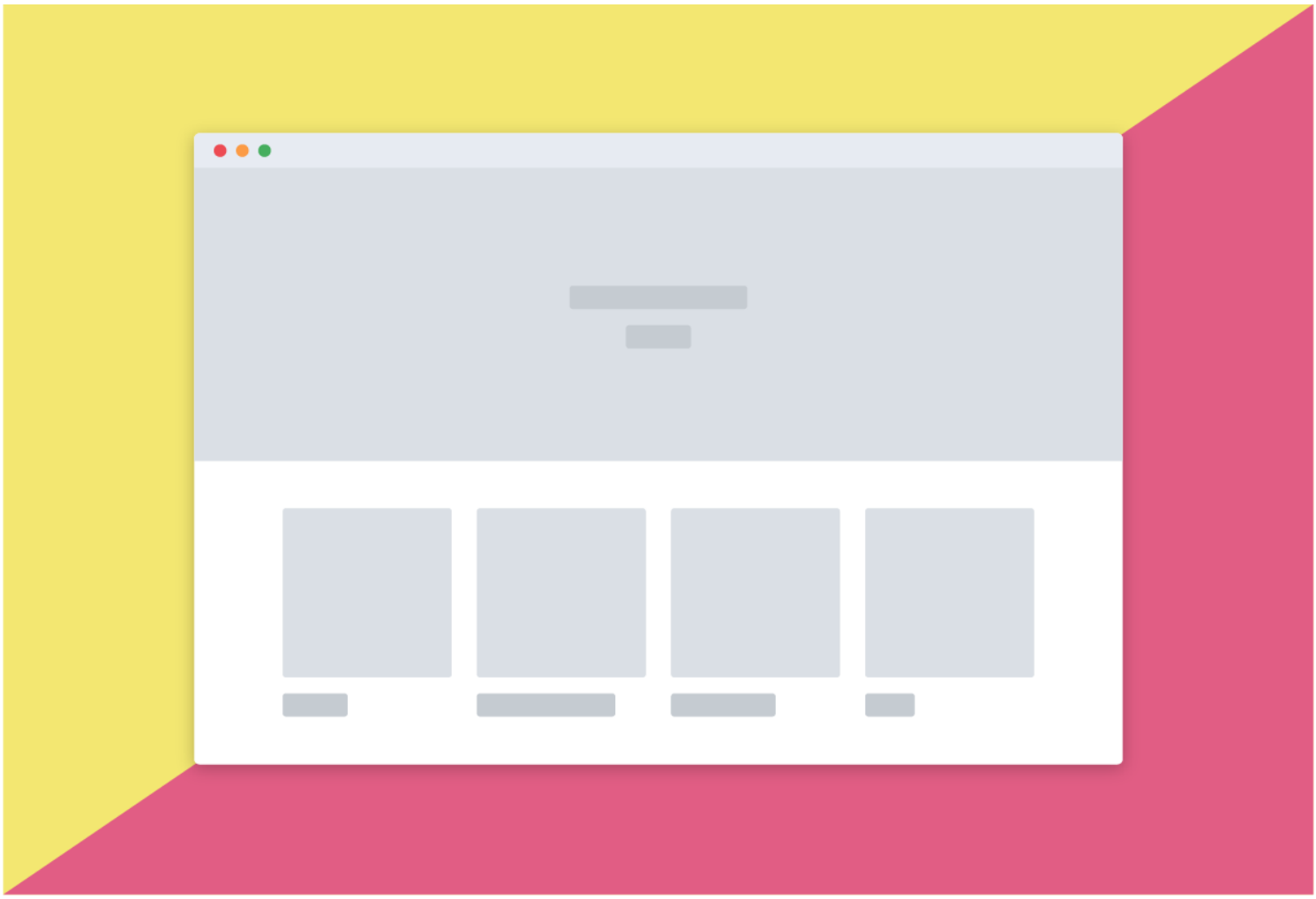
Jonathan Z. White Follow
Designer & developer | Writes about all things to do with building products | twitter.com/jonathanzwhite
May 3, 2016 · 5 min read

Cracking the front-end interview



By @jonathanzwhite

Technical front-end interviews are difficult. That's a fact. Not only do you need to have a solid grasp of computer science fundamentals, but also an understanding of things like web performance, build systems, and CSS layout engines.

While there are resources out there, I've found that there are only a few *complete* guides for helping you prepare for a front-end interview. So I decided to write a topic by topic outline that will hopefully help you ace your next interview.

Before the interview

So before your interview, ask your recruiter for the format of the interview. Some interviews might revolve around a whiteboard while others might use an online text editor like CoderPad. It's important to know so you can practice in the environment that your interview is going to take place in.

Also, ask your recruiter for tips on what topics to focus on when preparing. The reason behind this is because in addition to front-end specific question, some companies will ask you traditional computer science questions about topics like searching and sorting algorithms.

Front-end concepts

HTML and CSS, Javascript, and Javascript design patterns are the key concepts that you will be tested on during an interview. Go through the list and make sure you are comfortable with each topic.

