**HTML-CSS QC Questions**

1. What is HTML
   * HTML stands for Hypertext Markup Language and consists of various tags to describe the content of a document - utilized as the basis for all web pages, along with CSS and JavaScript
2. What is the structure of an HTML document? List some tags. What is <head> used for? <body>?
   * Start with the doctype declaration, then <html>, then <head> and <body>. The head contains the metadata for the page, while the body contains the content that is rendered to the screen. Other tags: div, span, p, ul, ol, li, strong, em, table
3. What is a doctype?
   * First tag in the document - defines what type of file it is - whether html 4 or 5, etc
4. What is the tag for an ordered list? Unordered list?
   * ordered list: ol, unordered list: ul. Both use li - list items
5. What are some HTML5 tags? Why were HTML5 tags introduced?
   * HTML5 introduced semantic tags to more accurately reflect the content of the tags. Examples: <strong> instead of <b>, <em> instead of <i>, <nav>, <header>, <footer>, <section>, <article>, and <aside> instead of reusing <div> tags everywhere
6. Do all tags come in a pair? What are the other things inside tags called? list some.
   * No - tags either have a closing tag or are self-closing (<tag />). Attributes are contained within tags - examples: id, class, style, height, width, etc
7. What is the syntax for a comment in HTML?
   * <!-- comments go in here -->
8. Give me the HTML markup for a table.

<table>

<caption>optional</caption>

<thead>

<tr>

<td>Heading 1</td><td>Heading 2</td>

</tr>

</thead>

<tr>

<td>Cell 1</td><td>Cell 2</td>

</tr>

<tbody>

</tbody>

</table>

1. What are some tags you would use in a form?
   * Form tags: <form>, <input>, <label>, <textarea>, <button>, <select>, <option>
2. What is CSS? what are the different ways of styling an HTML file? Which is best? why?
   * CSS stands for Cascading Style Sheets - it is a language for styling HTML documents by specifying certain rules for layout and display in key/value pairs. There are 3 ways of styling an HTML file:
   * (1) inline - in the style attribute
   * (2) internal stylesheet - in the <style> tag in the <head>
   * (3) external stylesheet - using external .css file, use <link> in the <head>
   * External stylesheet is best practice due to separation of concerns, reusability, modularity
3. Describe the CSS box model.
   * The box model consists of margin (outermost box), then border, then padding, then content (innermost). All box sizes / formatting can be styled with CSS
4. Which way has highest priority when styles cascade: inline, internal, and external.
   * Inline has highest priority, then internal/external depending on order. Cascading rules are determined by (1) importance (!important flag), (2) specificity of selector (inline has no selector, highest specificity), then (3) source order.
5. Syntax for styling an element? What is a class and how to style them? What is an id? how to style? difference?

div { property: value}

.class { property: value}

#id { property: value}

1. What if I want to select child elements, What syntax is that?
   * use direct descendant selector (>) or space for any level nested element
2. Can I select multiple elements at once? How?
   * yes, with a comma
3. What is a psuedo-class? What is syntax for selecting that?
   * psuedo-class selects an element in a certain state - for example, when hovered over. Use colon (:psuedoselector) syntax
4. What is Bootstrap?
   * Bootstrap is a CSS framework with pre-written CSS rules to easily style your page and incorporate responsive design seamlessly. Contains various utilities and components for making a great UI
5. Describe the Bootstrap grid system
   * The Bootstrap grid is divided into 12 columns. You wrap the columns in a .row which is in a .container or .container-flex. The columns are responsive - there are classes for different screen sizes which collapse on smaller windows

**JavaScript Language Fundamentals**

1. What is JavaScript? What do we use it for?

It is a scripting language used in webpages to add coding to them.

1. Can we run JavaScript in a web browser, on a server, or both?

It is runs on the web browser. You can run JavaScript on the server with an API though.

1. [What programming paradigm(s) does JS support?](https://medium.com/javascript-in-plain-english/what-are-javascript-programming-paradigms-3ef0f576dfdb)

JavaScript is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles.

1. What are the data types in JS?

Boolean,

Integer,

Functions,

Null,

Double,

Float,

Number,

String,

Undefined,

Array

* + What is the datatype of NaN?

Stands for: Not a Number, and is essentially a number

* + What is the isNaN function?

Checks the inputted value whether it is an actual number or not

* + What is the data type of a function?

It is a function, and an object

* + What about an array?

It is an object

* + What is the difference between undefined and null?

Null is a variable with no assigned value. IE: var a = null; <- null value

Undefined is a variable that has not been declared as a type of variable.

IE: var b; <- b is not defined as to what it is? (number, string, integer, Boolean)

1. What are JS objects? what is the syntax?

Objects are made with key:value pairs in a created object.

IE:

Var car =

{

//Key: Value

Type: “Fiat”,

Model: “500”,

Color: “White”

};

1. What is JSON? Is it different from JS objects?

Unlike JavaScript Object, a JSON Object has to be fed into a variable as a String and then parsed into JavaScript. A framework like jQuery can be very helpful when doing parsing.

1. What are some ways you can use functions in JS?

Function dividing(a,b){return a/b;}

Const dividing = function(a,b){return a/b;}

Const dividing = (a,b) => {return a/b;}

Const dividing = (a,b)=> a/b;

1. What are the different scopes of variables in JS?
   * What are the different ways to declare global variables?
   * Is it a best practice to use global variables? Why or why not?
2. What is function and variable hoisting?

Hoisting is where as long as the variable is declared in a retrievable location it is considered declared, and can be accessed anywhere in the file

1. What is the global object in client-side JavaScript? - What are some built-in functions (methods on the global object)?

Global objects are variables that are accessible anywhere in the file. It is at global scope.

Built-In global object functions:

Eval(),

isFinite(),

isNaN();

1. What are callback functions? What about self-invoking functions?

They are functions used in parameters in another function that get called first then set as that parameter.

Self-invoking functions simply invoke themselves. The function cannot have any parameters

1. What is closure and when should you use it?

Accessing outter function variables inside the nested function, we would use this for something like counters. Instead of resetting the counter to 0 each time, would instead increment the number by 1 each time.

1. Use the object literal syntax to create an object with some properties

Var car =

{

Type: “Fiat”,

Model: “500”,

Color: “White”

};

1. What is a truthy or falsy value? List the falsy values.

Falsy values: 0, false, null, “” (empty string), NaN (not a number), undefined

If it is not Falsy, it is Truthy (values for variables)

1. What is the difference between == and ===? Which one allows for type coercion?

== is type coercion, this one tests if the values of the two variables are equal. (5==”5” is true)

=== checks both equality and if both are of the same type

(5===”5” is false) 5 is a number, and “5” is a string

1. Explain the template literal syntax

It uses ‘`’ to make strings where you can input variables to output the value of the variable

Var age = 30;

Var templateLiteral = `I am ${age} years old. ` <- output: I am 30 years old.

1. What are arrays in JS? can you change their size?
2. Explain what “strict mode” does
3. Explain how inheritance works in JS
4. What does the this keyword refer to?
5. Explain the concept of lexical scope
6. What will happen when I try to run this code: console.log(0.1+0.2==0.3) ?

**ES6+**

1. What new features did ES6 introduce?
2. What is the difference between var, let, and const keywords?
3. Does JS have classes? If so, when were they introduced?

**Events and DOM**

1. What is the DOM? How is it represented as a data structure? What object in a browser environment allows us to interact with the DOM?
2. List some ways of querying the DOM for elements
3. How would you insert a new element into the DOM?
4. What are event listeners? What are some events we can listen for? What are some different ways of setting event listeners?
5. What are some methods on the event object and what do they do?