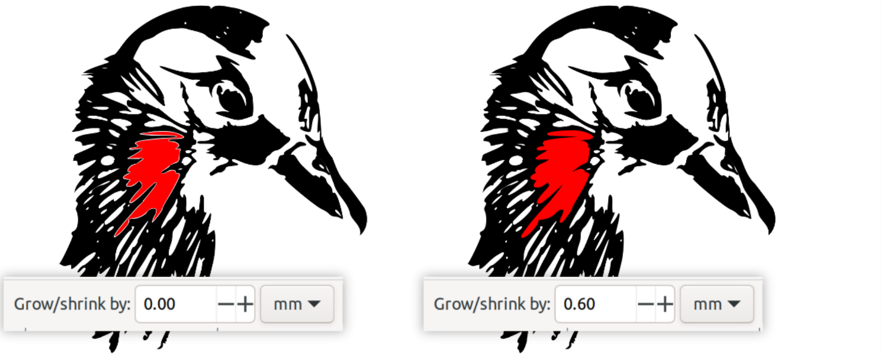


Figure 14.12 – Using the Fill Bucket tool to trace shapes in the duck's head

It's worth noting that like the other Inkscape tools, the tool controls bar contains options for the **Fill Bucket** tool when it's active. Turning up the **Threshold** value will capture more similar colors faster. This is particularly helpful when single-clicking an area (instead of click-dragging), as Inkscape will use the color directly under the mouse click to make a shape. Thus, the larger the **Threshold** value, the more will be added to that area.

You can also use the **Grow/shrink by** value to get a somewhat more globby expansion area, though at the cost of the sharpness and accuracy of the trace. This option is more useful when using the **Fill Bucket** tool to color comics, as it decreases the chance of white gaps between adjacent areas. *Figure 14.13* shows the difference, when we try to color in some of the feathers in our illustrative duck head trace.



Grow/shrink by: 0.00 mm

Grow/shrink by: 0.60 mm

Figure 14.13 – Adjusting the Grow/shrink by value to get rid of small gaps between shapes

In some instances, the **Fill Bucket** tool will fail to find a closed shape to fill in due to small gaps in the shapes, which let the fill area spill out into the canvas area. You can tell the **Fill Bucket** tool to ignore these small gaps with the **Close gaps** dropdown, also choosing the size of your gaps (**None**, **Small**, **Medium**, or **Large**).

Up to this point, we've been using the **Fill Bucket** tool in **Fill by: Visible Colors** mode. Also worth noting is that the **Fill by** dropdown at the far left of the tool controls bar offers other options that enable you to isolate different areas by different criteria, including **Red**, **Green**, and **Blue** color channels, **Lightness**, **Hue**, **Saturation**, and even **Alpha** (transparency) for PNG images.

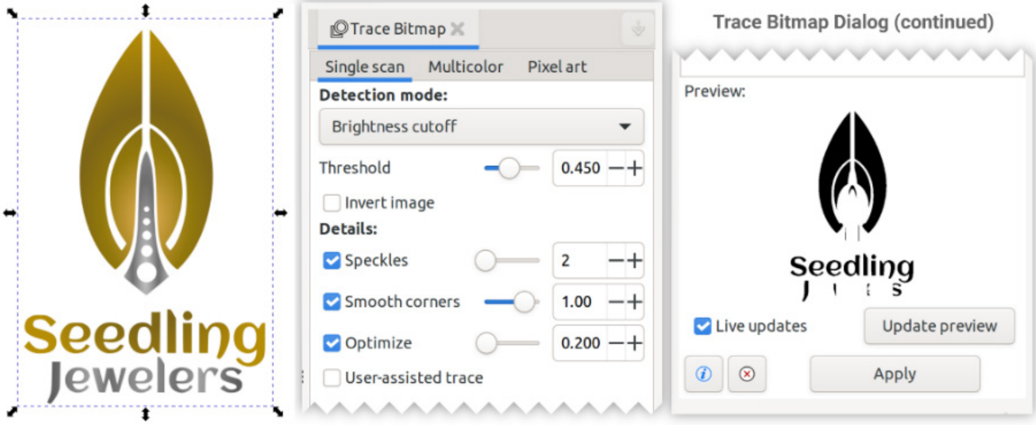
So now, we have a bunch of tools to trace an image and even pick out some details on the fly. Let's put some of these to use in a practice lesson!

Try it yourself

In this exercise, we'll do a complete vectorization of a logo. You can choose one from the internet for practice, or you can use the *Seedling Jewelers* logo provided in the practice materials for this chapter.

The process will be the same for most logos; just make sure that the logo you choose is high-quality, typically at least 1,000 x 1,000 px to preserve crisp corners and accurate curves. You can work with lower-resolution logos, but the results will need more cleanup using the **Node** tool to manually tweak your geometry:

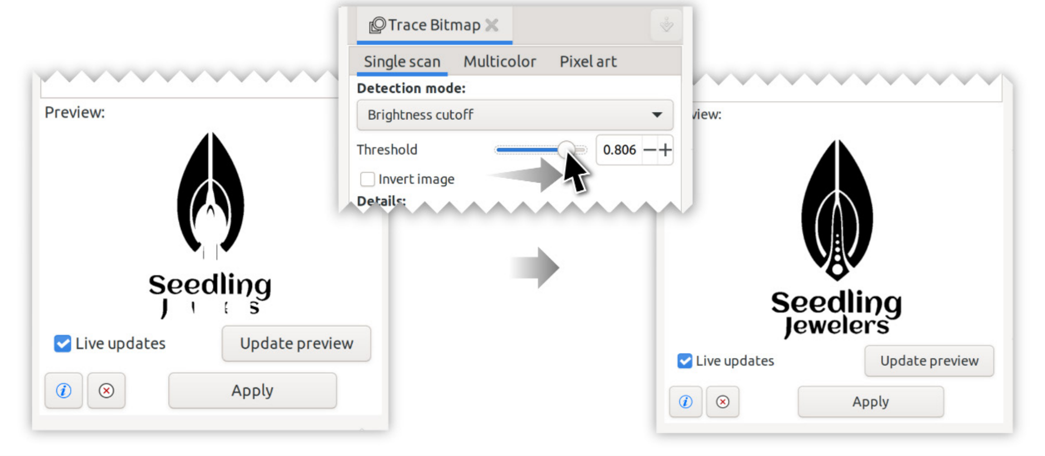
1. Import your bitmap logo to the canvas using **File** > **Import**, or drag it to Inkscape from your filesystem.
2. We will then call up the **Trace Bitmap** dialog by choosing **Path** > **Trace Bitmap**. We will choose the **Single scan** tab and select **Brightness cutoff** from the **Detection mode** dropdown. With our bitmap image selected, we should see a preview, as shown in *Figure 14.14*.



Detection mode: Brightness cutoff
Threshold 0.450
□ Invert image
Details:
☑ Speckles 2
☑ Smooth corners 1.00
☑ Optimize 0.200
□ User-assisted trace
☑ Live updates

Figure 14.14 – The Trace Bitmap dialog with a preview of the logo trace

3. We will then adjust the **Threshold** value until we get a complete silhouette of our logo, as shown in *Figure 14.15*. When we click the **Apply** button, this will give us a single vector shape that we can then break into separate shapes and colors to match the original gradients.



Trace Bitmap

Single scan | Multicolor | Pixel art

Detection mode: Brightness cutoff

Threshold 0.806

□ Invert image

Details:

Preview:

☑ Live updates

Update preview

Apply

Figure 14.15 – Increasing the Threshold value to trace more of the image

4. We want to break this logo apart for coloring; however, instead of selecting **Path** > **Break Apart**, we'll choose **Path** > **Split Path** to preserve the holes in our letters and logo. We will then select the similarly colored parts of the logo and then choose **Path** > **Combine** to combine each set into a total of two shapes, as shown in *Figure 14.16*.

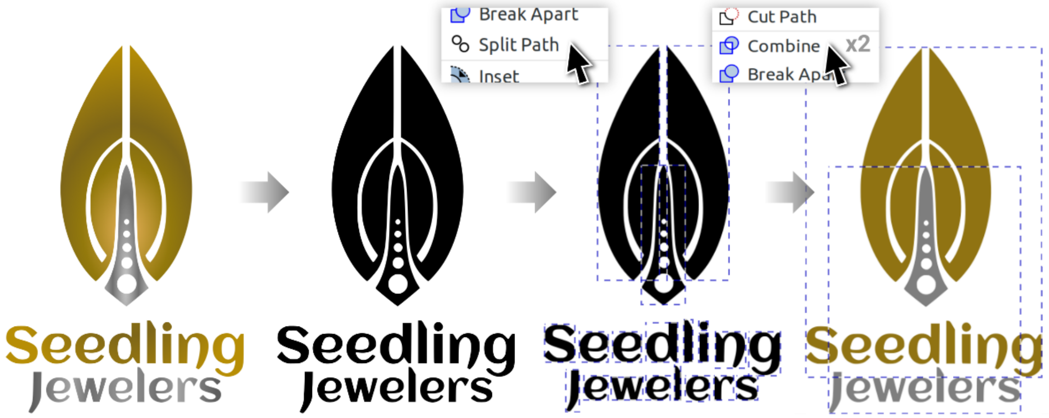


Figure 14.16 – Splitting and recombining paths into two objects

5. From here, it's just a case of reproducing the gold and silver gradients of the original. Remember that you can use the **Gradient** tool, select a color stop, and use the eyedropper tool to sample colors from the original, as shown in *Figure 14.17*.

