Advanced HTML5 and CSS3 Specialist:
CIW Web and Mobile Design Series
Student Guide
CCL02-CDHTCS-CK-1405 • version 1.0 • rd042214
Advanced HTML5 and CSS3 Specialist
Student Guide
# Table of Contents

<table>
<thead>
<tr>
<th>Lesson 1: HTML5 Essentials</th>
<th>1-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Assessment Questions</td>
<td>1-2</td>
</tr>
<tr>
<td>Introduction to HTML5 and CSS3</td>
<td>1-3</td>
</tr>
<tr>
<td>Migration to Mobile Devices</td>
<td>1-3</td>
</tr>
<tr>
<td>The Web Development Trifecta</td>
<td>1-5</td>
</tr>
<tr>
<td>The Evolution of HTML5</td>
<td>1-6</td>
</tr>
<tr>
<td>HTML5 Structure Elements</td>
<td>1-9</td>
</tr>
<tr>
<td>Validating HTML5 Code</td>
<td>1-10</td>
</tr>
<tr>
<td>The &lt;video&gt; Element</td>
<td>1-15</td>
</tr>
<tr>
<td>The &lt;audio&gt; Element</td>
<td>1-19</td>
</tr>
<tr>
<td>HTML5 APIs</td>
<td>1-21</td>
</tr>
<tr>
<td>Case Study</td>
<td>1-24</td>
</tr>
<tr>
<td>Lesson 1 Review</td>
<td>1-26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lesson 2: Using Cascading Style Sheets (CSS) Technology</th>
<th>2-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Assessment Questions</td>
<td>2-2</td>
</tr>
<tr>
<td>Cascading Style Sheets (CSS)</td>
<td>2-3</td>
</tr>
<tr>
<td>Style Guides</td>
<td>2-3</td>
</tr>
<tr>
<td>CSS and HTML</td>
<td>2-5</td>
</tr>
<tr>
<td>CSS Terms and Syntax</td>
<td>2-6</td>
</tr>
<tr>
<td>Applying CSS Styles</td>
<td>2-9</td>
</tr>
<tr>
<td>Page Layout with CSS</td>
<td>2-14</td>
</tr>
<tr>
<td>CSS Positioning Schemes</td>
<td>2-21</td>
</tr>
<tr>
<td>The CSS Box Model</td>
<td>2-22</td>
</tr>
<tr>
<td>Case Study</td>
<td>2-25</td>
</tr>
<tr>
<td>Lesson 2 Review</td>
<td>2-27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lesson 3: Introduction to CSS Version 3 (CSS3)</th>
<th>3-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Assessment Questions</td>
<td>3-2</td>
</tr>
<tr>
<td>Introduction to CSS3</td>
<td>3-3</td>
</tr>
<tr>
<td>CSS3 Selectors and Properties</td>
<td>3-3</td>
</tr>
<tr>
<td>CSS3 Background Properties</td>
<td>3-8</td>
</tr>
<tr>
<td>CSS3 Border Properties</td>
<td>3-14</td>
</tr>
<tr>
<td>CSS3 Font Properties</td>
<td>3-18</td>
</tr>
<tr>
<td>CSS3 Text Effects</td>
<td>3-20</td>
</tr>
<tr>
<td>Case Study</td>
<td>3-24</td>
</tr>
<tr>
<td>Lesson 3 Review</td>
<td>3-26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lesson 4: Using Advanced CSS3 Techniques</th>
<th>4-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Assessment Questions</td>
<td>4-2</td>
</tr>
<tr>
<td>Introduction to Advanced CSS3 Techniques</td>
<td>4-3</td>
</tr>
<tr>
<td>CSS3 2D and 3D Transformations</td>
<td>4-3</td>
</tr>
<tr>
<td>CSS3 Transitions</td>
<td>4-8</td>
</tr>
<tr>
<td>CSS3 Animations</td>
<td>4-11</td>
</tr>
<tr>
<td>CSS3 User Interfaces</td>
<td>4-15</td>
</tr>
<tr>
<td>Creating Menus and Buttons with CSS3</td>
<td>4-17</td>
</tr>
<tr>
<td>Case Study</td>
<td>4-21</td>
</tr>
<tr>
<td>Lesson 4 Review</td>
<td>4-23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lesson 5: Introduction to JavaScript</th>
<th>5-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Assessment Questions</td>
<td>5-2</td>
</tr>
<tr>
<td>HTML5 and JavaScript</td>
<td>5-3</td>
</tr>
<tr>
<td>Introduction to Scripting</td>
<td>5-4</td>
</tr>
<tr>
<td>Lesson 6: JavaScript Events, Functions and Methods</td>
<td>6-1</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Pre-Assessment Questions</td>
<td>6-2</td>
</tr>
<tr>
<td>User Events and JavaScript Event Handlers</td>
<td>6-3</td>
</tr>
<tr>
<td>Introduction to JavaScript Functions</td>
<td>6-4</td>
</tr>
<tr>
<td>Defining a Function</td>
<td>6-5</td>
</tr>
<tr>
<td>Calling a Function</td>
<td>6-6</td>
</tr>
<tr>
<td>Methods as Functions</td>
<td>6-19</td>
</tr>
<tr>
<td>Errors In JavaScript</td>
<td>6-22</td>
</tr>
<tr>
<td>Case Study</td>
<td>6-26</td>
</tr>
<tr>
<td>Lesson 6 Review</td>
<td>6-27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lesson 7: Using HTML5 APIs</th>
<th>7-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Assessment Questions</td>
<td>7-2</td>
</tr>
<tr>
<td>Introduction to HTML5 APIs</td>
<td>7-3</td>
</tr>
<tr>
<td>Document Object Model (DOM)</td>
<td>7-3</td>
</tr>
<tr>
<td>Common HTML5 APIs</td>
<td>7-4</td>
</tr>
<tr>
<td>The Canvas API</td>
<td>7-4</td>
</tr>
<tr>
<td>The Offline AppCache API</td>
<td>7-10</td>
</tr>
<tr>
<td>The Geolocation API</td>
<td>7-13</td>
</tr>
<tr>
<td>The Drag-and-Drop API</td>
<td>7-16</td>
</tr>
<tr>
<td>The File API</td>
<td>7-20</td>
</tr>
<tr>
<td>The History API</td>
<td>7-22</td>
</tr>
<tr>
<td>Retrieving Data with XMLHttpRequest</td>
<td>7-24</td>
</tr>
<tr>
<td>Manipulating Data with jQuery</td>
<td>7-24</td>
</tr>
<tr>
<td>Case Study</td>
<td>7-28</td>
</tr>
<tr>
<td>Lesson 7 Review</td>
<td>7-30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lesson 8: Developing HTML5 Forms</th>
<th>8-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Assessment Questions</td>
<td>8-2</td>
</tr>
<tr>
<td>Introduction to HTML5 Forms</td>
<td>8-3</td>
</tr>
<tr>
<td>Cross-Browser Compatible HTML5 Forms</td>
<td>8-3</td>
</tr>
<tr>
<td>HTML5 Form Input Types</td>
<td>8-4</td>
</tr>
<tr>
<td>New Form Elements in HTML5</td>
<td>8-9</td>
</tr>
<tr>
<td>HTML5 Global Attributes for Form Elements</td>
<td>8-13</td>
</tr>
<tr>
<td>Case Study</td>
<td>8-21</td>
</tr>
<tr>
<td>Lesson 8 Review</td>
<td>8-23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lesson 9: Completing, Submitting and Validating User Input Forms</th>
<th>9-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Assessment Questions</td>
<td>9-2</td>
</tr>
<tr>
<td>Improving Forms with HTML5</td>
<td>9-3</td>
</tr>
<tr>
<td>HTML5 Attributes for the &lt;form&gt; Element</td>
<td>9-3</td>
</tr>
<tr>
<td>HTML5 Attributes for the &lt;input&gt; Element</td>
<td>9-5</td>
</tr>
<tr>
<td>Submitting Forms with the &lt;button&gt; Element</td>
<td>9-8</td>
</tr>
<tr>
<td>Concepts and Techniques for Validating User Input</td>
<td>9-11</td>
</tr>
<tr>
<td>Validating User Input with HTML5 Attributes</td>
<td>9-13</td>
</tr>
<tr>
<td>Validating User Input with JavaScript</td>
<td>9-20</td>
</tr>
<tr>
<td>Case Study</td>
<td>9-22</td>
</tr>
<tr>
<td>Lesson 9 Review</td>
<td>9-24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lesson 10: Designing for Mobile Devices</th>
<th>10-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Assessment Questions</td>
<td>10-2</td>
</tr>
<tr>
<td>Introduction to Mobile Design</td>
<td>10-3</td>
</tr>
<tr>
<td>Mobile Web Sites vs. Mobile Apps</td>
<td>10-3</td>
</tr>
</tbody>
</table>
Course Description

