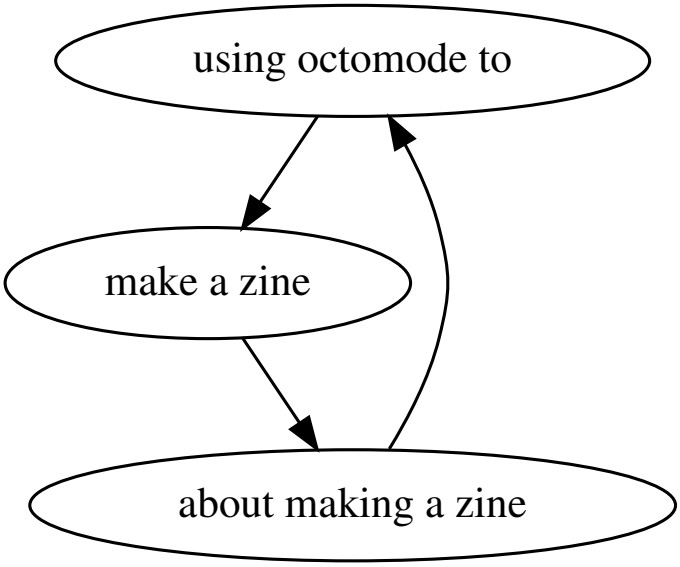
so, you decided to make a zine using *octomode*



contents

what's a zine?

"Zine" is short for "fanzine" or "magazine". Zines are a small-circulation publications, usually made on a copy machine or with whatever technology is available. They are also often characterised by their DIY (Do It Yourself) ethos. Historically, zines have been made about a vast array of topics, from personal stories, to queer and LGBTQI+ culture, to feminism(s), to social and political activism, to music scenes, to HOWTOs and tutorials…

Here are some examples:

printed zines

- https://psaroskalazines.gr/zines
- https://daap.network/zinesindex
- https://iffybooks.net/zines

web zines

- https://coolguy.website/web-zine/01
- https://htmlzineclub.usersentimentalexperience.net/

what's this zine about?

This zine is an ouroboric guide to making zines with *octomode*, a collective editing space for PDF making in a web browser (such as Firefox, Chrome/Chromium, etc).

Octomode is free and open-source software made by Varia, a collective-space for everyday technology in Rotterdam. We use it to make lots of collaborative documents, such as our occasional publication SomeTimes. And to feed the snake its own tail even more, this zine was also made using octomode!

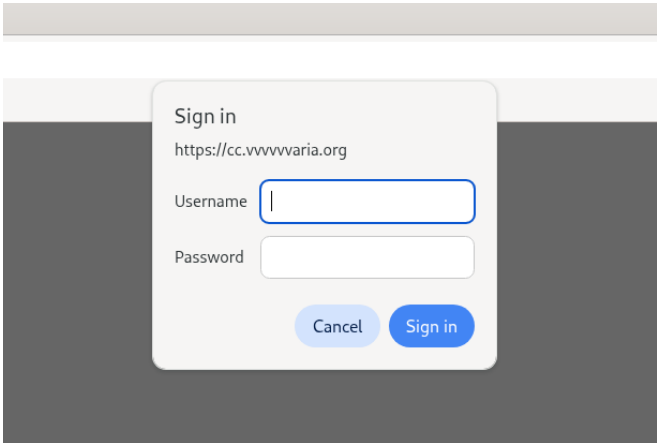
getting tentacular with *octomode*

Octomode is available to use on the CC (creative crowds) server. Please read the conditions and add your publication to the calendar:

https://cc.vvvvvvaria.org/pad/p/cc-conditions

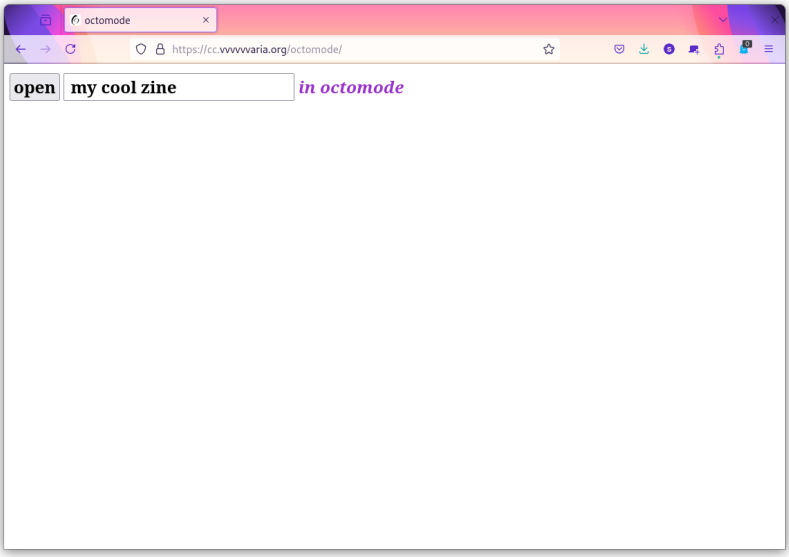
To use octomode, go to this URL (yes, that is with 6 “v”s). You will need login credentials, which are shared only in a network of trust (ask a friend):

