www.logicalimagination.training
800.657.1494

HTML5 Fundamentals

Course #: HT-102 **Duration:** 2 days

Prerequisites

Students should be familiar with HTML.

Details

This course introduces students familiar with writing HTML to the elements and attributes of HTML5 before moving into the web application scripting APIs. The course begins by introducing where HTML5 came from and what it is. It discusses how to use the new elements and attributes as well as how to detect if the browser supports them. The course then moves into new form input types and attributes before covering the audio/video tags. A deeper look into the Canvas element follows, and then the Geolocation API, new local data storage options that could make cookies obsolete, and finally new communications APIs that will greatly enhance online web applications.

Software Needed

- Any web browser (Chrome, Firefox, etc.)
- Any text editor (Notepad, Notepad++, Visual Studio, etc.)

Outline

HTML5 Fundamentals

- **HTML5 Overview**
  - HTML5 History/Timeline
  - HTML5 Design Principles
  - What Is (and Isn't) HTML5
  - HTML5 Review
  - Doctype
  - Root Element
  - <head> element
  - Syntax
- **HTML5 Elements**
  - Content Models
  - New Structural Elements
  - HTML4/HTML5 Comparison
  - Other New Elements
  - Redefined Elements
  - Obsolete Elements
  - HTML5 Outlines
  - When Can I Use It?
  - Feature Detection: Techniques
  - Feature Detection: Examples
  - Feature Detection: Modernizr
  - CSS Styling
  - Validating
  - Accessibility (WAI-ARIA)
- **Forms**

- HTML5 Forms Overview
- New Input Types: Contact Info
- New Input Types: Native Date Picking
- Opera's Rendering of Date Input Types
- New Input Types: Number and Range
- New Input Types: Search and Color
- New Attributes
- New Attributes: Placeholder and Required
- New Attributes: Autocomplete and Autofocus
- New Attributes: Min, Max, Step, and Pattern
- Detecting Support
- Accessibility: WAI-ARIA
- Styling Form Elements
- Avoiding Validation
- The Constraint Validation API
- Custom Validation Example

- **HTML5 Media**
  - Audio and Video Overview
  - Using the Media Elements
  - Attributes
  - Formats
  - Serving Device-Specific Files
  - Accessibility
  - Backward Compatibility
  - Media API

- **Canvas**
  - Canvas Overview
  - The Canvas Element
  - The 2D Context
  - The Coordinate System
  - Rectangles
  - Paths
  - Paths: Drawing Methods
  - Curves and Arcs
  - Colors and Styles
  - Gradients
  - Patterns
  - Transformations: Scale, Translate
  - Transformations: Rotate
  - Drawing States
  - Image Drawing
  - Animations
  - Responding to User Events: Keyboard
  - Responding to User Events: Mouse
  - Compositing
  - Text
  - Pixel Manipulation
  - toDataURL
  - Accessibility
  - Canvas and Internet Explorer

- **Geolocation**
  - Geolocation Overview
  - Privacy Concerns
  - API Methods
  - API Attributes
  - Using the Geolocation API: Success Handlers
  - Using the Geolocation API: Error Handlers
  - Using the Geolocation API: The Third Argument
  - watchPosition() and clearWatch()
  - Fallback Support: Geo.js

- **Local Data Storage**
  - Local Data Storage Overview
  - Web Storage Overview

- Web Storage API
- Data Types
- JSON (JavaScript Object Notation)
- Accessing Storage
- The Storage Event
- Database APIs
- Web SQL Databases
- Web Databases: Opening the DB, Creating Tables, and Inserting Data
- Web Databases: Selecting, Using, and Deleting Data
- IndexedDB

- **Web Messaging**
  - Cross-Document Messaging
  - Using the postMessage API: Step 1
  - Using the postMessage API: Step 2
  - Using the postMessage API: Step 3
  - Server-Sent Events
  - EventSource API
  - Using the EventSource API: Client-side
  - Event Stream Format: Server-side
  - Simple Implementation
  - XMLHttpRequest Level 2
  - Cross-Origin Requests
  - Making a Cross-Origin Request
  - Progress Events
  - Using Progress Events

- **Web Workers**
  - Web Workers Overview
  - What Can You Do with a Worker?
  - Message Passing
  - Handling Errors
  - Stopping Workers
  - Loading and Executing External Scripts
  - Workers within Workers
  - Subworkers: An Example
  - Security Notes
  - Shared Workers: The Parent Page
  - Shared Workers: Within the Worker

- **Microdata**
  - What Is It?
  - Why Use It?
  - Data Model
  - Typed Items
  - Using Microdata