

# **How to make a professional webpage:**

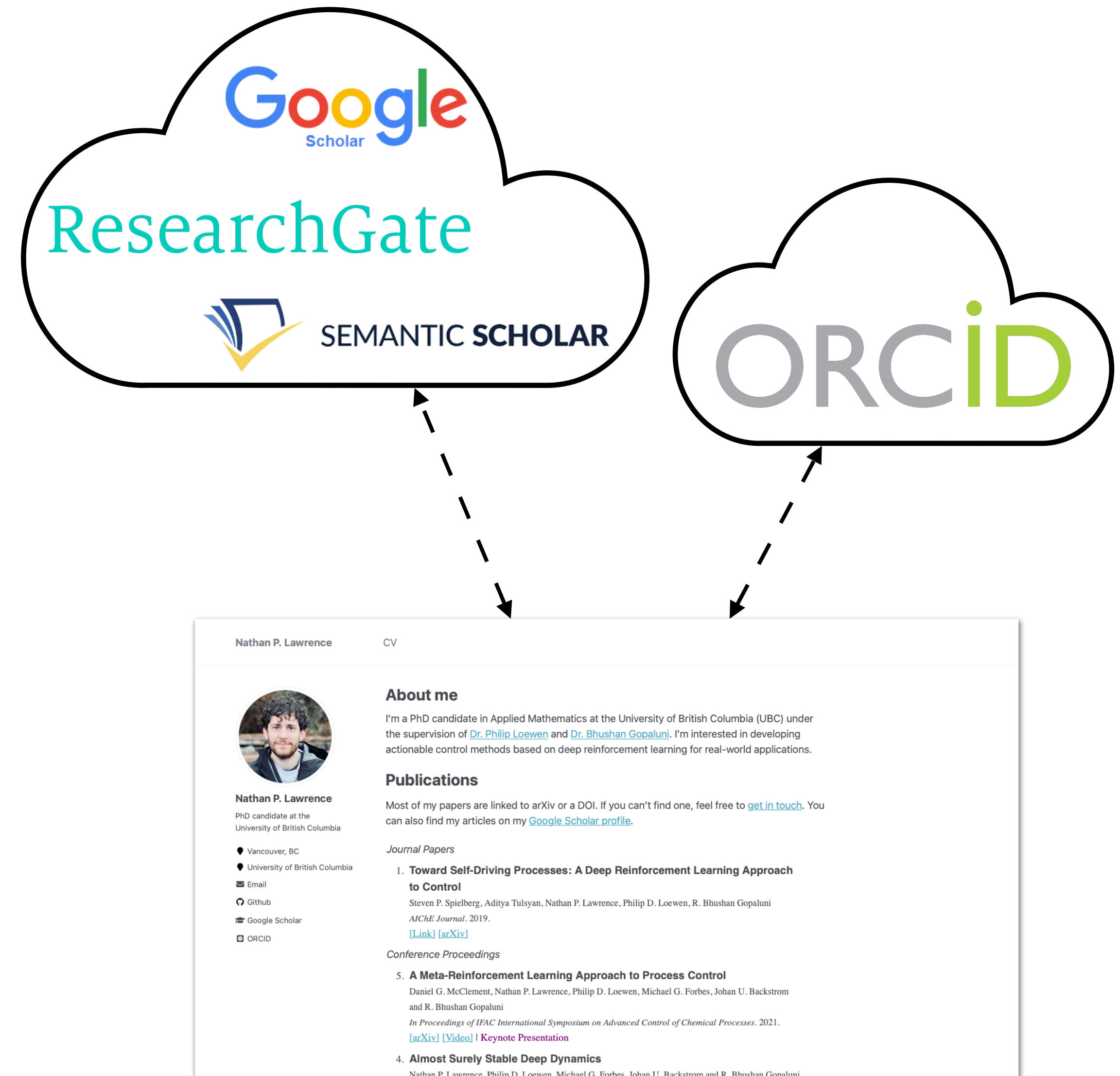
**“From 90% to Hero”**



**Nathan Lawrence ~ IAM workshop ~ November 2021**

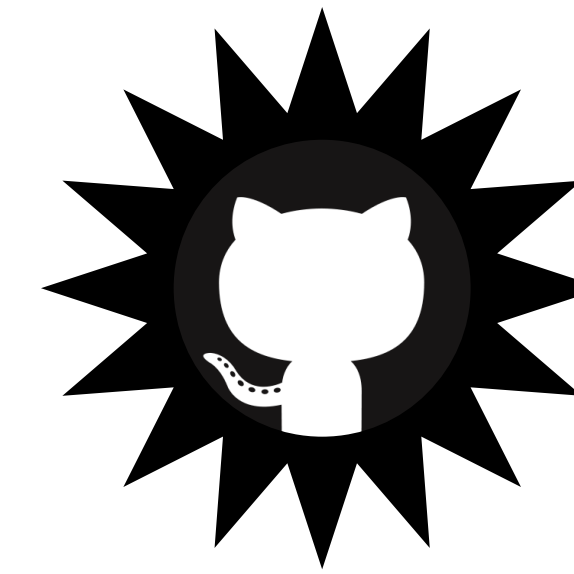
# Motivation and scope

- A professional website is your central space to convey a complete picture of your research and background
  - **Professionalism, centrality, flexibility, stability**
- We will start with template material and cover the basic skills needed to make it your own
- By the end of the workshop you should **have a functioning website and the tools needed to manage it**



