

# CIW JavaScript Specialist v3



## Certification Exam: 1DO-835

### Exam Objectives

#### Domain 1: Essential JavaScript Principles and Practices

##### 1.1: Identify characteristics of JavaScript and common programming practices.

- 1.1.1: List key JavaScript characteristics, including object-based nature, events, platform independence, and differences between scripting languages and programming languages.
- 1.1.2: Identify common programming concepts, including objects, properties, and methods.
- 1.1.3: Describe various JavaScript versions and flavors, including ECMA standards, JScript and similarities with proprietary scripting languages.
- 1.1.4: Distinguish between server-side and client-side JavaScript applications, including JavaScript interpreters and rendering engines.
- 1.1.5: Describe acceptable coding practices, including appropriate use of comment tags and the <noscript> tag.
- 1.1.6: Examine the evolution of the role of JavaScript in simple Web page design, such as gaming, virtual reality (VR), native development, mobile device scripting and backend development.

##### 1.2: Work with variables and data in JavaScript.

- 1.2.1: Use attributes and methods to communicate with users, including the type attribute.
- 1.2.2: Define variables, data types and scope.
- 1.2.3: List keywords and reserved words.
- 1.2.4: Store user input in variables and evaluate for appropriate responses using the console and built-in methods such as alert() and prompt().
- 1.2.5: Distinguish between concatenation and addition.
- 1.2.6: Apply operators, including string concatenation (+), strict comparison (===, !==), mathematical precedence and bitwise operators.
- 1.2.7: Demonstrate how to use expressions.
- 1.2.8: Implement simple event handlers, including keyboard, mouse and mobile (gestures or touch) events.

##### 1.3: Use JavaScript functions, methods, and events.

- 1.3.1: Define and use methods as functions.
- 1.3.2: Demonstrate the use of various types of functions and function elements including prototype functions, anonymous functions, closure functions, arguments and the use of return values.
- 1.3.3: Distinguish between global and local variables.
- 1.3.4: Use the conditional operator.
- 1.3.5: Identify user events and event handlers.
- 1.3.6: Demonstrate the use of function specific methods including calling, binding and applying.
- 1.3.7: Use built-in functions and cast variables.

#### Domain 2: Intermediate JavaScript Programming Techniques

##### 2.1: Debug and troubleshoot JavaScript code.

- 2.1.1: Demonstrate common steps for debugging JavaScript code, including reviewing code and testing code in different browsers and various devices.
- 2.1.2: Demonstrate how to use various native and supplemental debugging tools, including enabling/disabling displays.

##### 2.2: Use JavaScript statements to control program flow.

- 2.2.1: Use the if... statement.
- 2.2.2: Use the while... statement.
- 2.2.3: Use the do...while statement.
- 2.2.4: Use the for... statement.
- 2.2.5: Use the forEach statement.
- 2.2.6: Use the break and continue statements.
- 2.2.7: Use the switch... statement.

**2.3: Use the JavaScript Document Object Model (DOM).**

- 2.3.1: Use JavaScript to manipulate the Document Object Model (DOM).
- 2.3.2: Use the window object of the DOM.
- 2.3.3: Manipulate properties and methods of the document object within the DOM.
- 2.3.4: Use the image object of the DOM, including image rollover creation.
- 2.3.5: Use the history object of the DOM.
- 2.3.6: Evaluate and change URL information with the location object of the DOM.
- 2.3.7: Use the navigator object of the DOM.
- 2.3.8: Describe virtual DOM.

**2.4: Use JavaScript language objects and create expressions.**

- 2.4.1: Use the String object to test user input.
- 2.4.2: Evaluate strings, including use of the length property, and use of the indexOf(), lastIndexOf(), substring(), and charAt() methods.
- 2.4.3: Implement basic regular expressions and the RegExp object.
- 2.4.4: Use the Array object to create more efficient code.
- 2.4.5: Use the map() method.
- 2.4.6: Apply the Date and Math objects.

**2.5: Create and use custom JavaScript objects.**

- 2.5.1: Create a custom JavaScript object.
- 2.5.2: Define properties and methods of custom objects.
- 2.5.3: Create new object instances.
- 2.5.4: Create client-side arrays using custom objects.
- 2.5.5: Create functions and methods for manipulating client-side arrays.
- 2.5.6: Use the prototype property, concept of classes, constructors, iterators and generators

**Domain 3: Applied JavaScript****3.1 Modify HTML with JavaScript**

- 3.1.1: Identify steps and methods for changing HTML "on the fly," including the getElementById, getElementsByName, getElementsByTagName, and getElementsByClassName methods of the DOM.
- 3.1.2: Modify attributes in HTML using DOM elements.
- 3.1.3: Modify form object values.

**3.2: Use JavaScript to develop interactive forms**

- 3.2.1: Identify and use form controls, including HTML form elements.
- 3.2.2: Define the form object.
- 3.2.3: Refer to form objects, including input, text, textarea, radio, checkbox, select, button, password, hidden, file, and submit.
- 3.2.4: Use form objects, including radio, select, button, text, input, textarea, checkbox, password, hidden, file and submit.
- 3.2.5: Conduct form validation.
- 3.2.6: Identify common form security issues.

**3.3: Address JavaScript security issues involving browsers and cookies.**

- 3.3.1: Distinguish between the browser and the operating system in relation to the elements responsible for security.
- 3.3.2: Discuss browser security issues relevant to JavaScript. frame-to-frame URL changing, and document.write behavior differences among browsers.
- 3.3.3: Perform feature detection and determine browser compatibility.
- 3.3.4: Identify common issues and procedures for creating secure JavaScript code.
- 3.3.5: Define cross-site scripting and the associated security risks.
- 3.3.6: Define the functions and common uses of cookies.
- 3.3.7: Manipulate cookies effectively, including testing for presence of cookies, clearing cookies, enabling/disabling cookies in the browser, and deleting cookies from your hard drive. Use secure, HTTP-only cookies managed by the server.
- 3.3.8: Discuss ethics in collecting, storing, using and protecting user data.

## **Domain 4: Advanced JavaScript Development**

### **4.1: Implement JavaScript libraries and frameworks.**

- 4.1.1: Identify and evaluate the benefits and drawbacks of using predefined libraries and frameworks, such as React.js and Angular.js and Prototype.
- 4.1.2: Identify steps for using libraries, frameworks and available plug-ins, including optimization for faster JavaScript manipulation.
- 4.1.3: Identify steps for loading and referencing external scripts and pre-made external scripts.
- 4.1.4: Identify and evaluate the benefits and drawbacks of server-side JavaScript technologies.
- 4.1.5: Optimize page load time and user experience on various devices.

### **4.2: Use JavaScript and AJAX to create interactive web applications.**

- 4.2.1: Define synchronous and asynchronous, fundamental AJAX elements, and procedures.
- 4.2.2: Explain the Fetch API, Promises and callback functions.
- 4.2.3: Use the XMLHttpRequest object to retrieve data.
- 4.2.4: Describe typical AJAX-based requests.
- 4.2.5: Identify key server response issues related to AJAX-based requests.
- 4.2.6: Use JavaScript to send and receive data between a webpage and a server that communicates with databases.
- 4.2.7: Identify and compare XML and JSON.

### **4.3: Implementing Web APIs.**

- 4.3.1: Define Web API (Application Programming Interface) and benefits.
- 4.3.2: Identify and use Web APIs, including session storage, local storage, and GEO location.
- 4.3.3: Identify and use the Canvas API.