

Course content

JavaScript Fundamentals – Part 1

- Section Intro
- Hello World!
- A Brief Introduction to JavaScript
- Linking a JavaScript File
- Values and Variables
- Practice Assignments
- Data Types
- let, const and var
- Basic Operators
- Operator Precedence
- Coding Challenge #1
- Strings and Template Literals
- Taking Decisions: if / else Statements
- Coding Challenge #2
- Type Conversion and Coercion
- Truthy and Falsy Values
- Equality Operators: == vs. ===
- Boolean Logic
- Logical Operators
- Coding Challenge #3
- The switch Statement
- Statements and Expressions
- The Conditional (Ternary) Operator
- Coding Challenge #4
- JavaScript Releases: ES5, ES6+ and ESNext

JavaScript Fundamentals – Part 2

- Section Intro
- Activating Strict Mode
- Functions
- Function Declarations vs. Expressions
- Arrow Functions
- Functions Calling Other Functions
- Reviewing Functions
- Coding Challenge #1
- [Introduction to Arrays](#)
- Basic Array Operations (Methods)
- Coding Challenge #2
- Introduction to Objects
- Dot vs. Bracket Notation
- Object Methods
- Coding Challenge #3

- Iteration: The for Loop
- Looping Arrays, Breaking and Continuing
- Looping Backwards and Loops in Loops
- The while Loop
- [Coding Challenge #4](#)

Developer Skills & Editor Setup

- Section Intro
- Section Roadmap
- Setting up Prettier and VS Code
- Installing Node.js and Setting Up a Dev Environment
- Learning How to Code
- How to Think Like a Developer: Become a Problem Solver!
- Using Google, StackOverflow and MDN
- Debugging (Fixing Errors)
- Debugging with the Console and Breakpoints
- Coding Challenge #1

JavaScript in the Browser: DOM and Events Fundamentals

- Section Intro
- Section Roadmap
- PROJECT #1: Guess My Number!
- What's the DOM and DOM Manipulation
- Selecting and Manipulating Elements
- Handling Click Events
- Implementing the Game Logic
- Manipulating CSS Styles
- Coding Challenge #1
- Implementing Highscores
- Refactoring Our Code: The DRY Principle
- PROJECT #2: Modal Window
- Working With Classes
- Handling an "Esc" Keypress Event
- [PROJECT #3: Pig Game](#)
- Rolling the Dice
- Switching the Active Player
- Holding Current Score
- Resetting the Game

How JavaScript Works Behind the Scenes

- Section Intro
- Section Roadmap
- An High-Level Overview of JavaScript
- The JavaScript Engine and Runtime
- Execution Contexts and The Call Stack

- [Scope and The Scope Chain](#)
- Scoping in Practice
- Variable Environment: Hoisting and The TDZ
- Hoisting and TDZ in Practice
- The this Keyword
- The this Keyword in Practice
- Regular Functions vs. Arrow Functions
- Primitives vs. Objects (Primitive vs. Reference Types)
- Primitives vs. Objects in Practice

Data Structures, Modern Operators and Strings

- Section Intro
- Section Roadmap
- Destructuring Arrays
- Destructuring Objects
- The Spread Operator (...)
- Rest Pattern and Parameters
- Short Circuiting (&& and ||)
- The Nullish Coalescing Operator (??)
- Coding Challenge #1
- Looping Arrays: The for-of Loop
- Enhanced Object Literals
- Optional Chaining (?.)
- Looping Objects: Object Keys, Values, and Entries
- Coding Challenge #2
- Sets
- Maps: Fundamentals
- Maps: Iteration
- Summary: Which Data Structure to Use?
- Coding Challenge #3
- Working With Strings - Part 1
- Working With Strings - Part 2
- Working With Strings - Part 3
- Coding Challenge #4
- String Methods Practice

A Closer Look at Functions

- Section Intro
- Section Roadmap
- Default Parameters
- How Passing Arguments Works: Value vs. Reference
- First-Class and Higher-Order Functions
- Functions Accepting Callback Functions
- Functions Returning Functions
- The call and apply Methods
- The bind Method

- Coding Challenge #1
- Immediately Invoked Function Expressions (IIFE)
- [Closures](#)
- More Closure Examples
- Coding Challenge #2

Working With Arrays

- Section Intro
- Section Roadmap
- Simple Array Methods
- Looping Arrays: forEach
- forEach With Maps and Sets
- [PROJECT: "Bankist" App](#)
- Creating DOM Elements
- Coding Challenge #1
- Data Transformations: map, filter, reduce
- The map Method
- Computing Usernames
- The filter Method
- The reduce Method
- Coding Challenge #2
- The Magic of Chaining Methods
- Coding Challenge #3
- The find Method
- Implementing Login
- Implementing Transfers
- The findIndex Method
- some and every
- flat and flatMap
- Sorting Arrays
- More Ways of Creating and Filling Arrays
- Summary: Which Array Method to Use?
- Array Methods Practice
- Coding Challenge #4

Numbers, Dates, Intl and Timers

- Section Intro
- Section Roadmap
- Converting and Checking Numbers
- Math and Rounding
- The Remainder Operator
- Working with BigInt
- Creating Dates
- Adding Dates to "Bankist" App
- Operations With Dates
- Internationalizing Dates (Intl)