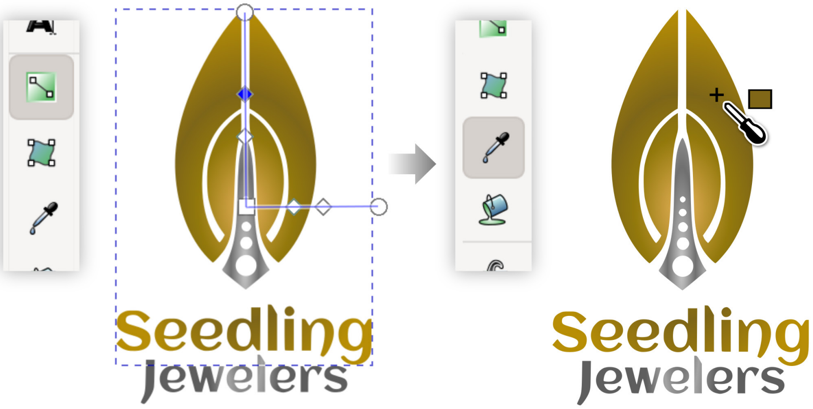


Figure 14.17 – Using the original bitmap logo colors to recreate the radial gradient

And that's it! We've recreated a vector version of this bitmap logo!

Summary

In this chapter, we learned lots of useful tricks to convert imported bitmaps to vector shapes. These are great time-savers and will serve us well to reduce tedious reconstruction. In the next chapter, we're going to wrap up this grand tour of Inkscape by looking at document properties and the many file-saving and export options that Inkscape supports.

15

Document Properties, Pages, Exporting, and Printing

Well, we've certainly come a long way in making all sorts of vector designs in Inkscape. But what now? How do we get those great pieces of art out of Inkscape and into the real world? In this chapter, we'll go over preparing and exporting our great Inkscape designs in a variety of different formats, for different purposes.

In this chapter, we will cover the following topics:

- The **Document Properties** dialog
- The **Pages** tool
- The **Export** dialog
- Save and export considerations
- Printing

Technical requirements

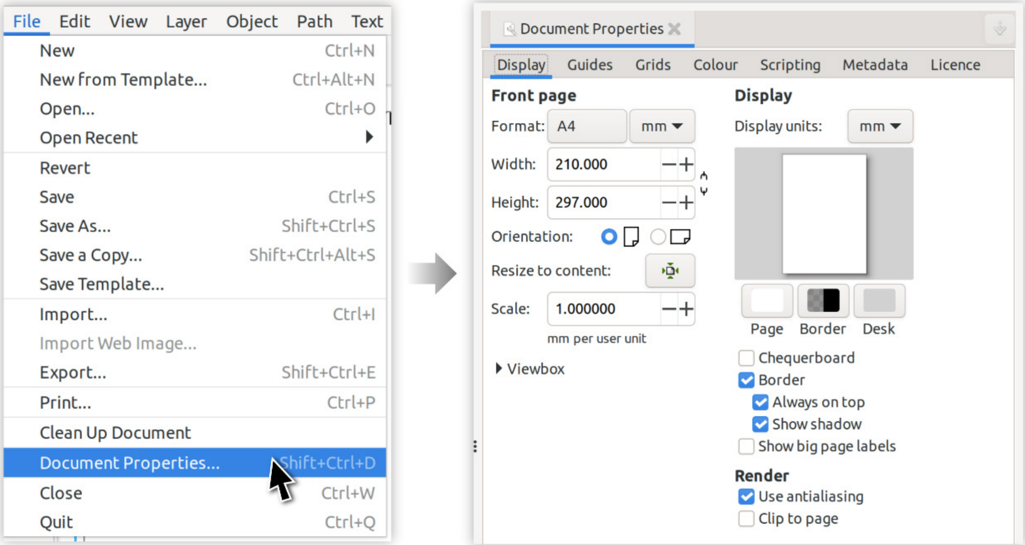
You can download the practice files for this chapter from GitHub at

`https://github.com/PacktPublishing/Design-Made-Easy-with-Inkscape/tree/main/Chapter15`

The Document Properties dialog

Up until now, we haven't been too concerned with the scale of things – how big is our canvas? How big are our shapes? What units are we using to measure? We've kind of taken for granted that Inkscape comes with a default A4 page-sized canvas and units in mm. For most purposes, this works fine.

But what if your measurements are in inches or even screen pixels? Well, we can change these things and much more in the **Document Properties** dialog. To bring it up, we simply choose **File** > **Document Properties…**. We are then greeted by something that looks like *Figure 15.1*.



Front page

Format: A4 mm

Width: 210.000

Height: 297.000

Orientation: ● ○

Resize to content:

Scale: 1.000000

mm per user unit

Viewbox

Display

Display units: mm

☐ Chequerboard

☑ Border

☑ Always on top

☑ Show shadow

☐ Show big page labels

Render

☑ Use antialiasing

☐ Clip to page

Figure 15.1 – Calling up the Document Properties dialog

As you can see, there are a wealth of different options, organized into tabs. We'll be looking through the **Display** tab options first. Notice that there are two columns here, one for **Front page** and another for **Display**. Both columns have a dropdown for the units (the default is **mm**).

In truth, I'm not entirely sure why anyone would want to use two different units for **Format** and **Display**, and indeed, I recommend that if you change one, you change the other too. Otherwise, you will spend a lot of time switching to your desired units on the fly when working in the Tool Control bar.

These options set what shows up by default throughout most of Inkscape's interface. So, why would we care what units show up? Well, if we're working with web graphics, for example, we don't know the physical dimensions of the screen visitors are using. We do have some idea of the different screen sizes in pixels, though, which is why by default I tend to work in pixels for website images and video graphics templates and use **mm** for just about everything else (like most of the world outside the US).

You can switch to **inches** if you like, but then you have to deal with 12 inches to a foot, and fractions of an inch, and it gets messy fast to convert between them. Of course, if your client sends measurements in inches, then you can set your initial size to **inches**, then call up **Document Properties** afterward to change to a much more convenient measurement system.

Going down the **Front page** column, we have a dropdown for the page size. Note that this conveniently changes the page and display units to inches if you choose a US page size. Going farther down, there are some self-explanatory fields for adjusting the height and width of the document (note the link icon, which you can click to resize proportionally based on the current dimensions), and the orientation, which swaps the height and width to make the document portrait or landscape.

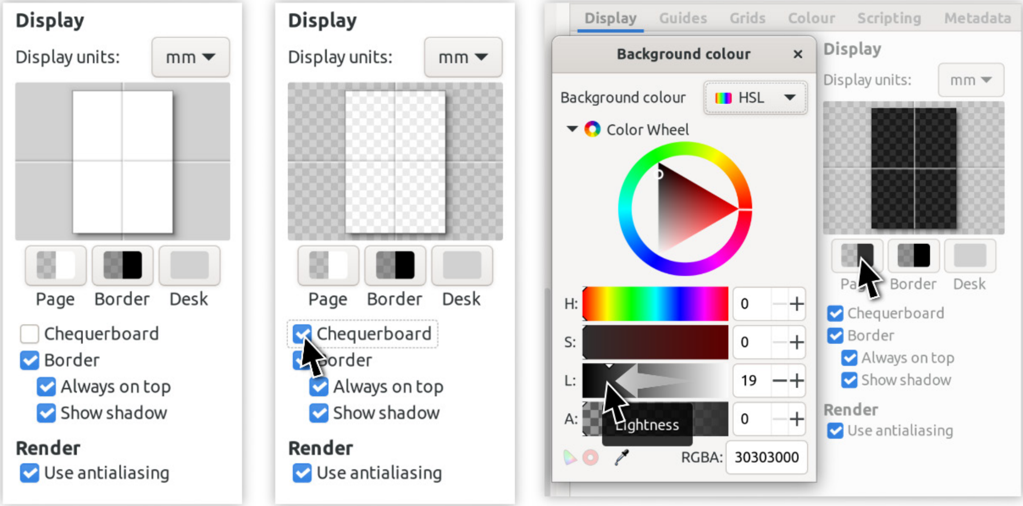
Below that, there's a useful button, **Resize to content**, meaning that it will auto-resize the document to fit everything on your canvas onto the page. Worth noting is that you can also resize the page to anything on your canvas by selecting objects and choosing **Edit** > **Resize Page to Selection** (or pressing *Shift* + *Ctrl* + *R*). I use this hotkey far more than opening up the document properties, but it's nice having a button for it in the **Document Properties** dialog too.

