# Chapter 1

# Introduction to web development

## Objectives

### Applied

1. Use a Chrome to run JavaScript applications that are on the Internet, your computer, or a local server.
2. Use Chrome's developer tools to find the JavaScript statement that caused an error in a JavaScript application.
3. Use an IDE or text editor such as VS Code to edit HTML, CSS, and JavaScript files.
4. If you're using an IDE or text editor such as VS Code that lets you run web applications from it, use your IDE or text editor to run an application.

### Knowledge

1. Describe the components of a web application.
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 application.
5. Describe the ECMAScript specification.
6. Describe browser support for the ECMAScript specification.
7. Describe the use of HTML and CSS.
8. Distinguish between the HTML5 semantic elements and the HTML div and span elements.
9. Describe the use of these HTML attributes: id, class, title, for, and name.
10. Describe the coding for these types of CSS selectors: type, id, and class.
11. Describe the components of a CSS style rule.
12. Describe the components of a URL.

# Chapter 2

# Get started fast with JavaScript

## Objectives

### Applied

1. Given the specifications for a JavaScript application that requires only the skills that you've learned so far, code, test, and debug the application.

### Knowledge

1. Describe two ways to include JavaScript in the body of an HTML document.
2. Describe how case-sensitivity, semicolons, and whitespace relate to the syntax for a JavaScript statement.
3. List the primary rules for creating a JavaScript identifier.
4. Describe the use of JavaScript comments, including "commenting out" portions of JavaScript code.
5. Describe these three primitive data types that are available from JavaScript: numeric, string, and Boolean.
6. Describe the use of variable and constant declarations and assignment statements with numeric, string, and Boolean data.
7. Distinguish between using the *let* and *const* keywords to declare constants and variables and using the *var* keyword to declare variables and constants.
8. Describe the use of the arithmetic operators and the rules for evaluating an arithmetic expression, including order of precedence and the use of parentheses.
9. Distinguish between using the + operator and a template string literal when working with strings.
10. Explain how to use an escape sequence such as \n to include a special character in a string.
11. Describe the syntax for referring to a method or property of an object.
12. Describe the use of the alert(), prompt(), parseInt(), parseFloat(), write(), and toFixed() methods.

## Chapter 3

# The essential JavaScript statements

## Objectives

### Applied

1. Given the specifications for a JavaScript application that requires only the skills that you've learned so far, code, test, and debug the application.

### Knowledge

1. Describe the rules for evaluating a conditional expression that consists of relational operators, including the use of the isNaN() method.
2. Describe the use of the logical operators and the rules for evaluating a compound conditional expression, including order of precedence and the use of parentheses.
3. Describe the flow of control for an if statement that has both else if and else clauses.
4. Describe two ways that you can code a conditional expression that tests whether a Boolean variable is true.
5. Describe the flow of control for while, do-while, and for loops.
6. Describe the use of a JavaScript array, including the use of indexes and the length property of an array.
7. Describe the use of a for, for-in, and for-of loop for processing each element in an array.

## Chapter 4

# How to work with JavaScript objects, functions, and events

## Objectives

### Applied

1. Given the specifications for a JavaScript application that requires only the skills that you've learned so far, code, test, and debug the application.

### Knowledge

1. Distinguish between these objects: window, document, Textbox, Number, Date, and String.
2. Describe the way Number and String objects are created.
3. Describe these methods of the window object: parseInt() and parseFloat().
4. Describe these methods of the document object: querySelector(), querySelectorAll(), and write().
5. Describe these methods and properties of a Textbox object: focus(), select(), value, and disabled.
6. Describe the toFixed() method of a Number object.
7. Describe the way Date objects are created.
8. Describe these methods of a Date object: toDateString(), getFullYear(), getDate(), and getMonth().
9. Describe the length property of a string and these methods of a string: indexOf(), substr(), substring(), toLowerCase(), and toUpperCase().
10. Describe the creation and use of function declarations, function expressions, and arrow functions.
11. Distinguish between function expressions and arrow functions.
12. Distinguish between global scope, local scope, and block scope.
13. Describe how to create and attach event handlers, including how to attach an event handler to the DOMContentLoaded event of the document object.
14. Describe the currentTarget property and the preventDefault() method of the Event object.

## Chapter 5

# How to test and debug a JavaScript application

## Objectives

### Applied

1. Develop test plans for the applications that you develop.
2. Use Chrome's developer tools to debug applications by setting breakpoints, viewing the current data values, and stepping through the execution of statements.
3. Trace the execution of an application with console.log() statements.
4. View the source code for a web page.