https://cc.vvvvvvaria.org/octomode



The login for octomode

Once you log in, you will see the following page:



making a new octomode environment

Write a name for your octomode production environment (for example `my cool zine`), and click **open**.

now, the magic happens 🪄

Octomode depends on the free and open-source software Etherpad, which is often used for collaborative text-editing in a web browser. This means that many people can write in the same “pad”, synchronously. At Varia, we use Etherpad for collective note-taking, and for writing code together. We also self-host the software on our server (at https://pad.vvvvvvaria.org) for members and friends of Varia to use.

octomode buttons (pad, stylesheet, html, pdf)

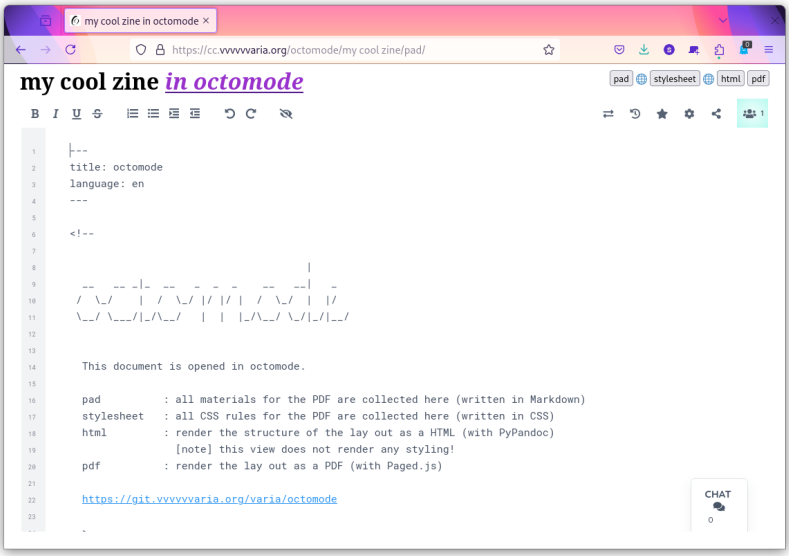
After you open an octomode environment, you should see some buttons at the top right of the browser window:

buttons for content, stylesheet, HTML preview and PDF preview

Each of these buttons opens a different part of an octomode environment. So, you will need to get used to a lot of clicking back and forth between the different parts!

Let’s take a tour of the interface, beginning with the “pad” button.

pad



pad for content

Clicking the “pad” button opens an Etherpad where you can write the **content** of the publication, in either HTML (HyperText Markup Language) or a relatively simpler language called Markdown. We are going to use Markdown in this guide, so here are some basic examples:

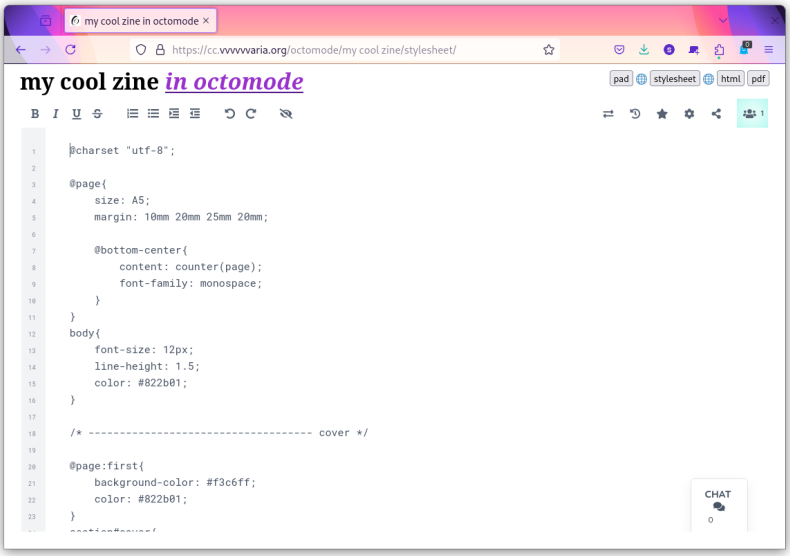
```
# heading 1
## heading 2
### heading 3
#### heading 4

**bold text**
*italics*

[linked text](https://mycoolwebsite.com)

![image caption](https://mycoolwebsite.com/image.jpeg)
```