The next item in this column is **Scale**. I recommend you don't touch this value – it's really only useful if you're importing something that isn't scaling correctly versus the original. Inkscape's user units are hidden from the user, so changing this is just asking for scaling problems on saving and loading.

Another thing to ignore completely is the **Viewbox** dropdown directly below this. This controls the viewbox area for the SVG file when you view it in a web browser, which should be the same as the page size, unless you change it…

Moving on to the **Display** column, we find controls for how the **Page** area looks. Note that there is a preview of the canvas (also known as **Desk**), the page, and page styles, such as the border (the thin line around the canvas). You can change a lot about the look of these parts of your document, but maybe the most useful is the **Chequerboard** checkbox.

Ticking this box will display transparent parts of your document as that checkerboard pattern, which makes it easy to see what is opaque and what has some transparency. Also, if you are working on light-colored illustrations that will be displayed over black, you are able to change the default page colour to match. *Figure 15.2* shows the **Display** column and the result of checking this box, as well as changing the page background colour.



Display

Display units: mm

Page Border Desk

☐ Chequerboard

☑ Border

☑ Always on top

☑ Show shadow

Render

☑ Use antialiasing

Display

Display units: mm

Page Border Desk

☑ Chequerboard

☑ Border

☑ Always on top

☑ Show shadow

Render

☑ Use antialiasing

Background colour

Background colour: HSL

Color Wheel

H: 0

S: 0

L: 19

A: 0

RGBA: 30303000

Display

Display units: mm

Page Border Desk

☑ Chequerboard

☑ Border

☑ Always on top

☑ Show shadow

Render

☑ Use antialiasing

Figure 15.2 – Changing various options in the Display column of the Display tab in the Document Properties dialog

You'll likely recognize the next two tabs in the **Document Properties** dialog from *Chapter 4*, *Automatic Shape Alignment in Inkscape*, when we went over guides and grids. Thus, we will skip those and go on to the **Colour** tab (or **Color** tab, depending on whether your Inkscape uses the British or American spelling of color).

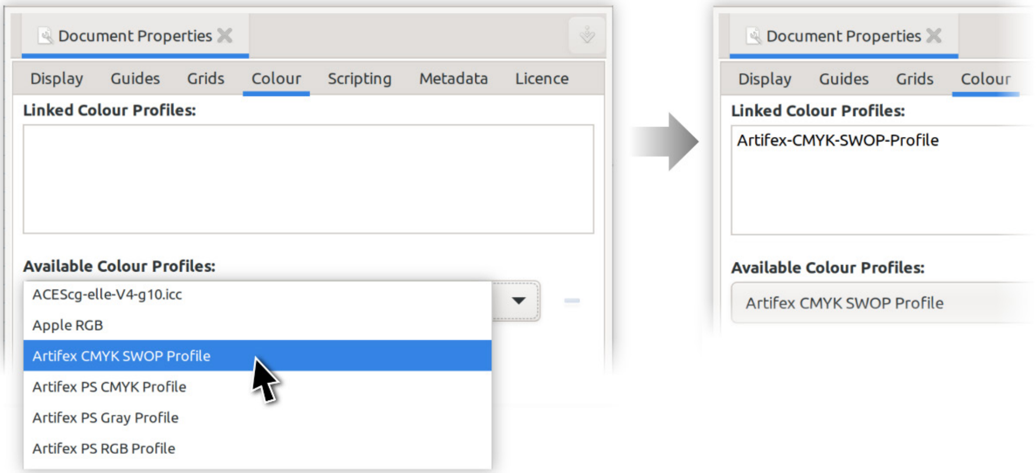
This tab contains linked color profile information. Simply put, color profiles are used to convert the colors you see on your screen to approximately matching colors on various outputs, including print (usually CMYK) and video (NTSC, PAL).

Generally speaking, you don't need to change or worry about these settings, as most modern printers will happily convert your sRGB PDFs (or even PNGs, or JPEGs) to the CMYK profiles used in their printers.

However, if the printer provides a color profile, you can link it here. Note, however, that linking in a CMYK color profile does not convert RGB values to CMYK automatically, and at the time of writing, Inkscape does not natively support this conversion either.

If none of this is making sense, know that color profiles are a very technical part of the printing process and are generally best handled by the printer anyway. Even using the video color profiles is usually unnecessary, as video editing software is more than capable of converting the default sRGB color profile to the output formats required.

Figure 15.3 shows the process of linking in a color profile, which can be chosen from the **Available Colour Profiles** dropdown.



Linked Colour Profiles:

Available Colour Profiles:

- ACEScg-elle-V4-g10.icc
- Apple RGB
- Artifex CMYK SWOP Profile
- Artifex PS CMYK Profile
- Artifex PS Gray Profile
- Artifex PS RGB Profile

Linked Colour Profiles: Artifex-CMYK-SWOP-Profile

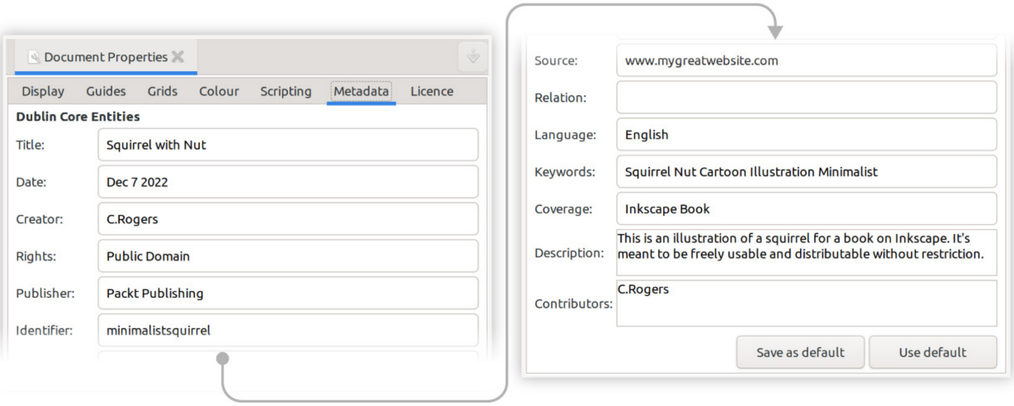
Available Colour Profiles: Artifex CMYK SWOP Profile

Figure 15.3 – Linking a CMYK SWOP profile to the document

The **Scripting** tab is the next stop on our tour and contains tools for managing linked and embedded scripts in your SVG file. A script is a program that does something within the SVG file when it's embedded in a website.

The web browser can run these programs to create interactivity, animation, and other things. Of course, scripting is outside the scope of this book, so for our purposes, it suffices to know that options exist if you decide to take the very technical plunge into scripting your SVGs in the future.

Since we're not taking the plunge into scripting, we'll go on to the **Metadata** tab. That sounds like something really technical, but it's really just a way to store information about the document inside it, for example, the author, license, search keywords, and even a description. As you can see from *Figure 15.4*, there are lots of different fields that you can use to add extra information to your document.



Dublin Core Entities

Title: Squirrel with Nut

Date: Dec 7 2022

Creator: C.Rogers

Rights: Public Domain

Publisher: Packt Publishing

Identifier: minimalistsquirrel

Source: www.mygreatwebsite.com

Relation:

Language: English

Keywords: Squirrel Nut Cartoon Illustration Minimalist

Coverage: Inkscape Book

Description: This is an illustration of a squirrel for a book on Inkscape. It's meant to be freely usable and distributable without restriction.

Contributors: C.Rogers

Figure 15.4 – Metadata tab filled out with sample information

Unlike many other dialogs, this one has no **Apply** or **Save** button. This is because simply changing a value in any of the boxes will automatically save the new/changed data. You'll notice the **Save as default** button at the bottom of the dialog, which can be used to save the current metadata for this document to be the default for new documents.

For example, you may want to save just the **Creator** information so that it automatically shows up for new documents. In this case, leave all other fields blank, and then click the **Save as default** button. You can then resume filling out the rest of the information for this particular document and close the dialog. Clicking the **Use default** button will replace the metadata with the default information you saved.