### Knowledge

1. Distinguish between the goals of testing and debugging.
2. Distinguish between syntax, runtime, and logic errors.
3. Describe the type of debugging problem that can occur when you use floating-point numbers in arithmetic expressions, and describe one way to fix this problem.
4. Describe the type of debugging problem that can occur if you don't use strict mode and JavaScript treats undeclared variables as global variables.
5. In general terms, describe how to create a test plan.
6. Describe the procedure for tracing the execution of an application with console.log() statements.

## Chapter 6

# How to script the DOM with JavaScript

## Objectives

### Applied

1. Develop DOM scripting applications that work with forms and controls, including applications that add nodes to the DOM.
2. Use Chrome's developer tools to view the changes that JavaScript has made to the DOM.

### Knowledge

1. Describe the use of the DOM in JavaScript applications.
2. Describe these properties of the Node interface for the DOM: parentNode, childNodes, firstChild, lastChild, nextElementSibling, nodeValue, and textContent.
3. Describe these methods of the Document interface for the DOM: querySelector() and querySelectorAll().
4. Describe these methods of the Element interface for the DOM: hasAttribute(), getAttribute(), setAttribute(), and removeAttribute().
5. Explain how the DOM HTML specification can simplify coding when compared to the DOM Core specification.
6. Describe how to use the classList property of the Element interface.
7. Describe the use of a form that has a submit button and a reset button.
8. Describe the use of Textbox, Textarea, Select, Radio, and Checkbox objects.
9. Describe these methods for working with forms and controls: submit(), reset(), focus(), blur(), and select().
10. Describe these events for working with controls: focus, blur, click, dblclick, change, and select.
11. Describe the process of using the appendChild() and insertBefore() methods of the Node interface to add nodes to the DOM.

# Chapter 7

# How to work with images and timers

## Objectives

### Applied

1. Preload the images for an application when the images aren't automatically loaded by the HTTP request for the web page.
2. Use one-time and interval timers in your JavaScript applications.

### Knowledge

1. Describe the use of images and preloaded images in applications like image swaps and slide shows.
2. Describe these timer methods: setTimeout(), clearTimeout(), setInterval(), and clearInterval().

# Chapter 8

# Get off to a fast start with jQuery

## Objectives

### Applied

1. Use jQuery to develop common DOM scripting applications like the Email List, FAQs, Image Swap, and Image Rollover applications that are presented in this chapter.

### Knowledge

1. Describe jQuery.
2. Describe two ways to include the jQuery library in your web pages.
3. In general terms, describe the use of jQuery selectors, methods, and event methods.
4. Describe the syntax for a jQuery selector.
5. Describe the use of these jQuery methods for working with forms and controls: val(), text(), next(), submit(), and focus().
6. Describe the use of object chaining.
7. Describe the use of these jQuery methods: next(), prev(), find(), attr(), css(), addClass(), removeClass(), toggleClass(), html(), hide(), show(), and each().
8. Describe the use of these jQuery event methods: ready(), click(), dblclick(), mouseover(), mouseout(), and hover().
9. Describe the use of the Event object, its currentTarget property, and its preventDefault() method.

# Chapter 9

# How to use effects and animations

## Objectives

### Applied

1. Use jQuery to add effects and animations to a web page.

### Knowledge

1. In general terms, describe the use of the jQuery methods for effects.
2. Describe how to stop and start a slide show that uses an interval timer.
3. In general terms, describe how the jQuery animate() method works.
4. Explain how the chaining effects and animations works and how callback functions can affect the results.
5. Describe the use of the delay(), stop(), and finish() methods with effects and animations.
6. Describe the use of easings with effects and animations.

# Chapter 10

# How to work with forms and data validation

## Objectives

### Applied

1. Use any of the jQuery selectors, methods, and event methods for working with forms and controls in a JavaScript application.

### Knowledge

1. In general terms, describe the HTML5 controls and attributes for working with forms.
2. In general terms, describe the HTML5 attributes and CSS3 pseudo-classes for data validation.
3. Describe the use of these jQuery selectors for working with forms and controls: :input, :text, :radio, :checkbox, :submit, :reset, :button, :disabled, :enabled, :checked, and :selected.
4. Describe the use of the jQuery trim() method for working with text boxes.
5. Describe the use of these jQuery event methods for working with forms and controls: focus(), blur(), change(), select(), and submit().
6. Describe the use of these jQuery methods for triggering events: focus(), blur(), change(), select(), and submit().

