Quiz 1 Review

20231010

About this slide deck

- Everything on the quiz can be found in this slide deck.
- There is more in this slide deck than what is on the quiz.
- I will not tell you which is which, so study everything here!
- Quiz is 100% open notes, open Internet.
    - You may not communicate with anyone other than me and the TA during the quiz.
    - Any code sharing plugins to VSCode **must be off during the quiz!**
    - You may be tempted to get back exams. They are of limited utility.

Accomodations

- If you shared with me a memo from DSS, now would be the time to remind me (privately) so that appropriate arrangements can be made.

How does the World Wide Web work?

- There are lots of protocols that undergird the World Wide Web.
- In this class, we talked about two protocols that take place every time you enter a URL into your browser's address bar: DNS and HTTP.
- You should be able to tell me in extreme detail **all of the steps** that happen during DNS resolution. You should also be able to tell me in extreme detail **all of the steps** that make up an HTTP transaction between a client and a server.
- You should be able to tell me the different components of a URL.
- You should be able to tell me (at least some) history of the Web.

HTML

- You should be able to tell me why it is we use HTML to mark up documents.
- You should be able to tell me about all the constituent parts of HTML: tags, attributes, structure, semantics.
- You should be able to explain to me topics and terms such as “progressive enhancement,” “separation of concerns,” and “three tier model.”

CSS

- You should be able to explain to me the constituent parts of CSS rules.
- You should be able to explain the rules that govern which CSS rules apply when you have multiple rules that could apply.
- You should be able to explain to me selector combinators and other variations such as using attributes that allow you fine-grained control over which items on a page are selected.

JavaScript

- You should be able to tell me what things in JavaScript are primitives.
- You should be able to tell me what the difference between primitives and objects are.
- You should be able to tell me what the difference between properties and methods are.
- You should be able to explain why you might use JavaScript on your pages (and why you might not).
- You should be able to tell me how objects work, including but not limited to how objects can be constructed.
- You should be able to tell me how JSON works.

AJAX

- You should be able to explain the steps involved in making an AJAX call **from the perspective of either jQuery or the fetch API (not the old vanilla JS)**.
- You **will not** have any questions about jQuery (but you may use jQuery when you get to the code writing section of the quiz).

APIs

- You should be able to tell me what a REST API is (the six constraints)
- You should be able to tell me how you can access a REST API and get data from it
- You should be able to describe in detail the six constraints, in particular the first one about universal interface

Git(Hub)

- You should be able to tell me about the history (generations) of Version Control Systems.
- You should be able to tell me what a repository is, how it works, and how you can move forwards and backwards through history.
- You should be explain to me in great detail the process by which a file that has been edited on your local machine goes through the git system and eventually appears in your GitHub repository.
- You should be able to explain to me when to use Issues versus Pull Requests.
- You should be able to walk me through good git workflow.

DevOps

- You should be pros at everything with your Azure VMs.
- Turning on, using the command line, syncing your repos with your VMs, everything.

Coding

- I will ask you to code me a website to specifications.
- You may use: HTML, CSS (Bootstrap), JavaScript (jQuery) to code the site.
- You may use libraries and other code you find, but:
  - If the code has an open source license, go for it. **You must have a comment that credits the author and lists the license of the code. You should also include a link to the original GitHub repo (if available)**. You should study the code and understand how it works (I reserve the right to spot-check your knowledge of a library!).
  - **If the code does not have an open source license, you may not use it directly**. But you may study it to learn how problems are solved and re-implement a solution.
- No Generative AI!

Reading code

- I may give you code to read.
- Your job will be threefold:
  - 1. Provide a high-level overview of the code, as if you were explaining it to someone who has never coded before.
  - 2. Provide a line-by-line heavily detailed commentary of the code.
  - 3. If you find mistakes in the code, you must fix those mistakes.
    - **Do not** come up to me on quiz day and tell me you found a mistake in the code because you will lose 5 points and I will tell you to go sit down.

Security

- Everything from the Google Gruyere site is fair game, so make sure you finish it!

That’s it!

If you can do all that, you’re ready for the quiz.