*Advanced HTML5 And CSS3 Specialist* is the first course in the CIW Web And Mobile Design series. This course teaches the essentials of HTML5, Cascading Style Sheets (CSS) and JavaScript. These technologies, sometimes known as the "HTML5 Family" or the "Web development trifecta," can be used together to create Web pages that easily adapt to display on smartphones, tablets, gaming devices and smart TVs, as well as to traditional desktop computers.

This course builds upon your manual coding skills to teach you how to develop Web documents using new elements, attributes and selectors introduced in HTML5 (Hypertext Markup Language version 5) and CSS3 (Cascading Style Sheets version 3). You will create Web pages using the HTML5 structure elements, embed video and audio, and develop cross-browser user-input forms. You will use CSS3 to position and format content, and to create effects such as transformations, transitions and animation.

You will also learn basic JavaScript coding, and use HTML5 APIs (application programming interfaces) to extend the functionality of Web pages with modern features such as geolocation, drag-and-drop, canvas and offline Web applications. In addition, you will learn techniques for code validation and testing, form creation, inline form field validation, and mobile design for browsers and apps, including Responsive Web Design (RWD).

All CIW courses offer Case Studies about real-world skills applications and job-related topics. Guided, step-by-step labs provide opportunities to practice new skills. You can challenge yourself and review your skills after each lesson in the Lesson Summary and Lesson Review sections. Additional skill reinforcement is provided in Activities, Optional Labs, Lesson Quizzes and a Course Assessment that are available from your instructor (for ILT training) or from CIW Online (for self-study).

This coursebook includes supplemental online material containing the lab files used in the course. To practice the skills presented in the course or to perform any labs that were not completed, refer to the Classroom/System Setup section for information about system requirements and using the lab files.

The CIW *Advanced HTML5 And CSS3 Specialist* course prepares you to take the high-stakes CIW Advanced HTML5 And CSS3 Specialist certification exam.

Series

*Advanced HTML5 And CSS3 Specialist* is the first course in the CIW Web And Mobile Design series of courses, which consists of seven courses:

- *Advanced HTML5 and CSS3 Specialist*
- User Interface Design Specialist
- Multimedia Specialist
- Social Media Strategy Specialist
- Mobile Applications Specialist
- E-Commerce Specialist
- Data Analyst

Prerequisites

The *Advanced HTML5 And CSS3 Specialist* courseware teaches intermediate and advanced HTML5 and CSS3 coding skills, as well as introductory JavaScript. Before taking this course, you should complete the *CIW Site Development Associate* course from the CIW Web Foundations series, or have equivalent basic HTML and CSS coding knowledge.
Certification

The CIW Advanced HTML5 And CSS3 Specialist course prepares you to take the high-stakes CIW Advanced HTML5 And CSS3 Specialist certification exam (1D0-620). Those who pass the CIW Advanced HTML5 And CSS3 Specialist exam earn the CIW Advanced HTML5 And CSS3 Specialist certification, which is recognized throughout the industry as validating essential HTML5 and CSS3 development skills for the workplace.

Candidates who pass all seven high-stakes exams in the CIW Web And Mobile Design series will also be awarded CIW Web And Mobile Design Professional certification status.

For information about taking this or any other CIW exam, visit www.CIWcertified.com.

Target audience

Individuals preparing to enter or continue in the workforce fields of Web site development and design can benefit from the CIW Advanced HTML5 And CSS3 Specialist course and/or certification:

- High school students
- College students
- Technical/trade school students

Professionals working in Web site development and design fields for any industry can benefit from the CIW Advanced HTML5 And CSS3 Specialist course and/or certification:

- Web and graphic designers
- Web marketing professionals
- Mobile application developers
- Creative directors
- Advertising professionals
- Entrepreneurs

Student Courseware

This coursebook was developed for instructor-led and/or self-study training. Along with comprehensive instructional text and objectives checklists, this coursebook provides easy-to-follow hands-on labs and a glossary of course-specific terms. It also provides Internet addresses needed to complete some labs, although due to the constantly changing nature of the Internet, some addresses may no longer be valid.

The student coursebook is organized in the following manner:
When you return to your home or office, you will find this coursebook to be a valuable resource for reviewing labs and applying the skills you have learned. Each lesson concludes with questions that review the material. Lesson review questions are provided as a study resource only and in no way guarantee a passing score on the high-stakes CIW Advanced HTML5 And CSS3 Specialist certification exam.

Coursebook versions

The CIW *Advanced HTML5 And CSS3 Specialist* courseware is designed for various learning environments: academic, learning center/corporate and self-study. Coursebooks are available in both instructor and student versions. Student versions are available for the academic and learning center/corporate environments, as well as self-study. Check your book to verify which version you have.

**Instructor (Academic, Learning Center/Corporate)**

- Example syllabi for 10-week and 16-week instruction periods are included with the instructor supplemental files available on CIW Online. Learning centers can teach this series at an accelerated pace; consult the implementation table that can be found on CIW Online.

- The instructor version of this book includes Instructor Notes in the margin, which provide additional tips and commentary for the instructor to supplement course narrative. Margin callouts also direct instructors to material that relates directly to specified CIW Advanced HTML5 And CSS3 Specialist exam objectives.

