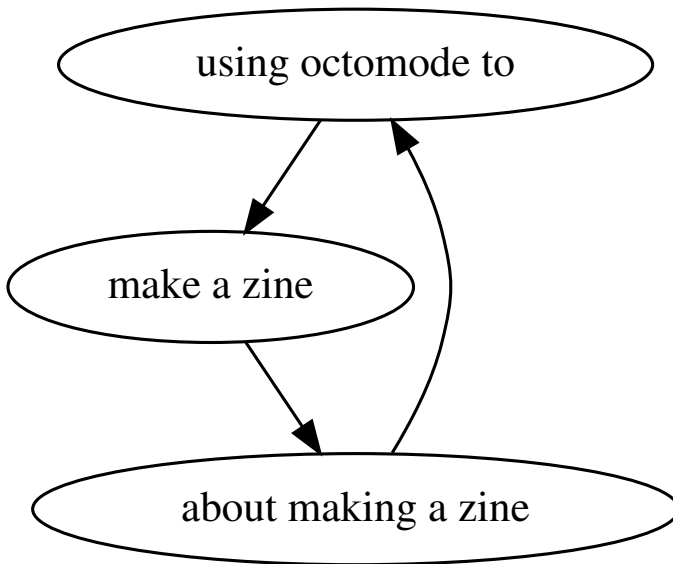


**so, you decided to
make a zine in
*octomode***



contents

4	what's this zine about?
4	getting tentacular with <i>octomode</i>
7	octomode buttons
12	changing the pdf filename
13	changing the title on the cover
14	adding some content
15	checking your PDF
16	setting image size with CSS
18	debugging with a web inspector
19	getting ready to save the pdf
20	so, you have a PDF, what now?
21	so, how does octomode <i>actually</i> work?
22	other (non-octomode) ways are possible!

what's this zine about?

This zine is an [ouroboric](#) guide to making zines in *octomode*, a collective editing space for PDF making in a web browser (such as Firefox, Chrome/Chromium, etc). And to feed the snake its own tail even more, this zine was also made in octomode!

getting tentacular with *octomode*

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.

Octomode is available to use on the CC (creative crowds) server, which is currently maintained by Manetta Berends and Simon Browne:

<https://cc.practices.tools>

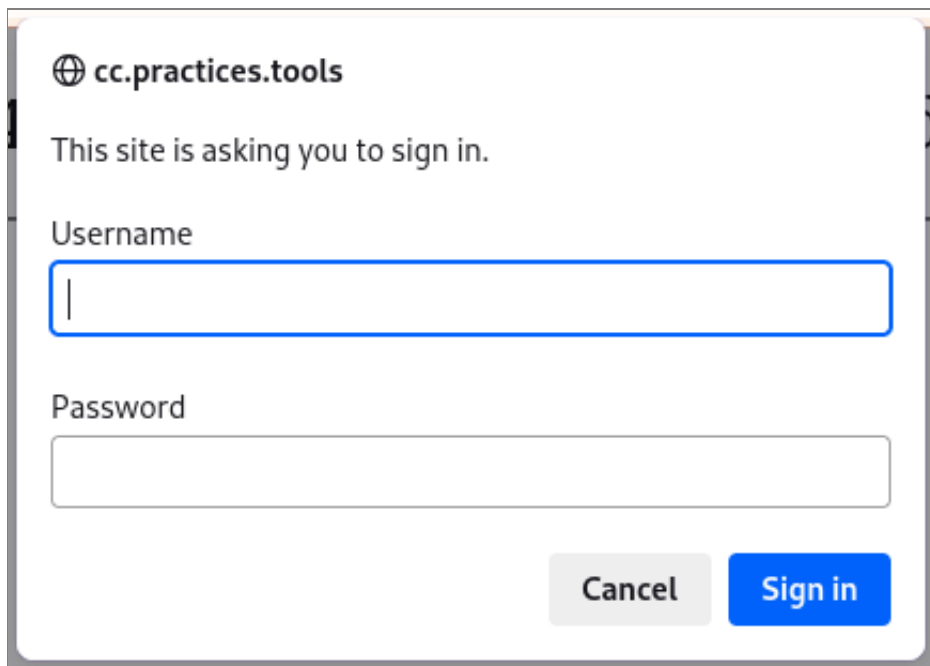
Please read the conditions and if you agree, add your publication to the calendar:

<https://cc.practices.tools/pad/p/cc-conditions>

To use octomode on the CC server, go to this URL:

<https://cc.practices.tools/octomode>

You will need login credentials, which are shared only in a network of trust (ask a friend):

A login form for the website cc.practices.tools. The form is displayed in a browser window. At the top, there is a globe icon followed by the text "cc.practices.tools". Below this, a message states "This site is asking you to sign in." The form contains two input fields: "Username" and "Password". The "Username" field is currently empty and has a blue border. The "Password" field is also empty. At the bottom right of the form, there are two buttons: a grey "Cancel" button and a blue "Sign in" button.

cc.practices.tools

This site is asking you to sign in.

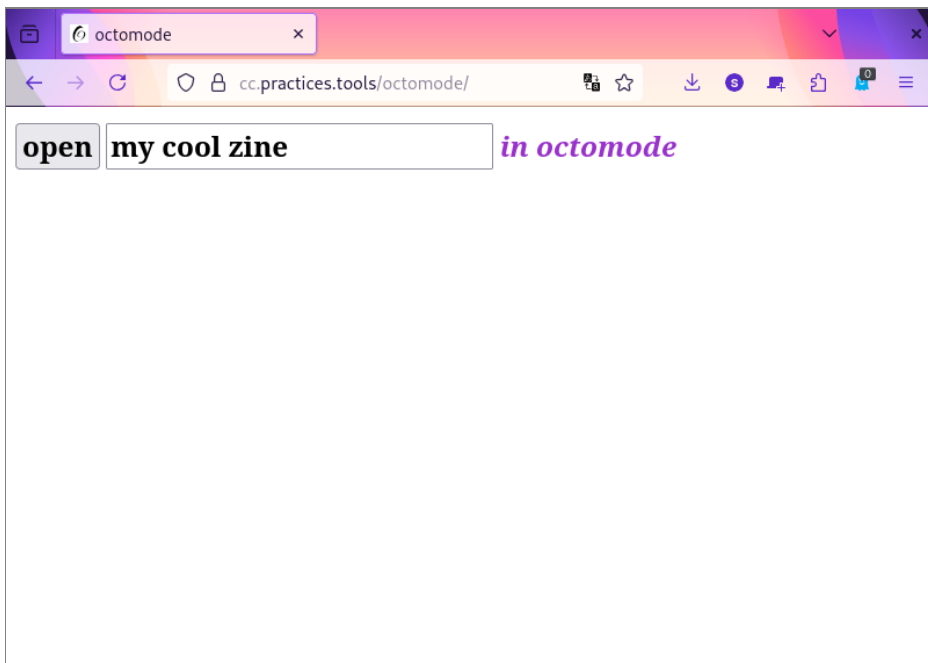
Username

Password

Cancel Sign in

The login for octomode

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

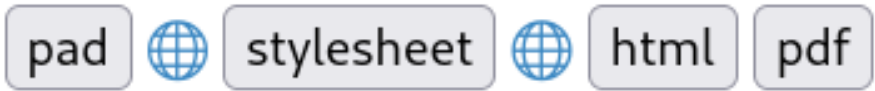


making a new octomode environment

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

octomode buttons

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. Here are some basic examples:

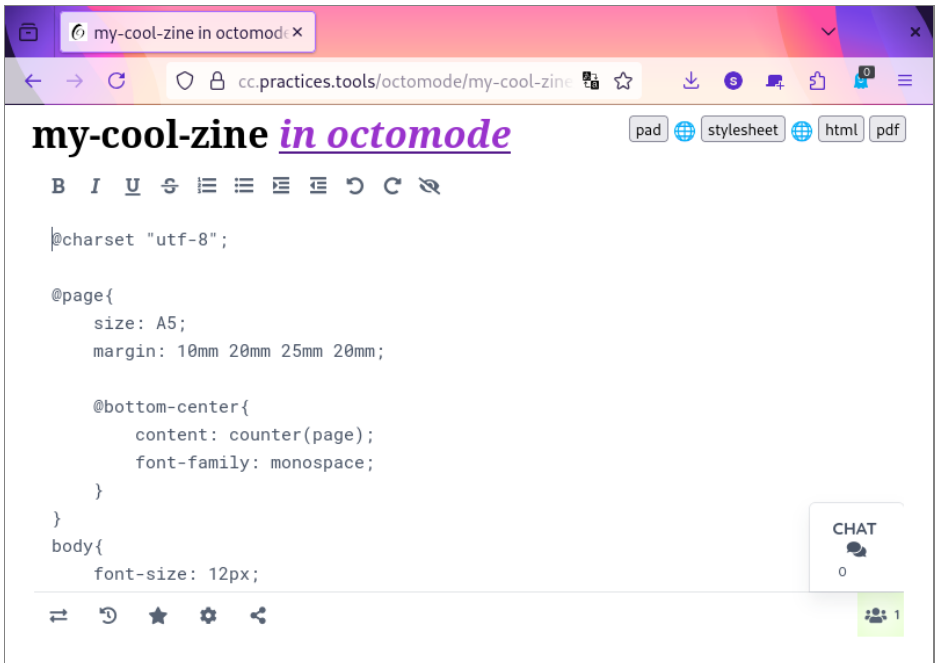
```
# heading 1
## heading 2
### heading 3
#### heading 4
```

```
**bold text**
*italics*
```