# Which software and host?

## Jekyll and GitHub Pages



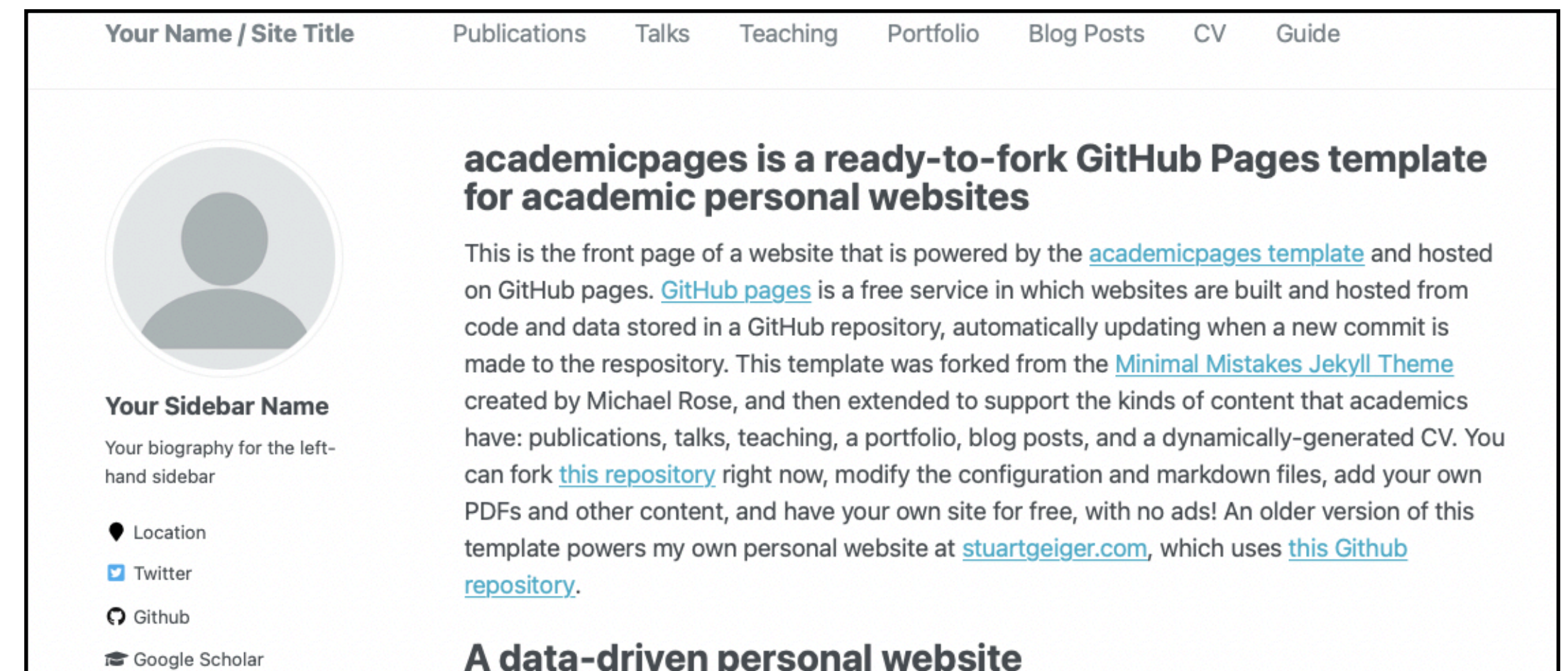
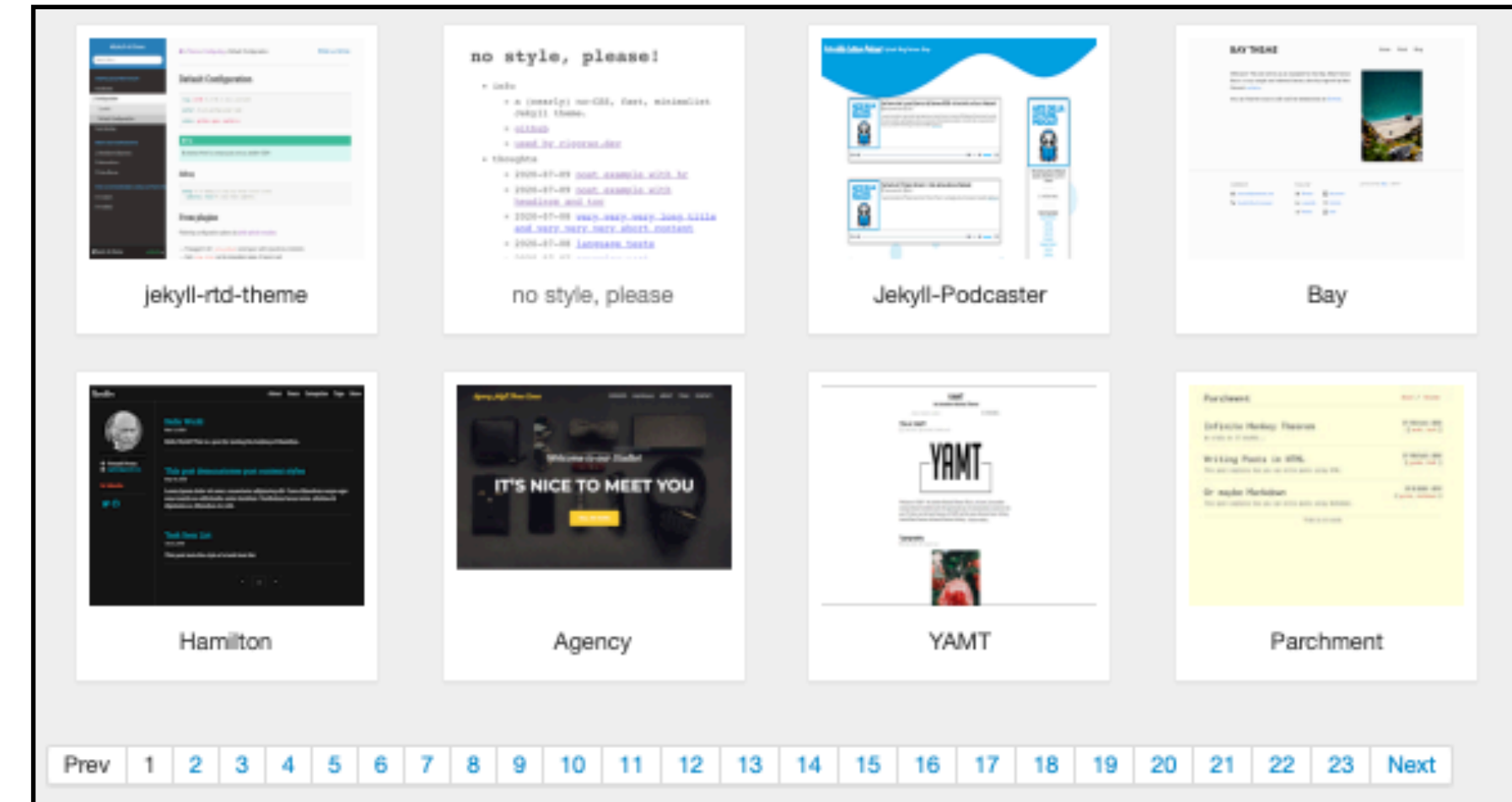
### GitHub Pages

- Free
  - Simple but flexible
    - Easy to integrate information
    - **Disentangles design and content**
  - Only requires some technical skills
    - **Somewhere between raw HTML+CSS and drag-and-drop type services like Wix or Wordpress**
- Takes Markdown, HTML, CSS and your template code to output a static website
  - Works with Liquid for processing your site's data
  - Works with Github Pages for hosting
  - Tons of templates/documentation if you aren't familiar with these
  - Overall, more room for growth, but still relatively easy to learn
  - Examples: DAIS

# Jekyll templates

“Just add your data”\*

- Hundred of themes to choose from
- Every theme has an associated GitHub repo
  - Once you install it, you have complete access to the source code and can modify your webpage however you please
- I modified Academic Pages which itself is based off of Minimal Mistakes
- \*We'll cover the basics of how these templates work



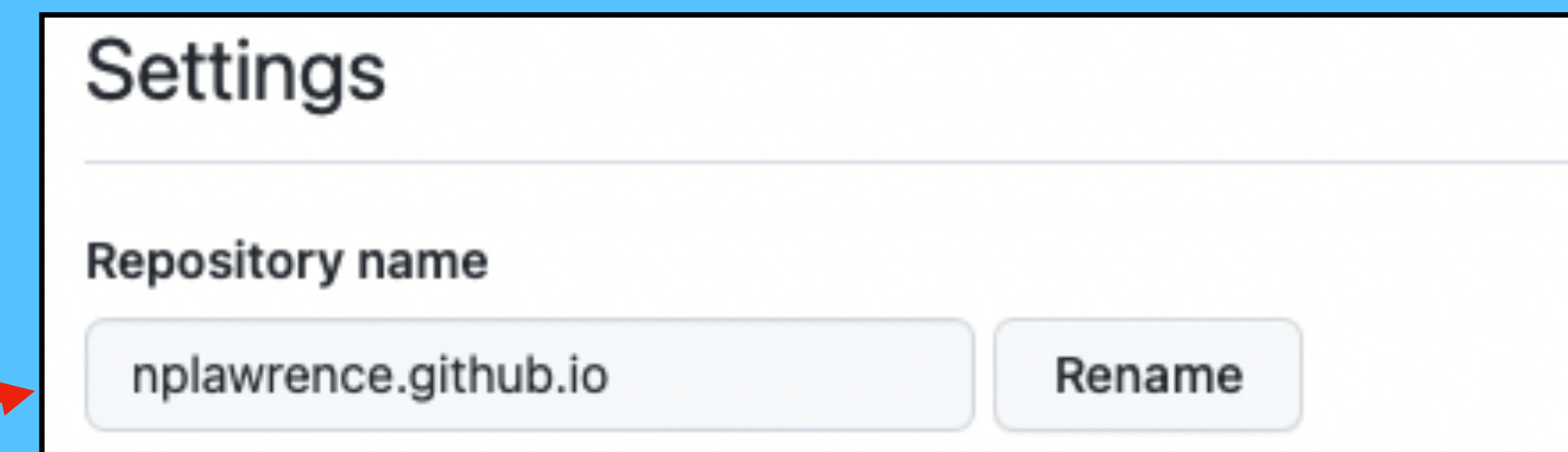
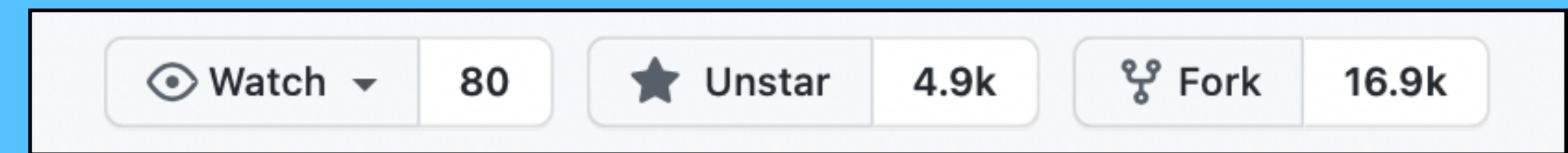