- The instructor book and supplemental files contain all answers to Optional Labs, Lesson Quizzes and the Course Assessment.
• The supplemental files also include handout versions of all Optional Labs, Lesson Quizzes and the Course Assessment, which the instructor can print and assign during class or as homework.

• The supplemental files also include an appendix listing the CIW Advanced HTML5 And CSS3 Specialist certification exam objectives and locations of corresponding material in the coursebook.

Lesson Quizzes and Course Assessments are provided as study and course-grading resources only; success on these materials in no way guarantees a passing score on the CIW Advanced HTML5 And CSS3 Specialist certification exam.

**Student (Academic, Learning Center/Corporate, Self-Study)**

• The student book and supplemental files include Pre-Assessment and Lesson Review questions for each lesson. However, the student book does not provide answers to these questions.

• The student book also does not include any Activities, Optional Labs, Quizzes or the Course Assessment. Students can obtain these elements and answers from the instructor (for ILT training) or from CIW Online (for learning center/corporate or self-study training).

• The student supplemental materials include appendixes and files used to perform many of the labs in the coursebook.

• The supplemental files also include an appendix listing the CIW Advanced HTML5 And CSS3 Specialist certification exam objectives and locations of corresponding material in the coursebook.

Lesson Quizzes and Course Assessments are provided as study and course-grading resources only; success on these materials in no way guarantees a passing score on the CIW Advanced HTML5 And CSS3 Specialist certification exam.

**Online resources**

You can visit CIW Online at [http://education.certification-partners.com/ciw/](http://education.certification-partners.com/ciw/) to access supplemental course materials and to get help in preparing for the CIW Web And Mobile Design series certification exams. CIW Online provides a variety of online tools you can use to supplement the Official CIW Courseware.

**CIW courseware supplemental files**

This coursebook includes supplemental material that can be accessed from CIW Online. Online materials are provided for both instructors and students, and include some elements required to complete the coursework and other optional elements that are provided for your interest or further study.

Student materials include:

• Lab files used to complete the course labs

• Answers to student exercises and quizzes (Learning Center/Corporate and Self-Study only)

• Appendixes with related information (including the CIW Advanced HTML5 And CSS3 Specialist Objectives and Locations Appendix).

Instructor materials include:

• Course syllabi and implementation tables

• Answers to students exercises and quizzes

• Appendixes with related information (including the CIW Advanced HTML5 And CSS3 Specialist Objectives and Locations Appendix).

See the CIW Supplemental Files section under Classroom/System Setup for information about accessing these files.
**CIW Movies**
The CIW Advanced HTML5 And CSS3 Specialist course offers movie clips that provide supplementary instruction in a multimedia format, and enhance the coursebook narrative and labs. However, movie content does not comprehensively address CIW Advanced HTML5 And CSS3 Specialist certification exam objectives and is not intended to replace coursebook content.

Instructors in a classroom environment are strongly encouraged to present movies to the entire class using a computer screen projector. Group presentations enable instructors to present and discuss movie content when appropriate. Controlling the presentation of movies also minimizes distractions from course material and essential lecture or lab time.

Students are strongly encouraged to watch the movie clips on their own if they are unable to view them in a class. Each student is provided access to CIW Online to view the movies.

**CIW Online Exercises**
These interactive activities are instructional supplements to the official print and online books, designed to offer a blended-learning approach. Mapped directly to the Official CIW Courseware, the CIW Online Exercises enable you to review important concepts from the CIW Advanced HTML5 And CSS3 Specialist course and measure your proficiency on content relevant to the CIW Advanced HTML5 And CSS3 Specialist certification exams. CIW Online Exercises challenge you with a wide range of activities that all provide immediate feedback, including:

- Glossary flashcards.
- Matching exercises.
- Fill-in-the-blank exercises.
- Crossword puzzles.
- True/false questions.

**CIW Course Mastery**
CIW Course Mastery questions are designed to assess your knowledge of the concepts, skills and best practices of Web technology taught in the Official CIW Courseware. The CIW Course Mastery questions assess lesson knowledge, reinforce classroom learning and enhance instruction. This online review program contains multiple-choice questions that cover CIW Advanced HTML5 And CSS3 Specialist courseware content lesson by lesson. The Course Mastery program is based on a unique method that maximizes knowledge retention.

**CIW Certification Practice Exams**
After you have mastered the CIW Advanced HTML5 And CSS3 Specialist course material, you are ready to prepare for the high-stakes CIW Advanced HTML5 And CSS3 Specialist certification exam. The online CIW Certification Practice Exams program helps you build confidence with your knowledge of the CIW exam objectives. This program provides you with:

- Timed practice exams that simulate the high-stakes testing environment and help predict actual performance on CIW certification exams.
- A feedback review mode that allows you to check answers while taking the practice exam and gain valuable feedback that relates each question to a CIW exam objective and a lesson in the Official CIW Courseware.
- Exam results that report on your mastery of each CIW exam objective.
- Personalized performance reports and study plans to track individual progress and view overall class trends.
Course Objectives

After completing this course, you will be able to:

- Implement HTML5 coding solutions using HTML5 elements, attributes and values.
- Apply CSS3 functionality to Web documents using various properties, selectors and techniques.
- Integrate basic JavaScript coding into a Web page to create HTML5 APIs.
- Consider HTML5 and CSS3 techniques to apply to both traditional and mobile delivery platforms, including mobile apps.
- Transform traditional Web pages into mobile Web pages.

Classroom/System Setup

If you are taking this course in an instructor-led classroom, then your instructor has probably set up the classroom computers based on the system requirements listed in the following sections. Most software configurations on your computer are identical to those on your instructor's computer. However, your instructor may use additional software to demonstrate network interaction or related technologies.

To implement this course as a self-study student, you will need to set up your computer based on the hardware, software and connectivity requirements listed in the following sections. However, you may want to use additional software to further explore network interaction or related technologies.

System Requirements

This section lists the hardware, software and connectivity requirements to implement this course.

Hardware

Each student and instructor needs access to an individual computer workstation. The following table summarizes the hardware requirements for all courses in the CIW program.

*Note: The CIW hardware requirements are similar to the minimum system requirements for Microsoft Windows 8.1 implementation.*

<table>
<thead>
<tr>
<th>Hardware Specifications</th>
<th>Minimum Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>1 GHz 32-bit (x86) or 64-bit (x64) processor</td>
</tr>
<tr>
<td>Hard disk</td>
<td>16 GB available hard disk space (32-bit) or 20 GB available hard disk space (64-bit) for Windows 8 installation. 16 GB of additional space must be available for course applications and files.</td>
</tr>
<tr>
<td>RAM</td>
<td>1 GB RAM (32-bit) or 2 GB RAM (64-bit)</td>
</tr>
<tr>
<td>Network interface card (NIC)</td>
<td>Wireless, 10/100 Ethernet, or Gigabit Ethernet</td>
</tr>
<tr>
<td>Sound card/speakers</td>
<td>Required</td>
</tr>
<tr>
<td>Video adapter</td>
<td>Microsoft DirectX 9 graphics device with WDDM driver</td>
</tr>
<tr>
<td>Network connectivity</td>
<td>Enough wireless nodes, hubs or switches to allow classroom computers to communicate and access the Internet.</td>
</tr>
<tr>
<td>Monitor</td>
<td>1024 x 768 screen resolution using a VGA, DVI or HDMI connector</td>
</tr>
<tr>
<td>Web camera (Webcam)</td>
<td>Any type of Web camera. Some monitors include an internal Webcam. USB Web cams are a good choice.</td>
</tr>
</tbody>
</table>
Software

The recommended software configurations for computers used to complete the labs in this book are as follows.