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

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

stylesheet



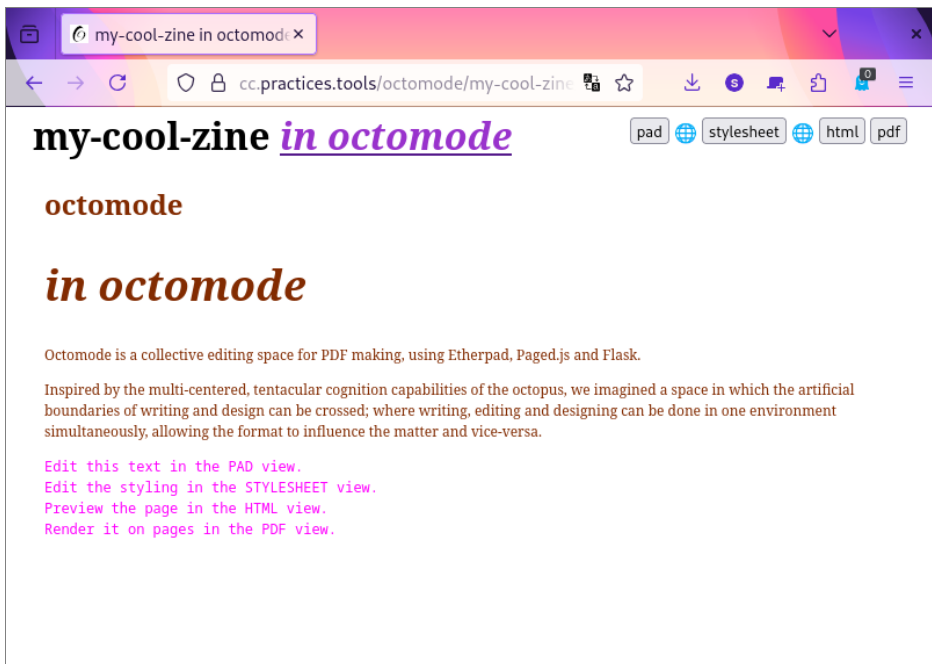
pad for stylesheet

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;
}
```

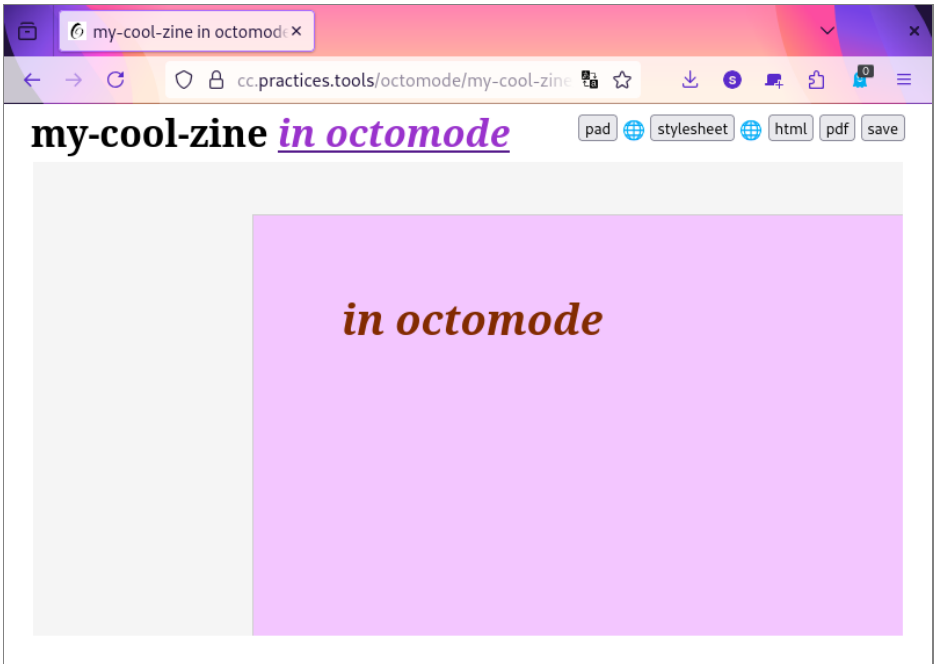
html



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

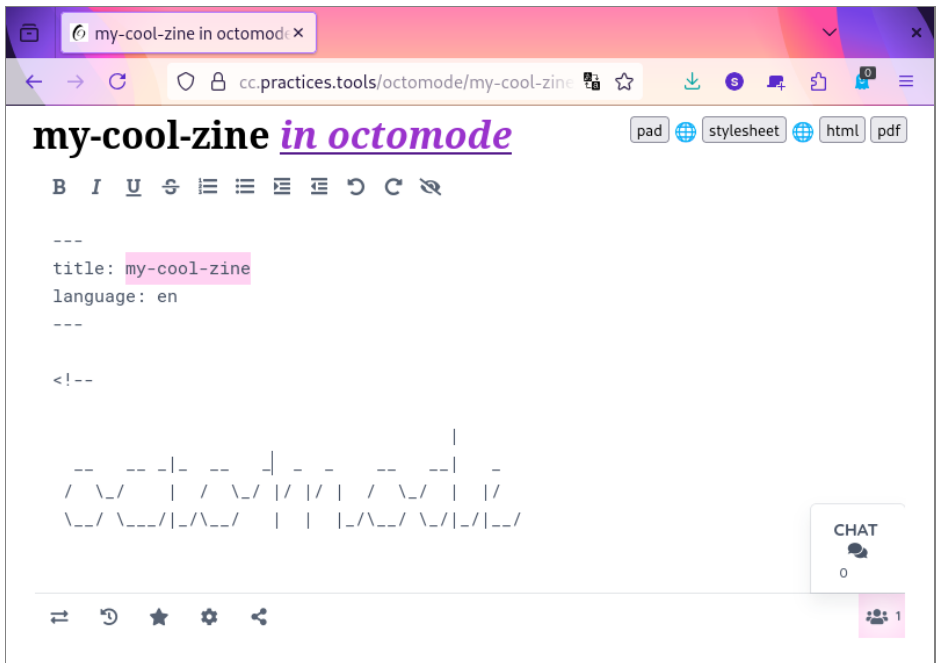


PDF preview

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 that the title is in.

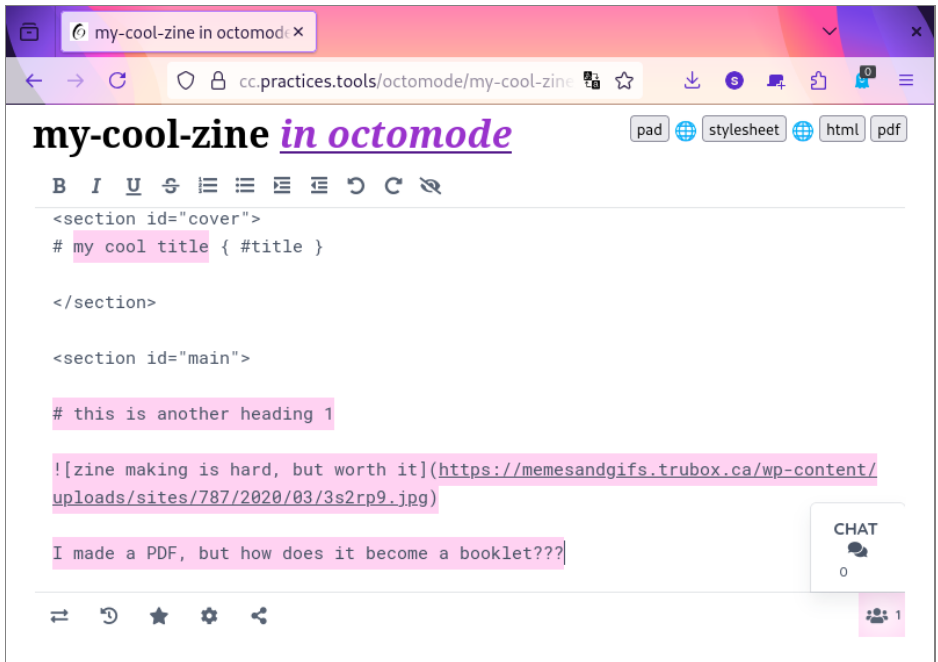
Change the text to whatever you want. In this example we will use “my cool title”.

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