stylesheet



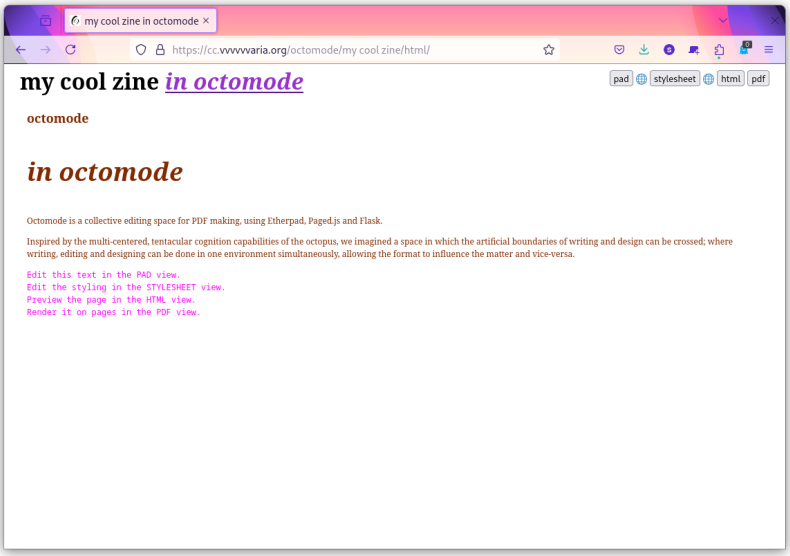
pad for stylesheet

Clicking the “stylesheet” button opens up an Etherpad where you can write CSS (Cascading StyleSheets). This affects the **style** of a document.

Each rule written in CSS includes a *selector*, a declaration of a *property* and the *values* it is given. Each selector corresponds to an HTML element. For example, if you want the text value to be `monospace`, use the `font-family` property on the `body` selector:

```
body {
    font-family: monospace;
    }
```

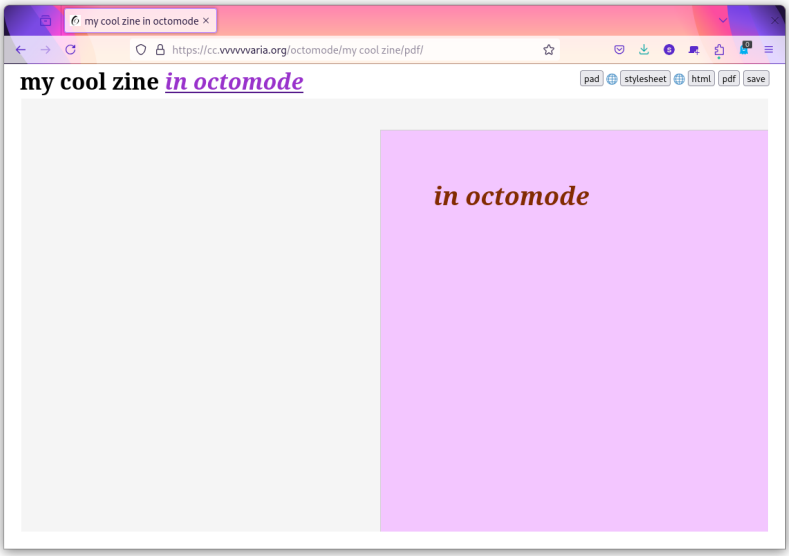
html



HTML preview

HTML (HyperText Markup Language) is one of the building blocks of the web. It provides a **structure** for a webpage. By clicking on the `html` button, you can preview how the octomode document would look as a webpage.