HTML and CSS is like the bread and butter of front-end development. During interviews, you will most likely be asked questions about the nuances of HTML and CSS. Also, be prepared to be asked to code up a layout based on a mockup.

Just in case you need an HTML and CSS refresher, here are a few basic concepts to look over.

- CSS animations
- CSS sprites
- Pseudo classes
- Grid systems
- Semantic markup

In addition to these concepts, know about CSS preprocessors like SASS or LESS and their benefits. Also be familiar with CSS naming conventions like BEM and OOCSS.

Another important point is that interviewers look for candidates who champion CSS best practices. As a good reference, this guide written by a front-end at Medium provides insight into how Medium iterated to their current CSS architecture.

I mentioned earlier that some interviewers will ask you to recreate layouts in HTML and CSS. Practice doing so in a playground like CodePen. Check out Dribbble since it has lots of simple yet nice designs that would be fun to recreate.

Finally, as front-ends, we are so used to making changes in our editor and then verifying the change in our browser. Often times during interviews, you won't have this luxury. When you're preparing for your interview, try code most of your layout *without* looking at the result till the end.

. . .

Javascript Concepts

If HTML and CSS are the bread and butter of front-end development, then Javascript is the knife. Companies will spend a good amount of time during your interview testing your knowledge of Javascript. A lot of the questions will revolve around the following concepts.

- Prototypal inheritance
- Scoping
- Closures
- The event loop
- Event bubbling
- Apply, call, and bind
- Callbacks and promises
- Variable and function hoisting
- Currying

When given a Javascript question, figure out which of these concepts you're being tested on and it'll be much easier to figure out the right solution. If you feel confident of your Javascript prowess, test your knowledge here, here, and here.

. . .

Design patterns in Javascript provide you with repeatable solutions to common problems. These are a few of the design patterns that are important to know.

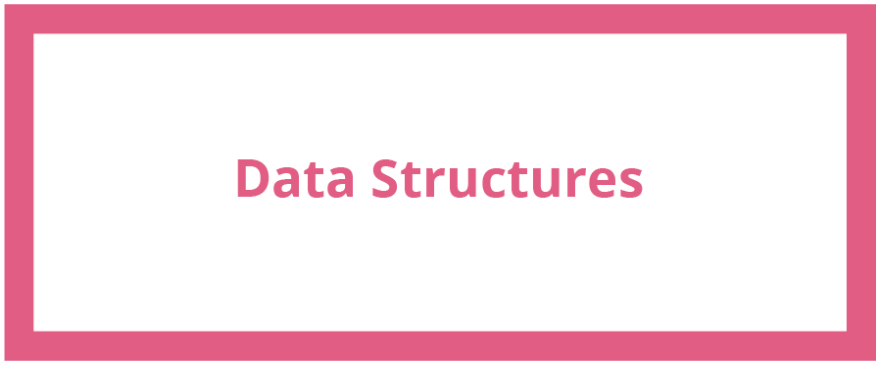
- Decorator
- Factory
- Singleton
- Revealing module
- Facade
- Observer
- MVC, MVP, MVVM

Asides from Javascript design patterns, it's good to be familiar with Javascript frameworks. This does *not* mean that you have to go learn another framework before your interview. Instead, know *when* and *why* front-end teams use frameworks. Also, if you're interviewing for something like a React + Flux or an Angular position, review some of the documentation for the architecture of the framework beforehand.

Computer science concepts

Some companies hire software engineers before front-end developers. What this means is that these companies expect you to be well grounded in topics like good software design principles, scalable code architecture, and testing.

If your recruiter suggests you review your knowledge of data structures and algorithms, this section is for you. *Otherwise, feel free to skip this section.* If you don't have a computer science background, that's okay. Most of these concepts are pretty straightforward to learn.



These are the basic data structures that I would suggest knowing off the top of your head. Don't just read about them, take the time to implement them in Javascript as well. If you're not familiar with unit testing, test your data structure with a library like Mocha to learn.

- Linked lists
- Hashtables
- Stacks and queues
- Trees (binary trees and heaps)
- Graphs

Note: For graphs, also know how to implement depth-first and breadth-first search traversals.

For implementations of these data structures, you can take a look at SanFoundry. All their examples are in Java but re-implementing them in Javascript is pretty easy.

. . .

Once you feel confident about data structures, you can move onto sorting. Go through the list, implement them in Javascript, and then make note of their time and space complexity.

- Binary search
- Bubble sort
- Insertion sort

- Merge sort
- Quick sort
- Selection sort

After reviewing data structures and algorithms, test your knowledge with Leetcode and these technical Javascript questions.

Wrapping it up

I know it's a lot of material to take in, especially if you're new to front-end development. Start preparing for your interview ahead of time, so you can move at a comfortable pace.

Also, remember that interviews are not a measure of your worth as a programmer. Some people are good at interviews, other are not. Sometimes you click with your interviewer, other times you don't.

If you have any questions, feel free to leave a note or Tweet out to me. I would also **love** to know how your interview experience went. Good luck!

P.S. If you liked this article, it would mean a lot if you hit the recommend button or share with friends.

. . .

If you want more, you can follow me on Twitter where I post non-sensical ramblings about design, front-end development, bots, and machine learning.