To be installed before beginning the course:

- Microsoft Windows 8.1 or higher (typical installation)
- Windows Internet Explorer 11 or higher (typical installation)
- Firefox 27 or higher (typical installation)
- Google Chrome 34 or higher (typical installation)

Connectivity

Internet connectivity is required for this course. You will experience optimal performance with a dedicated Internet connection (e.g., a cable/DSL modem or a T1 line).

CIW supplemental files

Each coursebook includes supplemental materials that are referenced and used throughout the course. These supplemental materials are provided online at http://education.certification-partners.com/ciw/.

You will need to create a directory for all supplemental materials for the course. The default location is C:\CIW\[Course_Title]. To view or download the materials, go to CIW Online, click the link for each file and save to this directory. You can then create a shortcut to this directory on your Desktop. As you conduct the course labs, you can use this shortcut to quickly access your lab files.
Conventions and Graphics Used in This Book

The following conventions are used in this coursebook.

**Terms**
Technology terms defined in the margins are indicated in **bold type** the first time they appear in the text. However, not every word in bold type is a term requiring definition.

**Lab Text**
Text that you enter during a lab appears in *italic bold type*. Names of components that you access or change in a lab appear in **bold type**.

**Notations**
Notations or comments regarding screenshots, labs or other text are indicated in *italic type*.

**Program Code or Commands**
Text used in blocks of program code or operating system commands appears in the *Lucida Sans Typewriter* font. Code keywords used in sentences appear in *italic type*.

The following graphics are used in this coursebook.

- **Tech Notes** point out exceptions or special circumstances that you may find when working with a particular procedure. Tech Notes that occur within a lab are displayed without the graphic.

- **Tech Tips** offer special-interest information about the current subject.

- **Warnings** alert you about cautions to observe or actions to avoid.

This graphic signals the start of a lab or other hands-on activity.

Each lesson summary includes an *Application Project*. This project is designed to provoke interest and apply the skills taught in the lesson to your daily activities.

Each lesson concludes with a summary of the skills and objectives taught in that lesson. You can use the *Skills Review* checklist to evaluate what you have learned.

This graphic indicates a line of code that is completed on the following line.
Lesson 1: HTML5 Essentials

Objectives

By the end of this lesson, you will be able to:

- 1.1: Consider HTML5 development skills you can apply to both traditional and non-traditional delivery platforms, including mobile apps.
- 1.2: Identify the Web development trifecta and explain its significance.
- 1.3: Explain the evolution of HTML and its relevance to modern design techniques.
- 1.4: Create and deploy HTML5 structure tags.
- 1.5: Perform HTML5 code validation, which includes explaining the practical value of code validation and listing the steps to validate HTML5 code.
- 1.6: Explain the importance of universal markup creation.
- 1.7: Apply the HTML5 <video> element appropriately.
- 1.8: Use the HTML5 <audio> element.
- 1.9: Define common elements of HTML5 application programming interfaces (APIs).
Pre-Assessment Questions

1. The terms "HTML5 family" and "Web development trifecta" are sometimes used to refer to which combination of Web technologies?
   a. HTML5, CSS3 and JavaScript
   b. HTML5, XML and CSS3
   c. HTML5, JavaScript and Flash
   d. HTML5, Flash and XML

2. Which HTML5 structure element was designed to enclose Web site content such as company services, blogs, images and videos?
   a. <header>
   b. <footer>
   c. <article>
   d. <aside>

3. Which element introduced in HTML5 saves mobile device resources by avoiding the use of plug-ins to play MP4 files?
Introduction to HTML5 and CSS3

The skills of Web page and app creation have become vital to many careers. If you work in an office or with a computer, then you are likely to need skills with Web-based app technologies for various job-related tasks. If you want to work as a Web developer, app developer or site designer, with responsibility for the Web resources of an entire company or organization, then you need to master the most current standard languages for Web authoring: Hypertext Markup Language version 5 (HTML5) and Cascading Style Sheets version 3 (CSS3).

Whatever job role you pursue, this course will teach you how to create Web pages and apps using the latest HTML5 and CSS3 techniques. You will use a text-editor and write code manually for all development in this course. Graphical user interface (GUI) software, such as Adobe Dreamweaver, is not used here, although it can be helpful for some on-the-job tasks after you learn the underlying code.

Mastering code and the coding process enables you to understand what happens behind the GUI of a Web editor. This knowledge is invaluable, because it enables you to troubleshoot code issues when the GUI Web editor fails.

Migration to Mobile Devices

Consider that mobile computing now means more than just being able to check e-mail and Web pages on the move from a mobile device. Today, users also move from one mobile environment to another, using a variety of Web-enabled devices from various companies that use different computing platforms.

The principle of responsive design is essential for today's user interface (UI) designers. The term "responsive design" describes the ability to create pages that respond to user screen size and that work in multiple environments. You will learn about responsive design techniques later in this course.

Users no longer view Web pages only through standard Web browsers. Your smartphone, tablet, smart TV and gaming console are all capable of reading Web-based markup languages (Figure 1-1).
As a designer, it is your responsibility to ensure that your pages render as your audience expects, regardless of the device used to access them. Therefore, your code must work in many different environments without requiring a different solution for each one. HTML5 and CSS3 allow most devices to properly render the code, regardless of the size of the device’s screen.

Consider the HTML5 and CSS3 technologies as an attempt to develop a "one size fits all" page development solution for all devices. However, this approach does not work in all situations. You must determine when it makes sense, and when it does not.

App development

The term "app" has become widely used to describe relatively small applications developed exclusively for mobile devices. Smartphones and tablets created a huge market for apps. Many organizations now have an app for download in Google Play™, the Windows Store and the Apple App Store (Figure 1-2).

Most organizations also have Web pages for traditional and mobile browsers to access. The traditional Web page, mobile Web page and mobile app for eBay are all shown below for comparison (Figure 1-3).
HTML5 and CSS3 are capable of creating both apps and Web pages. This course will teach you the skills required to create both types.

