Interview Question Week 3

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

JavaScript is a programming language developed initially and still primarily used for client-side scripting on the web. JS runs natively in the browser, but has expanded into desktop applications and server-side scripting with Node.js

1. **Where is the best place to put a script tag in your HTML document?**

Best practice is to put it just before the close of the <body> tag because so that it won't block the rendering of the rest of the page

1. **What are the data types in JS?**

Object, String, number, Boolean, null, undefined, Symbol

1. **What are the variable scopes in JS?**
   1. *Globa*l - variable can be used everywhere within the script.
      1. see hoisting...
   2. *Local* - variable can be used anywhere in the function they were defined in.
      1. The var keyword indicates a variable will be limited to the local scope.
   3. *Block* - variable can only be used within the block it was defined in.
      1. The let keyword indicates that a variable is of block scope.
2. **What are JS objects? What is the syntax?**

In JavaScript, objects consist of properties which are key/value pairs

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

JSON stands for JavaScript Object Notation and is used as a data interchange format - it consists of key/value pairs similar to JS objects, but it cannot contain functions or circular references.

1. **How to convert JS object to/from JSON?**

Use the JSON.parse() to convert from JSON to JS object, and use JSON.stringify() to convert object to JSON

1. **What does truthy/falsy mean?**

Non-Boolean data types can take on Boolean values when evaluated. For example, 0, empty strings, null, and undefined evaluate to falsy; everything else is truthy

1. **What is type coercion?**

Type coercion refers to the ability of JavaScript to change the type of the variable being evaluated - e.g. 5 == '5' will evaluate to true

1. **What is the difference between == and ===?**

== allows type coercion

1. **How does inheritance work in JS?**

JS has prototypical inheritance - all objects have a link to a prototype object which can link to another prototype object in the prototype chain until the Object prototype is reached

1. **What is unique about functions in JS?**

Functions are objects in JavaScript - they can be passed as parameters, attached to other objects, and nested inside other functions

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

Callback functions are those passed as parameters into other functions and can then be invoked within. Self-invoking functions are wrapped in parentheses and then invoked immediately - (function () {...}) ()

1. **What is closure? Why use closures?**

Closures are the ability for a function to retain access to an outer function's scope even after returning

1. **What is the “this” keyword?**

this refers to the lexical context in which it is used - it is determined by how a function is called. When called in a method, refers to the object the method is called on.

1. **Does JS have classes? when were they introduced?**

ES6 introduced the class syntax, but it is only syntactic sugar for prototype inheritance

1. **What is new with ES6? What are template literals, let keyword, Symbols, Promises?**

Template literals are a new way of denoting strings with backticks, allowing string interpolation let is a new way to declare a variable - will be block-scoped Symbol is a new primitive - used as the key for an object property when intended to be private for internal use Promises are a way of doing asynchronous operations - the Promise object represents the eventual completion or failure of said operation

1. **Arrow function? What makes it different from normal function?**

Arrow function is a shorthand notation for a function which inherits the lexical context of its enclosing scope

1. **What is strict mode?**

ES5 introduced strict mode - it enforces certain rules like preventing accidental global variables. Syntax: 'use strict'

1. **What will happen when I run this code: console.log(0.1+0.2==0.3) ?**

This will print false due to floating point values - the sum will be close to 0.3 but not exact

1. **What are arrays in JS? can you change their size?**

Arrays are an ordered collection of data. Size can be changed dynamically by adding/removing elements. Indexing starts at 0

1. **What is a CDN? What are the benefits?**

CDN stands for Content Delivery Network - we often use them to download libraries by setting the URL in the src attribute of a <script> tag

1. **What is AJAX? why do we use it?**

AJAX stands for Asynchronous JavaScript and XML - it is used to make HTTP requests without blocking execution of other code or rendering on the web page

1. **What are steps to sending an AJAX request?**

Create the XmlHttpRequest object (XHR) 2. Attach onreadystatechange event handler. 3. Open the XHR object with .open(method, url, async) 4. Send XHR object with .send()

1. **What is the DOM?**

The DOM stands for document object model and refers to the tree-like structure of the HTML document.

1. **How to select elements from the DOM? How to insert elements dynamically?**

Use document.getElementById() to select elements by their id. Can also get with classname, tagname, queryselector. Use document.createElement() or document.createTextNode() then invoke .appendChild() or .insertBefore() on an element to add to the DOM

1. **What are events / event listeners? What are some events we can listen for? ways of setting event listeners?**

Events represent actions that occur in the DOM, whether from user input or from an API. Some examples: mouseclick, dblclick, mouseenter, mouseleave, keyup

1. **What is bubbling and capturing and what is the difference?**

Bubbling refers to events propagating up through the DOM from innermost elements to outermost; capturing refers to the opposite process

1. **How do you stop an event from bubbling further?**

use event.stopPropogation()