pdf

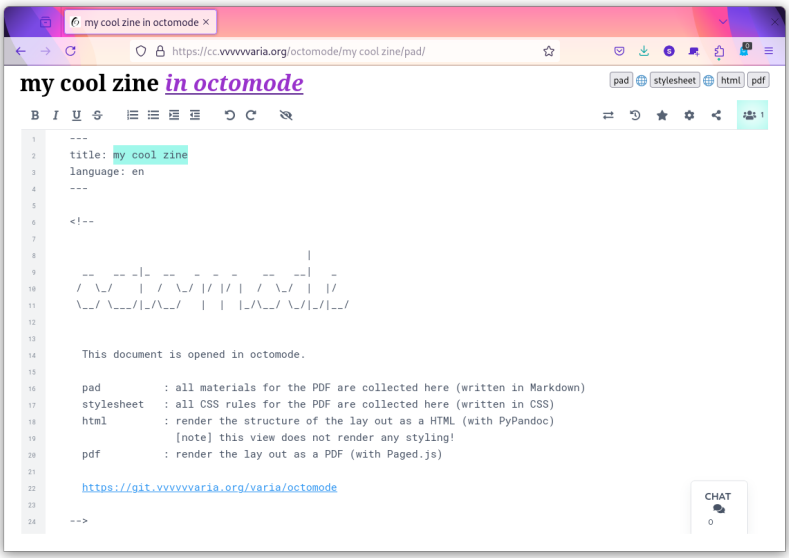


PDF preview

When you click the pdf button, you can see a preview. Here you can see how the PDF would look with a cover and left and right pages. Remember, this is just a preview in the web browser! You will need to save the PDF first, to see exactly how it will look when it is exported. There is more on that later in this zine, in the section on getting ready to save the pdf.

But first, let's make some changes, such as giving a new filename, changing the title that will appear on the cover, and adding some headings, images and text.

changing the pdf filename



changing the name of the PDF file

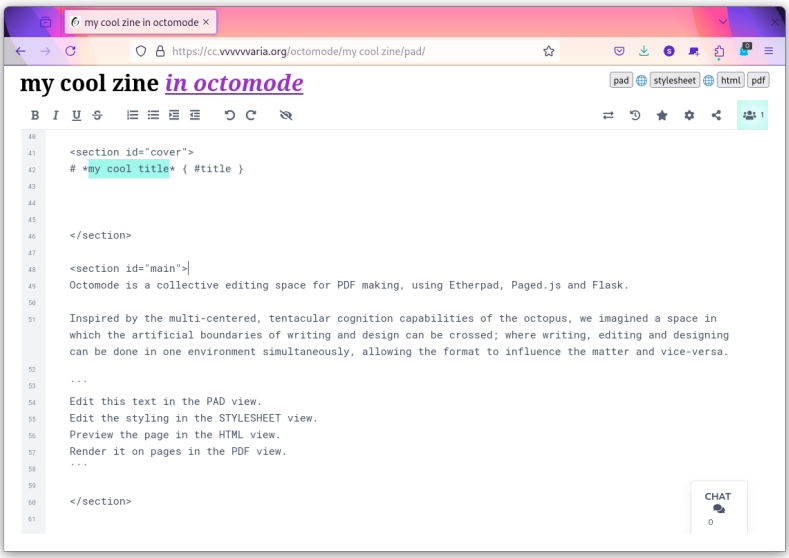
At the top of the the pad for content is a metadata block. By default, the title is `octomode` and the language is English (`en`). `title` is only there to make the filename of the PDF when it is exported, and `language` determines hyphenation handling. But there **must be metadata for the title and language**, and there **must be three hyphens** (`---`) opening and closing the metadata block, with **no text before them**.

Change the metadata block to give a different filename to your PDF:

```
---
title: my cool zine
language: en
---
```

⚠ If something goes wrong when you try to view the PDF, check that this metadata block is written correctly! ⚠

changing the title on the cover



changing the title of the zine

Let's go ahead and change the title that will appear on the cover of the zine. Scroll down to the `section` the `cover` `title` is in.