(10 minutes)

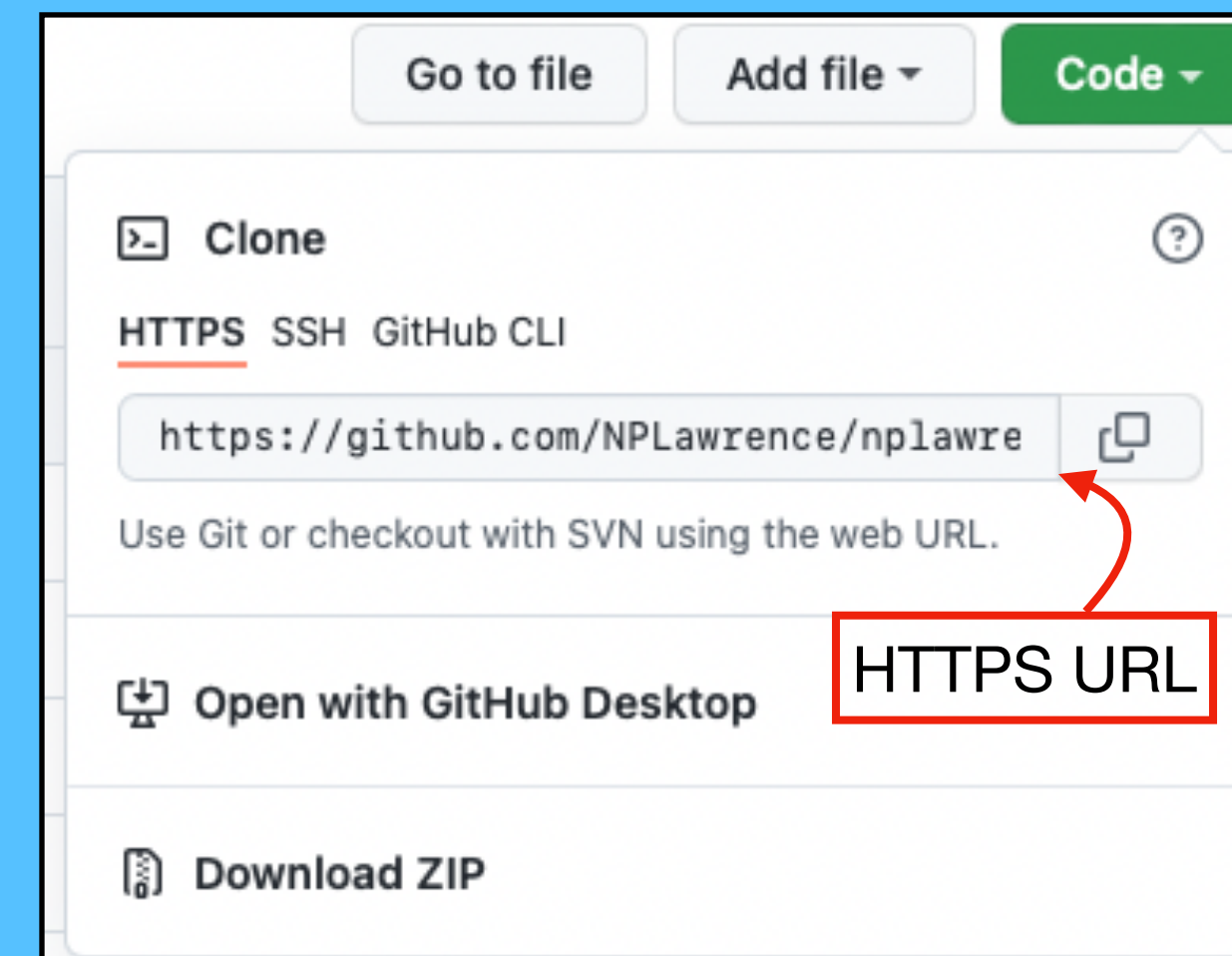
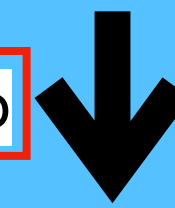
# Exercise

## Getting started

- You should already have Jekyll installed
- Fork Academic Pages template (<https://github.com/academicpages/academicpages.github.io>)
- Rename your forked version to "username.github.io" where "username" is your GitHub username
- Clone it locally and open the folder in your favorite IDE
- Navigate to your project folder in your terminal (command prompt or whatever)