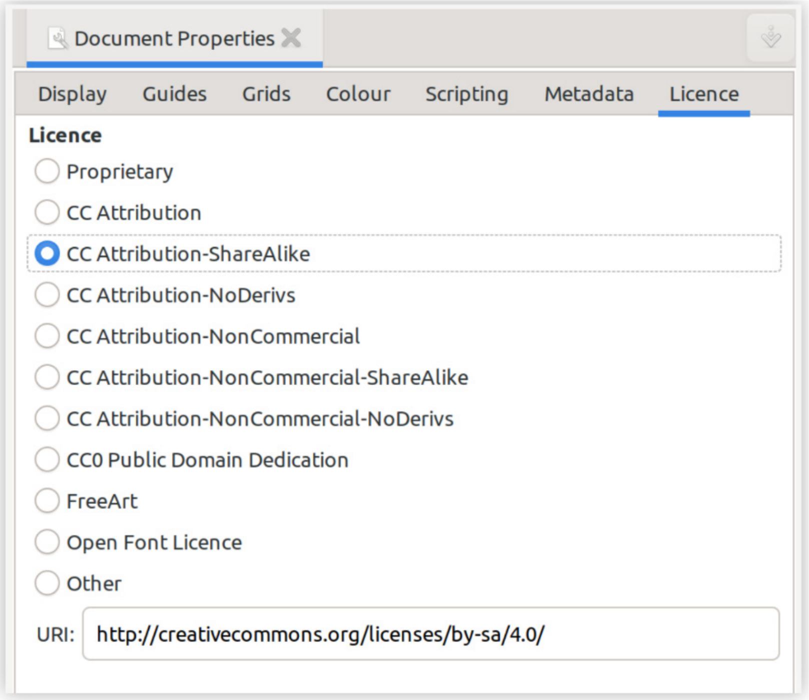
What good is all this metadata? Well, for one, when someone opens your SVG file, they will be able to see all this metadata saved in the file, which will provide useful information about who made it and how it can be used.

This information can also be seen and used by programs on the internet, which can search for the keywords you enter in the **Keywords** field. Thus, your file will show up in more search results for those terms.

Along the same lines, the last tab in the dialog is the **License** tab, which can be used to select a license for your file. The default is **Proprietary**, which assumes you want to reserve all rights to your own artwork for yourself – people must come to you and acquire a license if they intend to use your graphic.

This dialog includes many other options for more permissive licenses, for those who want to release some of their rights in order to permit others to use the graphics without needing permission. These alternative licenses are called **Creative Commons** (**CC**) licenses and allow people to share your graphics under certain circumstances.

For example, **CC Attribution-ShareAlike** gives everyone permission to use and distribute your document as long as they give you credit for making it, and also share what they make with it under the same license. When you choose a license from this tab, a web address that links to the wording of the license appears in the **URI** field at the bottom, as shown in *Figure 15.5*.



Licence

○ Proprietary

○ CC Attribution

● CC Attribution-ShareAlike

○ CC Attribution-NoDerivs

○ CC Attribution-NonCommercial

○ CC Attribution-NonCommercial-ShareAlike

○ CC Attribution-NonCommercial-NoDerivs

○ CC0 Public Domain Dedication

○ FreeArt

○ Open Font Licence

○ Other

URI: http://creativecommons.org/licenses/by-sa/4.0/

Figure 15.5 – Chosen license and corresponding URI (web address)

You can read all the details of the license you choose by entering that web address into your web browser of choice. I recommend you read them all carefully before choosing something that fits your particular circumstances. For unlimited sharing with no strings attached, choose **CC0 Public Domain Dedication**. You can still include your information as the author; the license just doesn't require anyone to give you credit for making the graphic or restrict how it may be used.

That concludes our tour of **Document Properties**. But what if you want more than one page in your document? As of Inkscape 1.1, we have a brand-new tool for making multipage documents. Let's have a look at the **Pages** tool!

The Pages tool

This long-requested feature brings Inkscape to a whole new level of usefulness. Previously, it was only possible to make single-page documents as separate PDFs, which then had to be put together in a program outside Inkscape. With the **Pages** tool, we can now add and remove pages, which will get saved as separate pages when we save our document as a PDF.

When we click on the **Pages** tool, we get options in the Tool Control bar, which allows us to add and modify the currently selected page. The tool itself can be used to simply click and select pages on the canvas as well as move them around. *Figure 15.6* shows the **Pages** tool and the options in the Tool Control bar.

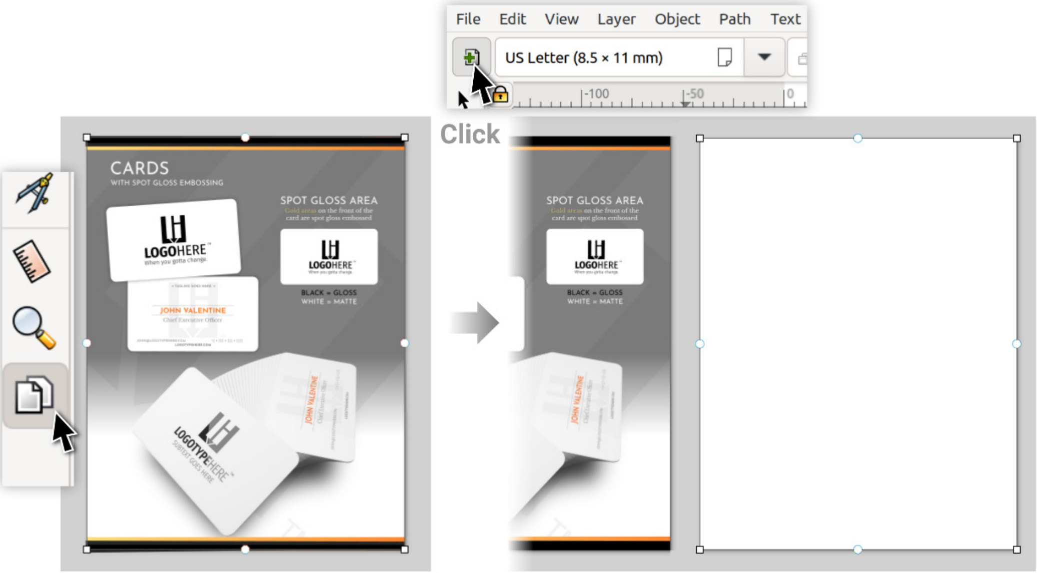


Figure 15.6 – Adding a new page of the same size with the Pages tool

You'll notice the dropdown contains many different standard page sizes and selecting a different one will resize the currently selected page. If you want to draw a new page free hand, simply click and drag on an empty part of the canvas with the pages tool active. Note that this will not change the order of the pages in the output file, however.

Say that we'd like to instead have our new blank page before the cards page in our example. Once we have a new page, we can change the order by using the **Move Page Backwards** button in the Tool Control bar. As shown in *Figure 15.7*, this automatically places the selected (new) page before the cards page.

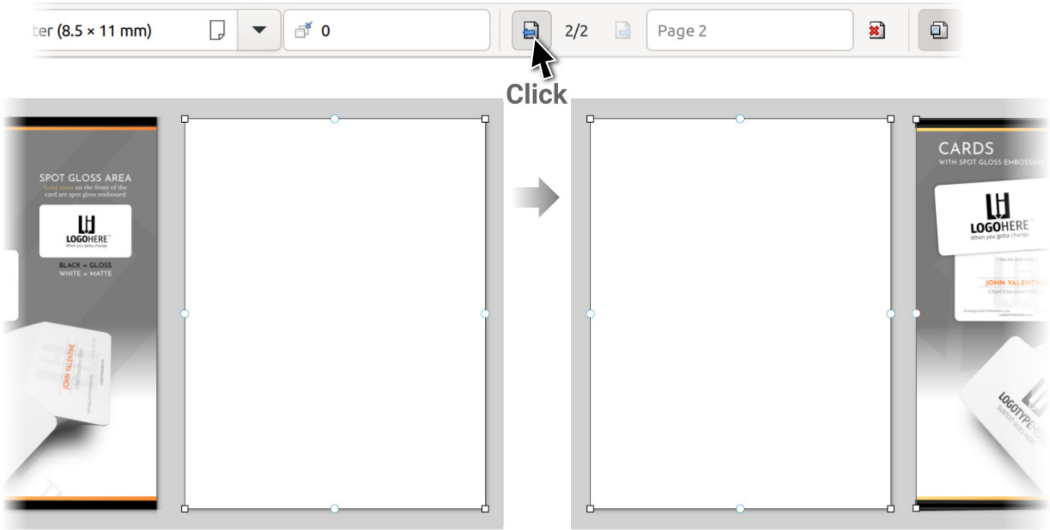


Figure 15.7 – Moving our new blank page before the cards page

Note that you can change the page names in the **Page Label** box, as the labels do not affect the order of the pages. You can also delete the selected page by clicking the **Delete Selected Page** button (the one with the **X**), and if, by chance, you don't want to move or delete the contents of the page along with the page, you can toggle off the **Move Overlapping Objects** toggle, which, at the time of writing, appears last in the Tool Control bar.

There's one more handy Inkscape 1.3 feature in the Tool Control bar for the **Pages** tool, and that's the **Margins** box, where you can set margins for the currently selected page. Simply replace the `0` value in the text box with `10`, for example. You can see from *Figure 15.8* that this adds a margin at a distance of `10 mm` from each side of the page. You can also drag the round handles at the edges of the page to do this manually.