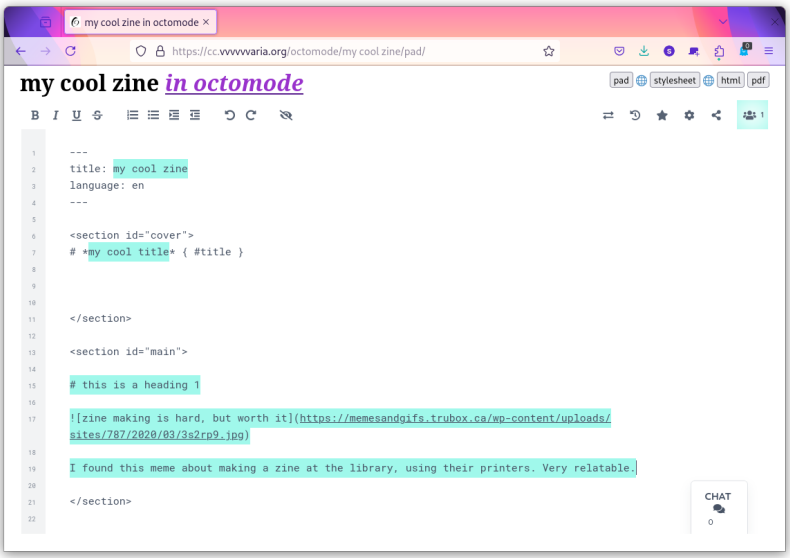
Change the text to whatever you want, in this case we will use "my cool title". This is a heading 1, which will appear in italics:

```
# *my cool title* { #title }
```

The `{ #title }` part gives an id (a unique identifier) of "title" to the heading, which we can style differently from the other heading 1 text using CSS:

```
h1#title {
    font-size: 400%;
    }
```

adding a heading, an image and some text



adding some things in the content pad

Let's try adding a heading 1 inside the `main` section:

```
# this is a heading 1
```

Octomode does not store any images. These must be hosted somewhere online at a URL. An image link is made with the following syntax:

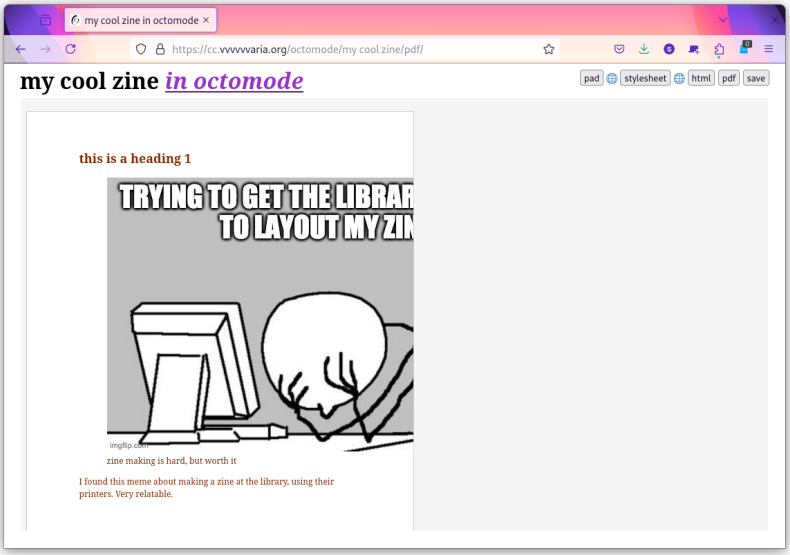
```
![image caption](https://path/to/the/URL/image.jpeg)
```

In this example, `image caption` represents the caption below the image, and `https://path/to/the/URL/image.jpeg` represents the URL to the image.

⚠ The image link must begin with an exclamation mark `!`, followed by square brackets `[]` and the image URL inside parentheses `()` ⚠

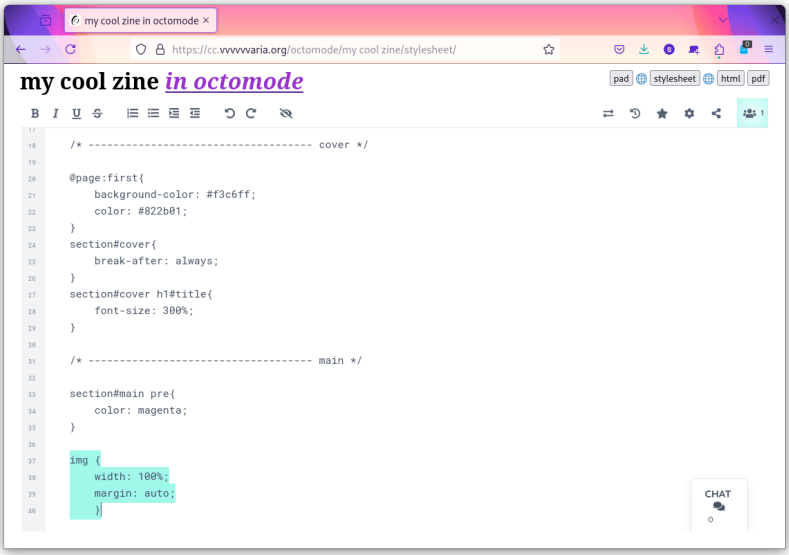
checking your PDF

Click the pdf button, to see your changes!



the image is trying to escape the page...

