

# Academic Portfolio Websites

Why & How



# Save time

- Create Github account at

<https://github.com>

- Download Github Desktop from

<https://desktop.github.com/>

# Why have one?

- Cultivate your online professional identity
- Reach broader audience
- Supplement your hiring or promotion package

# What about Academia.edu, ResearchGate, or Google Scholar ?

- Vendor lock in
- Academics provide free data to for-profit companies
- No substitute for open access repositories

\*\*\*

- Enthusiasm varies by domain and institution

# What about my.vanderbilt.edu ?

- Not trapped in Vanderbilt system when you move institutions
- Vanderbilt limits the plugins, themes, and color schemes you can use to maintain consistent branding
- Go ahead and have one as a member of your department, but don't have it be your main one

# Why have your own website?

## Maintain Control

- You decide content
- You decide format
- Easy to move institutions

# One method: Use a CMS

- WordPress
- SquareSpace
- and more

\*\*\*

- Pay the extra to get rid of ads
- Buy your own domain name
- Keep it professional

# Go beyond the CV

- Add course websites and syllabi
- Add your teaching statement
- Add news or blog – news mentions or blog about projects you are working on
- Add project websites



# Why *build* your own website?


## Maintain Control

- You decide content
- You decide format
- Easy to move institutions

## **Plus**

## Sustainability

- Easy to migrate as web evolves
- Escape proprietary tools
- Future-proofing



**Plus signaling digital  
competency**

# Start building foundational DH skills

- Command line basics
- Markdown syntax
- HTML & CSS basics
- Git basics

# Use a text editor

- Save in plaintext
  - Word processing programs add hidden formatting that breaks things
- Syntax highlighting help you see your mistakes
- Look the part

Recommended:

- Atom.io
- Sublime Text

# Writing in plaintext with Markdown

- Plaintext files are
  - Platform independent: readable on almost any device, operating system, or word processor
  - Sustainable: more future-proof than proprietary file types (remember Word Perfect? Microsoft Works)
- Markdown is a lightweight syntax for formatting your file
  - Can easily convert to HTML to display on web
  - With pandoc, you can also convert Markdown files to other file types: PDF, Word, LaTeX, slides

<https://daringfireball.net/projects/markdown/>

# What makes up a webpage?

3 types of files that work together to make webpage

HTML	Content
CSS	Styling
Javascript	Interactivity

Only HTML is strictly required

# How it works

Files reside on a server

Clicking a url tells the browser to connect to the web server and retrieve that page

The server sends (serves) the page to the browser (the client)

Browser formats and displays the page to the user

# Building your own static website

- Professional profile
  - Course website
  - Project website
  - Lab website
- 
- Not a good option when interactivity or database querying is required



# Dynamic websites

- Pages composed dynamically -- on the fly -- from a template
- Dependent on a content management system (CMS)
- Requires a database for file storage
- Can query a database and display results
- Examples: WordPress, ecommerce site

# Static websites

- Composed of flat, fully-built files, which means no database
- No database means better protection from hackers
- Static pages are pre-built and so load faster
- Version control
- More sustainable; easier and cheaper to maintain
- Limited interactivity

See more: <https://about.gitlab.com/blog/2016/06/03/ssg-overview-gitlab-pages-part-1-dynamic-x-static/>

# Much better explanation

<http://nilclass.com/courses/what-is-a-static-website>

And with pictures

# Enter the static website generator

- Combines the ability to build webpages from a template (like a CMS) to automate the boring stuff (headers, footers, navigation bar, etc) with advantages of a static website
- You can edit your website content knowing only Markdown

Popular static site generators:

Jekyll (<https://jekyllrb.com/>)

Hugo (<https://gohugo.io/>)

# Github Pages Demo

<https://pages.github.com/>

You don't have to host your static site with Github, but it is free and is well-integrated with Jekyll.

# Hosting options

- Static sites are free to host on Github or Netlify
- Reclaim Hosting (\$30/year including free domain registration)
  - Popular with academics
  - Good pricing and service

# Next steps - 1

Do this tutorial – should take 2-3 hours

“Building Static Websites with Jekyll & Github Pages” by  
Amanda Visconti

<https://programminghistorian.org/en/lessons/building-static-sites-with-jekyll-github-pages>

# Next steps - 2

- Practice using Markdown and your text editor

<https://guides.github.com/features/mastering-markdown/>

- Consider using Markdown for all your writing

<https://programminghistorian.org/en/lessons/sustainable-authorship-in-plain-text-using-pandoc-and-markdown>

- Learn some HTML & CSS

<https://learn.shayhowe.com/html-css/>

<https://www.w3schools.com/>



# Use Developer Mode

To peek under the hood of other websites and foundational for doing web scraping

*Chrome:* View → Developer

*Firefox:* Tools → Web Developer → Toggle

*Safari:* Preferences → Advanced → Show Develop Menu

# Next steps - 3

Explore more Jekyll functionality

- Try out includes: <https://jekyllrb.com/docs/includes/>
- Try out layouts: <https://jekyllrb.com/docs/layouts/>

Or try the Hugo static site generator instead

Academic theme: <https://academic-demo.netlify.com/>