70-480
Programming in HTML5 with JavaScript and CSS3

Candidates for this exam are developers with at least one year of experience developing with HTML in an object-based, event-driven programming model, and programming essential business logic for a variety of application types, hardware, and software platforms using JavaScript.

Candidates should also have a thorough understanding of the following:

- Managing program flow and events
- Asynchronous programming
- Data validation and working with data collections including JQuery
- Handling errors and exceptions
- Arrays and collections
- Working with variables, operators, and expressions
- Working with prototypes and methods
- Decision and iteration statements

Objective Domain

Note: This document shows tracked changes that are effective as of December 14, 2017.

Implement and Manipulate Document Structures and Objects (20-25%)

Create the document structure by using HTML

Structure the UI by using semantic markup, including markup for search engines and screen readers, such as Section, Article, Nav, Header, Footer, and Aside; create a layout container in HTML

Write code that interacts with UI controls

Programmatically add and modify HTML elements; implement media controls; implement HTML5 canvas and SVG graphics

Apply styling to HTML elements programmatically

Change the location of an element; apply a transform; show and hide elements

Implement HTML5 APIs

Implement storage APIs, AppCache API, and Geolocation API

Establish the scope of objects and variables

Define the lifetime of variables; keep objects out of the global namespace; use the "this" keyword to reference an object that fired an event; scope variables locally and globally

Create and implement objects and methods

Implement native objects; create custom objects and custom properties for native objects using prototypes and functions; inherit from an object; implement native methods and create custom methods

Implement Program Flow (25-30%)

Implement program flow

Iterate across collections and array items; manage program decisions by using switch statements, if/then, and operators; evaluate expressions

Raise and handle an event

Handle common events exposed by DOM (OnBlur, OnFocus, OnClick); declare and handle bubbled events; handle an event by using an anonymous function

Implement exception handling

Set and respond to error codes; throw an exception; request for null checks; implement try-catch-finally blocks

Implement ~~a callback~~asynchronous programming

Receive messages from the HTML5 WebSocket API; use JQuery to make an AJAX call; wire up an event; implement a callback by using anonymous functions; handle the "this" pointer

Create a web worker process

Start and stop a web worker; pass data to a web worker; configure timeouts and intervals on the web worker; register an event listener for the web worker; limitations of a web worker

Access and Secure Data (25-30%)

Validate user input by using HTML5 elements

Choose the appropriate controls based on requirements; implement HTML input types and content attributes to collect user input

Validate user input by using JavaScript

Evaluate a regular expression to validate the input format; validate that you are getting the right kind of data type by using built-in functions; prevent code injection

Consume data

Consume JSON and XML data; retrieve data by using web services; load data or get data from other sources by using XMLHTTPRequest

Serialize, deserialize, and transmit data

Handle binary data; handle text data such as JSON and XML; implement the JQuery serialize method; handle web forms with Form.Submit; parse data; send data by using XMLHTTPRequest; sanitize input by using URI/form encoding

Use CSS3 in Applications (25-30%)

Style HTML text properties

Apply styles to text appearance; apply styles to a text font, including WOOF, @font-face, size, and understudy fonts; apply styles to text alignment, spacing, and indentation; apply styles to text hyphenation; apply styles for a text drop shadow

Style HTML box properties

Apply styles to alter appearance attributes, including size, borders, rounded corners, outline, padding, and margin; apply styles to alter graphic effects, including transparency, opacity, background image, gradients, shadow, and clipping; apply styles to establish and change an element's position

Create a flexible content layout

Implement a layout using a flexible box model; implement a multi-column layout; implement a layout using position floating and exclusions; implement a layout using grid alignment; implement a layout using regions, grouping, and nesting

Create an animated and adaptive UI

Animate objects by applying CSS transitions; apply 3-D and 2-D transformations; adjust UI based on media queries, including device adaptations for output formats, displays, and representations; hide or disable controls

Find elements by using CSS selectors and JQuery

Choose the correct selector to reference an element; define element, style, and attribute selectors; find elements by using pseudo-elements and pseudo-classes

Structure a CSS file by using CSS selectors.

Reference elements correctly; implement inheritance; override inheritance by using !important; style an element based on pseudo-elements and pseudo-classes