# Chapter 11

# How to use jQuery plugins and UI widgets

## Objectives

### Applied

1. Use jQuery plugins and UI widgets in your applications.

### Knowledge

1. In general terms, describe a jQuery plugin.
2. In general terms, describe the process of finding and using a plugin.
3. From the website user's point of view, describe the operation of any of these plugins: Lightbox, bxSlider, or Cycle 2.
4. In general terms, describe a jQuery UI widget.
5. Describe the files and folders that jQuery UI requires, and explain how to include them in your web pages.
6. In general terms, describe the process of finding and using a jQuery UI widget.
7. From the website user's point of view, describe the operation of any of these jQuery UI widgets: Accordion, Tabs, Button, Dialog, or Datepicker.

# Chapter 12

# How to work with numbers, strings, and dates

## Objectives

### Applied

1. Use the properties and methods of Number, String, and Date objects in your applications.
2. Use the random() method of the Math object to generate a random integer within a specific range.

### Knowledge

1. Describe these special numerical values: Infinity, -Infinity, NaN, Number.MAX_VALUE, Number.MIN_VALUE, Number.MAX_SAFE_INTEGER, Number.MIN_SAFE_INTEGER, and Number.EPSILON.
2. Describe these methods of a Number object: toFixed() and toString().
3. Describe these static methods of the Number type: isNaN(), isFinite(), isInteger(), and isSafeInteger().
4. Describe these properties and methods of the Math object: PI, abs(), round(), ceil(), floor(), trunc(), pow(), sqrt(), min(), max(), and random().
5. Describe these properties and methods of a String object: length, charAt(), concat(), indexOf(), substr(), substring(), toLowerCase(), toUpperCase(), startsWith(), endsWith(), includes(), trimStart(), trimEnd(), trim(), padStart(), padEnd(), and repeat().
6. Describe how to use the split() method of a String object to create an array.
7. Describe how to create a Date object for the current date and time or for a specified date and time.
8. Describe these methods of the Date object: toString(), toDateString(), toTimeString(), getTime(), getFullYear(), getMonth(), getDate(), getDay(), getHours(), getMinutes(), getSeconds(), getMilliseconds(), setFullYear(), setMonth(), setDate(), setHours(), setMinutes(), setSeconds(), and setMilliseconds().
9. Describe the coding that's required to add or subtract dates.
10. Describe how to use the Internationalization API to format numbers and dates.

## Chapter 13

# How to work with control structures, exceptions, and regular expressions

## Objectives

### Applied

1. Use the identity operators in your control structures.
2. Use break and continue statements in your while, do-while, and for loops.
3. Use switch statements, including those that use fall through and default cases.
4. Use the conditional operator for simple logic requirements.
5. Use the AND and OR operators for selections.
6. Use try-catch statements to catch errors.
7. Create and throw Error objects.
8. Create regular expressions and use them to match patterns in strings.

### Knowledge

1. Describe type coercion and distinguish between the equality and identity operators.
2. Describe the use of break and continue statements in loops.
3. Describe the use of a switch statement.
4. Describe the use of a conditional operator as a replacement for an if statement.
5. Describe how to use non-Boolean values in conditions.
6. Explain how the short-circuit evaluations of the AND and OR operators can be used in selections.
7. Describe the use of the try, catch, and finally blocks in a try-catch statement.
8. Describe the how to create and throw Error objects.
9. Describe the use of regular expressions for matching patterns with strings.

## Chapter 14

# How to work with browser objects, cookies, and web storage

## Objectives

### Applied

1. Use the properties and methods of the location and history objects in your applications.
2. Use cookies in your applications.
3. Use web storage in your applications.
4. Use Chrome's developer tools to work with cookies and web storage.

### Knowledge

1. Describe these properties and methods of the location object: href, reload(), and replace().
2. Describe these properties and methods of the history object: length, back(), forward(), and go().
3. Describe the use of cookies.
4. Distinguish between session and persistent cookies.
5. Describe the use of web storage.
6. Distinguish between session storage and local storage.

# Chapter 15

# How to work with arrays, sets, and maps

## Objectives

### Applied

1. Use arrays in your applications.
2. Use associative arrays and arrays of arrays in your applications.
3. Use sets and maps in your applications.