Oh no! the image is too wide! Let's fix that with some CSS.



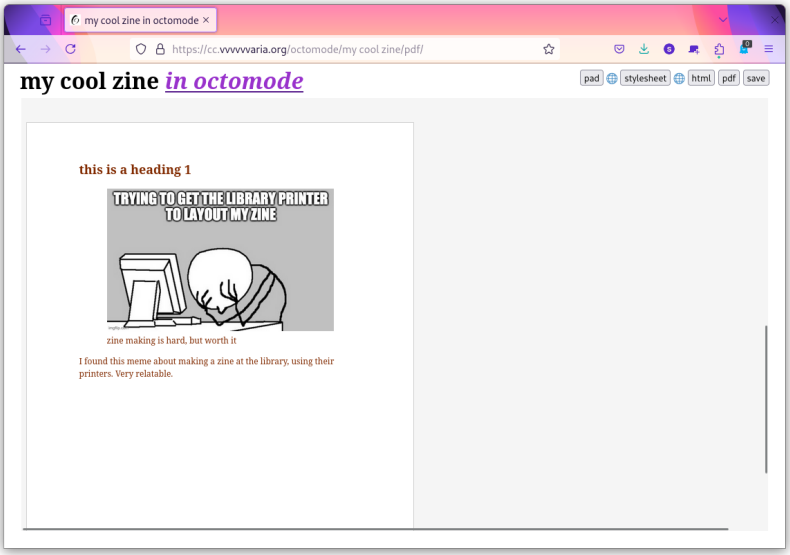
making the image fit inside the page by editing the stylesheet

To make an image fit, we should give it a width of `100%` and set the margin to `auto`:

```
img {
    width: 100%;
    margin: auto;
    }
```

setting image size with CSS

Check your changes by clicking the pdf button again:



now the image fits!

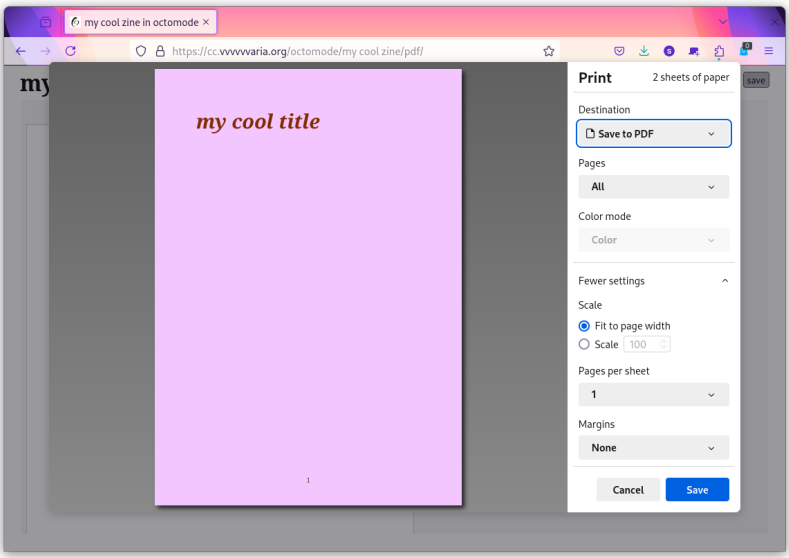
Go ahead and add some more images and text. You can write paragraphs in markdown without any special syntax. Just make line breaks between each paragraph by typing the “return” or “enter” key. If you don’t use any line breaks, the text will automatically wrap to the next line:

```
I found this meme about making a zine at the library, using their
printers. Very relatable!
```

If you want to go deeper into CSS-print, the pagedjs web design for print documentation is a good place to start.

When you are ready, you can save the PDF!