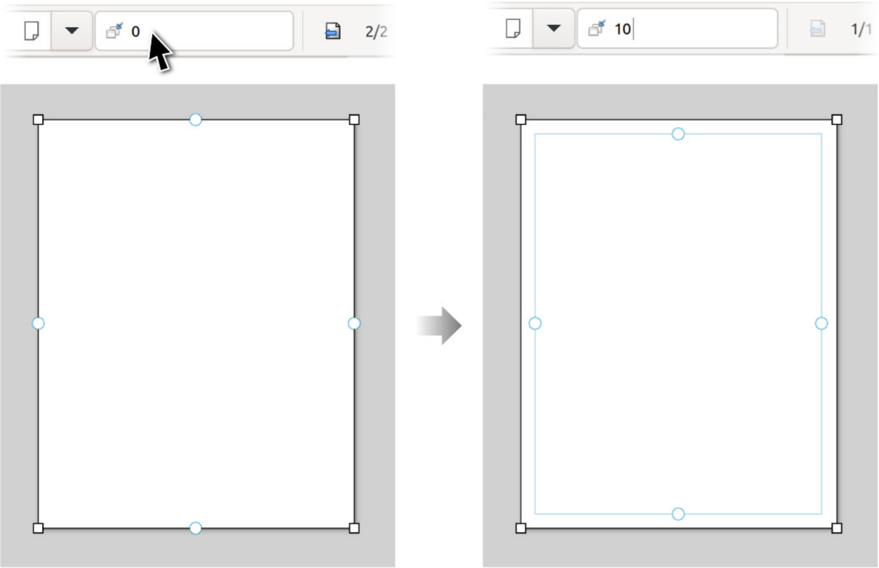


Figure 15.8 – Adding a 10 mm margin to each side of the page

Note that the units of the margins correspond to the units selected in the **Document Properties** dialog. So, `10` mm would be `0.39` if the page units are set to inches.

Now that we've added pages, we can use them in our save and export settings. Let's explore those and other options as we tour the **Export** dialog.

The Export dialog

When it comes to using your Inkscape artwork for a variety of purposes, no dialog is quite as useful as the **Export** dialog. From it, you can export the page area, a selection, or even the entire artwork (on and off canvas) to a variety of useful file formats.

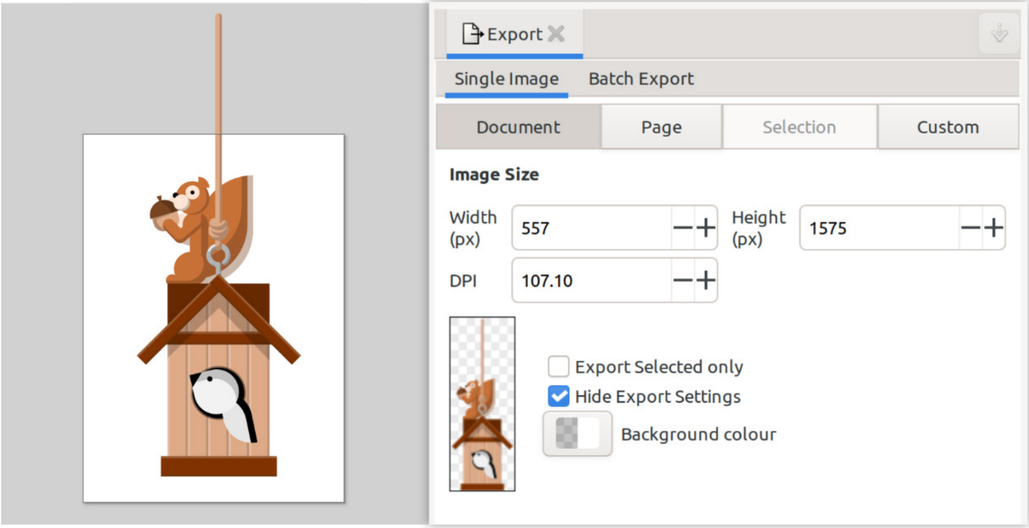
To call up this dialog, you can choose **File** > **Export…** or use the *Ctrl* + *Shift* + *E* hotkey combination. We're then presented with the almighty **Export** dialog, with two tabs, one for exporting a single image and another, **Batch Export**, for exporting multiple images at the same time.

Let's start with the **Single Image** tab, which is usually the default tab that appears when exporting. You'll notice four buttons at the top, just under the tab, which allow you to choose the area that will be exported. The options are as follows:

- **Document**: Exports everything in the document, on and off the page area. This option ignores the page area and exports everything you've drawn.
- **Page**: Exports an area within the page area, trimming off anything that falls outside the page area.

- **Selection**: Uses the bounding box of the currently selected object or objects as the export area, clipping off anything outside that area.
- **Custom**: Allows you to specify specific page coordinates of the rectangular area that will be exported.

Figure 15.9 shows the **Document** option in the **Export** dialog and related settings.



Width (px) 557
Height (px) 1575
DPI 107.10

☐ Export Selected only
☑ Hide Export Settings
Background colour

Figure 15.9 – The Export dialog

As you can see, the rope above the birdhouse extends beyond the **Page** area. With **Document** selected, it is still shown in the preview area of the **Export** dialog as we would expect. You can change the width and height in pixels of the exported image as well as the **Dots Per Inch** (**DPI**) of the resulting image.

DPI is useful if your document units are inches or millimeters and you don't know how big (in pixels) your image should be. Most printers, for example, require a `300` DPI image for a good-quality print. Thus, you can use your physical measurements (**mm** or **inches**) and set **DPI** to `300`. Inkscape will then automatically calculate the height and width of the image in pixels. Handy, isn't it?

Most of the time, however, you'll probably want to use the **Page** option to export as these are the most convenient, using the page area for clipping. This is also why the default document in Inkscape is an A4 page, which is the standard desktop printer paper size across most of the world.

You can of course set the **Page** area to any size you require, and it gives you a reference for how large your illustrations should be while creating them. *Figure 15.10* shows the same graphic as before, but with the **Page** option selected instead of **Document**.

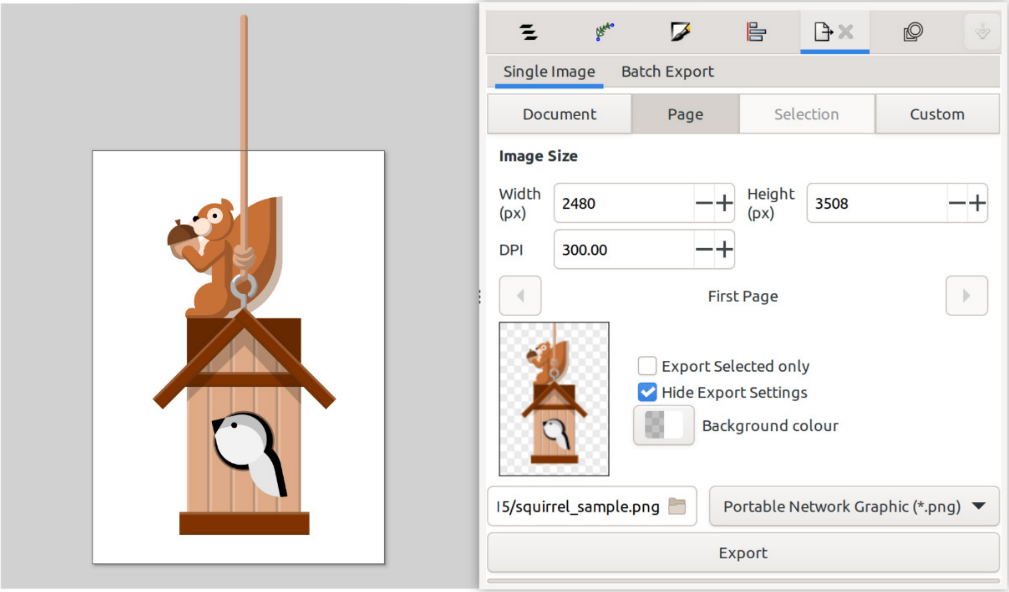


Image Size

Width (px) 2480

Height (px) 3508

DPI 300.00

First Page

☐ Export Selected only

☑ Hide Export Settings

Background colour

15/squirrel_sample.png

Portable Network Graphic (*.png)

Figure 15.10 – Page option selected with DPI set to 300

You'll notice there are a couple of extra options: **Pages**, with a couple of arrows, and **First Page**, which lets you choose which page you want to export from by clicking the arrow. So, if you've made extra pages, this dialog is not grayed out.

You may also have noticed that there are a few options that all four export options share. By ticking the **Export Selected only** option, you can choose what gets exported. This is handy especially in a case where you want to exclude the objects behind and/or in front of something, and only include the foreground objects.

Simply select the objects you want to export and tick the **Export Selected only** checkbox. *Figure 15.11* shows the results of doing this with the **Export Selected only** option chosen, which will additionally crop the resulting image to the selected objects.

