

JavaScript Syllabus (ES6+)

Time: 20 hrs. [B] / 40 hrs. [I, A]

This workshop covers JavaScript content for all levels and discusses JS fundamentals, ES6 and beyond ES6 features. We can apply this knowledge on web development, NodeJS and for scripting on some automation tools.

Labels:

[B] -> Beginner, [I] -> Intermediate, [A] -> Advanced

[Number] -> No. Of hours to complete the topic

M 1. Introduction and setting up environment [B][2]

- i. Introduction
- ii. History
- iii. Releases
- iv. Setting up Code editor
- v. Installing Node.js
- vi. Installing GIT
- vii. Learning How to Code
- viii. How to Think Like a Developer: Become a Problem Solver!
- ix. Using Google, Stack-Overflow and MDN
- x. Debugging (Fixing Errors)
- xi. Debugging with the Console and Breakpoints

M 2. JavaScript Fundamentals [B][6]

- i. Hello World!
- ii. Values and Variables
- iii. Data Types
- iv. let, const and var
- v. Basic Operators and Precedence
- vi. Strings and Template Literals
- vii. Taking Decisions: if / else and switch Statements
- viii. Type Conversion and Coercion
- ix. Truthy and Falsey Values
- x. Equality Operators: == vs. ===
- xi. Logical Operators
- xii. Statements and Expressions
- xiii. The Conditional (Ternary) Operator
- xiv. Functions
- xv. Function Declarations vs. Expressions

- xvi. Arrow Functions
- xvii. Functions Calling Other Functions
- xviii. Introduction to Arrays
- xix. Basic Array Operations (Methods)
- xx. Introduction to Objects
- xxi. Dot vs. Bracket Notation
- xxii. Object Methods
- xxiii. Iteration: The for Loop
- xxiv. Looping Arrays, Breaking and Continuing
- xxv. Looping Backwards and Loops in Loops
- xxvi. The while Loop

M 3. JavaScript in the Browser: DOM and Events[B][4]

- i. What's the DOM and DOM Manipulation
- ii. Selecting and Manipulating Elements
- iii. Handling Events
- iv. Manipulating CSS Styles
- v. Refactoring Our Code: The DRY Principle

M 4. How JavaScript Works Behind the Scenes [A][5]

- i. A High-Level Overview of JavaScript
- ii. The JavaScript Engine and Runtime
- iii. Execution Contexts and The Call Stack
- iv. Scope and The Scope Chain
- v. Scoping in Practice
- vi. Variable Environment: Hoisting and the TDZ
- vii. Hoisting and TDZ in Practice
- viii. This Keyword
- ix. Regular Functions vs. Arrow Functions
- x. Primitives vs. Objects (Primitive vs. Reference Types)

M 5. Data Structures, Modern Operators and Strings [B][6]

- i. Destructuring Arrays
- ii. Destructuring Objects
- iii. The Spread Operator (...)
- iv. Rest Pattern and Parameters
- v. Short Circuiting (&& and ||)
- vi. The Nullish Coalescing Operator (??)
- vii. Logical Assignment Operators
- viii. Looping Arrays: The for-of Loop
- ix. Enhanced Object Literals
- x. Optional Chaining (?.)
- xi. Looping Objects: Object Keys, Values, and Entries
- xii. Sets
- xiii. Maps: Fundamentals
- xiv. Maps: Iteration
- xv. Summary: Which Data Structure to Use?

- xvi. Working With Strings
- xvii. String Methods in Practice

M 6. Functions - A Closer Look [I][2]

- i. Default Parameters
- ii. How Passing Arguments Works: Value vs. Reference
- iii. First-Class and Higher-Order Functions
- iv. Functions Accepting Callback Functions
- v. Functions Returning Functions
- vi. The call and apply Methods
- vii. The bind Method
- viii. Immediately Invoked Function Expressions (IIFE)
- ix. Closures

M 7. Working With Arrays [I][3]

- i. Simple Array Methods
- ii. The new at Method
- iii. Looping Arrays: forEach
- iv. forEach With Maps and Sets
- v. Data Transformations: map, filter, reduce
- vi. The map Method
- vii. The filter Method
- viii. The reduce Method
- ix. The Magic of Chaining Methods
- x. The find Method
- xi. some and every
- xii. flat and flatMap
- xiii. Sorting Arrays
- xiv. More Ways of Creating and Filling Arrays
- xv. Summary: Which Array Method to Use?
- xvi. Array Methods Practice

M 8. Numbers, Dates, Intl and Timers [B][2]

- i. Converting and Checking Numbers
- ii. Math and Rounding
- iii. The Remainder Operator
- iv. Numeric Separators
- v. Working with BigInt
- vi. Creating Dates
- vii. Operations with Dates
- viii. Internationalizing Dates (Intl)
- ix. Internationalizing Numbers (Intl)
- x. Timers: setTimeout and setInterval
- xi. Implementing a Countdown Timer

M 9. Object-Oriented Programming (OOP) With JavaScript [I][3]

- i. What is Object-Oriented Programming?

- ii. OOP in JavaScript
- iii. Constructor Functions and the new Operator
- iv. Prototypes
- v. Prototypal Inheritance and The Prototype Chain
- vi. Prototypal Inheritance on Built-In Objects
- vii. ES6 Classes
- viii. Setters and Getters
- ix. Static Methods
- x. Object.create
- xi. Inheritance Between "Classes": Constructor Functions
- xii. Inheritance Between "Classes": ES6 Classes
- xiii. Inheritance Between "Classes": Object.create
- xiv. Encapsulation: Protected Properties and Methods
- xv. Encapsulation: Private Class Fields and Methods
- xvi. Chaining Methods

M 10. Asynchronous JavaScript: Promises, Async/Await, and AJAX [I, A][4]

- i. Asynchronous JavaScript, AJAX and APIs
- ii. Our First AJAX Call: XMLHttpRequest
- iii. How the Web Works: Requests and Responses
- iv. Welcome to Callback Hell
- v. Promises and the Fetch API
- vi. Consuming PromisesChaining Promises
- vii. Handling Rejected Promises
- viii. Throwing Errors Manually
- ix. Asynchronous Behind the Scenes: The Event Loop
- x. The Event Loop in Practice
- xi. Building a Simple Promise
- xii. Consuming Promises with Async/Await
- xiii. Error Handling With try...catch
- xiv. Returning Values from Async Functions
- xv. Running Promises in Parallel
- xvi. Other Promise Combinators: race, allSettled and any

M 11. Modern JavaScript Development: Modules, Tooling, and Functional [I, A][3]

- i. An Overview of Modern JavaScript Development
- ii. An Overview of Modules in JavaScript
- iii. Exporting and importing in ES6 Modules
- iv. Top-Level await (ES2022)
- v. The Module Pattern
- vi. CommonJS Modules
- vii. A Brief Introduction to the Command Line
- viii. Introduction to NPM
- ix. Bundling With Parcel and NPM Scripts
- x. Configuring Babel and Polyfilling

- x i. Review: Writing Clean and Modern JavaScript
- x ii. Declarative and Functional JavaScript Principles