Any questions on this list are fair game for your final technical interview.
Resources with most answers at the end.

Behavioral Questions

1. Give me an example of the project or initiative that you started on your own. It can be a non-business one. What prompted you to get started?
2. Tell me about a time you had to work on several projects at once. How did you handle this?
3. Describe a situation in which you felt you had not communicated well enough. What did you do? How did you handle it?
4. Tell me about when you had to deal with conflict within your team. How was the conflict solved? How did you handle that? How would you deal with it now?
5. Give me an example of a time you had to take a creative and unusual approach to solve coding problem. How did this idea come to your mind? Why do you think it was unusual?
6. Describe a situation in which you worked diligently on a project and it did not produce the desired results. Why didn't you get the desired results? What did you learn from the experience?
7. Give an example of an important project goal you reached and how you achieved it.
8. Describe a situation in which you experienced difficulty in getting others to accept your ideas? What was your approach? How did this work? Were you able to successfully persuade someone to see things your way
9. Tell me about a situation when you were responsible for project planning. Did everything go according to your plan? If not, then why and what kind of counteractions did you have to take?
10. Tell me about a situation when you made a mistake at work. What happened exactly and how did you deal with it? What steps did you take to improve the situation?
11. Tell me about a time when you worked with someone who was not completing his or her share of the work. How did you handle the situation? Did you discuss your concern with your coworker? With your manager? If yes, how did your coworker respond to your concern? What was your manager's response?
12. Describe a situation when you worked effectively under pressure. How did you feel when working under pressure? What was going on, and how did you get through it?
13. Tell me about yourself.
14. Tell me about your experience at Resilient Coders.
15. What do you know about our company?
16. Why do you want to work for us?
17. Why are you interested in this opportunity.
18. Tell me about your dream job? What do you really want to do with your career?
19. Tell me a time when you failed.
20. What do you read on a regular basis?
21. What's some critical feedback you've gotten recently?
22. Do you have any questions?

Technical Questions

HTML

- What does a doctype do?
- How do you serve a page with content in multiple languages?
- What kind of things must you be wary of when design or developing for multilingual sites?
- What are `data-` attributes good for?
- Consider HTML5 as an open web platform. What are the building blocks of HTML5?
- Describe the difference between a `cookie`, `sessionStorage` and `localStorage`.
- Describe the difference between `<script>`, `<script async>` and `<script defer>`.
- Why is it generally a good idea to position CSS `<link>`s between `<head></head>` and JS `<script>`s just before `</body>`? Do you know any exceptions?
- What is progressive rendering?
- Why you would use a `srcset` attribute in an image tag? Explain the process the browser uses when evaluating the content of this attribute.
- Have you used different HTML templating languages before?

CSS

- What is CSS selector specificity and how does it work?
- What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?
- Describe floats and how they work.
- Describe z-index and how stacking context is formed.
- Describe BFC (Block Formatting Context) and how it works.
- What are the various clearing techniques and which is appropriate for what context?
- Explain CSS sprites, and how you would implement them on a page or site.
- How would you approach fixing browser-specific styling issues?
- How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?
- What are the different ways to visually hide content (and make it available only for screen readers)?

- Have you ever used a grid system, and if so, what do you prefer?
- Have you used or implemented media queries or mobile specific layouts/CSS?
- Are you familiar with styling SVG?
- Can you give an example of an @media property other than screen?
- What are some of the "gotchas" for writing efficient CSS?
- What are the advantages/disadvantages of using CSS preprocessors?
- Describe what you like and dislike about the CSS preprocessors you have used.
- How would you implement a web design comp that uses non-standard fonts?
- Explain how a browser determines what elements match a CSS selector.
- Describe pseudo-elements and discuss what they are used for.
- Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.
- What does * { box-sizing: border-box; } do? What are its advantages?
- What is the CSS display property and can you give a few examples of its use?
- What's the difference between inline and inline-block?
- What's the difference between a relative, fixed, absolute and statically positioned element?
- What existing CSS frameworks have you used locally, or in production? How would you change/improve them?
- Have you played around with the new CSS Flexbox or Grid specs?
- Can you explain the difference between coding a web site to be responsive versus using a mobile-first strategy?
- How is responsive design different from adaptive design?
- Have you ever worked with retina graphics? If so, when and what techniques did you use?
- Is there any reason you'd want to use translate() instead of absolute positioning, or vice-versa? And why?

Javascript

- Explain event delegation
- Explain how this works in JavaScript
- Explain how prototypal inheritance works
- What do you think of AMD vs CommonJS?
- Explain why the following doesn't work as an IIFE: function foo(){ }();. What needs to be changed to properly make it an IIFE?
- What's the difference between a variable that is: null, undefined or undeclared? How would you go about checking for any of these states?
- What is a closure, and how/why would you use one?