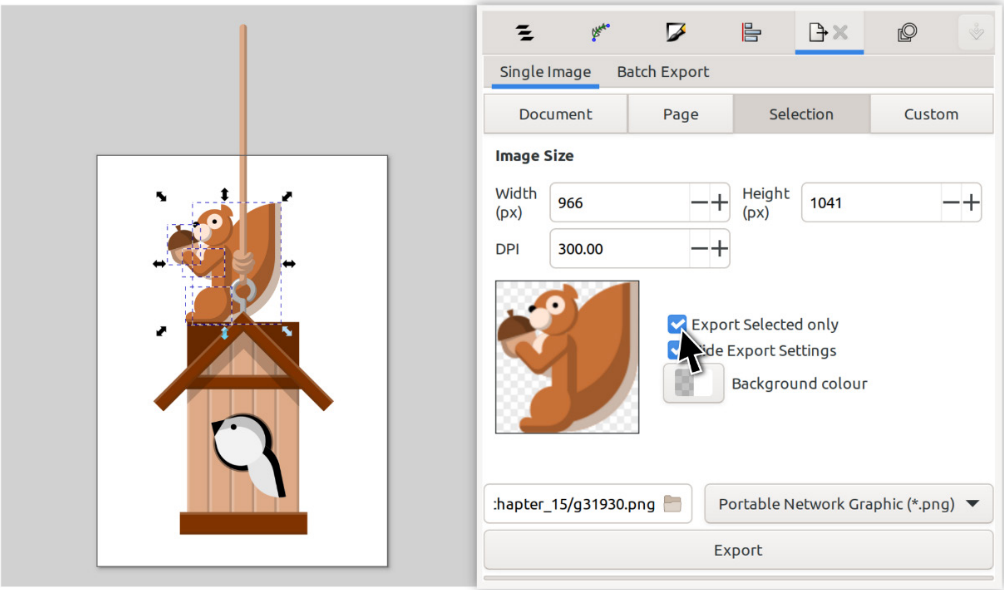


Image Size

Width (px) 966

Height (px) 1041

DPI 300.00

☑ Export Selected only

☑ ide Export Settings

Background colour

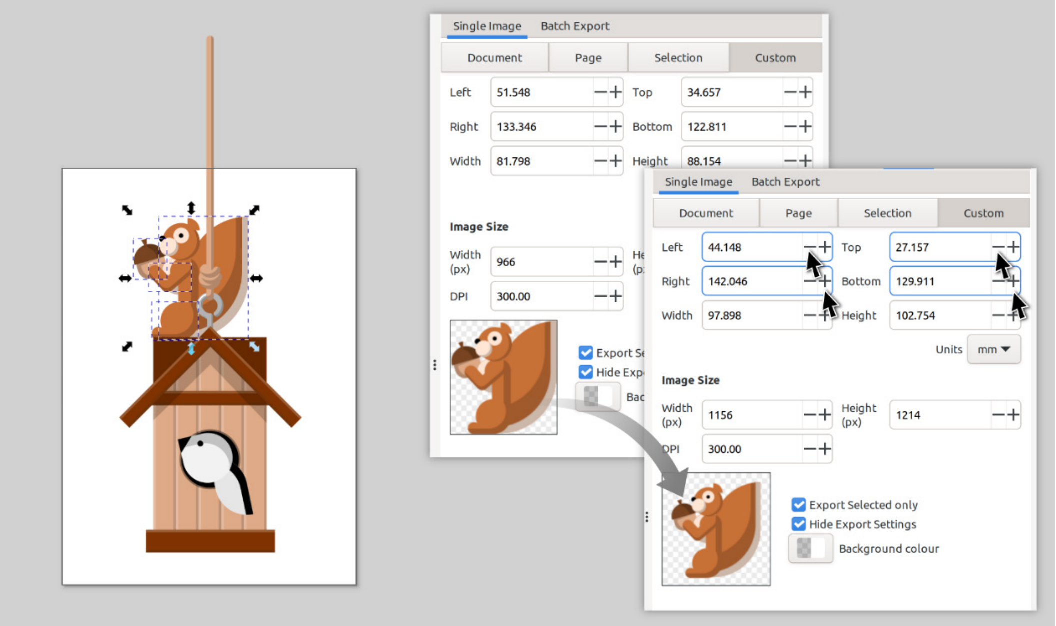
:hapter_15/g31930.png

Portable Network Graphic (*.png)

Figure 15.11 – Ticking Export Selected only exports only our squirrel and not the rope in front of him

Generally speaking, we will not be using the **Custom** export option, since the **Page** and **Selection** options give us lots of control over what's exported. However, let's say you want to export the squirrel, which is selected, but add some empty space around the resulting image.

Selecting things on the canvas automatically populates the **Custom** options with the current selection, so all you have to do is adjust the **Left**, **Right**, **Top**, and **Bottom** values to add the required padding. The preview will be updated to show you the results, as shown in *Figure 15.12*.



Left 51.548 Top 34.657
Right 133.346 Bottom 122.811
Width 81.798 Height 88.154

Image Size

Width (px) 966

DPI 300.00

☑ Export Se
☑ Hide Exp
Bac

Left 44.148 Top 27.157
Right 142.046 Bottom 129.911
Width 97.898 Height 102.754
Units mm

Image Size

Width (px) 1156 Height (px) 1214

DPI 300.00

☑ Export Selected only
☑ Hide Export Settings
Background colour

Figure 15.12 – Using Custom export options to add space around the selection

After choosing your settings, you'll of course want to fill in the filename field at the bottom of the dialog. Here, you can also choose your file format from the dropdown. The default is PNG, which is a web format that supports transparency.

More and more printers and print-on-demand websites are accepting (or requiring) JPEG or PNG as a format to upload for printing. So, the default of PNG works fine for our purposes. Once you've chosen your formats, simply click on the **Export** button and Inkscape will export an image file with the specifications you've chosen.

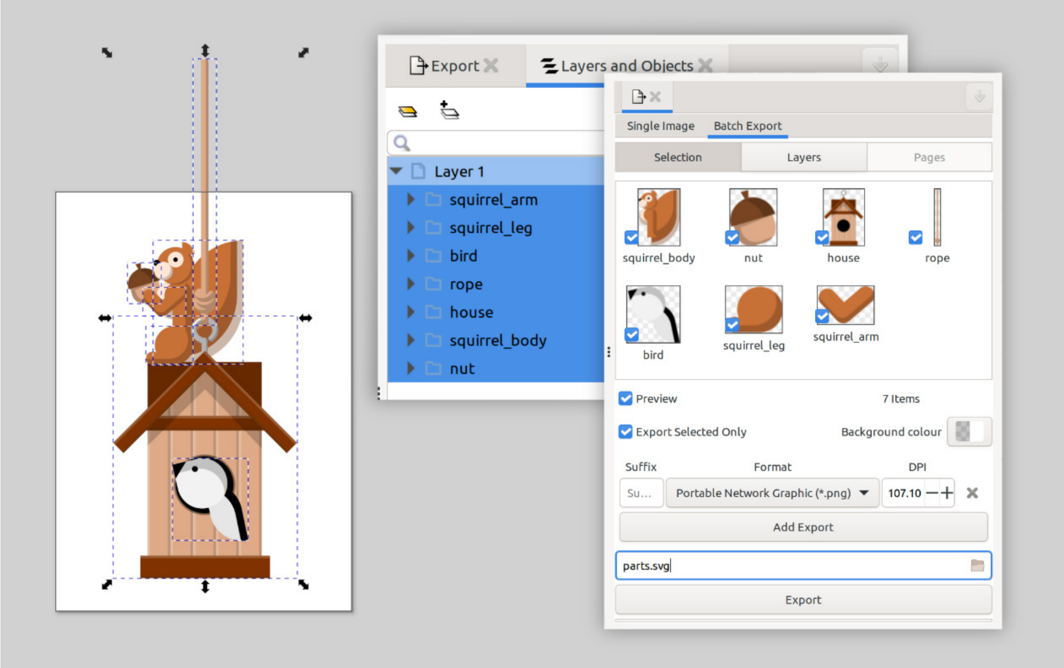
Note that if you delete the filename and file path from the filename field, you can click on the file folder icon to pop up your file browser and saves at the location in which your SVG is already saved.

Moreover, it will name the PNG file after the SVG file if you don't change the filename. This process saves you from having to type a new filename each time you save and works well if you're only saving one file. But what if you want to save more than one file? This brings us to the **Batch Export** tab!

Switching to the **Batch Export** tab in the **Export** dialog presents us with three export options:

- **Selection**: Export all selected items as individual files
- **Layers**: Export all layers, each as an individual file
- **Pages**: Export each page as an individual file

The **Selection** option is perhaps the most generally useful one for exporting, as it takes each of the selected objects and exports a file for each. In combination with the **Export Selected Only** checkbox, **Selection** will show a preview of each object, named after each object's name in the **Layers and Objects** dialog (yet one more handy reason to name your objects!). *Figure 15.13* shows my selected squirrel parts, which I need to export to animate.