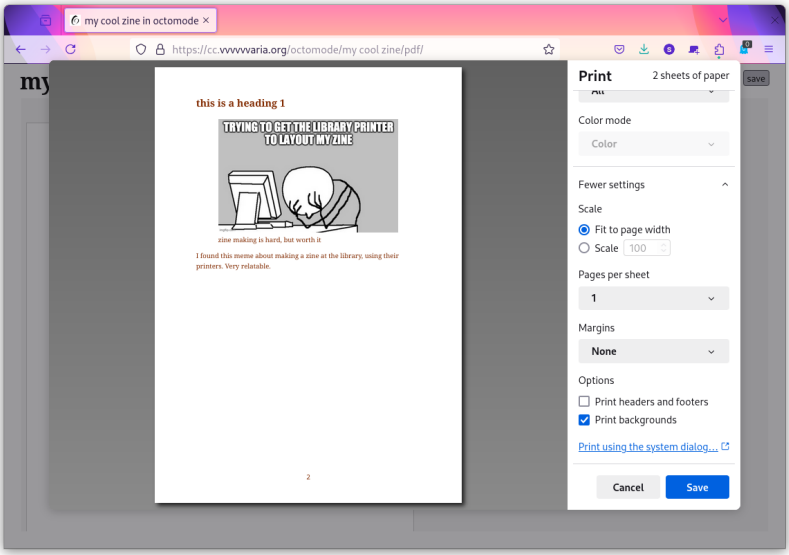
getting ready to save the pdf



exporting the zine to PDF

In the PDF preview, you should see a new button appear to save the PDF. Clicking save does something similar to pressing `ctrl/cmd + P` - a common way to export a webpage as PDF, or send a PDF file to a printer.

One nice thing about web browsers is that they come with a lot of software that not only knows how to render web pages, but also gives you options for how to print or export them as PDFs. Many web browsers let you choose the destination (such as to save to PDF, or send the PDF to a printer), how many pages to export, the color mode, the scale, how many pages per sheet, the size of margins, and if you want to include background colors or not.



checking “Print backgrounds”

exporting a pdf file

We are going to choose the option `Save to PDF`, as we want to export a PDF file.

For the **scale**, choose `Fit to page width`.

For **Pages per sheet**, choose `1`.

For **Margins**, choose `None`.

If your PDF has color backgrounds, choose `Print backgrounds`.

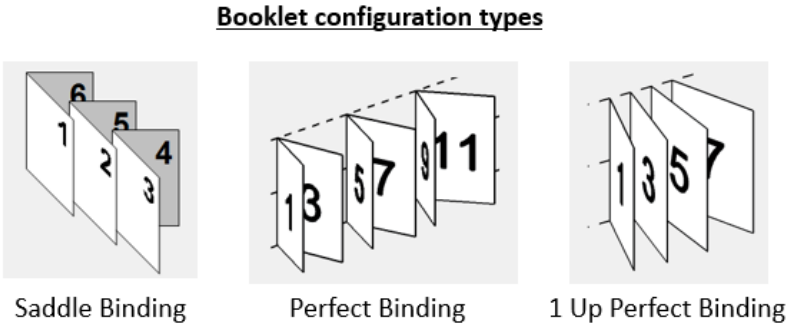
Then click `Save`.

so, we have a pdf, what now?

That is up to you!

A couple of possibilities for printing and binding the zine:

- as a **stack of sheets**, stapled at the top left corner, or along the left edge (perfect binding)
- as a **booklet**, folded in half and stapled in the center (saddle binding)
    - for a booklet you will need to figure out the imposition, which means how multiple pages are arranged on a sheet of paper when it is printed. Here are some free and open-source tools to help you do that:
        - pdfbook2: https://github.com/jenom/pdfbook2
        - pdfcpu: https://pdfcpu.io/
        - pdfjam: https://gitlab.com/quentinjuhel/Bookletor