- Can you describe the main difference between a .forEach loop and a .map() loop and why you would pick one versus the other?
- What's a typical use case for anonymous functions?
- How do you organize your code? (module pattern, classical inheritance?)
- What's the difference between host objects and native objects?
- Difference between: function Person(){}, var person = Person(), and var person = new Person()?
- What's the difference between .call and .apply?
- Explain Function.prototype.bind.
- When would you use document.write()?
- What's the difference between feature detection, feature inference, and using the UA string?
- Explain Ajax in as much detail as possible.
- What are the advantages and disadvantages of using Ajax?
- Explain how JSONP works (and how it's not really Ajax).
- Have you ever used JavaScript templating? If so, what libraries have you used?
- Explain "hoisting".
- Describe event bubbling.
- What's the difference between an "attribute" and a "property"?
- Why is extending built-in JavaScript objects not a good idea?
- Difference between document load event and document DOMContentLoaded event?
- What is the difference between == and ===?
- Explain the same-origin policy with regards to JavaScript.
- Make this work: duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
- Why is it called a Ternary expression, what does the word "Ternary" indicate?
- What is "use strict";? what are the advantages and disadvantages to using it?
- Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5
- Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?
- Why would you use something like the load event? Does this event have disadvantages? Do you know any alternatives, and why would you use those?
- Explain what a single page app is and how to make one SEO-friendly.
- What is the extent of your experience with Promises and/or their polyfills?
- What are the pros and cons of using Promises instead of callbacks?
- What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?
- What tools and techniques do you use debugging JavaScript code?

- What language constructions do you use for iterating over object properties and array items?
- Explain the difference between mutable and immutable objects.
- Explain the difference between synchronous and asynchronous functions.
- What is event loop? What is the difference between call stack and task queue?
- Explain the differences on the usage of foo between function foo() {} and var foo = function() {}
- What are the differences between variables created using let, var or const?
- What are the differences between ES6 class and ES5 function constructors?
- Can you offer a use case for the new arrow => function syntax? How does this new syntax differ from other functions?
- What advantage is there for using the arrow syntax for a method in a constructor?
- What is the definition of a higher-order function?
- Can you give an example for destructuring an object or an array?
- ES6 Template Literals offer a lot of flexibility in generating strings, can you give an example?
- Can you give an example of a curry function and why this syntax offers an advantage?
- What are the benefits of using spread syntax and how is it different from rest syntax?
- How can you share code between files?
- Why you might want to create static class members?

Javascript General

- Can you name two programming paradigms important for JavaScript app developers?
- What is functional programming?
- What is the difference between classical inheritance and prototypal inheritance?
- What are the pros and cons of functional programming vs object-oriented programming?
- What are two-way data binding and one-way data flow, and how are they different?
- What is asynchronous programming, and why is it important in JavaScript?

Node

- What is Node.js? Where can you use it?
- Why use Node.js?

- What are the features of Node.js?
- How do you update NPM to a new version in Node.js?
- Why is Node.js Single-threaded?
- Explain callback in Node.js.
- What is callback hell in Node.js?
- How do you prevent/fix callback hell?
- Explain the role of REPL in Node.js.
- Name the types of API functions in Node.js.
- What are the functionalities of NPM in Node.js?
- What is the difference between Node.js and Ajax?
- What are “streams” in Node.js? Explain the different types of streams present in Node.js.
- Explain chaining in Node.js.
- What are Globals in Node.js?
- What is Event-driven programming?
- What is Event loop in Node.js work? And How does it work?
- What is the purpose of module.exports in Node.js?
- What is the difference between Asynchronous and Non-blocking?
- What is Tracing in Node.js?
- How will you debug an application in Node.js?
- Difference between setImmediate() vs setTimeout()?
- What is process.nextTick()
- What is package.json? What is it used for?
- What is libuv?
- What are some of the most popular modules of Node.js?
- What is EventEmitter in Node.js?

CS Theory

- What is recursion and give an example using javascript?
- What are types?
- What are data structures?
- What is an algorithm?
- What is scope / lexical scope in javascript?
- What is polymorphism?
- What is encapsulation?
- What is a Linked List
- What is a Doubly Linked List
- What is a Queue

- What is a Stack
- What is a Hash Table
- What is a Heap
- What is a Trie
- What is a Tree
- What is a Binary Search Tree
- What is a Disjoint Set
- What is a Bloom Filter
- Demonstrate Bubble Sort and explain when you might use it?
- Demonstrate Insertion Sort and explain when you might use it?
- Demonstrate Merge Sort and explain when you might use it?
- Demonstrate Quicksort and explain when you might use it?
- Auto Pass

Whiteboard

Expected to solve, determine edge cases, and explain complexity

- Every morning challenge
- Every codewars you have solved / pushed to github

Resources:

https://eloquentjavascript.net/
https://github.com/getify/You-Dont-Know-JS
https://github.com/yangshun/front-end-interview-handbook
https://medium.com/javascript-scene/10-interview-questions-every-javascript-developer-should-know-6fa6bdf5ad95
https://www.simplilearn.com/node-js-interview-questions-and-answers-article
https://medium.com/@vigowebs/frequently-asked-node-js-interview-questions-and-answers-b74fa1f20678