

**Comprehensive HTML5 Development**

**Course Number:** HTML-108  
**Duration:** 5 days

**Overview**

Accelebrate's Comprehensive HTML5 Development Training teaches attendees everything they need to know to build interactive web pages using the latest version of HTML. Students take their skills to the next level with powerful APIs to leverage device orientation, audio and video, drag and drop, speech recognition, geolocation, the onboard battery, vibrations, and much more. This course is current to version 5.2.

**Prerequisites**

No prior experience is presumed.

**Materials**

All students receive comprehensive courseware covering all topics in the course.

**Software Needed on Each Student PC**

* Web or text editor of your choice
* Web browsers - a recent version of one or more of the following:
  + Google Chrome
  + Mozilla Firefox
  + Microsoft Edge
  + Safari
* If space on a web server is available, students may also wish to bring their mobile devices so they can test how pages appear on these devices.

**Objectives**

* Learn how to write well-formed HTML using proper syntax
* Understand the basic structure of a web page
* Create lists
* Incorporate images
* Use HTML5 elements and templates
* Create HTML5 Forms
* Incorporate audio and video
* Use the Canvas element and API
* Obtain the charge status of a device’s battery
* Gain end users’ attention via their sense of touch
* Send the end user notifications that appear outside the web browser
* Tap into the device’s physical orientation
* Make portions of your web page editable by the end user
* Display portions of the web page so that they occupy the users’ entire screen
* Implement voice recognition and speech synthesis
* Leverage the new media elements (audio and video)
* Work with the new <form> elements and attributes
* Work with the WebStorage API
* Send and receive messages from web pages, including web pages from a different domain
* Associate HTML elements with data
* Make any object on the web page draggable
* Use the new features of XMLHttpRequest (XHR) level 2
* Open full duplex connections from client to server with the WebSocket specification
* Create single unidirectional channels between the server and the client
* Create a 2D drawing surface with the <canvas> element and "draw” on the canvas with JavaScript.
* Use geolocation information in your applications
* Use web workers to boost the performance of your web applications

**Outline**

* Introduction
  + HTML5 summary
  + Abbreviated syntax guide to writing HTML5
  + Best practices
* Syntax
  + HTML Elements: how to write HTML tags
  + Syntax rules
  + Page structure
  + Basic structure of a web page
  + The HTML5 syntax summary
* Structure
  + Introduction to the structure of HTML documents
  + The HTML Content Models
  + The Content Models
  + Deprecated elements
  + Elements with new meanings
  + Deprecated attributes
  + Structural and semantic elements
  + Overview of lists
  + Overview of links
  + Images
  + The remaining HTML5 elements
  + Using HTML5 templates
  + Using the HTML5 boilerplate and mobile boilerplate
* Elements
  + What are the new elements?
  + How do I use the new elements?
  + HTML5 new attributes
* Forms
  + Introduction form elements and attributes
  + New form input types
  + New form attributes
  + New form elements
  + When do I use the new elements?
* Video and Audio
  + Introduction to HTML5 media elements
  + Audio and Video
* HTML5 Canvas
  + What are the new elements
  + Why use the Canvas element and API?
* BatteryStatus API
  + Why use the BatteryStatus API?
  + How does the BatteryStatus API work?
  + The BatteryStatus API
    - BatteryStatus API methods
    - BatteryManager properties
    - BatteryManager events
  + Browser support
* Vibration API
  + Why use the Vibration API?
  + How does the Vibration API work?
  + Vibration methods
* Notifications API
  + Why use the Notifications API?
  + How does the Notifications API work?
  + Notifications API
    - Notification object attributes
    - Notification events
    - Notification API methods
* DeviceOrientation Event API
  + Why use the DeviceOrientation API?
  + DeviceOrientation Event API
    - DeviceOrientation events
    - How does the DeviceOrientation Event API work?
    - DeviceOrientation
    - DeviceMotion
    - DeviceOrientation Event API
* Fullscreen API
  + Why use the Fullscreen API?
  + The Fullscreen API
    - Fullscreen API methods
    - Fullscreen API properties
    - Fullscreen API events
  + Notes on the fullscreen view
* Speech APIs
  + Speech Recognition and Speech Synthesis
  + Speech Synthesis
    - SpeechUtterance object
    - SpeechUtterance object properties
    - SpeechUtterance object events
  + Speech Recognition
* HTML5 Media elements and API
  + Audio and Video
    - The media elements
    - What are the new elements used for?
    - Browser support for the new media elements
    - <audio> element attributes
    - <video> element attributes
    - Browser support <video>
    - Browser support Ogg/Theora video format\*
    - Browser support WebM/VP8 video format\*
    - Browser support MPEG-4/H.264video format\*
    - Browser support audio codec
    - When do I use these new elements?
  + Video and Audio APIs
    - Video/audio methods
    - Video/audio object properties
    - Video/audio object events
* Forms/Constraint Validation API
  + New form input types
  + What are the new input types?
    - New attributes for input elements
    - New form elements
  + When do I use the new elements?
    - The placeholder attribute
    - The autocomplete attribute
    - The autofocus attribute
    - The list attribute and the datalist element
    - The spellcheck attribute
  + HTML5 form validation
    - Preventing default validation from occurring
  + Constraint Validation API
    - Form properties
    - Form control methods
    - Form controls validity and validityState
* WebStorage API
  + Browser support for WebStorage API
  + WebStorage API
    - The Storage Object
  + When
  + Cookies vs. session storage vs. local storage
* Web Messaging API
  + Web messaging’s postMessage() method
    - Establishing trust between domains
    - The origin
  + What is cross-origin resource sharing?
    - CORS support and the server
  + Ports and channel messaging
    - MessagePort methods
    - The Message event object properties
* Dataset API
  + data-\* syntax rules
  + The dataset API
* Drag and Drop API
  + What is drag and drop?
    - What is drag and drop used for?
    - Browser support for drag and drop
    - Understanding drag and drop
    - The draggable attribute
    - Drag and drop with a data “payload”
  + Drag events
  + Drop events
* XMLHttpRequest Level 2 API
  + Objectives
  + The XMLHTTPRequest object
  + XHR API
    - What is XHR used for?
    - Benefits of XHR level 2
    - The XHR object: readyState property
  + The XHR object
    - The XHR object: Events
    - The XHR Object: Methods
    - The XHR object: Properties
* Web Sockets API
  + HTTP
    - WebSocket properties
    - WebSocket event handlers
    - WebSocket methods
  + Using a WebSocket server with the WebSocket API
* Server-sent Events API
  + The server messages
  + Creating an EventSource
    - The EventSource properties
    - The EventSource events
    - The advantages of server-sent events
    - The disadvantages of server-sent events
    - Server-sent events summary
* Canvas API
  + What are the new elements?
  + Why use the Canvas API?
  + Canvas quick summary
  + Uses for the new Canvas API
  + Canvas Demos
* Geolocation API
  + What is the Geolocation API used for?
  + Browser support for Geolocation
  + The Geolocation API
    - Geolocation object methods
    - Position object properties
    - Position interface attributes
* Web Workers API
  + Features available to Web Workers
  + Why use Web Workers?
  + When should Web Workers be used?
  + How do Web Workers work?
  + The Web Worker API
    - The abstractWorkerk interface properties
    - The WorkerGlobalScope object
    - The WorkerGlobalScope object properties
    - The WorkerGlobalScope object methods
    - The WorkerGlobalScope events
  + Browser support for Web Workers
  + Types of Web Workers
    - Dedicated Worker
    - Shared Worker
  + Creating Shared Workers
* Conclusion