**The Web Development Trifecta**

The future of Web design and app development will be ruled by three technologies: HTML5, Cascading Style Sheets (CSS) and JavaScript (see Figure 1-4). Used together, these technologies create Web pages that easily adapt to smartphones, tablets, gaming devices and smart TVs, as well as to traditional computers.
Apple co-founder Steve Jobs provided a great argument for using these technologies in his famous “Thoughts on Flash” blog (Adobe Flash provides multimedia such as video, animation, interactive games and audio):

“…we strongly believe that all standards pertaining to the web should be open. Rather than use Flash, Apple has adopted HTML5, CSS and JavaScript — all open standards. Apple’s mobile devices all ship with high performance, low power implementations of these open standards. HTML5, the new web standard that has been adopted by Apple, Google and many others, lets web developers create advanced graphics, typography, animations and transitions without relying on third party browser plug-ins (like Flash). HTML5 is completely open and controlled by a standards committee, of which Apple is a member.”

HTML5, CSS and JavaScript as a group are sometimes called the "HTML5 family" or the "Web development trifecta." Table 1-1 explains the functions of each technology.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML5</td>
<td>Markup language used for structuring and describing Web page content</td>
</tr>
<tr>
<td>Cascading Style Sheets (CSS)</td>
<td>Style sheet language that provides the formatting and 'look' of a Web page or document written in a markup language</td>
</tr>
<tr>
<td>JavaScript</td>
<td>Scripting language that provides dynamic, interactive capabilities to Web pages</td>
</tr>
</tbody>
</table>

This course teaches HTML5 and CSS. JavaScript will be introduced to show interactive HTML5 elements, but JavaScript is fairly complex and is covered in detail in its own separate CIW course.

**The Evolution of HTML5**

As you should already know, HTML links one document to another via pointers called **hyperlinks**. A hyperlink is a set of instructions embedded within a file that calls another location in the file or a separate file when the link is clicked. The global set of linked documents across the existing Internet framework grew into what is known as the World Wide Web.

**Hypertext Markup Language (HTML)**

Web pages are created using **Hypertext Markup Language (HTML)**. HTML is the markup language that defines page structure, hyperlinks, graphics and more to enable pages to render in Web browsers and other devices. The W3C regulates the development of HTML and CSS standards (Figure 1-5).
**HTML 3.2 and HTML 4 recommendations**

HTML 3.2 and 4 are older but functional HTML standards. Some Web pages and HTML editors still contain this older code.

HTML 4 contained many improvements over HTML 3.2, most notably the addition of Cascading Style Sheets (CSS) for formatting. The HTML 4.01 Recommendation was released in 1999 and brought some minor modifications. You can read the HTML 4 and HTML 4.01 specifications at [www.w3.org/TR/html4](http://www.w3.org/TR/html4) and [www.w3.org/TR/html401](http://www.w3.org/TR/html401) respectively.

**HTML 4.01 flavors**

As Web pages were developed in HTML 4.01, they had three distinct variants, or “flavors.” The HTML 4.01 flavors ensured that you could use the specification and still remain backward-compatible with older Web browsers. Following is a short description of each flavor.

- **HTML 4.01 Transitional** — allowed developers to insert formatting using either CSS or traditional layout instructions.
- **HTML 4.01 Strict** — required the separation of presentation and content. Deprecated tags were disallowed and generated validation errors.
- **HTML 4.01 Frameset** — required for pages that used HTML frames, which placed Web pages inside each other to create separate panes in the browser window.

Some older Web sites still contain code created with HTML 4.01.

**XHTML**

Extensible HTML (XHTML) is a version of HTML that incorporates the strict syntax rules of Extensible Markup Language (XML) with the existing set of HTML 4.01 tags to create Web documents. Like HTML 4.01, XHTML has three flavors: Transitional, Strict and Frameset.

XHTML introduced syntax rules that must be followed in order for a document to validate, including the following:

- The XHTML DOCTYPE statement is required.
- Document structure tags are required.
- The document must have one root element.
- All elements and attributes must be typed in lowercase letters.
- Each element must be properly closed, either with a separate closing tag or with a closing backslash (\), as appropriate.
- All elements must be properly nested.
- Attribute values must be enclosed in quotation marks.

The W3C Recommendation for XHTML was published in 2000. The goal for XHTML is to provide a bridge of backward- and forward-compatibility for HTML documents to easily adapt to XML technologies in future use. Many developers upgraded their Web page code from HTML to XHTML by applying the stricter syntax to existing code.
In 2005, the W3C began work on HTML5 to advance regular HTML, and in 2009 it acknowledged that HTML5 would be the only next-generation version of HTML, although it would include both XML and non-XML serializations. Many sites still use XHTML. In this course, however, you will create pages using HTML5.

**HTML5**

HTML version 5 (HTML5) is the latest version of HTML under development by the W3C. This course will focus on the HTML5 specification. At the time of this writing, HTML5 was a W3C working draft. The W3C has created an HTML5 logo, shown in Figure 1-6, to market the technology.

![HTML5 logo from the W3C](image)

**Figure 1-6: HTML5 logo from the W3C**

HTML5 provides modern design techniques for the Internet while requiring fewer plug-ins. For example, HTML5:

- Standardizes how video and audio are presented on a Web page.
- Introduces the `<video>` element, which is designed to eliminate the need to install third-party plug-ins (such as those for Adobe Flash or Microsoft Silverlight).
- Adds the `<audio>` element, which allows pages to seamlessly add audio files for events such as podcasts.
- Establishes ways to enable drag-and-drop capability for Web pages without using third-party add-ons.
- Gives developers more native tools to use on a page, such as download progress indicators, image captioning options and form validation.
- Provides developers with a native option for offline storage, and enables applications to run as expected even without network connectivity.
- Allows developers to retrieve the geographical location information for a Web site visitor. This technology is called geolocation. Examples include using the global positioning system (GPS) of a mobile device to determine the device’s location, which allows Web services to be provided based on the client’s location.

You will learn more about HTML5 throughout this course. To view the HTML5 specifications, go to: [http://dev.w3.org/html5/spec/Overview.html](http://dev.w3.org/html5/spec/Overview.html)

---

**NOTE:**
Research HTML5-compatible browsers and be sure you have at least one installed. Then search for and visit HTML5 Web sites and older HTML 4.01 sites. What are the similarities and differences between the sites?

**offline storage**
The ability for Web browsers and online services to download and access content and services without being connected to the Internet.

**CIW Online Resources – Online Exercise**
Visit CIW Online at [http://education.Certification-Partners.com/CIW](http://education.Certification-Partners.com/CIW) to complete an interactive exercise that will reinforce what you have learned about this topic.

*Exercise 1-2: The evolution of HTML5*
Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) technology determines how to display HTML elements in your Web pages. CSS commands contain formatting instructions that can define the font, color and display of phrase elements used on a particular markup page.

If all pages on your site are linked to the same external style sheet, then one simple change to the style sheet will change all associated elements across the site. Thus, if you want to change some formatting site-wide (for example, increase the size of a document heading), you need not change every page manually. You need only change a line in the style sheet file, then all your headings will change their appearance to conform to the style sheet. This technology can save a great deal of development and maintenance time, as well as make a more consistent, accessible interface.

