## Chapter 1

# How to code and run your first apps

## Objectives

### Applied

1. Use VS Code to edit HTML, CSS, and JavaScript files.
2. Use an extension with VS Code to run an app in a web browser.
3. Use Chrome to run JavaScript apps that are on your computer or the internet.
4. Use Chrome's developer tools to find the statement that caused an error in an app.

### Knowledge

1. Describe the components of a web app.
2. Describe HTTP requests and responses.
3. Distinguish between static web pages and dynamic web pages.
4. Describe the use of JavaScript in a web app.
5. Describe the ECMAScript specification.
6. Describe a basic process for detecting syntax errors and runtime errors.

## Chapter 2

# How to get started with JavaScript

## Objectives

### Applied

1. Code block comments and single-line comments.
2. Declare and initialize variables and constants of the number, string, and Boolean data types.
3. Code expressions that use arithmetic operators and methods from the Math and Number objects.
4. Use the console object to display data to a browser's debugging console.
5. Combine strings of characters using the + operator, += operator, and template literals.
6. Use escape sequences to include special characters in strings.
7. Convert strings to numbers using parseInt(), parseFloat(), and Number().
8. Use dialogs to get input and display output.

### Knowledge

1. Describe how case-sensitivity, semicolons, and whitespace relate to the syntax for a JavaScript statement.
2. Describe these three primitive data types: numeric, string, and Boolean.
3. Describe what an object is and give an example of how one can be used.
4. Describe the syntax for referring to a method or property of an object.

# Chapter 3

# How to code control statements

## Objectives

### Applied

1. Code conditional expressions that use relational and logical operators.
2. Control the flow of an app using if statements, switch statements, and the conditional operator.
3. Repeatedly execute blocks of code using while, do-while, and for loops.

### Knowledge

1. Differentiate between the equality and strict equality operators.
2. Name two short-circuit operators and explain how they work.
3. Describe the order of precedence for evaluating a compound conditional expression.
4. Describe how nesting statements works.
5. Differentiate between true and false values and truthy and falsy values.
6. Describe the use of break and continue statements within a loop.

## Chapter 4

# How to work with arrays and strings

## Objectives

### Applied

1. Create arrays and add, replace, or delete its elements.
2. Use for, for-in, and for-of loops with arrays.
3. Use the methods of an Array object to modify, copy, inspect, and transform an array.
4. Use the methods of a String object to inspect and modify a string.
5. Split a string into an array and convert an array to a string.
6. Chain method calls.

### Knowledge

1. Describe when you would use for, for-in, and for-of loops with arrays.

## Chapter 5

# How to code functions and handle events

## Objectives

### Applied

1. Code function declarations and arrow functions.
2. Define default values for function parameters.
3. Pass parameters to functions and methods using the rest and spread operators.
4. Create event handlers and attach them to events.
5. Use the currentTarget property and the preventDefault() method of the Event object to work with events.

### Knowledge

1. Distinguish between arrow functions and function expressions.
2. Distinguish between global scope, local scope, and block scope.
3. Briefly describe the Document Object Model (DOM).
4. Describe when the DOMContentLoaded event occurs and what you typically use it for.

## Chapter 6

# How to script the DOM

## Objectives

### Applied

1. Use the querySelector() and querySelectorAll() methods of the document object to get elements from the DOM.
2. Script elements and their attributes using their properties and methods.
3. Script form and control elements including text boxes, text areas, select lists, radio buttons, and check boxes.
4. Preload images.

### Knowledge

1. Describe the use of these properties for working with nodes: parentNode, childNodes, firstChild, lastChild, nextElementSibling, nodeValue, and textContent.
2. Distinguish between the DOM API and the HTML DOM API.
3. Describe how DOM scripting can be used for data validation.
4. Identify what triggers the following events: focus, blur, click, dblclick, change, and select.

# Chapter 7

# How to test and debug an app

## Objectives

### Applied

1. Develop a test plan for an app.
2. Use Chrome's developer tools to debug an app by setting breakpoints, viewing the current data values, and stepping through the execution of statements.
3. Trace the execution of an app by using methods of the console object.
4. View the HTML and CSS for a web page.

### Knowledge

1. Distinguish between testing and debugging.
2. Distinguish between syntax, runtime, and logic errors.
3. Describe the debugging problem that can occur if you don't use strict mode and JavaScript treats undeclared variables as global variables.

## Chapter 8

# How to work with dates, times, and timers

## Objectives

### Applied

1. Create a Date object from a string, numbers, another Date object, or a timestamp.
2. Get and set the parts of a Date object, such as year, month, day, minutes, and seconds.
3. Format a Date object.
4. Perform calculations on a Date object.
5. Start and stop one-time timers and interval timers.

