This 2-day course introduces students familiar with writing HTML to the new 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. Appendices covering Microdata, offline applications, and WebSockets are included for further study.

Course Objectives:

- Describe the history and design principles of HTML5.
- Implement the new structural elements of HTML5 to mark up an HTML5 page.
- Use HTML5 form input types and attributes to reduce the need for scripting in your pages.
- Embed media in a web page with the new video and audio tags.
- Draw 2D graphics on the canvas element.
- Determine your user's location with the Geolocation API.
- Use client-side storage technologies to persist and retrieve content.
- Communicate between the browser and the server using Server-Sent Events and XMLHttpRequest Level 2.
- Run multiple threads simultaneously within your browser with Web Workers.

Audience: Experienced web developers who will be designing, creating, and deploying HTML5 web applications.

Prerequisites: Experience in HTML 4 and JavaScript is required. CSS knowledge is helpful.

Number of Days: 2 days

1 HTML5 Overview

HTML5 History/Timeline
HTML5 Design Principles
What is (and isn't) HTML5
HTML Review
Doctype
Root Element
<head> Element
Syntax

2 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)

3 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

© Batky-Howell, LLC

New Attributes
Detecting Support
Accessibility (WAI-ARIA)
Styling Form Elements
Avoiding Validation
The Constraint Validation API
Custom Validation Example

4 HTML5 Media
Audio and Video Overview
Using the Media Elements
Attributes
Formats
Serving Device-Specific Files
Accessibility
Backwards Compatibility
Media API
Rolling Custom Controls

5 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 Manipulations
toDataURL()
Accessibility
Canvas and Internet Explorer

6 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

7 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

8 Web Messaging
Cross Document Messaging
Using the postMessage API
Server-Sent Events Overview
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

9 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
10 **Appendix A – Microdata**
What Is It?
Why Use It?
Data Model
Typed Items
Using Microdata
11 **Appendix B – Offline**
Offline Web Applications Overview
Adding and Service a Cache Manifest
Cache Manifest Sections
Updating the Cache
The ApplicationCache Object: Methods and Status Values
The ApplicationCache Object: Events
Programmatically Updating the Cache
Detecting Connectivity
12 **Appendix C – Web Sockets**
Web Sockets Overview
Using the WebSockets API
WebSockets on the Server
Fallback Solutions