

## JavaScript Bootcamp Training Course

Duration: 4.5 months

### MODULE 1: THE JAVASCRIPT BASICS

#### LVL 1: Introduction to software programming. Web application architecture. First Steps in JS

1. Introduction. Software Engineer mindset. Course outline.
2. What is a computer program? Writing a program using Armenian language. The “Kebab maker program” (this is the Armenian version of famous “*How to make a peanut butter and jelly sandwich*” task).
3. High level languages. Assembler language. Input and Output (I/O). Numeric representation with bits: binary number system (*There are only 10 different kinds of people in the world: those who know binary and those who don't*).
4. The architecture of a web application. Front-end, Back-end, Databases: how do they communicate? Programming languages used in web. The era of “write once - run anywhere” applications.
5. JavaScript basics. The role of JS in the modern software development. Brief history.
6. DEMONSTRATION: Writing a simple I/O JS application

#### LVL 2: Introduction to JavaScript

1. JavaScript concepts. The JS script environment. Using a browser as an advanced user. Browser Developer Tools (mainly focusing on console).
2. PRACTICAL: Writing the first console application.
3. Environmental setup. Tips and tricks.
4. JavaScript variable types. Declaring variables. The Boolean, Number and String
5. JavaScript operators.
6. JavaScript control flows: If/Else, Switch
7. JavaScript Loops: For, While
8. PRACTICAL: Writing a more complex JS application using conditional statements and loops

#### LVL 3: Arrays, Objects and Functions

1. What is an array? Mutable and immutable arrays (examples from python)
2. Creating and accessing arrays
3. What is a function? How to declare and call them in JS
4. Function I/O: function arguments, returning values from functions
5. Global and user defined functions
6. Anonymous/arrow functions

7. What is a SCOPE? Global and local scopes: function scope
8. PRACTICAL: Write a program using multiple functions
9. The JavaScript Object Notation (JSON)
10. Creating and accessing Objects in JavaScript. Objects inside Objects (nested objects)

#### **LVL 4: Interacting with the page**

1. The Document Object Model (DOM)
2. The Browser Object Model (BOM)
3. Interacting with DOM. Locating DOM elements by writing selectors
4. JavaScript Event Loop and data communication
5. Basic DOM events explained: click, focus, blur, mouseover, mouseenter, keydown, input etc
6. PRACTICAL: writing a 'Try to catch <ANY ANIMAL>' game using JavaScript event loop
7. JavaScript Style object
8. Creating and styling content dynamically

#### **LVL 5: JavaScript Trends. Where does the language go?**

1. JavaScript Standards. ECMAScript 6. The problems with browser incompatibilities.
2. JavaScript libraries and frameworks: JQuery, Angular, React.Js, Vue.js
3. JavaScript in backend: Node.js

### **MODULE 2: ADVANCED JAVASCRIPT**

#### **LVL 1: Scope**

1. Scope and JS Compiler. Compiling function scope
2. Function declaration, function expressions and function execution
3. Block scope
4. Lexical scope. Lexical scope cheating: eval
5. Block scope in ES6. Let keyword: some problems
6. Dynamic scope
7. Hoisting
8. *this* keyword
9. Binding. Explicit and Implicit binding
10. the *new* keyword

#### **LVL 2: Closure**

1. Closures
2. Closure examples
3. Module patterns

### **LVL 3: Object Oriented JavaScript**

1. Is JavaScript an object oriented language?
2. Prototype and prototype linkages
3. Prototype: Objects linked
4. Linked Prototype diagram
5. Prototype behavior
6. Inheritance. OLOO (Object linked to other objects)
7. PRACTICAL: Write a JS application that mimics an OOP app

### **LVL 4: Async Patterns**

1. Functions inside functions. Pure functions
2. Callbacks: solving callback problems
3. Generators
4. Promises
5. asynquence
6. Observable pattern
7. PRACTICAL: Write an application following async patterns