### Knowledge

1. Name the different parts of a Date object.
2. Define UTC and describe how time zones work with Date objects.

## Chapter 9

# How to work with data validation and exceptions

## Objectives

### Applied

1. Use HTML to perform basic data validation.
2. Customize HTML data validation.
3. Use try-catch statements to catch exceptions.
4. Create and throw exceptions.

### Knowledge

1. Identify two limitations of HTML data validation.
2. Describe the use of the following HTML attributes: autofocus, placeholder, required, title, pattern, novalidate, autocomplete.
3. Describe how you can use regular expressions to check for a pattern in a string.
4. Identify three reasons to deliberately throw an exception in your code.

# Chapter 10

# How to work with web storage

## Objectives

### Applied

1. Set items in web storage and get items from web storage.
2. Use the nullish coalescing operator to provide a default value for an item in web storage.
3. Convert objects to and from the JSON format.
4. Use Chrome's developer tools to view, add, edit, or delete items in web storage.

### Knowledge

1. Distinguish between session cookies and persistent cookies.
2. Describe three differences between cookies and web storage.
3. Distinguish between session storage and local storage.

## Chapter 11

# More skills for working with arrays

## Objectives

### Applied

1. Inspect an array using the isArray(), find(), findIndex(), every(), and some() methods.
2. Sort an array by using the sort() method with a callback function.
3. Modify an array by using the forEach() method with a callback function.
4. Filter, map, and reduce the elements in an array.
5. Work with an array of arrays.

### Knowledge

1. Describe how to destructure an array.
2. Distinguish between a shallow copy of an array and a deep copy.

## Chapter 12

# How to work with objects

## Objectives

### Applied

1. Use object literals to define objects.
2. Use classes to define objects.
3. Create and use JavaScript libraries.
4. Create private properties and methods.
5. Create accessor properties.
6. Create static properties and methods.
7. Create and use nested objects.
8. Use a generator function to make an object iterable.

### Knowledge

1. Distinguish between object literals and classes.
2. Explain how two variables can refer to the same object.
3. Define encapsulation and describe one of its benefits.
4. Distinguish between data properties and accessor properties.
5. Distinguish between inheritance and object composition and describe when you would use each technique.
6. Describe how to create and use cascading methods.
7. Describe how you can destructure an object.

# Chapter 13

# How work with modules

## Objectives

### Applied

1. Export and import items from an ES module.
2. Declare a script as a module.
3. Rename imports and exports.
4. Import and export classes.
5. Create a default export.
6. Use an import map.

### Knowledge

1. Describe three benefits of using modules.
2. Recognize closures, IIFEs, and the module pattern in legacy code and understand how they work.

## Chapter 14

# How to work with Ajax

## Objectives

### Applied

1. Use your browser to make a GET request to a web service.
2. Use the Fetch API to make GET, POST, PUT, and DELETE requests that update a web page without reloading it.
3. Use the *async* and *await* keywords to work with asynchronous functions.
4. Use an API for a web service to get data from a website.
5. Use a server-side proxy to get data from a web service that doesn't allow cross-origin requests.

### Knowledge

1. In general terms, describe how Ajax works.
2. Distinguish between XML and JSON.
3. Name and describe the three states of a Promise object.
4. Describe how a callback function works and why callback functions are essential to asynchronous programming.
5. Describe how a top-level await works.

# Chapter 15

# How to work with Node.js

## Objectives

### Applied

1. Use Node.js to interactively test JavaScript outside of a web browser.
2. Use Node.js to run a server-side script that has been saved in a file, including passing arguments to the script.
3. Use the built-in modules available from Node.js to write server-side scripts.
4. Create a module that works with Node.js.
5. Use NPM to install modules that can help you write server-side scripts.
6. Use Express to create a web-based API with routes for GET, POST, PUT, and DELETE requests.
7. Create a server-side proxy that you can use to make cross-origin requests.

### Knowledge

1. Describe how the process object provided by Node.js allows you to access the arguments that are passed to a script from the command prompt.
2. Distinguish between the CommonJS module system and the ES module system.
3. Describe what NPM is and how it can help you.
4. Distinguish between a package and a module.
5. Describe how package.json files work.
6. Describe semantic versioning and how to use special characters with dependencies to indicate compatible versions.

## Chapter 16

# How to work with drawing and animation

## Objectives

### Applied

1. Use the Canvas API to draw rectangles, text, lines, circles, and arcs on a web page.
2. Change the color of lines and shapes.
3. Create basic animations.
4. Use the HTML Drag and Drop API to make it possible to drag elements from one location on a web page and drop them in another.

### Knowledge

1. Generally, describe what the Canvas API does and how it works.
2. Describe the use of fallback text for providing accessibility.
3. Define path in the context of the Canvas API.