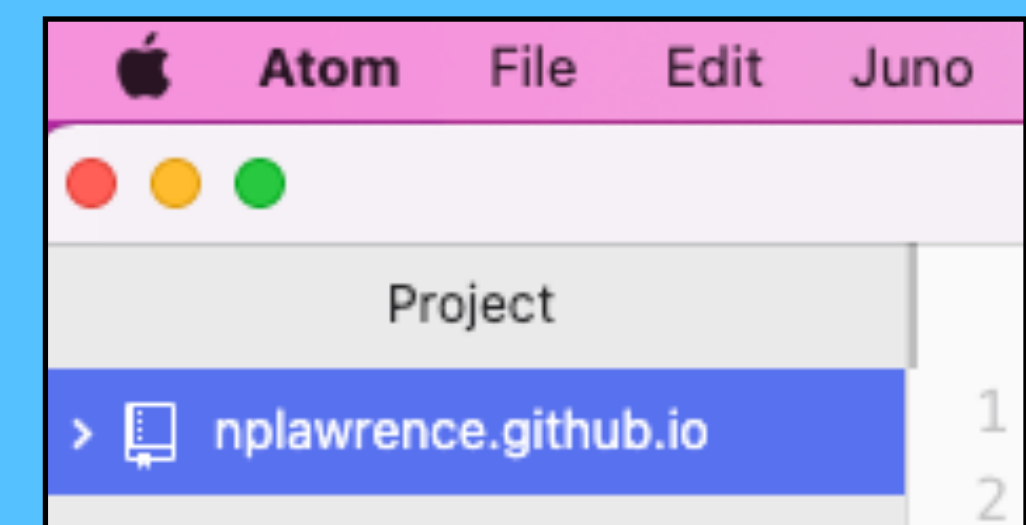


username.github.io



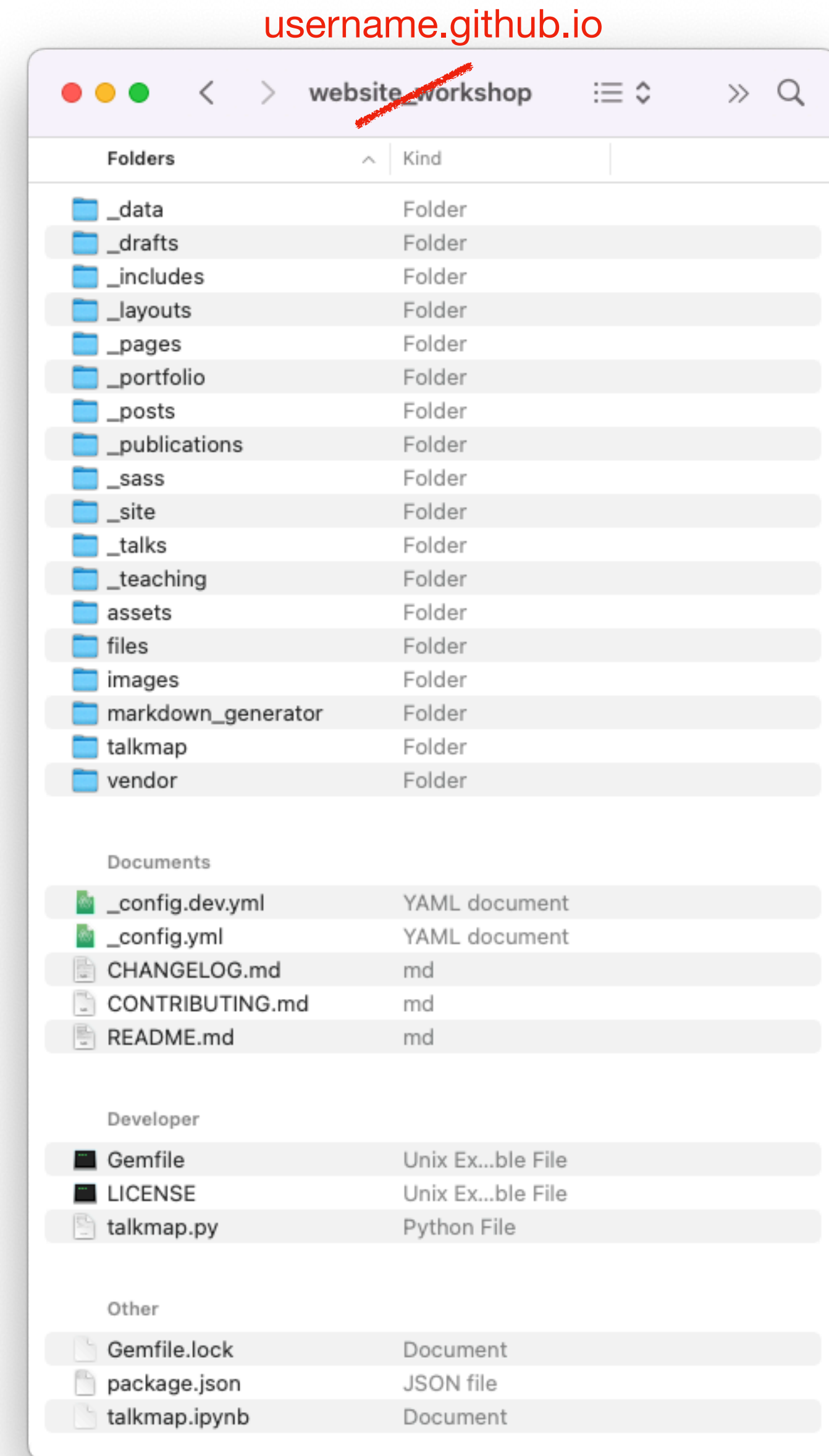
In your desired directory....

git clone HTTPS URL



# An overview

- This presentation is a broad overview of the key pieces for operating a Jekyll-based website
- We will focus on:
  - General website structure
  - Basic syntax for Liquid
  - Managing your data
- We will *not* get into:
  - HTML, CSS
  - Writing custom layouts
- However, we will provide additional references



This is the structure your directory should have

# An overview

## What makes these themes work?

- The backbone of the webpage is the **\_config.yml** file
  - This is a structure that stores global data about your site, such as your name, email, URLs to various platforms
- The **\_includes** folder is a collection of code snippets that you may want to use repeatedly, for example to format a single publication
- The **\_layouts** folder contains individual templates for page types, such as homepage, cv page, blog, etc — basically the outline for how a page type should look
- **\_pages** and **\_layouts** are closely related
  - A page contains raw text (content) and a section for specifying its meta-data (like which layout to use)
  - A layout is code that takes the content and imposes its own structure (like formatting or including a sidebar)
  - Conceptually somewhat similar to writing content in LaTeX
- **Focus on writing your pages without worrying about formatting**



# An overview

## Website data

- Any static or generally fixed-structure data you have (publications, CV info, etc) can be saved into YAML files
- **Your data is accessible across your whole website under a “site” variable**
- Practically speaking, when you update your data file, the changes are present in your compiled website
  - For example, if you change employer, anywhere you reference that “object” gets updated

YAML = “YAML Ain’t Markup Language”



<https://twitter.com/drveronikach/status/1405438332428038150>



(10 minutes)

# Exercise

## Fill out \_config.yml

- Add your profile pic to “images” folder, then...
- # Site Settings:
  - Title
  - Name

} These can both just be your name

  - Url (“https://username.github.io”)
  - Repository (“username/username.github.io”)
- # Site Author:
  - Name
  - Avatar
  - Bio
  - Social links, etc...

**Extra:** Following the format of the \_config.yml file, add you own fields under # Site Author, such as:

- discipline
- department

### When finished:

- Click “save”
- Open your terminal (command prompt or whatever)
- Navigate to your website’s directory:

“cd path/to/directory”
- Run:

“bundle exec jekyll serve --livereload”

(Try “bundle install” first if it didn’t work)
- Go to “http://localhost:4000” in your favorite browser
  - It may take a while to load at first, but you should see your compiled website
- **Extra:** Read the content on the homepage if you’re waiting around!