### Knowledge

1. Describe the creation and use of an array.
2. Describe the use of indexes and the length property for working with an array.
3. Distinguish between the use of for, for-of, and for-in loops for working with an array.
4. Describe how destructuring can assign the elements in an array to individual constants or variables.
5. Describe these methods of a Array object: push(), pop(), unshift(), shift(), splice(), slice(), indexOf(), lastIndexOf(), includes(), entries(), values(), and keys().
6. Describe these methods of an array that accept callback function as parameters: find(), findIndex(), filter(), every(), some(), forEach(), sort(), reduce(), map(), and flatMap().
7. Describe the static Array.isArray() method.
8. Describe how to split a string into an array.
9. Describe at least two ways to copy and array.
10. Distinguish between an array, an associative array, and an array of arrays.
11. Describe JSON.
12. Describe how you can convert an array to JSON and back.
13. Distinguish between an array, a set, and a map.

# Chapter 16

# How to work with objects

## Objectives

### Applied

1. Use object literals to create objects that have properties and methods.
2. Use classes to create objects that have properties and methods.
3. Use the properties and methods of the objects you create.
4. Use JavaScript libraries in your applications.

### Knowledge

1. Distinguish between creating objects with object literals and creating objects with classes.
2. Distinguish between the concise method syntax for coding methods and the traditional syntax.
3. Explain how two variables can refer to the same object.
4. Distinguish between data properties and accessor properties.
5. Describe how getters and setters work with read-only and write-only properties.
6. Describe the creation, use, and benefits of JavaScript libraries.
7. Describe how you can use a class to define objects that have properties and methods.
8. Describe the inheritance hierarchy for these JavaScript object types: Object, String, Number, Boolean, Date, Array, and Function.
9. Distinguish between inheritance and object composition and describe when you would use each technique.
10. Distinguish between a class-based language and a prototypal language.
11. Describe how to use a symbol and a generator function to create an iterator for an object.
12. Describe the creation and use of cascading methods.
13. Describe how you can use destructuring to assign the values stored in an object to individual variables or constants.

## Chapter 17

# How to work with functions, closures, and modules

## Objectives

### Applied

1. Use default parameters and rest parameters in your functions.
2. Use closures, IIFEs, the module pattern, and ES modules in your applications.

### Knowledge

1. Describe how default parameters work.
2. Distinguish between the rest operator and the spread operator.
3. Describe how the *this* keyword works in arrow functions when compared to function expressions and function declarations.
4. Describe how the scope chain works in JavaScript.
5. Describe the creation and use of closures, including how to use closures to create private state.
6. Describe the creation and use of IIFEs, including how to use an IIFE to create private state for an object literal.
7. Describe the use of the module pattern.
8. Describe the creation and use of namespaces.
9. Describe the use of ES modules.

# Chapter 18

# How to work with Ajax

## Objectives

### Applied

1. Use your browser to review the response that's returned from a request to a web service.
2. Use the Fetch API to make Ajax requests that update a web page without reloading it.

### Knowledge

1. In general terms, describe how Ajax works.
2. Distinguish between XML and JSON.
3. Describe how to use an API for a web service to get data from a website.
4. Describe how to use a browser to request data from a web service and review its response.
5. Distinguish between the XMLHttpRequest object and the Fetch API.
6. Describe how a callback function works and why callback functions are essential to asynchronous programming.
7. Describe these Fetch API methods: fetch(), then(), catch(), and json().
8. Name and describe the three states of a Promise object.
9. Describe the "callback hell" problem and how the Fetch API solves that problem.
10. Describe how named functions can make a chain of promises easier to read, understand, and maintain.
11. Describe how you can use the *async* and *await* keywords to work with asynchronous functions.
12. Describe how you can use a server-side proxy to get data from a web service that doesn't allow cross-origin requests.

# Chapter 19

# 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 JavaScript file.
3. Use the built-in modules available from Node.js to help you write server-side scripts.
4. Use NPM to install and use modules that can help you write server-side scripts.

### Knowledge

1. In general terms, describe Node.js and when you might want to use it.
2. Distinguish between the console object provided by Node.js and the console object provided by a browser.
3. 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.
4. Distinguish between the CommonJS module system used by Node.js and the ES module system.
5. Describe how you can use the require() function that's available from Node.js to import a module.
6. Distinguish between the readFile() function that's available from the fs module and the readFile() function that's available from the fs.promises API.
7. Describe how you can use the exports property that's available from Node.js to export a module.
8. Name and describe the two main components of NPM.
9. Distinguish between a package and a module.
10. Describe semantic versioning.