Let's break this down. First comes one hash symbol #, making the title a heading 1. Following this is { #title }, which is the Markdown syntax that Octomode uses to give an id (a unique identifier) of “title”. We can then style it differently from the other heading 1 text using CSS:

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

adding some content



adding some things in the content pad

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

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

An image link in Markdown 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 where the image is hosted online.

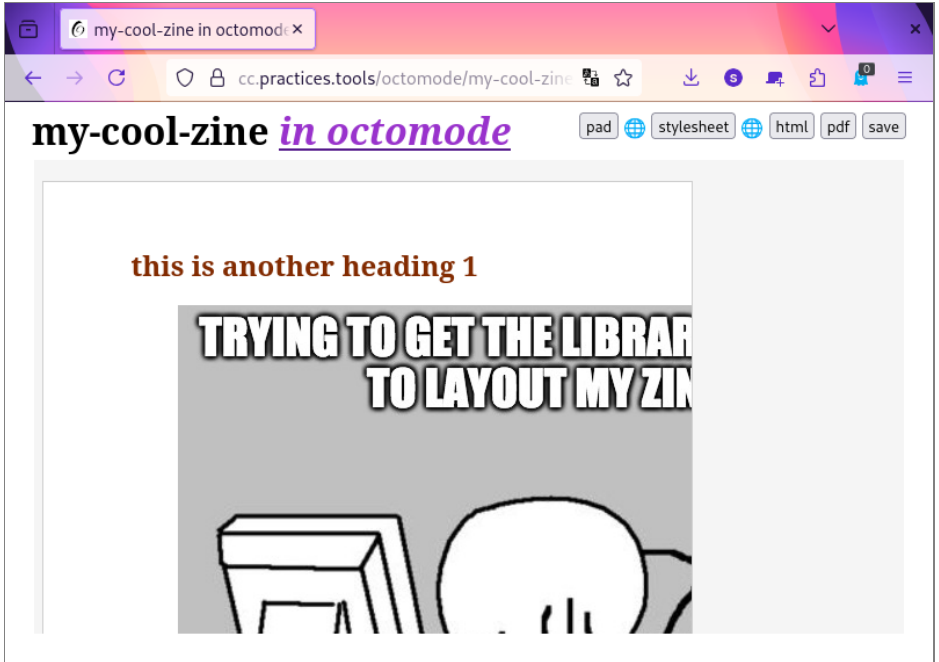
⚠ **The image link must begin with an exclamation mark !, followed by square brackets [] and the image URL inside parentheses ()** ⚠

Now, add some text below the image link:

```
I made a PDF, but how does it become a booklet???
```

checking your PDF

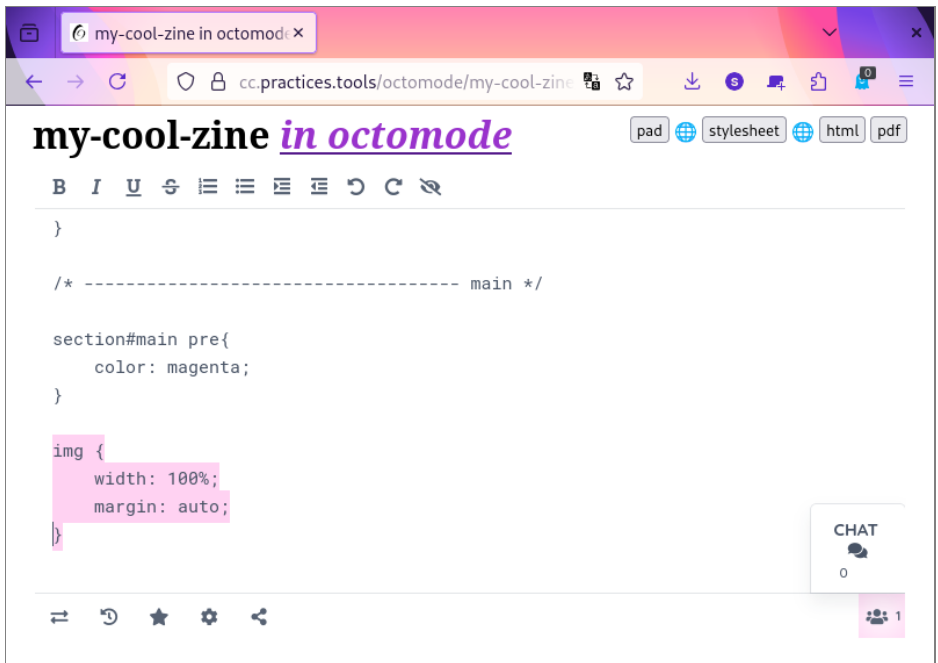
Click the `pdf` button, to see your changes!



the image is trying to escape the page...

Looks ok, but there is an issue with the image width. Without any CSS to define how wide the image should be, it will load at its native resolution, which may be wider than what you need. Let's try fixing that!

setting image size with CSS

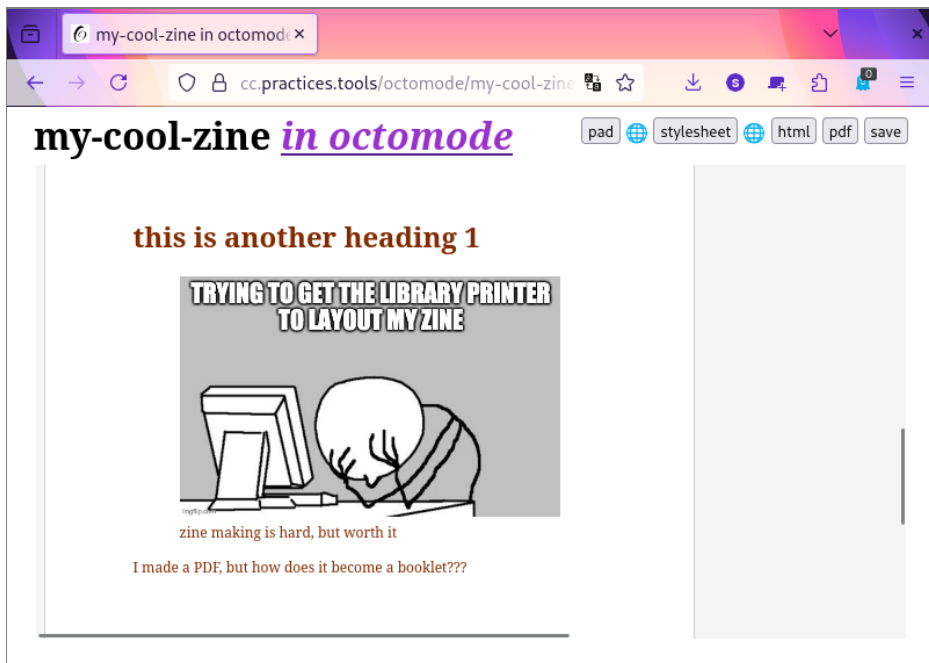


making the image fit inside the page by editing the stylesheet

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

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

Check your changes by clicking the `pdf` button again:



now the image fits!

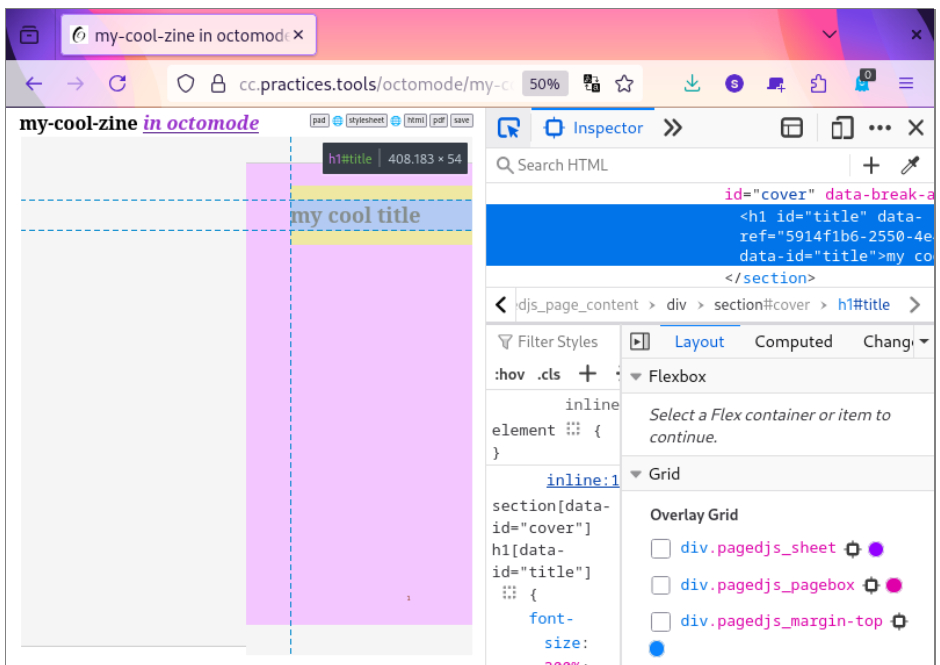
If you want to go deeper into using CSS for paged media, the [pagedjs documentation on web design for print](#) is a good place to start.

When you are ready, you can save the PDF!

debugging with a web inspector

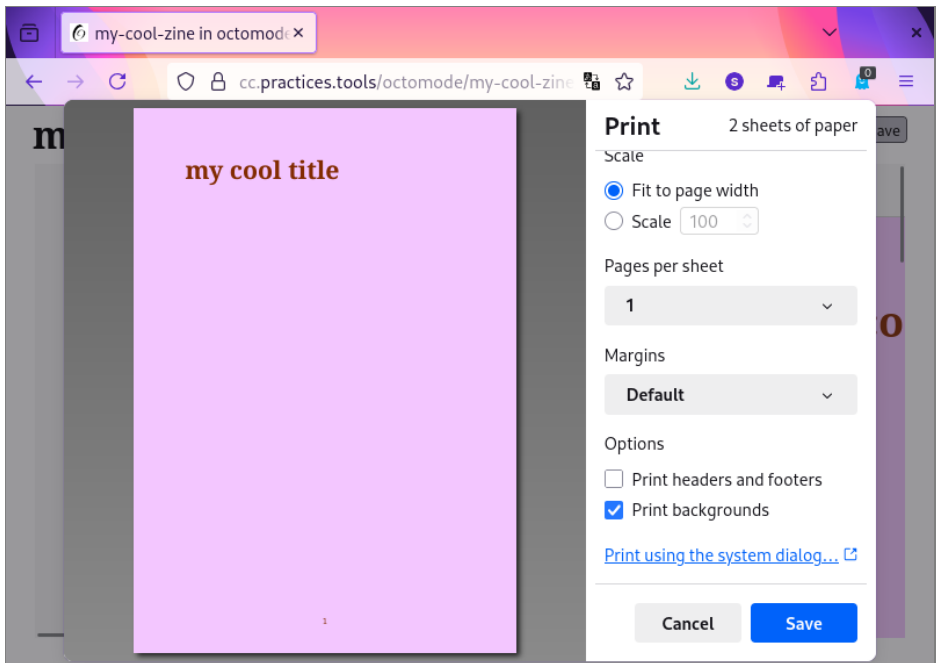
If your design looks a bit strange and you want to check why, or try out other CSS quickly, try using a **web inspector** to inspect the source code of a web page, and edit CSS without committing the changes. Just remember, any edits to the page are only visible on your computer, and they will disappear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