**5 min break**

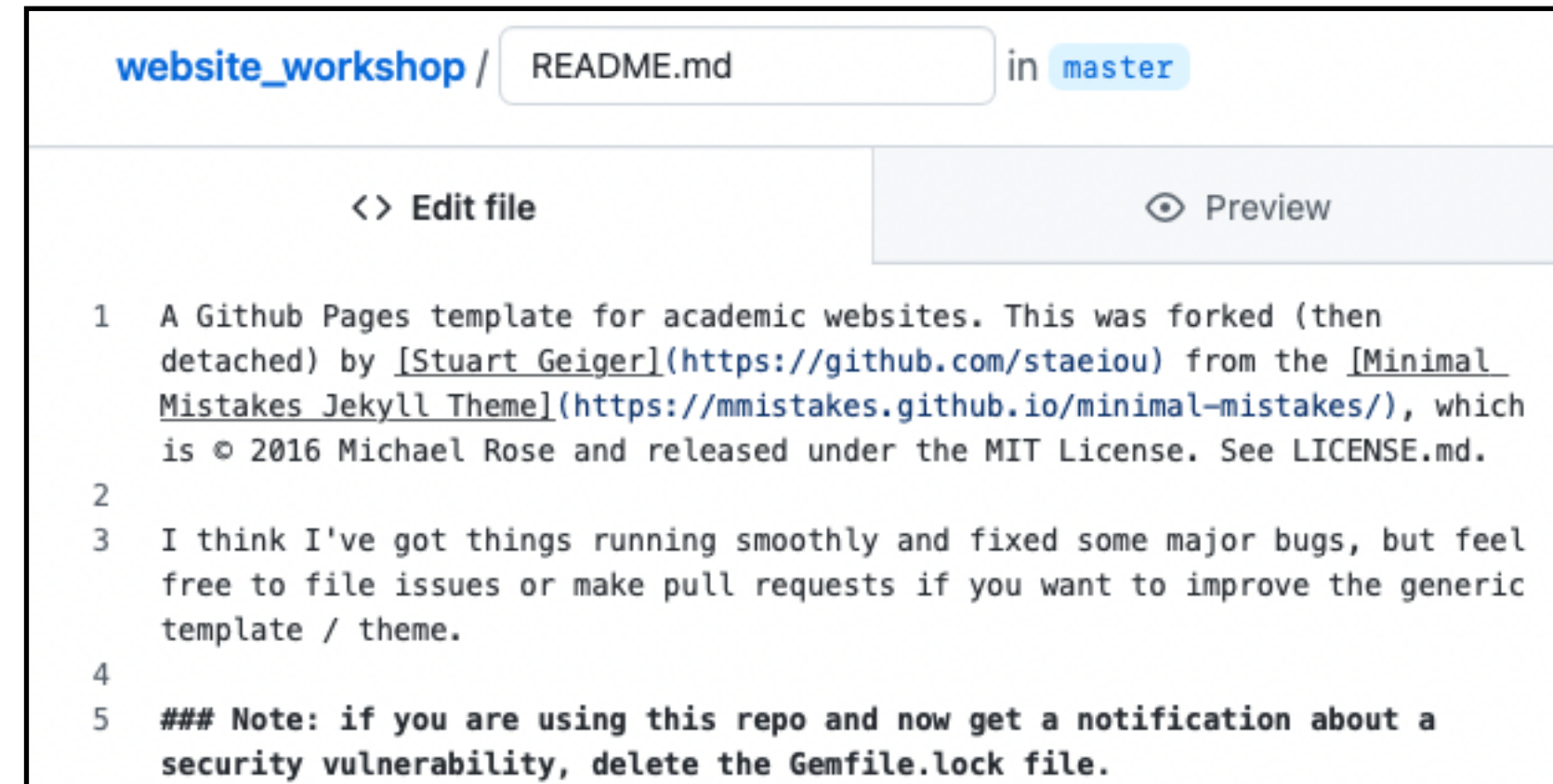




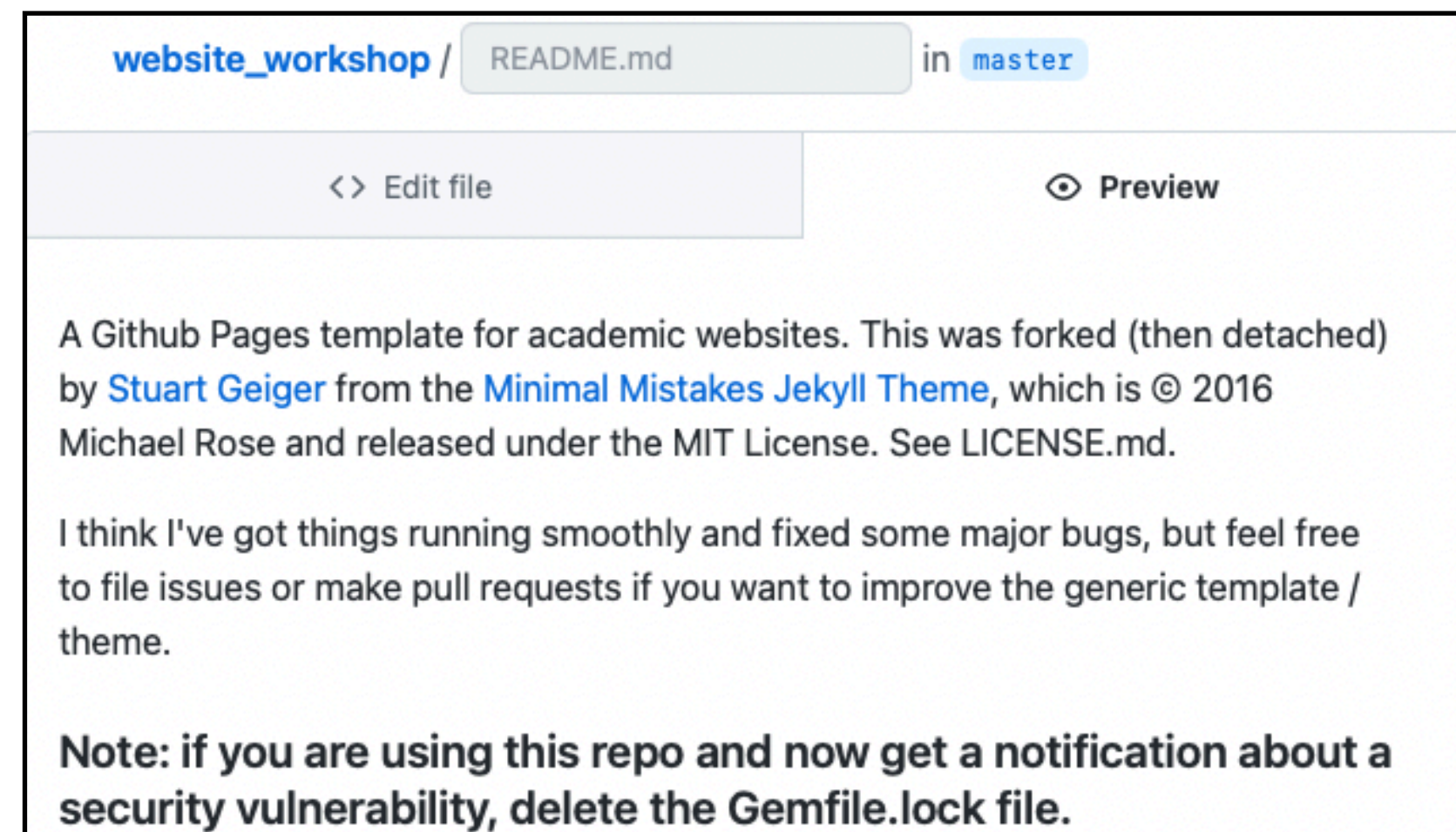
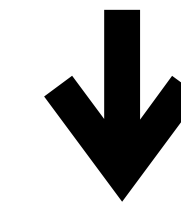
# Markdown

- Markdown provides a simple syntax for formatting plain text
- Have this cheat sheet handy:  
<https://www.markdownguide.org/cheat-sheet/>
- All your *content* can be written in .md files
  - The formatting is handled by separate code for page “Layouts” (a combination of HTML and CSS)

Somewhere you may have seen markdown before:



```
website_workshop / README.md in master  
  
<> Edit file Preview  
  
1 A Github Pages template for academic websites. This was forked (then  
detached) by [Stuart Geiger](https://github.com/staeiou) from the [Minimal  
Mistakes Jekyll Theme](https://mmistakes.github.io/minimal-mistakes/), which  
is © 2016 Michael Rose and released under the MIT License. See LICENSE.md.  
2  
3 I think I've got things running smoothly and fixed some major bugs, but feel  
free to file issues or make pull requests if you want to improve the generic  
template / theme.  
4  
5 ### Note: if you are using this repo and now get a notification about a  
security vulnerability, delete the Gemfile.lock file.
```



```
website_workshop / README.md in master  
  
<> Edit file Preview  
  
A Github Pages template for academic websites. This was forked (then detached)  
by Stuart Geiger from the Minimal Mistakes Jekyll Theme, which is © 2016  
Michael Rose and released under the MIT License. See LICENSE.md.  
  
I think I've got things running smoothly and fixed some major bugs, but feel free  
to file issues or make pull requests if you want to improve the generic template /  
theme.  
  
Note: if you are using this repo and now get a notification about a  
security vulnerability, delete the Gemfile.lock file.
```



# Exercise

## Modifying the homepage

- Open `_pages` —> `about.md`
- Look over its **structure**, for example:
  - *How are new sections/headings created?*
  - *How do you create a list?*
  - *How do you hyperlink text?*
- Substitute the **sections names** for your own and delete the content therein — “About me”, “Research interests”, etc... — don’t worry about the content yet!

When finished:

- Click “save”
- Your webpage should reload in your browser
  - Aside: the **only** exception to this rule is `_config.yml`

**Extra:** (removing tabs on homepage)

- Open `_data` —> `navigation.yml`
- Comment out sections you don’t need
- Click “save”

# Liquid

- Liquid is a templating language
- When Jekyll builds your website, your template Liquid code is substituted for actual content
  - Great for iterating over large, redundant data files!
- Have this open  
<https://shopify.github.io/liquid/basics/introduction/>
- Liquid code goes in a markdown (or HTML) file

Input

```
{% if user %}  
  Hello {{ user.name }}!  
{% endif %}
```

Output

Hello Adam!

