**Johnny Chiang - 04/30/2019**

# Epic Games, Frontend Developer Exercise

## Questions:

1. **What is responsive design all about?**

*The purpose of responsive design is to create a design that fits all devices. Mobile, desktop or tablet, the website needs to be legible and also performant on all platforms.*

2. **Explain some of the pros and cons for CSS animations versus JavaScript animations.**

*The biggest difference between CSS animations vs JavaScript animations is the complexity. While you can get very far and do a lot of with just CSS animations some animations are impossible to do without JavaScript. To me the best features come from a combination of both CSS and JavaScript.*

## HTML Questions:

1. **What are data- attributes good for?**

*Data- attributes are used to embed a value into an HTML element, this value can be accessed with JavaScript and used. I can maybe put a list of heroes on a page and store their username in the data- attribute to be accessed later. Maybe a fetch request to retrieve their data.*

2. **What kind of things must you be wary of when developing for multilingual sites?**

*One of the things I notice when I sometimes visit an eCommerce site is when prices are listed in a different currency or the time-zone is different. Most of the time when I visit a foreign domain, it will automatically redirect me to my own country however this is not always a feature in every site.*

3. **What is progressive rendering?**

*Progressive rendering allows render in visual content the user sees first before running scripts or loading in other images that aren't in view of the viewport.*

# CSS Questions:

1. **Explain CSS sprites, and how you would implement them on a page or site?**

*While I have used sprites to animate my 2D characters in a video game, this type of feature is rarely used on the web. I mostly use CSS sprites to cut or crop a piece of an image where I only need a section of it to display. Pages with lots of images take long to load so sometimes CSS sprites can reduce server requests.*

2. **How would you implement a web design comp that uses non-standard fonts?**

*In many of my projects, I was able to download a custom font and set it with @font-face. I have also imported fonts from Google fonts. A few examples from my GitHub below.*

```
@font-face {
   font-family: ESO;
   src: url("../fonts/Planewalker.otf") format("opentype");
}
```

```
<link
href="https://fonts.googleapis.com/css?family=Cabin|Roboto:400,700|Roboto+Condensed:400,700"
rel="stylesheet">
```

3. **Is there any reason you'd want to use translate() instead of absolute positioning, or vice-versa? And why?**

*The browser usually renders from JavaScript => Style => Layout => Paint => Composite. When you use absolute position it usually affects the layout and onward causing the browser to do more calculations. Transform: translate() only affects the browser on a composite level which is why it's quicker.*

# Javascript Questions:

1. **Explain why the following doesn't work as an IIFE: function foo(){ }();? What needs to be changed to properly make it an IIFE?**

*I thought an IIFE needed to be anonymous functions to invoke immediately but after testing in Node, I learned that it still works even if you name the function. All you need to do is wrap the function in parentheses.*

```
(function foo() { console.log("foo")})();
```

2. **What's the difference between an "attribute" and a "property"?**

*Attributes are usually strings and are values set inside an HTML element. A property is an object that belongs in the DOM that you get and with the property you can set the value of the attribute. These are things that I manipulated heavily with jQuery.*

3. **Explain the difference between mutable and immutable objects.**

*Mutable objects can be modified after it is created while immutable can not be.*

4. **What is a MVC? and how does it differ from a MVV**
   **Can you explain what a promise is in Javascript and when it comes in handy**

*I have worked with Model-View-Controller the most and I see the controller as middleware or ORM. The controller is in a sense the middle man that routes data back and forth from the model to the view and vice-versa.*

*While I have never used the MVVM, I know they are very similar but it's biggest drawback comes from testing.*

*A promise is very handy when you need to wait for pending data to be fully retrieved from an API before an action takes place. Some public APIs are slower and you may want to use a promise to make sure all data was pulled in before you run an asynchronous function because that data is not available yet.*

5. **Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?**

*In my personal experience after I learned ES6 and started using let and const, I ran into problems with scope. After changing some vars into let or const, some of my functions no longer worked! Var is hoisted into the global scope and global scope is generally a bad thing because all code shares the same global space in JavaScript and should be avoided.*

# Test Questions:

1. **What tools do you use to test your code's functionality?**

*I have used Mocha and Jest to test my code, React usually uses Jest to test their code and I believe Angular uses Jasmine or Karma.*

2. **What is the outcome of the two alerts below?**

```
var foo = "Hello";
(function() {
 var bar = " World";
 alert(foo + bar);
})();
alert(foo + bar);
```

*The first result will be “Hello World” however the second will result in bar being undefined because the value of bar is lost after the IIFE is finish invoking.*