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.

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, you have a PDF, what now?

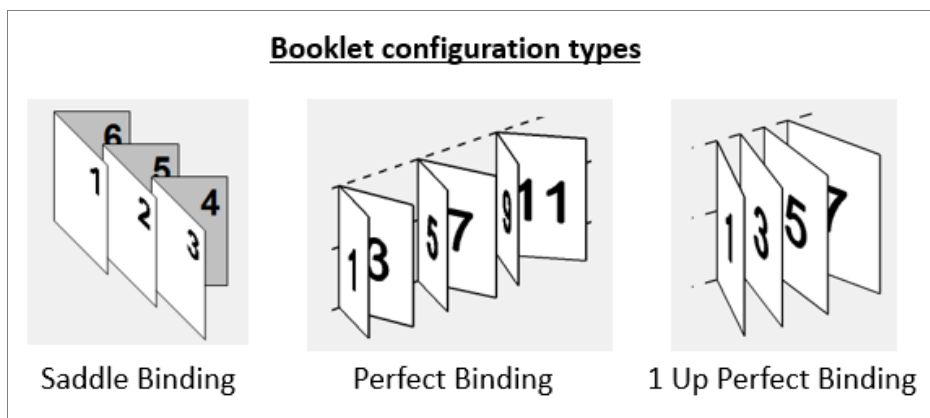
That is up to you!

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, or 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/>
- bookletor: <https://gitlab.com/quentinjuhel/Bookletor>



different types of booklet binding

If you decide not to print and bind it, you can upload the PDF zine somewhere. Or, email it around to your friends. The nice thing about making PDF zines is that you can include [hyperlinks](#) which will take a reader off to other places (other pages in the same zine, or web pages online). Choose your own adventure!

so, how does octomode *actually* work?

Octomode's 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

Octomode is free/libre open-source software that was made within the context of the collective [Varia](#), a collective-space for everyday technology in Rotterdam. Since then, it has been versioned by [Hackers & Designers](#) in Amsterdam and [Constant VZW](#) in Brussels.

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

- <https://cc.practices.tools/wiki/octomode>
- <https://git.vvvvvvaria.org/CC/octomode>

other (non-octomode) ways are possible!

There are quite a few FLOSS (free/libre open-source software) design and publishing projects that also make use of a HTML + CSS to PDF pipeline.

Here are a few other F/LOSS *web-to-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>
 - A collection of starter kits for paged.js <https://pagedjs.org/posts/2020-04-15-starterkits-for-pagedjs>
 - PPP Demos <https://gitlab.com/prepostprint/demos>
 - The Web 2 Print Library <http://web.2print.org/fr>

Colophon

This zine was made by Simon Browne in November 2025, with *octomode* in Linz, Austria, as part of a CC residency at the 2025 edition of d*signweek (<https://dsignweek.servus.at>)

Published under the conditions of the **CC4r * COLLECTIVE CONDITIONS FOR RE-USE** <https://constantvzw.org/wefts/cc4r.en.html>

Printed by Potato Publishing <https://potatopublishing.at/>

A digital version of this zine is available at:

<https://cc.practices.tools/pdf/so-you-decided-to-make-a-zine-in-octomode.pdf>

<https://cc.practices.tools>

shape tools
shape
practices