☑ Preview 7 Items

☑ Export Selected Only Background colour

Suffix: Su... Format: Portable Network Graphic (*.png) DPI: 107.10

parts.svg

Figure 15.13 – Selected objects, the Layers and Objects dialog, and the resulting previews in Batch Export in the Export dialog, in Selection mode

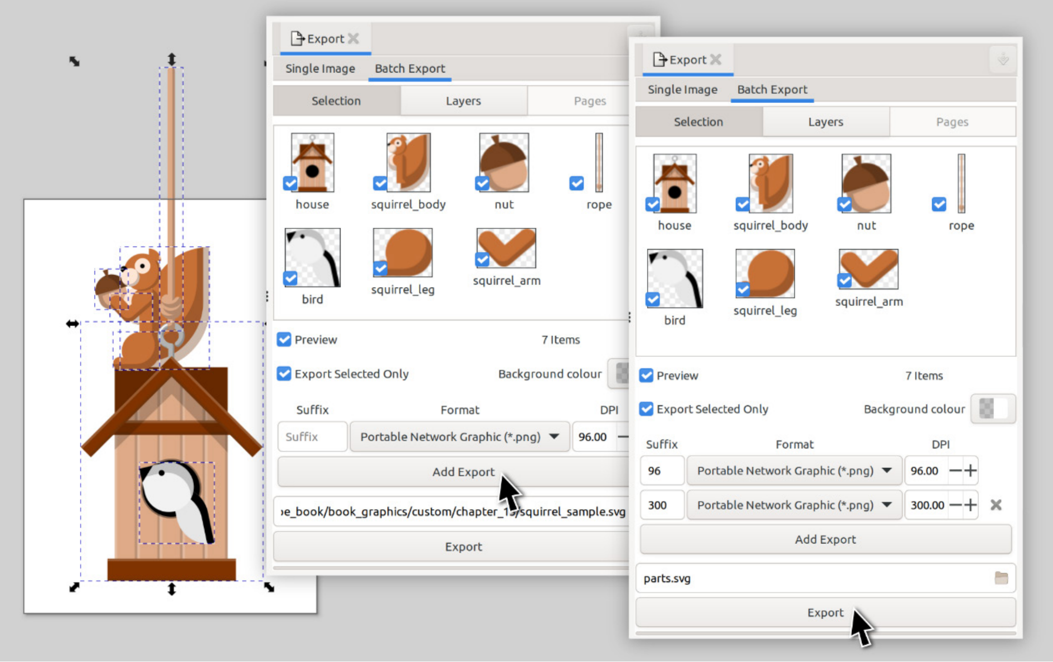
From this point on, **Export** works the same way, except that Inkscape will combine the filename with the object name and any text you put in the **Suffix** field. Thus, in the case of our example, our `squirrel_arm` will be exported to `parts_squirrel_arm.png`.

Inkscape will follow this naming convention for all the selected objects, so you'll wind up with a folder full of images. I use this feature a lot with assets for animations, as the parts need to be imported separately. If you want to see the result, check out the `squirrel_birdhouse_animation.mp4` file in the folder for this chapter.

How precisely I've made this animation is beyond the scope of this book, but you can see the parts that move are all separate, and Inkscape provides us with this great way to save time exporting and re-exporting assets separately for a variety of uses, including animation.

You may have also noticed the **Add Export** button. This allows you to add more than one output at a different DPI. This is particularly useful if you want to output, say, differently sized logos to provide to your client for a website or to print.

A setting of `96` DPI generally works well for website graphics (if the website framework doesn't support SVG), while a DPI of `300` to `600` may work well for printing. Make sure you add a suffix indicating the DPI if you're exporting more than one set of graphics. So, if we were to add an output for `96` DPI and `300` DPI, we might add `96` for the first one, and `300` for the second, as shown in *Figure 15.14*.



☑ Preview 7 Items
☑ Export Selected Only

Suffix	Format	DPI
Suffix	Portable Network Graphic (*.png)	96.00

›e_book/book_graphics/custom/chapter_1?/squirrel_sample.svg

☑ Preview 7 Items
☑ Export Selected Only

Suffix	Format	DPI
96	Portable Network Graphic (*.png)	96.00
300	Portable Network Graphic (*.png)	300.00

parts.svg

Figure 15.14 – Adding an export and appropriate suffixes for different DPI settings

This configuration will export files such as `parts_squirrel_arm_96.png` and `parts_squirrel_arm_300.png` each at the indicated DPI resolution. But what about SVG files? We can indeed choose `.svg` from the dropdown and get a series of SVG graphics.

In fact, you'll probably want to export SVG for the web as well, in case SVGs are supported by the website framework (such as WordPress, for example). It's always better to use SVGs if possible because no matter how far you zoom in, you always get a crisp edge. In the case of tablets and mobile devices, as well as high-resolution screens, one can zoom in quite a bit, for example.

We'll wrap up this chapter with some export considerations so you can choose the format that best works for your intended use of the artwork.

Save and export considerations

To round out this book, we'll now cover some of my own recommended settings for outputs of various types. This is more of a summary than a tour, so I'll make it relatively brief in the form of a bulleted list, which we will expand on in the following subsections:

- Saving for printed magazine adverts
- Saving and exporting for email-friendly PDFs
- Exporting for the web
- Saving for CNC cutters and engravers
- Exporting for print-on-demand websites

Saving for printed magazine adverts

Most magazines want `300` DPI or better for pixel graphics. I usually send exactly `300` DPI at whatever physical dimensions the page of the magazine is. Generally, they will want a PDF document, which you can choose from the **Export** dialog, or **File** > **Save As**, if you want a multipage document.

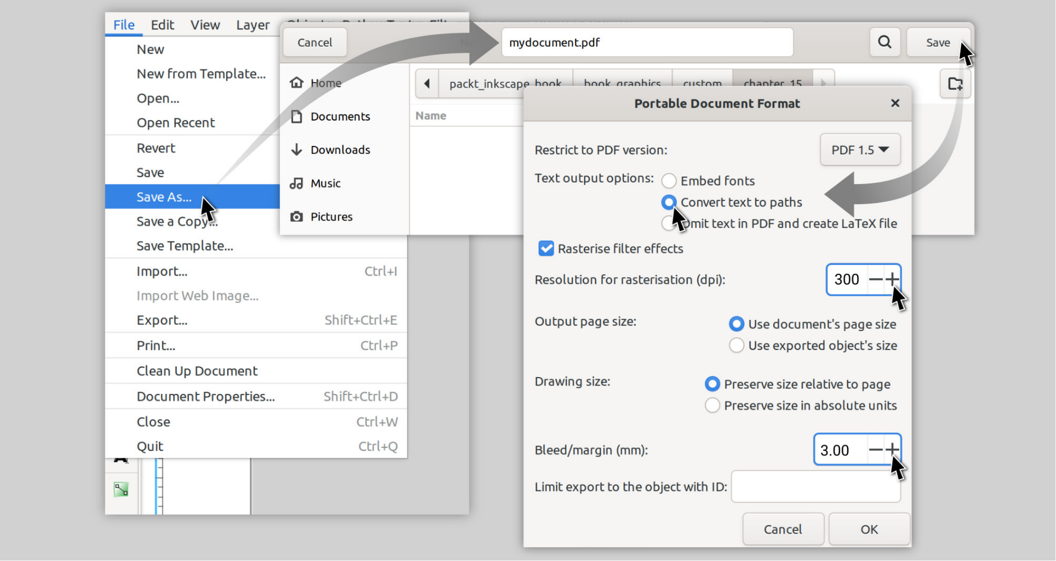
If the publisher requires CMYK conversion prior to sending, you'll need to export your file as a `300` DPI PNG and do your PDF export in a software program that specializes in print formats. Scribus is one such free and open source program that I recommend. You can choose `FOGRA27` as the colour profile for conversion and Scribus will pop out a PDF ready to go to the printers.

If you want to save as a PDF from Inkscape, you can do so by simply replacing the `.svg` part of the filename with `.pdf`. Upon hitting the **Save** button, you'll be greeted with a dialog with PDF saving options.

Here, you can include all your vector shapes as they are and tick the **Convert text to paths** option, as most printers don't want to deal with fonts they may or may not have. Ticking this option converts the text to shapes, which will make the text non-editable, but will preserve the look of your font.

Likewise, if you have any filter effects (**Blur** included), you might want to up **Resolution for rasterization (dpi)** to `300` DPI for printing. Also, please note the **Bleed/margin (mm)** box, which will add the specified length to the outside of your page. This is a good thing to set if your graphics extend all the way to the edge of the printed page, as it gives extra area for the printers to trim down the overprinted areas.

Most printers require a `3mm` bleed area, which you can add in this same **PDF** dialog. These options are shown in *Figure 15.15*.



mydocument.pdf

Portable Document Format

Restrict to PDF version: PDF 1.5

Text output options: ○ Embed fonts ● Convert text to paths ○ Omit text in PDF and create LaTeX file

☑ Rasterise filter effects

Resolution for rasterisation (dpi): 300

Output page size: ● Use document's page size ○ Use exported object's size

Drawing size: ● Preserve size relative to page ○ Preserve size in absolute units

Bleed/margin (mm): 3.00

Limit export to the object with ID:

Figure 15.15 – Saving as a PDF for printing

Saving and exporting for email-friendly PDFs

If you're exporting a PDF to attach to an email, you will naturally want to make your PDF as small as possible. You will likely also want to keep your text as text instead of converting to paths, as it's more easily readable by screen reader software and copyable by selection in PDF readers.

