

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

Portfolio Websites

UW Bothell ACM Workshop Proposals
November 2018



Why Should ACM Have A Workshop?

- Having a portfolio site is a great way to market students to employers
- Builds rudimentary skills in web development
- Not many web dev resources on the internet have:
 - a step-by-step introduction to HTML, HTML+CSS, or script injection
 - simple boilerplate code that helps students understand the basics
 - information about producing excellent C O N T E N T that is meaningful for CS students who are looking for employment
 - information about the mechanics of website maintenance, hosting, and domain registration
- ACM officers and volunteers can help students follow along with demonstrations



Proposed Structure

1. Presentation about Portfolio Creation - 45 minutes to 1 hour
2. 10 minute break
3. Follow-Along Introduction to Website Creation

Important Prereqs (but not essential):

- Working knowledge of git + existing github account
- A desire to learn
-



Part I: Presentation

1. Why does this matter to you, a CS student?
 - a. Complements and expands on your resume
 - b. Embed audio, video, photographs, gifs, and more! External links! Embed Google calendars (if that's your thing)!
 - c. Helps build one more skill set to have in your arsenal - web development is everywhere
 - d. Takeaway: it's a good idea to have a website, no?
2. Show some sweet examples from current / former Bothell students for inspiration
 - a. <https://etcadinfinity.github.io/index.html> - my website
 - b. <https://chris-johnston.me/> - our illustrious ACM cohort's portfolio site
 - c. Takeaway: slick, interesting websites make you slick and interesting!



Part I: Presentation (Continued)

3. Writing content for your portfolio website

- a. Use the site as a platform to talk about your CS expertise, projects, work experience, and interests.
- b. The website is an extension of your resume - you can be more detailed (and less formal if you so choose).
- c. Similarly, if you have non-textual media that is interesting and relevant, USE IT.
- d. Being concise is advised - do you enjoy reading long blocks of semicoherent text?
- e. Use the strategies and techniques you learned in CSS 301 - engage your audience!
- f. Takeaway: draft your content carefully before you add it to your site; a pretty website is meaningless if it has “word vomit”.



Part I: Presentation (Continued)

4. Now that you have some great content, make the darn website!


- a. Discuss editors: features, downsides, previewing, etc.
 - i. Author recommends vim for local use, www.codeanywhere.com for non-local
- b. When you are just starting out, **Keep **It Simple, Stupid****
- c. Easy mode: use a template :)
- d. Regular mode: create your own static pages in HTML and CSS. Use some script libraries (bootstrap, jQuery) to organize and display content nicely. Maybe use some javascript to provide custom functionality.
- e. Hard mode: use an advanced framework like Angular or Flask.
- f. Takeaway: you can make your website as complicated, feature-rich, and interesting as you want it to be. To start out, make something simple to maintain and don't lose sight of your goal. What is your goal in this context? To describe your CS experience and make yourself an appealing applicant!



Part I: Presentation (Continued)

5. Now that you have a website, you have to publish it to the world!

- a. In this context, “publishing” means two things:
 - i. Providing outside entities a URL to access your site (aka “domain”)
 - ii. Dedicating a machine or service to host and serve your content when someone accesses the URL (aka “host”)
- b. Two options: free, and NOT free
- c. Free: host through GitHub (!!!!!)
 - i. Make a repository called <myUserName>.github.io (eg SpaceKatt.github.io)
 - ii. Commit some files that you want to be displayed
 - iii. Presto magesto, you have a website!
- d. Not free: buy a custom domain, find or create a hosting service, etc.
 - i. Domain names: www.namecheap.com is a good source for available domains
 - ii. Hosting services: You can set up your own machine to host (even a Raspberry Pi, if you *really* want to), or engage a hosting service (such as GoDaddy, Digital Ocean, etc.)



Part II: Workshop Outline & Objectives

- Structured follow-along workshop for creating a .github.io website
- Will include some boilerplate code for folks to mess around with and to introduce:
 - HTML elements (both displayed and undisplayed)
 - CSS tags and properties
 - jQuery - simple use case of injecting HTML into an element from a content file (nice for organizing content, no?)
- Attendees should walk out of the workshop with basic understanding of HTML structure and a template file to build a basic website off of.
- A dummy photograph and fill-in-the-blank text doc will be provided to have a simple “About Me” statement to publish by the end of the workshop if so desired.