Additional Web page elements

Web pages can incorporate more than just HTML and CSS. You can use other languages to enhance a page, such as JavaScript, JScript and VBScript. You can also enhance Web pages by inserting specialized content created with programs and technologies such as Java (www.java.com), ActiveX (www.microsoft.com), Microsoft Silverlight (www.microsoft.com/silverlight/) and Adobe Flash (www.adobe.com/products/flash/). You will learn about all these technologies throughout this course.

HTML5 Structure Elements

HTML5 with CSS provides an effective and simple way to structure Web pages. The developers of HTML5 created specific elements to define the document structure. These elements include <header>, <footer>, <main>, <nav>, <section>, <article> and <aside>, shown in Figure 1-7.

As the figure illustrates, a developer can structure the Web page with basic elements that are easily interpreted and native to any HTML5-compliant browser, regardless of whether the browser is on a mobile device, laptop or tablet. Table 1-2 describes the HTML5 page structure elements.
Table 1-2: HTML5 structure elements

<table>
<thead>
<tr>
<th>HTML5 Structure Element</th>
<th>Description</th>
<th>Example Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;header&gt;</td>
<td>Defines the top of the Web page, similar to the header in a word-processing document. Also can be contained in article, aside, nav, main and section elements to provide a header in those areas.</td>
<td>General topic description for the content below it. Company logo. News headlines.</td>
</tr>
<tr>
<td>&lt;main&gt;</td>
<td>Defines the main content of the document body. Cannot be the child of the header, nav, article, aside or footer elements.</td>
<td>Content is directly related to the subject of the document.</td>
</tr>
<tr>
<td>&lt;nav&gt;</td>
<td>Defines an area for navigation links.</td>
<td>A hypertext menu to access various pages of the Web site.</td>
</tr>
<tr>
<td>&lt;section&gt;</td>
<td>Defines portions or areas of a document as necessary. Can group other element types.</td>
<td>A group of articles or images. Any portion of the Web page document.</td>
</tr>
<tr>
<td>&lt;article&gt;</td>
<td>Defines site content accompanying the main content.</td>
<td>Company services, news articles, blogs, images, videos, tweets, social networking posts.</td>
</tr>
<tr>
<td>&lt;aside&gt;</td>
<td>Defines content that is aside from or additional to the main article content. Not just for sidebars, it can be used for any content.</td>
<td>Advertisements, news feeds, highlighted resource links.</td>
</tr>
<tr>
<td>&lt;footer&gt;</td>
<td>Defines the bottom of the Web page, similar to the footer in a word-processing document. Also can be contained in article, aside, nav, main and section elements to provide a footer in those areas.</td>
<td>Site copyright, owner, contact information.</td>
</tr>
</tbody>
</table>

These HTML5 basic structure elements may or may not render in older browsers that do not support HTML5, such as Internet Explorer versions prior to IE9. This should not hinder your use of HTML5 structure tags, but you should be aware of it.

**CIW Online Resources – Online Exercise**

Visit CIW Online at [http://education.Certification-Partners.com/CIW](http://education.Certification-Partners.com/CIW) to complete an interactive exercise that will reinforce what you have learned about this topic.

**Exercise 1-3: HTML5 structure elements**

**Validating HTML5 Code**

Code validation is the process of checking your code to verify that it complies with the syntax rules for your chosen standard. After you have checked your code and corrected any discrepancies, a code validator confirms that your code is compliant, and thus "validated."

Validating your markup code is worthwhile because validated code is most likely to be interpreted accurately by the majority of user agents. As a result, you have a better chance of your pages rendering as you expect and to a larger audience.
Make sure you adopt a single W3C standard (e.g., HTML5 or XHTML) and apply it consistently so that when you validate your markup code, the code will match the DTD (standard syntax rules) that you specify.

**Using a code validator**

You can validate your markup code automatically with online tools. Several code validators exist, but the most authoritative is the W3C Markup Validation Service ([http://validator.w3.org](http://validator.w3.org)). Using this service, you can upload local HTML files for validation, or provide the URL of a Web page to validate it.

Note that code validators check your file’s code for standard compliance, and some will specify problems or direct you to problem locations. However, validators do not correct the code for you — you must correct your code manually.

Remember to validate your code again after a round of corrections, as many times as necessary until your code validates as fully compliant.

In this course, you will use the W3C validation service (Figure 1-8) to validate your HTML5 code. However, it is important to note that this validator reads the `<DOCTYPE>` declaration on an HTML page and validates according to the specified DTD. So, if your document references an older HTML 4.01 Strict DTD, for example, then the validator will validate your code according to the HTML 4.01 Strict specifications.

> **NOTE:** You will practice validating markup code later in the course.

Following are some tips to consider when validating your markup code:

- Do not be discouraged when you see multiple problems reported for a page. Sometimes one small flaw can cause the remaining code on the page to fail validation, even if the remaining code is actually valid.

- When errors are reported, search through the code carefully to find the true problem. Sometimes when a validation program finds a problem, it does not report the correct cause or it may not report the cause clearly.

- Make sure that you are validating the correct file.
Universal markup creation

Good Web page coding generally involves ensuring that the content is rendered appropriately regardless of the browser used to render it. To ensure this type of consistent, or universal, rendering, Web page developers can apply certain practices to their coding that result in universal markup:

- Follow W3C standards carefully.
- Choose one standard version of any given language you use, and apply that standard consistently throughout your document, pages or site.

Creating universal markup code is important for several reasons:

- Your pages will be ensured to render in future versions of most browsers.
- Your pages will be more scalable. This means that as you add more sophisticated content, make pages searchable or use the content in ways you have not yet imagined, you can still use markup you created without having to revise the code extensively.
- Your pages will be more accessible to all users, including those with disabilities. You will be able to more easily make your pages compliant so that your site is available to the widest possible audience and does not present a liability to your organization.

In the following lab, you will enhance a simple Web page with HTML5 structure elements. Suppose you have been assigned to create an HTML5-compliant Web page that will describe the mission of Habitat For Humanity. This HTML file should be named index.html and should validate as HTML5. What steps would you take to create this page?

Lab 1-1: Structuring and validating a Web page

In this lab, you will add structure elements to a Web site, then validate it as HTML5. You can use any Web browser. In fact, you are encouraged to view code in multiple browsers to ensure that you are creating pages that render well in various environments.

1. If necessary, configure your operating system so that you can read the full extensions of all file names. This will allow you to find your HTML files more easily.

   a. In Windows 8, open Start by pointing to the upper-right corner of the screen, moving the mouse pointer down, and selecting Start. If you are using a touch screen, swipe in from the right edge of the screen and select Start.

   b. Select Control Panel | Appearance And Personalization. In the Folder Options section, select Show Hidden Files And Folders. Ensure that the View tab is selected and choose the Show Hidden Files, Folders And Drives radio button.

   c. Deselect the Hide Extensions For Known File Types check box, then click OK to close the Folder Options dialog box. Close the Control Panel.

2. Copy the Lesson01 folder from your student lab files to your Desktop.
3. Open the Lab 1-1 folder and right-click the index.htm file. Select Open With and choose Notepad.

   Note: If Notepad does not appear when you select Open With, then select Choose Default Program. Click the down arrow next to Other Programs then double-click Notepad.

4. In the index.html file, insert the following structure elements (shown in bold) exactly as written. Be sure to close your tags properly.

```html
...  
</head>
<body>
<header>
</header>
<nav>
<h4>Simple, decent, affordable housing</h4>
<ul>
<li><a href="http://www.habitat.org/how/about_us.aspx">Learn more</a></li>
<li><a href="http://www.habitat.org/getinv/volunteer_programs.aspx">Volunteer</a></li>
<li><a href="https://www.habitat.org/cd/giving/donate.aspx?r=r&amp;link=1">Donate</a></li>
</ul>
</nav>
<article>
<h1>Our Mission</h1>
<p>Habitat for Humanity believes that every man, woman and child should have a decent, safe and affordable place to live. We build and repair houses all over the world using volunteer labor and donations. Our partner families purchase these houses through no-profit, no-interest mortgage loans or innovative financing methods.</p>
</article>
<article>
<h1>Why We Build</h1>
<p>There are nearly 2 billion people around the world who live in slum housing and more than 100 million are homeless. Families left homeless by natural disasters, war and civil unrest often face dire housing situations as they struggle to rebuild their lives. We provide shelter and housing assistance to help these families recover.</p>
</article>
<article>
<h1>Advocacy in Action</h1>
<p>Volunteer Amy Miles lends her voice in support of Habitat's advocacy efforts and Global Village trips. Use the audio controls below to hear her stories.</p>
</article>
```
5. Once you have inserted the structure elements, save your changes.

6. Now, open index.htm in a Web browser by right-clicking the file, selecting Open With, and choosing an HTML5-compliant browser. It should resemble Figure 1-9.

![Figure 1-9: File index.htm in Firefox with structure elements](image)

Tech Note: If you viewed the page in a browser before adding structure elements, you will notice that the page's appearance does not change. The HTML5 structure elements add context information to the file, rather than visual formatting.

7. As you can see, you have created a basic Web page structure that will validate as HTML5, as long as you have entered the code correctly. To verify this, visit [http://validator.w3.org](http://validator.w3.org). You will see the W3C Markup Validation Service Web page, as shown in Figure 1-10.
8. Click the **Validate By File Upload** link. To the right of the File text box, click the **Browse** button. Navigate to the *index.htm* file you have created and select it by double-clicking.

9. Click the **Check** button.

10. If your code does not validate, make appropriate changes. Warnings are OK and do not mean you made an error. In Figure 1-11, the warning states that the W3C validator checked the document with an experimental HTML5 conformance checker.

11. After your code validates, leave your Web browser and text editor open. In this lab, you created and validated an HTML5 document.

---

### The `<video>` Element

HTML5 has introduced the `<video>` element to provide developers with a standard method to embed video into their Web pages (see Figure 1-12). Prior to the `<video>` element, end-users needed browser plug-ins (such as Adobe Flash, Microsoft Silverlight or Apple QuickTime) in order to view video. Not all browsers or devices support these plug-ins, so a standardized way to include video was needed.
NOTE:
Plug-ins are no longer required to embed video into Web pages. In previous years, entire courses were dedicated to teaching developers how to create content for proprietary plug-ins such as Flash, because plug-ins were necessary for videos to appear and play on a Web page.

Figure 1-12: Page using HTML5 <video> element

Embedding video in an HTML5 document is straightforward. See the following code:

```html
<video width="360" height="270" controls="controls" poster="image.png">
  <source src="video.mp4" type="video/mp4" />
  <source src="video.webm" type="video/webm" />
  <source src="video.ogg" type="video/ogg" />
</video>
```

Your browser does not support the HTML5 video element.

Table 1-3 describes the elements and attributes used in this example code.
Table 1-3: HTML5 <video> elements and attributes

<table>
<thead>
<tr>
<th>HTML5 &lt;video&gt; Element or Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;video&gt; element</td>
<td>Defines a video to embed in the Web page</td>
</tr>
<tr>
<td><strong>width and height attributes</strong></td>
<td>Specifies the width and height (in pixels) of the video window If not specified, the video size will be determined by the source video file when it loads, which could change the Web page layout considerably</td>
</tr>
<tr>
<td>controls attribute</td>
<td>Adds video controls such as the Play, Pause, Rewind and Volume controls These video controls are native to HTML5</td>
</tr>
<tr>
<td><strong>poster attribute</strong></td>
<td>Identifies an image to be displayed until the Play button is clicked or while the video is downloading If the poster attribute is not specified, the first frame of the video is displayed instead</td>
</tr>
<tr>
<td>&lt;source&gt; element</td>
<td>Defines the media resource Multiple sources can be listed, such as different types of video formats, to support a variety of devices and browsers</td>
</tr>
<tr>
<td><strong>src attribute</strong></td>
<td>Identifies the location and file name of the media resource</td>
</tr>
<tr>
<td><strong>type attribute</strong></td>
<td>Identifies the format, or MIME type, of the video The &lt;video&gt; element supports three formats: MP4, WebM and Ogg</td>
</tr>
<tr>
<td>Text</td>
<td>Text enclosed in the &lt;video&gt; element will appear on the page if the browser or device cannot support any of the video formats available</td>
</tr>
</tbody>
</table>

Multiple sources can be identified with the <source> element to ensure various video formats are supported. The HTML5 specification does not require a video codec to be supported by all user agents, but it does support the following formats:

- **MPEG-4 (MP4)** — generally uses the H.264 video codec, which is native to most browsers that support HTML5. This codec uses far less processor and battery power because it does not require a plug-in. YouTube recently reformat most videos away from Flash (which requires a plug-in) to MP4.

- **WebM** — generally uses the VP8 codec, which is an open video compression format owned by Google.

- **Ogg** — uses the Theora format for HTML5 video, which is a free video compression format that can be distributed without licensing fees.

To ensure all browsers and devices can access your video, you should format your video files to all three of these formats and identify them in the <source> element. If that is not possible, then choose one (such as the MP4 format) as the default format.

In the following lab, you will add video to a Web page using the HTML5 <video> element. Suppose your supervisor asks you to provide a step-by-step instructional video in your company Web site that customers can access from anywhere. The video must be able to play on all smartphones, tablets, desktop computers and gaming consoles that have an HTML5-compliant browser. No plug-ins can be required to view it.
In this lab, you will add an MP4 video to your Habitat For Humanity Web page. The video cannot require a browser plug-in, so you must use the HTML5 <video> element.

1. **Windows Explorer:** Open the Lab_1-2 folder in your student lab files.

2. **Editor:** Open `index.htm`.