- The syntax {% stuff %} indicates a **tag**
  - Assign variables
  - Control flow (if, else statements)
  - Iteration (for loops)
- {{ things }} indicates an **object**
  - Its content gets rendered (unlike content in tags)



(10 minutes)

# Exercise

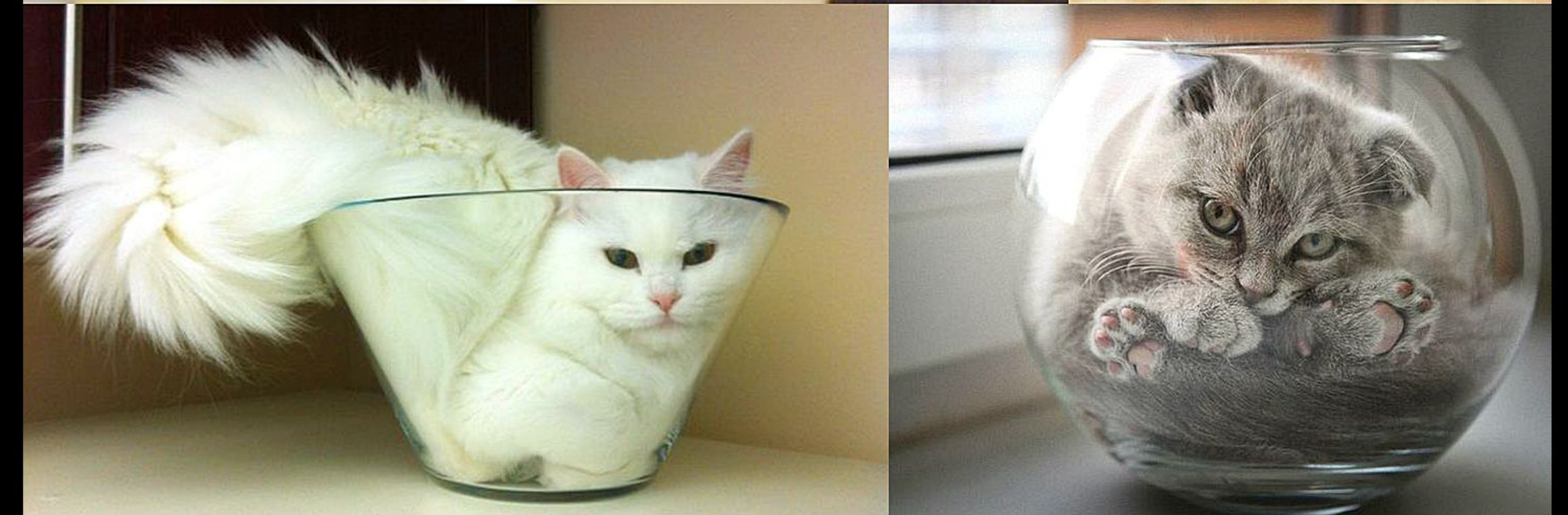
## Study Liquid

- Spend a few minutes looking over the basic syntax for Liquid

<https://shopify.github.io/liquid/basics/introduction/>

- We will practice using it in the next exercises!
- **Extra:** Browse through the tags *Control Flow*, *Iteration*, *Variable* on the left panel

**A liquid is able to flow and take the shape of a container**



**CATS ARE A LIQUID!**

<https://www.pinterest.ca/pin/435582595179490977/>



# Exercise

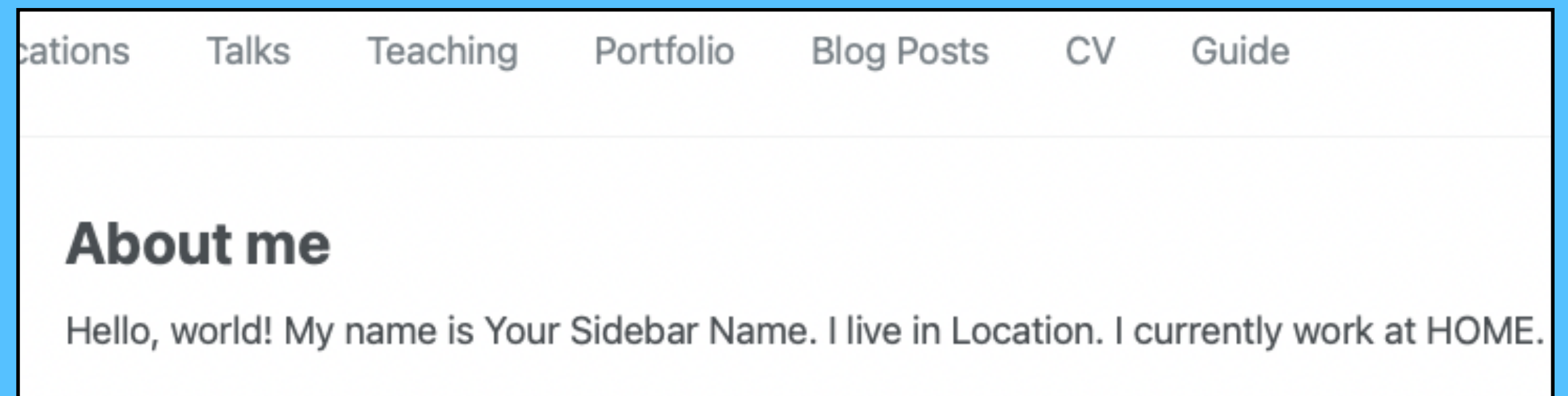
## Accessing site data with Liquid

- Write an “About me” section on your homepage (a couple sentences), but use objects in Liquid to insert your data, rather than manually entering it
- Example of an object
  - `{{ site.author.name }}`

(Global site variable —> `_config.yml` Site Author —> Name object)

- **Extra:** use control flows to enter a sentence *if* its subject exists
  - Example: Write where you work *if* “employer” field is filled out

Try to get something like this to display:



# Yet Another Mini Lesson

- *Don't* manually type your data into a page!
- Often times information fits into a **common underlying structure**
- Store this information under `_data` in **YAML files**
- It is then accessible globally under “**site.data.FileName**” and can be processed using Liquid
- Dashes (-) indicate the *start* of a **list item**
- Indents are used to **nest items** — either other lists or objects (recall Site Author in `_config.yml`)

For our purposes YAML is simply a syntax for a collection of key-values pairs

Filename: `_data/skills.yml`

```
- category: Blogging
  skills:
    - name: Writing
      rating: 5
    - name: SEO
      rating: 4
```

Sub-list item { List item 1

```
- category: Languages
  skills:
    - name: English
      rating: 5
    - name: Spanish
      rating: 3
```

List item 2

<https://www.aleksandrhovhannisyan.com/blog/getting-started-with-jekyll-and-github-pages/#example-1-skills-and-abilities>

# Another example

- To access “louie”, write

site.data.example.items[0].things.thing3

- To access “value”, write

site.data.example.items[1].["other things"].key

Filename: \_data/example.yml

```
items:
  - things:
      thing1: huey
      things2: dewey
      thing3: louie
  - other things:
      key: value
```



**5 min break**





# Exercise (part 1)

## CV content

1. Create a file called `education.yml` file in `_data`
2. Write a *general* structure that contains the content shown on the right (substitute your own data)
  - Create a list item for each bullet
  - Give each object a consistent key name, for example:  
  
degree : "B.S."

Create a YAML file with this information

### Education

- B.S. in Github, GitHub University, 2008 – 2012
- M.S. in Jekyll, GitHub University, 2012 – 2014
- Ph.D in Version Control Theory, GitHub University, 2014 – Present
  - Thesis: "How to Clone your Dragon"
  - Advisors: Professor Git, Professor Hub

# Exercise (part 2)

## CV content

1. Navigate to cv.md under \_pages
2. Replace the content under “Education” with Liquid code that produces the list on the right
  - Iterate over your education.yml data!

Try to get something like this to display:

### Education

- B.S. in Github, GitHub University, 2008 – 2012
- M.S. in Jekyll, GitHub University, 2012 – 2014
- Ph.D in Version Control Theory, GitHub University, 2014 – Present
  - Thesis: “How to Clone your Dragon”
  - Advisors: Professor Git, Professor Hub

Extra: Make sure your algorithm is general enough by considering...

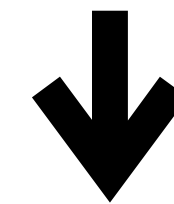
- What if you had a M.S. or B.S. advisor?
- How can you guarantee these bullets are displayed in the correct order?
- What if you have multiple degrees of the same type?



