**JavaScript & Web Development Training Content**

| **DAY I**   * **Introduction to JavaScript**   + What JavaScript Is   + What JavaScript Is Not   + What JavaScript Is Used For   + JavaScript and Its Place in a Web Page * **JavaScript and Events** * **Standardizing JavaScript and the W3C**   + JavaScript Objects   + The Document Object Model * **Where to Put JavaScript**   + JavaScript from External Files * **The HTML Document and JavaScript**   + Script Execution * **Debugging Tools (Should be covered in depth)** * Firefox * Debugging in Chrome , Firefox   **Functions**   * **What Is a Function?**   + Function Declaration and Invocation   + Return Values   + Anonymous Functions as Variables   + Closures   + Recursion   + Functions As Objects   + Using call, bind, apply * **Debugging Techniques (Should be covered in depth)**   + Function Syntax   + Exception Handling with *try/catch*and *throw* * **User-Defined Functions**   + What Is *this*?   + Inline Functions as Methods   **Objects (OOJS)**   * **What Are Objects?**   + Objects and the Dot Syntax   + Creating an Object with a Constructor   + Properties of the Object   + Methods of the Object * **Object Literals** * **Manipulating Objects** * **Loops**   + The *for/in*Loop   **Extending Objects with Prototypes**   * Adding Properties with the Prototype Property * The Prototype Lookup Chain * Adding Methods with Prototype * Properties and Methods of All Objects * Creating Subclasses and Inheritance * Extending Objects via prototype * Using the Correct Constructor with call() and apply() * Extending Objects via Object.create() * **JavaScript Core Objects** * **What Are Core Objects?** * **Javascript closures**   **The Browser’s Object Model (BOM)**   * Browser Compatibility * The document, window & navigator Objects * Browser Objects as Arrays * setTimeout() and window.clearTimeout() * setInterval() and window.clearInterval()   **ECMA 2015/ES6**   * Javascript language specifications- ECMAScript 5 & 6. New language features * Introduction to ES6, Features * Template Literals * Block Scope, Let, Constant, Variables * Rest, Spread, Deconstruct * ES6 Classes, Inheritance * Generator & Iterator functions * Arrow Operator * Object / Array Destructuring * Map / Set * ES6 modules, export, import, module level scope * ES6 Promise, resolve, catch, then * Asynchronous Programming   1. Promises, Async-Await, Error handling in Asynchronous Programming   **DAY II**  **The W3C DOM and JavaScript**   * + How the DOM Works with Nodes   + Parents and Children, Siblings   + The nodeName and nodeType Properties   + Walking with the DOM   + DOM Inspectors   + Methods to Shorten the DOM Walk   + The document.getElementById() Method   + The document.getElementsByTagName() Method * JavaScript Properties to Represent HTML Attributes * Modifying the DOM (Appending, Copying, and Removing Nodes) * Modifying the Content of an Element * Creating New Elements with the DOM * inserting Before a Node * Creating Attributes for Nodes * Cloning Nodes, Removing a Node   **Event Handling and the DOM**   * + The HTML Inline Way, The Scripting Way, The DOM Way   + Bubbling and Capturing   + Event Listeners with the W3C Model   + Adding an Event   + Registering More Than One Event   + Removing an EventListener   + Event Properties Revisited   **JavaScript Object Notation (JSON)**   * + Creating An Object Out Of Thin Air With {}, : and []   + Processing Result Strings with eval()   + Why eval() is Evil!   + The JSON Object   + The parse and stringify Methods   **DAY III**  **AJAX using XMLHttpRequest**   * + What is AJAX?   + Preparing & Sending the Request   + Processing the Results   + Callbacks vs. Promises * Javascript IIFE - Immediately invoked function expression * Javascript design patterns & anti-patterns (bad parts)   + Module Pattern   + Singleton Pattern   + Observer Pattern * AMD - Asynchronous Module Definition, concept of modules, exporting modules. Frameworks providing them- Requirejs framework, Browserify**,**WebPack[Focus more on Webpack] * Javascript MVC/ MVVP/ MVVM /MV\* concepts, their examples [Angular]   High Level Overview Of DOJO Syntax   * Introduction To WebGL * Webworker   1. **Multithreading with web workers**   2. **Communicating with postMessage() and onmessage** * Guidelines around High Performance JavaScript * QUnit & Karma * Code Coverage   **DAY IV**   * W3C DOM   1. HTML Canvas manipulation   2. Improve performance api for canvas   **Advanced CSS**   * Specifity * Position, float, clear, overflow, resize * Margin, padding, z-index * CSS 3 media Queries * Overview of CSS Pre-Processors [LESS,SASS] * Grids, Flexbox, Custom Properties for Responsive Web Design   **Memory Management and Performance tuning**   * What is heap * heap snapshot; memory snapshot; heap allocations * What is Memory Management? Why to worry? * How to find memory utilization * How to find memory leaks/loitering objects * Dealing properly with event handlers * Taking corrective actions in case of problems * Best practices to decrease memory consumption * Code optimization techniques * Using Chrome memory profiler * Identifying bottle necks in code * Memory and Performance Profiling, * Using Chrome Dev tools * **Accessibility in JavaScript Applications**  | Introduction | | --- | | Why Design for Accessibility | | Types of Disabilities | | An Introduction to WCAG 2.0 | | The 4 principles of WCAG 2.0 [POUR] | | A,AA,AAA | | W3C Validator and testing tools (WAVE, Lighthouse) | | What is Semantic HTML? | | Designing Color Accessible Pages |   **DAY V**  **Overview of Web Components and Custom Elements**   * Default HTML5 Web Components * Creating Custom Web Components * Using Templates and Slots in WebComponents     **Interactive reports and WebAssembly**   * Intro to Webpack and BabelJS(Just for the team to get a feel of what webpack does as it is a widely used tool)   **Web workers & Service Workers**   * Offline applications using Service Worker * Service workers events : Install, Activate, Fetch * Using cache |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |