Quiz 2 Review

Same format as last time

- Everything on the quiz is somewhere in here (or in the first quiz review slide deck)
- But there may be more in here than on the quiz
- Nothing wrong with knowing more than you need!
- **Reminder:** quizzes are cumulative

Reading code...

- Reading code is a highly important skill that you should have.
- If I give you code, you should be able to explain the code at a high level, meaning without jargon. As if you were explaining it to someone who doesn't know anything about coding.
- You should also be able to explain it in exacting detail, line by line.
- Remember we have encountered 5 languages: HTML, CSS, JS, PHP, and MySQL. I can give you code in any of those languages, but it will almost certainly be either JS or PHP or MySQL. Review your labs, and review the code snippets and sample code from previous lectures.

Optimization

- Name and explain at least **two** different ways to measure the performance of your web pages. What tools would you use? How do you use them?
- Discuss the logarithmic scale of load times that we talked about in class.
- Name and explain at least **five** different ways to improve web page performance.

PHP

- What are the “dangerous operators” we covered in class? Why are they dangerous? How can these operators still be used well, even if they are dangerous?
- Write a recursive PHP function that reads in three variables, a, b, and c, and multiplies them together but only if a is less than b and b is less than c. c should have a default parameter value of 5.
- What is the difference between a private and protected member in a PHP object?
- Explain the differences between interfaces and abstract classes in PHP.

Why write backend code at all?

- One thing the Free Bee game should have made you realize is that you can do a whole lot with only the frontend. So why even bother writing backend code at all? Describe at least **four** possible features of a web application that necessitates the need for backend code (or at least, is made easier with backend code).

Security

- Explain how you can fail to implement basic security measures (frontend and backend), and what the ramifications of doing so are.

Authentication & Authorization

- Explain the differences between authentication and authorization.
- What are salts? Why use them? How do you use them? What attacks do they defend against?
- Why should you hash passwords? With what algorithms should you hash passwords? Any algorithms that you should *not* use?
- Won't have you write any code for authentication or authorization, but you should know the answers to the above questions.

A simpler version of a lab

The big coding question is just going to be a slightly reworded (and slightly extended) lab. It’s up to you to figure out which lab it is, and you are free to use that lab as your starting point.