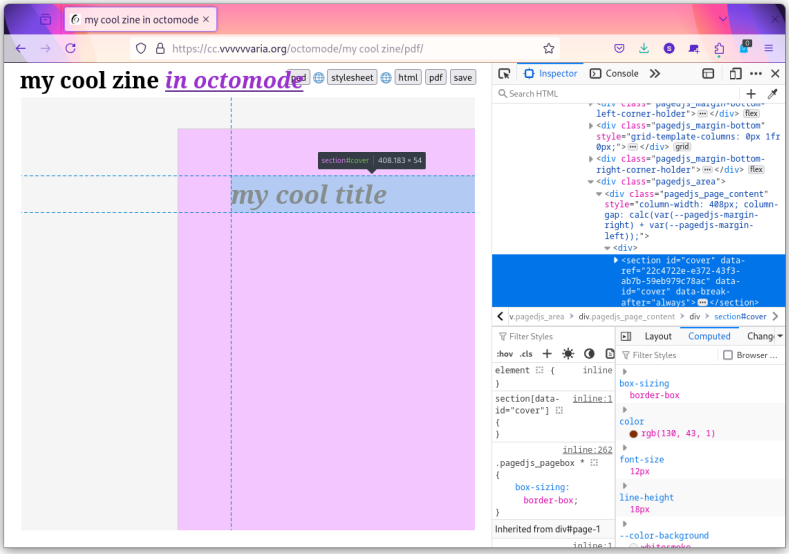
different types of booklet binding

Or, upload the zine somewhere it can be downloaded from. Or, email it around to your friends. The nice thing about making digital zines is that you can include links (like the ones Olia Lialina writes about) which will take a reader off to other places (other pages in the same zine, or web pages online). Choose your own adventure!

using a web inspector for debugging

If your design looks a bit strange and you want to check why, or try out other CSS quickly, try using a **web inspector** to inspect the source code of a web page, and edit CSS without committing the changes. Just remember, any edits to the page are only visible on your computer, and they will disappear when you refresh the page.

Most web browsers have a web inspector tool that can be accessed by doing a `right click` or `ctrl + click` or `cmd + click` on a web page, then selecting `Inspect`. You will then see a panel open with many tools, for example an editor which shows the HTML structure of the web page, and also any CSS that is used to style it. You can click on an element to highlight it, or pick an element in the page. This is useful for inspecting how the browser is actually rendering the content and style you gave your PDF.



inspecting the web page in Firefox

so, how does octomode *actually* work?

Octomode is a web-to-print publishing *environment* for making collaborative PDFs in a web browser, where different types of writing (editorial, technical) come together in the same space. Inspired by the multi-centered, tentacular cognition capabilities of the octopus, we imagined a space in which the artificial boundaries of writing and design can be crossed; where writing, editing and designing can be done in one environment simultaneously, allowing the format to influence the matter and vice-versa.

The technical stack of software includes:

- Flask, a web application framework which glues all the parts of octomode together
- Etherpad-Lite, a collaborative, browser-based text-editing software, so that many people can write content and design the document in the same environment
- PyPandoc, a Python wrapper for the document conversion software Pandoc to convert the content pad to HTML, so that content written in Markdown can be rendered to HTML in a browser
- CSS standards for paged media, which are organised by the W3C (World Wide Web Consortium), so that the rendered HTML looks like a page, with a fixed size, margins, etc
- paged.js, a JavaScript library to render a PDF preview in a web browser, so that it can then be exported as a file

To read more about how octomode works, how it has been used already, or to self-host your own octomode:

- https://cc.vvvvvvaria.org/wiki/octomode
- https://git.vvvvvvaria.org/CC/octomode

other (non-octomode) ways are possible!

There are quite a few projects that also make use of what is being known as *web-to-print*, or perhaps more accurately, *CSS-print* (using CSS and a web browser) to make PDFs. Many of these projects come from FLOSS (free/libre open-source software) design and publishing, a culture that has existed for almost 20 years, mostly in the Netherlands, Belgium and France.

Here is a list of a few other CSS-print softwares and projects. This list is by no means comprehensive, but instead serves as a sampler to whet the appetite:

By OSP (Open Source Publishing) http://osp.kitchen

- ether2html https://osp.kitchen/tools/ether2html
- html2print http://osp.kitchen/tools/html2print
- ethertoff http://osp.kitchen/tools/ethertoff

By Luuse https://luuse.io

- pad2print https://gitlab.com/Luuse/pad2print

By many in the PrePostPrint https://prepostprint.org network

- pinkmypad https://pinkmypad.net
- WeasyPrint https://weasyprint.org
- A collection of starter kits for paged.js https://pagedjs.org/posts/2020-04-15-starterkits-for-pagedjs
- PPP Demos - including CSS print with paged.js, and also without JavaScript https://gitlab.com/prepostprint/demos
- The Web 2 Print Library http://web.2print.org/fr

This zine was made by Simon Browne in June 2025

with *octomode* in Manorhamilton/Cluainín, County Leitrim, Ireland

on the occasion of the workshop “so, you decided to make a zine”

as part of his exhibition residency at the Leitrim Sculpture Center

a digital version of this zine is available at:

https://cc.vvvvvvaria.org/pdf/so-you-decided-to-make-a-zine-using-octomode.pdf