One trick is to export all your pixel images as JPEGs (choose `.jpg` from the **Export** dropdown rather than `.png`). Then, you will want to link these back into your document in place of the originals. You don't have to delete the originals, though. You can simply put the original PNGs on a separate layer and hide those layers – when you save as a PDF, Inkscape will automatically exclude any hidden layers, reducing the file size considerably in the process.

Exporting for the web

I tend to switch to **px** as the document units when working with web graphics. This is because often, I find I have to design different pixel-dimensioned screens to create what are called *responsive websites* – websites that load different graphics and layouts based on the size of the screen being displayed.

Of course, if you have the luxury of designing for a website framework that supports uploading and using SVGs, you can largely ignore any and all measurements, as you can set dimensions in the website code and the SVG will produce the best resolution for the screen on which it ends up showing. Thus, I export to SVG when possible and PNG otherwise, since PNG supports transparency.

This might be a bit heavy on file size for large images with lots of gradients or details, so if you can do without transparency, it's also worth considering exporting as JPEG to cut down the load times of your websites. As of Inkscape 1.1, exporting to the upcoming `.webp` format is also an option, which gives you better quality at lower file sizes, so it's worth considering that as a possible alternative to JPEG or PNG.

Saving for CNC cutters and engravers

Generally speaking, you'll want to export an SVG to be read into whatever software controls your CNC machine or engraver. Inkscape can produce something called **GCode**, which is basic instructions for how to move the engraving/cutting head and can be sent directly to the printer.

However, you will generally get better results using the software that comes with your device, and in certain cases, that software may save you from exporting instructions that may damage your cutter. Still, if you're feeling brave, you can use **Extensions** > **Export** > **Plot** and fiddle with the many settings there.

Otherwise, just export to a simple SVG (no gradients or fills), and load the SVG file in your CNC software. Typically, this should be compatible with most CNC machines, as SVG is a standard format.

Exporting for print-on-demand websites

This is one area where PNGs or JPEGs are the preferred format. At some point in the last decade, large commercial printers figured out that they can get more customers if they accept common formats such as JPEG, which most consumer-grade cameras provide by default, and most certainly do *not* support CMYK for printing.

So, if your printer demands a CMYK file, you're well within your rights to choose a different printer that can handle sRGB-to-CMYK color conversion. After all, what kind of printer can't print a JPEG photo? Most websites such as VistaPrint provide excellent quality prints on a variety of items and accept PNG as the format.

To print PNGs or JPEGs, there are a few considerations, though. Getting the exact color you see on your screen in print is difficult, because most computer screens are not color calibrated. Thus, that nice shade of purple you're seeing may in fact be dark blue on the color-calibrated monitors that printers are likely to use. So, what do we do? Let's look at some options in the final section of this chapter.

Printing

One solution is to get color calibration equipment, but before you rush out and get an expensive kit, there are a few tricks you can try instead to get close enough to the correct colors for most eyeballs out there.

Most smartphones have color-calibrated screens, so one way to check whether the colors you're seeing are accurate is to email the PNG file to yourself and open it on your phone. If the colors match, you'll be good to go 99% of the time. Another option is to print this color test chart I've made (see *Figure 15.16*), which is included in the materials for this chapter.

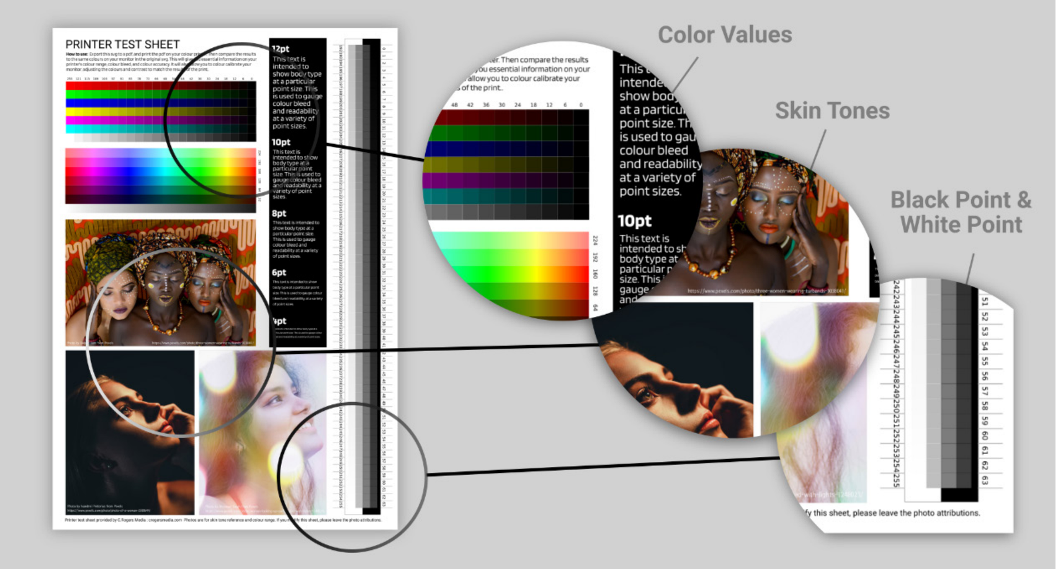


Figure 15.16 – Color test sheet to send to the printer

The trick is, you send this PDF (made in Inkscape) to the printer and have it print you one copy of it. When you receive the print in the mail, you can then compare it to the same file opened in Inkscape to see whether there's any color shift. You can then manually adjust your monitor color settings to match, or simply choose a color from the sheet that better fits what you're expecting.

This sheet contains a lot of useful tests other than just color, such as skin tone accuracy and the black point and white point your printer's printer can handle. The *white point* is how light a gray can be printed before it looks pure white in print, and likewise, the *black point* shows how dark a color can be before it looks black in print. This will vary from printer to printer, so it's a good way to know whether, say, that light gray background will turn out the way you want it to.

I've also included a bleed test for white-on-black text, which should give you a good idea of the smallest white-on-black font size you can use before the black ink bleed starts to make the text unreadable.

If you intend to use different kinds of paper, I recommend printing this sheet out on one of each, including gloss, silk, and matte finishes. These different surfaces affect the saturation of the resulting color in quite striking ways, allowing you to adjust the colors based on the results.

One last thing: if you're having difficulty printing your Inkscape files on your own desktop printer directly out of Inkscape, try saving your Inkscape file to a PDF and then opening the PDF in a PDF reading program, or even your web browser. Since PDF is a print delivery format, it contains all the sizing information you need to print.

Indeed, this is why printers often require `.pdf` documents for magazines. Make sure when you print that there is no automatic scaling; otherwise, your measurements will be a bit off. If this happens, simply untick the **Scale to fit page** option (or similar) in the print settings on your system.

Summary

Well, that just about wraps it up! In this chapter, we learned about the many tools Inkscape has to format documents and get the output files you need for your projects, and a bit about getting the best results for your printed artwork.

Congratulations on making it all the way through this book, and feel free to come and join us on the Inkscape forums if you have any questions and to share your creations with the rest of the Inkscape community. There are a ton of very knowledgeable people there (including myself), eager to share what they know about Inkscape. Good luck with your projects!

Index

Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:

‹packt›
1ST EDITION
Inkscape
by Example
A project-based guide to exploring the endless features
of Inkscape and upgrade your skills
ISTVÁN SZÉP

Inkscape by Example

István Szép

ISBN: 978-1-80324-314-6

- Create your own professional solutions following the project blueprints in this book
- Integrate Inkscape and other free design programs in a professional workflow efficiently
- Use Inkscape for more than illustration - design a logo, edit photos, and even design websites in vector
- Get to grips with troubleshooting, export the proper format, and find and correct errors in vector files
- Gain no-fluff practical knowledge of the tools in Inkscape and vector design methods
- Broaden your understanding of the fundamentals of vector designing

‹packt›
1ST EDITION
Real-time animation with Adobe Character Animator
Animate characters in real-time with webcam, microphone, and custom actions
CHAD TROFTGRUBEN

Real-Time Animation with Adobe Character Animator

Chad Troftgruben

ISBN: 978-1-80324-694-9

- Assemble rigs in Photoshop for Character Animator
- Add tags, handles, draggable limbs, swap sets, and physics effects
- Use a webcam and microphone for lip-syncing, head turns, and other real-time actions
- Work with multiple characters, scene layers, and audio files
- Add loopable motions such as walk cycles
- Record a short animation for export

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit `authors.packtpub.com` and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Share Your Thoughts

Now you've finished *Design Made Easy with Inkscape*, we'd love to hear your thoughts! If you purchased the book from Amazon, please select `https://www.amazon.in/review/create-review/error?asin=1801078777` for this book and share your feedback or leave a review on the site that you purchased it from.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere?

Is your eBook purchase not compatible with the device of your choice?

Don't worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don't stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below

`https://packt.link/free-ebook/9781801078771`

2. Submit your proof of purchase
3. That's it! We'll send your free PDF and other benefits to your email directly