### **LVL 5: Network Communication. AJAX. REST Approach. API**

1. How a client application interacts with servers. HTTP Protocol
2. Request → response path
3. HTTP verbs: GET, POST, DELETE, PUT
4. What is an API and how to interact with them?
5. REST approach explained
6. Creating http calls. AJAX requests
7. PRACTICAL: Implement a public API in a simple JavaScript application

### **LVL 6: Working with HTML 5 APIs**

1. Web Storage
2. Web Workers
3. Web Sockets
4. Canvas 2D

5. **PRACTICAL:** Write a JS app using public APIs and some of the aforementioned HTML 5 APIs

### **LVL 7: You are a PIXAR artist or we can make it move. JS animations**

1. Animation types: CSS 3 animations
2. Basics of JavaScript animations. Famous JS animation libraries
3. GreenSock or GSAP: the new standard for HTML and JavaScript animations
4. **DEMONSTRATION:** Using GSAP to move things in the browser
5. Wanna be a 3D artist? Brief introduction to WebGL (Web Graphics Library)

## **MODULE 2: DEVELOPING WEB APPLICATIONS USING ANGULAR 2 FRAMEWORK** (*During this module we will develop a hotel booking application*)

### **LVL 1: Introduction to Angular Framework**

1. Brief history. What is Angular? What is Single Page application
2. Technologies and principles
3. Environmental Setup
4. Data communication
5. Component pattern
6. React.js, Angular and Vue.js comparison
7. **DEMONSTRATION:** Writing a small angular application

### **LVL 2: TypeScript Fundamentals**

1. Declarations and Annotations
2. Type Interface
3. Static and Dynamic Typing
4. Compile Time or Run Time
5. Applying Types
6. Objects
7. Functions: Arrow Functions
8. Functions and Interfaces

#### **2.1. *TypeScript: Typing, Variables and Functions***

9. Declarations and Annotations
10. Type Interface
11. Static and Dynamic Typing
12. Compile Time or Run Time
13. Applying Types

14. Objects
15. Functions: Arrow Functions
16. Functions and Interfaces

## **2.2. *TypeScript: Classes, Interfaces, Decorators***

17. Defining and using classes
18. Casting and Type Definition
19. Inheritance: Extending types
20. Defining and using Interfaces
21. Multiple Implementation and single inheritance
22. Differences between interfaces and classes
23. Decorators
24. Property Decorators
25. Class Decorators
26. Parameter Decorators

## **LVL 3: Components**

1. The Component Pattern in The Front End Technologies
2. Creating an Angular Component
3. Component Decorator and Component Metadata
4. Imports. Component Inheritance
5. Bootstrapping Components

## **LVL 4: Template. Data Binding**

1. Building a template for a component
2. Data Binding Explained
3. Data Binding: From Model to View (Interpolation, property binding)
4. Data Binding: From View to Model (Event Binding)
5. Directives: Directive decorator and metadata
6. Building a directive and using in a component
7. \*ngIf and \*ngFor structural directives
8. Two-way binding: ngModel
9. Pipes

## **LVL 5: Data Flow and Communication in Angular**

1. @Input decorator
2. @Output decorator

3. Introduction to Redux
4. Angular Component Lifecycle and Lifecycle Hooks

### **LVL 6: Dependency Injection. Services. Http**

1. How DI works?
2. @Injectable decorator and metadata: creating an Injectable service
3. Registering and injecting an Injectable
4. Angular HTTP Service
5. Creating an Injectable HTTP Service

### **LVL 7: Navigation and Routing**

1. How routing works
2. Configure routing
3. Handling exceptions

### **LVL 8: Template driven and Reactive forms**

1. The differences between template driven and reactive forms
2. Creating a Template driven form: adding validation to it
3. Creating a Reactive or Model Driven form
4. FormBuilder, FormGroup, FormControl
5. Adding Validation to Reactive forms

### **LVL 9: Angular Modules**

1. What is a module? Registering a module
2. Imports, Exports, Providers, Bootstrap, Declarations
3. Using multiple modules in a single application
4. Communication between modules

### **LVL 10: Unit Testing**

1. Why we need unit tests in front end?
2. TestBed.configure()
3. Isolating testable functions with mocking the data
4. Running unit tests
5. Testing private functions