# “Include” files

- “Includes” simply copy-pastes code wherever you put the command
- Great for reducing redundancy and clutter
  - Example: Create a list of all publications using only a for loop and template code
- These are markdown or HTML files that live in the \_includes directory

```
Publications
=====
<ul>{% for post in site.publications %}
  {% include archive-single-cv.html %}
{% endfor %}</ul>
```



**Publications**

[Paper Title Number 3](#)

Published in *Journal 1*, 2015

This paper is about the number 3. The number 4 is left for future work.

Recommended citation: Your Name, You. (2015). "Paper Title Number 3." *Journal 1*. 1(3).  
<http://academicpages.github.io/files/paper3.pdf>

[Paper Title Number 2](#)

(5 minutes)

# Exercise (part 3)

## CV content

1. Navigate to `_includes` and create a Markdown file `education-item-cv.md`
2. Cut the content of your “for loop” and put it in this new file
3. Modify your “for loop” with an “include” to produce the same content!

Try to get something like this to display:

### Education

- B.S. in Github, GitHub University, 2008 – 2012
- M.S. in Jekyll, GitHub University, 2012 – 2014
- Ph.D in Version Control Theory, GitHub University, 2014 – Present
  - Thesis: “How to Clone your Dragon”
  - Advisors: Professor Git, Professor Hub

# Recap

- Jekyll provides an elegant framework for separating content from design
- All the user has to do is enter data into .yml files and write content in .md files
  - Your data is accessible across the entire website
  - Your CV and publications are automatically integrated and compiled into your website
  - No manual formatting; relatively easy to transition all your data to a new theme
  - Tons of themes means you don't have to worry about creating your own layouts — at least not from scratch
- The result is a directory containing your static HTML website, which can be hosted on Github Pages for FREE



(∞ minutes)


# Exercise

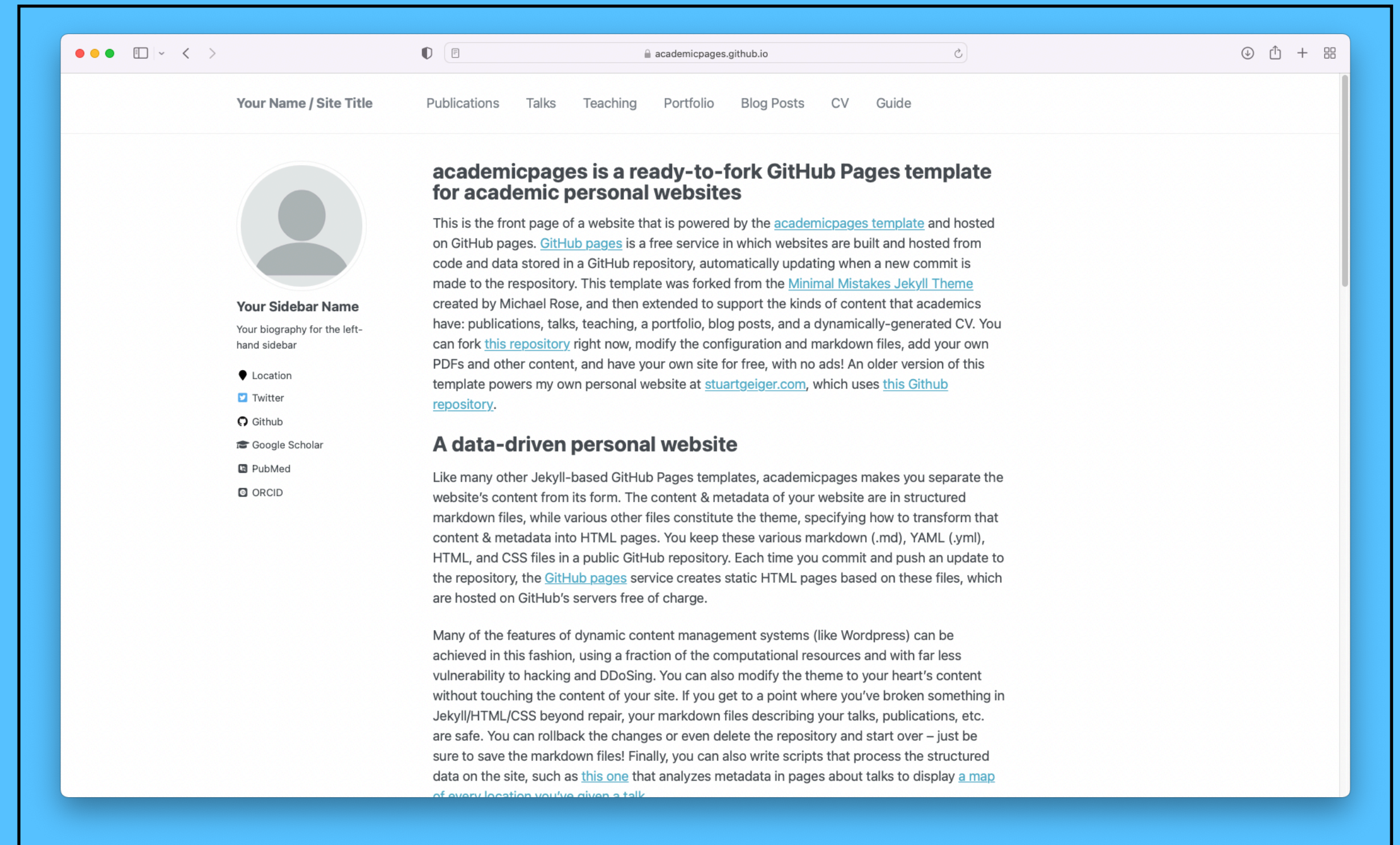
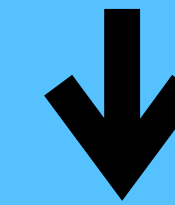
## Fill in the rest!

- Things to consider:
  - Finishing your CV or homepage
  - Adding \_publications content
  - Deleting unwanted content in the template
  - Adding/deleting profiles (Google Scholar, Twitter, etc)

```
git add . && git push
```