- Internationalizing Numbers (Intl)
- Timers: setTimeout and setInterval
- Implementing a Countdown Timer

Advanced DOM and Events

- Section Intro
- Section Roadmap
- [PROJECT: "Bankist" Website](#)
- How the DOM Really Works
- Selecting, Creating, and Deleting Elements
- Styles, Attributes and Classes
- Implementing Smooth Scrolling
- Types of Events and Event Handlers
- Event Propagation: Bubbling and Capturing
- Event Propagation in Practice
- Event Delegation: Implementing Page Navigation
- DOM Traversing
- Building a Tabbed Component
- Passing Arguments to Event Handlers
- Implementing a Sticky Navigation: The Scroll Event
- A Better Way: The Intersection Observer API
- Revealing Elements on Scroll
- Lazy Loading Images
- Building a Slider Component: Part 1
- Building a Slider Component: Part 2
- Lifecycle DOM Events
- Efficient Script Loading: defer and async

Object-Oriented Programming (OOP) With JavaScript

- Section Intro
- Section Roadmap
- What is Object-Oriented Programming?
- OOP in JavaScript
- Constructor Functions and the new Operator
- Prototypes
- [Prototypal Inheritance and The Prototype Chain](#)
- Prototypal Inheritance on Built-In Objects
- Coding Challenge #1
- ES6 Classes
- Setters and Getters
- Static Methods
- Object.create
- Coding Challenge #2
- Inheritance Between "Classes": Constructor Functions
- Coding Challenge #3
- Inheritance Between "Classes": ES6 Classes

- Inheritance Between "Classes": Object.create
- Another Class Example
- Encapsulation: Protected Properties and Methods
- Encapsulation: Private Class Fields and Methods
- Chaining Methods
- ES6 Classes Summary
- Coding Challenge #4

Mapty App: OOP, Geolocation, External Libraries, and More

- Section Intro
- Section Roadmap
- [Project Overview](#)
- How to Plan a Web Project
- Using the Geolocation API
- Displaying a Map Using Leaflet Library
- Displaying a Map Marker
- Rendering Workout Input Form
- Project Architecture
- Refactoring for Project Architecture
- [Managing Workout Data: Creating Classes](#)
- Creating a New Workout
- Rendering Workouts
- Move to Marker On Click
- Working with localStorage
- Final Considerations

Asynchronous JavaScript: Promises, Async/Await, and AJAX

- Section Intro
- Section Roadmap
- Asynchronous JavaScript, AJAX and APIs
- Our First AJAX Call: XMLHttpRequest
- [OPTIONAL] How the Web Works: Requests and Responses
- Welcome to Callback Hell
- Promises and the Fetch API
- Consuming Promises
- [Chaining Promises](#)
- Handling Rejected Promises
- Throwing Errors Manually
- Coding Challenge #1
- Asynchronous Behind the Scenes: The Event Loop
- The Event Loop in Practice
- Building a Simple Promise
- Promisifying the Geolocation API
- Coding Challenge #2
- Consuming Promises with Async/Await
- Error Handling With try...catch

- Returning Values from Async Functions
- Running Promises in Parallel
- Other Promise Combinators: race, allSettled and any
- Coding Challenge #3

Modern JavaScript Development: Modules, Tooling, and Functional

- Section Intro
- Section Roadmap
- An Overview of Modern JavaScript Development
- An Overview of Modules in JavaScript
- Exporting and Importing in ES6 Modules
- The Module Pattern
- CommonJS Modules
- A Brief Introduction to the Command Line
- Introduction to NPM
- Bundling With Parcel and NPM Scripts
- Configuring Babel and Polyfilling
- Review: Writing Clean and Modern JavaScript
- Let's Fix Some Bad Code: Part 1
- Declarative and Functional JavaScript Principles
- Let's Fix Some Bad Code: Part 2

Forkify App: Building a Modern Application

- Section Intro
- Section Roadmap
- [Project Overview and Planning \(I\)](#)
- Loading a Recipe from API
- Rendering the Recipe
- Listening For load and hashchange Events
- The MVC Architecture
- Refactoring for MVC
- Helpers and Configuration Files
- Event Handlers in MVC: Publisher-Subscriber Pattern
- Implementing Error and Success Messages
- Implementing Search Results - Part 1
- Implementing Search Results - Part 2
- Implementing Pagination - Part 1
- Implementing Pagination - Part 2
- Project Planning II
- Updating Recipe Servings
- Developing a DOM Updating Algorithm
- Implementing Bookmarks - Part 1
- Implementing Bookmarks - Part 2
- Storing Bookmarks With localStorage
- Project Planning III
- Uploading a New Recipe - Part 1

- Uploading a New Recipe - Part 2
- Uploading a New Recipe - Part 3
- Wrapping Up: Final Considerations

Setting Up Git and Deployment

- Section Intro
- Section Roadmap
- Simple Deployment With Netlify
- Setting Up Git and GitHub
- Git Fundamentals
- Pushing to GitHub
- Setting Up Continuous Integration With Netlify