Part II: Workshop Agenda

- Step 0: create GitHub account (if it doesn't already exist)
- Step 0.5: Discuss text editors and IDEs
 - Atom, VSCode, etc. Will also briefly demo www.codeanywhere.com (my preferred dev env for web stuff due to great preview link support)
 - Workshop will work with simple text editor and pushing changes to newly created repo
- Step 1: create the new .github.io repository
- Step 2: curl / clone / import sample files (precise mechanism TBD?)
- Step 3: push changes to see "Hello world!" in simplest HTML markdown. Discuss HTML elements, headings.
- Step 4: Uncomment 2nd block of "Hello world" to see and discuss CSS stylings being applied. Discuss style tags, introduce bootstrap.
- Step 5: Uncomment 3rd block of "Hello world" to see and discuss jQuery functionality as it is applied here.

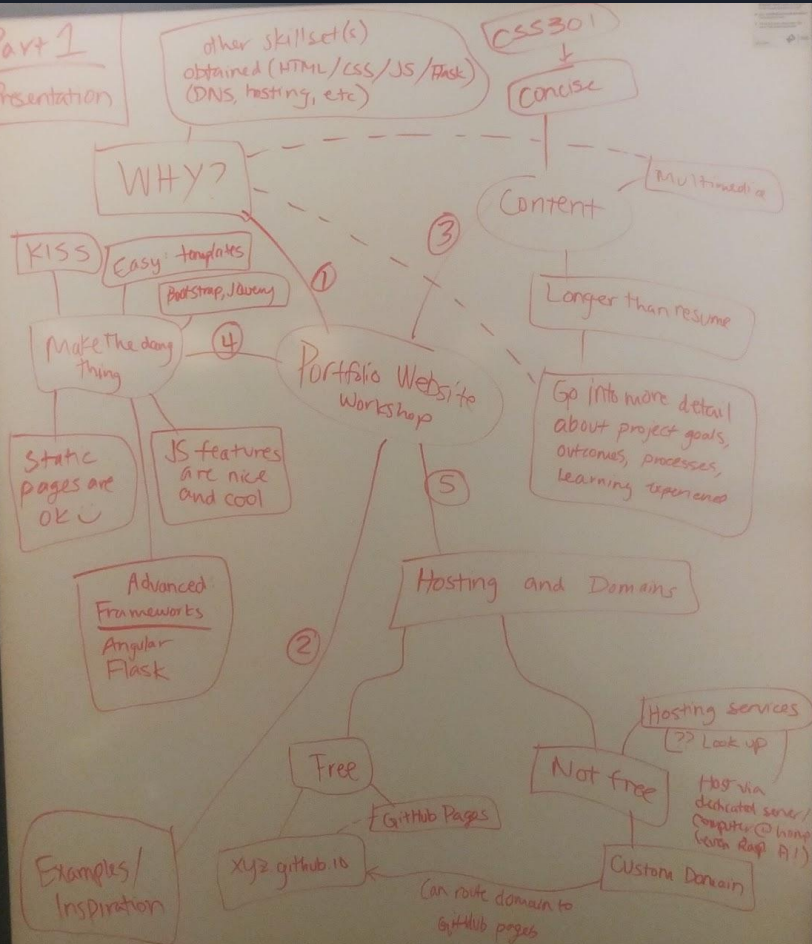


Part II: Workshop Agenda

- Step 6: 15 minutes to design a layout for an “About Me” page section on paper. Think about fonts, color schemes, where to put your photo, text size, div margins, etc.
- Step 7: 30 minutes to implement desired layout for “About Me”, using provided photo & description (focus only on getting the layouts and divs correct). Emphasize use of Bootstrap to create tiled layout that is responsive.
- Step 8: Replace the stock content with your own content (if you so wish).
- Step 9: Push your changes to your repo and see the website come to life!

Part I

Presentation



Part II: Create a .github.io site (~1hr)

① Create the Repository

② Add bootstrap.min.js

② Add JQuery.min.js

③ Hello World - HTML

④ Hello World - CSS

⑤ Sketch a layout design for "About Me"

⑥ Use bootstrap to create the layout

⑦ Add content/Photo (custom or stock)

⑧ Commit + publish

Needs:

- ① sample index.html
- ② .min.js files
- ③ Sample content (Freddie Fiddlesticks!)