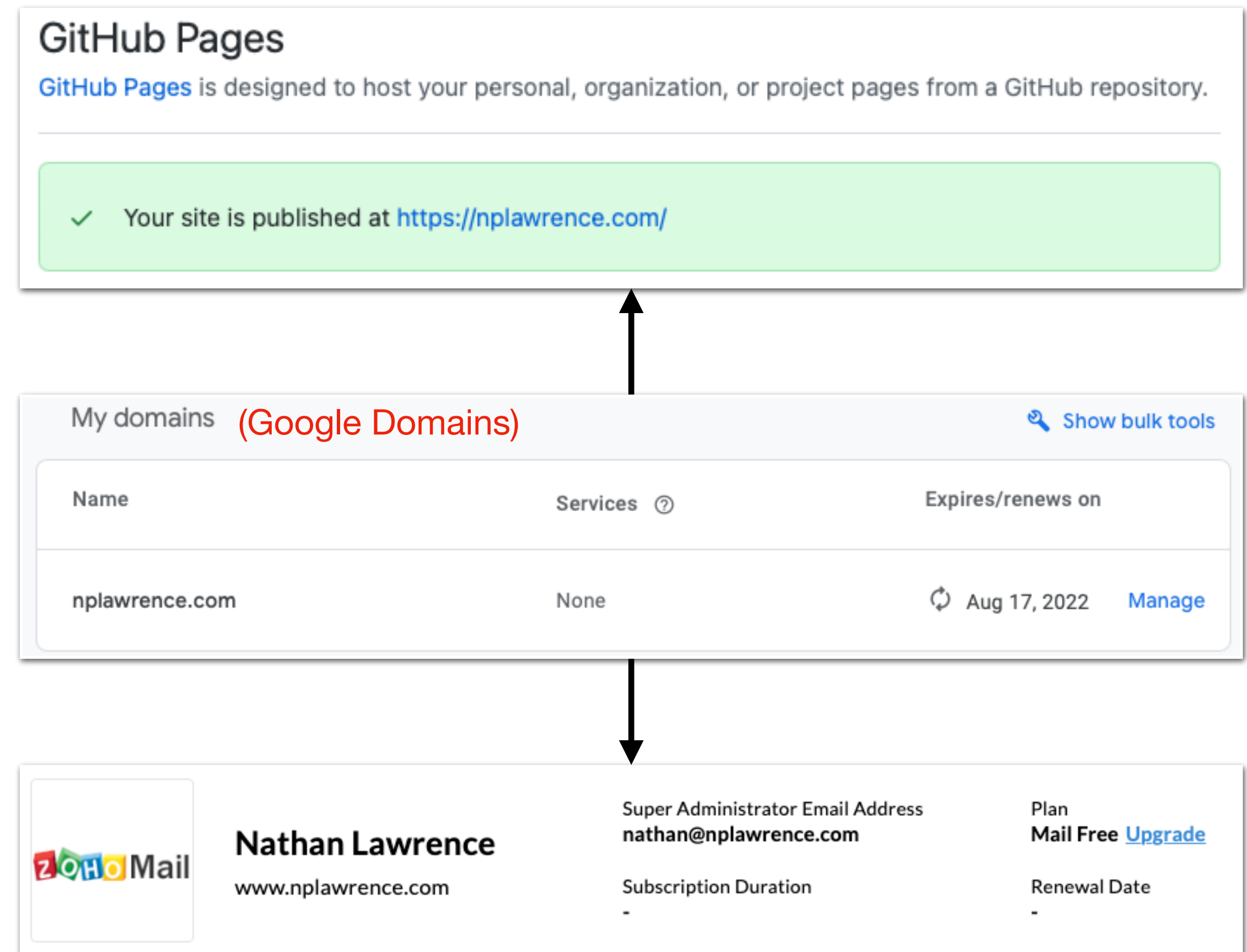
 <https://yourUsername.github.io>



# Bonus

## Domains and email

- With Github Pages, your personal webpage's domain is automatically "username.github.io"
- Purchase a domain that closely matches your name (I chose Google Domains)
- Configure settings such that your domain points to the website hosted at "username.github.io"
- For email, setup a mail server under your domain, for example, through Google Workplace or Zoho (free and secure)



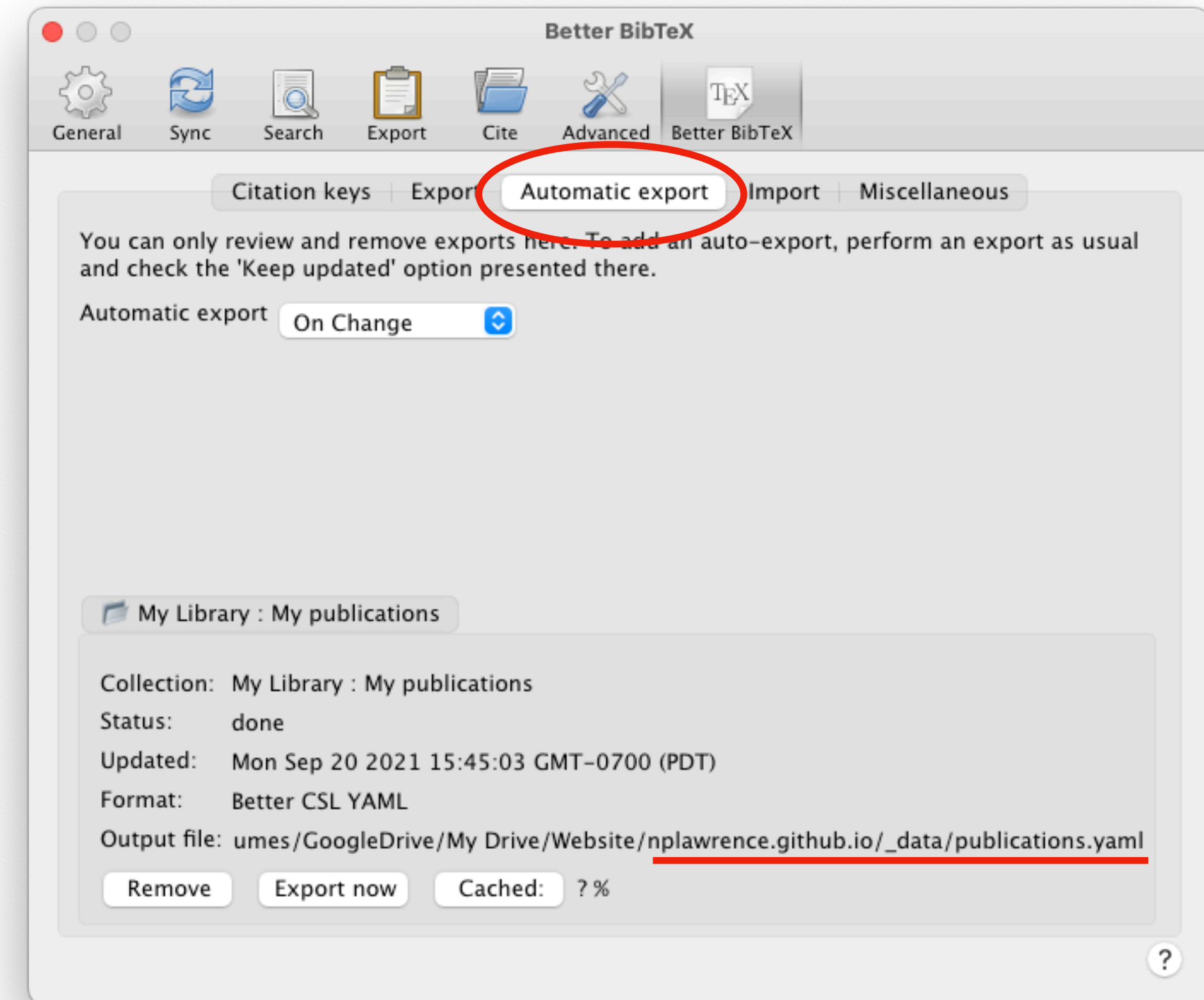


# Bonus

Can we automatically enter new data, too?

## ★ Enter Zotero & Better BibTeX

- Zotero is great on its own as a research tool
- BBT is an add-on for managing bibliography data
  - Export your publications as a yaml file
  - Keep it up to date in your local repo folder with *automatic export*
  - Only downside is that Zotero doesn't (yet) support custom textfields
    - For example, can't easily add a "Code" field for a GitHub link to a reference
- **Changes in Zotero get integrated into your website automatically**





# Useful resources

## Getting started

- Github Pages and Jekyll website (installation, find resources):
  - <https://pages.github.com>
  - <https://jekyllrb.com>
- ★ A very thorough introduction to Jekyll and Github Pages (“zero to hero”):
  - <https://www.aleksandrhovhannisyan.com/blog/getting-started-with-jekyll-and-github-pages/>



# Useful resources

## Themes

- Browse here: <https://jekyllrb.com/docs/themes/>
- I started from this one: <https://github.com/academicpages/academicpages.github.io>

## Misc:

- A nice CV template: <https://github.com/elipapa/markdown-cv>
- Editing the favicon (the icon that might show up in your browser tab): <https://ptc-it.de/add-favicon-to-mm-jekyll-site/>



# Useful resources

## Domains and email

- Google Domains: [https://domains.google/intl/en\\_ca/](https://domains.google/intl/en_ca/)
  - Find your perfect domain
- Use your domain for your webpage:
  - <https://docs.github.com/en/pages/configuring-a-custom-domain-for-your-github-pages-site/managing-a-custom-domain-for-your-github-pages-site>
  - <https://hossainkhan.medium.com/using-custom-domain-for-github-pages-86b303d3918a>
- Free mail server: <https://www.zoho.com/mail/zohomail-pricing.html>
  - Scroll to the bottom



# Useful resources

## Example repos

- Thank you to Siang Lim. The source code for the DAIS website was super helpful.
  - <https://dais.chbe.ubc.ca>
  - <https://github.com/daisubc/daisubc.github.io>
- My webpage:
  - <https://nplawrence.com>
  - <https://github.com/NPLawrence/nplawrence.github.io>



# **Thank you**

**Slides and solutions:**

**[https://github.com/NPLawrence/website\\_workshop](https://github.com/NPLawrence/website_workshop)**