3. In the `index.html` file, insert the `<video>` element exactly as written in bold below:

```html
<article>
  <h1>Why We Build</h1>
  <p>There are nearly 2 billion people around the world who live in slum housing and more than 100 million are homeless. Families left homeless by natural disasters, war and civil unrest often face dire housing situations as they struggle to rebuild their lives. We provide shelter and housing assistance to help these families recover.</p>
  
  <video height="210" class="center" controls="controls">
    <source src="media/habitat_video_web_720.mp4" type="video/mp4"/>
    Your browser does not support the HTML5 video element.
  </video>
</article>
```

4. After you have inserted this code, save your changes.

5. Refresh `index.htm` in your Web browser by selecting the F5 key, or by right-clicking the Web page and selecting Refresh (depending on your browser or whether you have a touch screen). The page should resemble Figure 1-13.

*Figure 1-13: Video added to Habitat site*

7. After your code validates, save and close the file.

In this lab, you added a video to your HTML5 page using the `<video>` element.

## The `<audio>` Element

Similar to the `<video>` element, HTML5 has introduced the `<audio>` element to provide developers with a standard method to embed audio into Web pages (see Figure 1-14). Prior to the `<audio>` element, end-users needed browser plug-ins or separate applications (such as Windows Media Center, Apple iTunes, Apple QuickTime, Adobe Flash and Microsoft Silverlight) to listen to audio in Web pages. Not all browsers or devices supported these plug-ins, so a standardized way to include audio was needed.

![Figure 1-14: Page using the HTML5 `<audio>` element](image)

Embedding audio in an HTML5 document requires the following code:

```html
<audio controls="controls">
  <source src="media/audio.mp3" type="audio/mpeg" />
  <source src="media/audio.wav" type="audio/wav" />
  <source src="media/audio.ogg" type="audio/ogg" />
</audio>
```

As with the `<video>` element, the `controls` attribute identifies the default audio controls: Play, Pause, Volume, etc. Any text enclosed within the `<audio>` element will appear in browsers that do not support it.

Like the `<video>` element, the `<audio>` element allows you to identify multiple sources with the `<source>` element to ensure various audio formats are supported. The HTML5 specification does not require a specific audio codec to be supported, but it does support three audio formats:

- MP3
- WAV
- Ogg

**OBJECTIVE**

1.8: HTML5 `<audio>` element

**NOTE:**

It is not necessary to include three file formats of each audio file you provide. Choose a file format that is more popular, such as MP3 or WAV. The Ogg format has been slow to gain popularity.
Not all HTML5-compliant browsers support all of these audio formats. Table 1-4 lists the HTML5 audio formats and shows which HTML5-compliant browsers support each format. (Information is current at the time of this writing.)

<table>
<thead>
<tr>
<th>Audio Format/Codec</th>
<th>Supported by Browser?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Chrome</td>
</tr>
<tr>
<td>MP3</td>
<td>Yes</td>
</tr>
<tr>
<td>Ogg</td>
<td>Yes</td>
</tr>
<tr>
<td>WAV</td>
<td>Yes</td>
</tr>
</tbody>
</table>

In addition to the controls attribute, several other attributes are common to both the <video> element and the <audio> element. Table 1-5 describes two widely used attributes.

<table>
<thead>
<tr>
<th>HTML5 &lt;audio&gt; and &lt;video&gt; Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>loop=&quot;loop&quot;</td>
<td>Specifies that the audio or video file will play over and over again without stopping</td>
</tr>
<tr>
<td>autoplay=&quot;autoplay&quot;</td>
<td>Specifies that the video will automatically play immediately upon loading</td>
</tr>
</tbody>
</table>

In the following lab, you will add audio to a Web page using the HTML5 <audio> element. Suppose your supervisor asks you to embed an audio tour into your company Web site. The audio must be able to play on any device with an HTML5 browser, without requiring any plug-ins.

**Lab 1-3: Embedding audio with the HTML5 <audio> element**

In this lab, you will add an MP3 audio file to your Habitat For Humanity Web page using the HTML5 <audio> element.

1. **Windows Explorer:** Open the Lab_1-3 folder in your student lab files.
2. **Editor:** Open index.htm.
3. In the index.html file, insert the <audio> element exactly as written in bold below:

```html
...<article><h1>Advocacy in Action</h1><p>Volunteer Amy Miles lends her voice in support of Habitat's advocacy efforts and Global Village trips. Use the audio controls below to hear her stories.</p><audio controls="controls" class="center"><source src="media/habitat_audio.mp3" type="audio/mpeg" />
Your browser does not support the HTML5 audio element.</audio></article>...
```

**OBJECTIVE**

1.8: HTML5 <audio> element
4. After you have inserted this code, save your changes.

5. Refresh your Web browser. Your page should now resemble Figure 1-15.

![Figure 1-15: Audio file added to Habitat page](image)


7. After your code validates, leave your Web browser and text editor open.

In this lab, you added an audio file to your HTML5 page using the `<audio>` element.

### HTML5 APIs

The future of Web development will probably focus on HTML5 APIs. An **application programming interface (API)** is a source code specification that enables components of an application or program to work together to produce the desired functionality. HTML5 APIs provide an open environment for developing Web applications that does not rely on proprietary browser plug-ins.

HTML5 APIs consist of the trifecta technologies: HTML5, CSS and JavaScript. These technologies are used together to provide Web pages that can easily adapt to smartphones, tablets, gaming devices and smart TVs, as well as to traditional desktop computers.

HTML5 APIs are also used to create apps for mobile devices, not just Web pages. For example, the mobile apps for Pandora and LinkedIn use HTML5 APIs (Figure 1-16). Many
expect mass adoption of HTML5 APIs in the next few years as mobile devices continue to proliferate.

Figure 1-16: LinkedIn mobile app developed with HTML5

Some functions of the HTML5 APIs include:

- Media (audio and video).
- Document editing.
- Cross-document messaging.
- MIME type and protocol handler registration.
- Web storage.

For a complete list of HTML5 APIs from one of the W3C members, visit http://platform.html5.org/. You should bookmark this Web page.

Popular APIs

Some popular APIs include Offline AppCache, Geolocation, Canvas and Drag-and-Drop. You will study these APIs in detail later in the course.

To give you an idea what APIs can do, consider the Canvas API. Canvas provides a place on a Web page (a “canvas”) where developers can display graphics, animation, video and games “on the fly” (dynamically) without the need for a plug-in. By itself, canvas is a simple pixel-based drawing API that produces a bitmap image. You can draw objects on a canvas using JavaScript. You can also use it for more advanced tasks that include interactivity, such as video games, simulations, video editing or image configurations (see Figure 1-17).
Previously, end-users had to install plug-ins such as Adobe Flash Player or Microsoft Silverlight to view these types of files. Moving forward, any user who has an HTML5-compliant browser will be able to experience multimedia without a plug-in.

**API compatibility**

Canvas and other APIs use fewer resources than a plug-in does (such as battery power and CPU memory). This is especially important with regard to mobile devices, such as smartphones and tablets, which rely upon battery power.

However, older browsers do not fully support APIs. Therefore, you should always provide alternative content or controls. The older browsers will ignore API elements (such as `<canvas></canvas>` tags) and will render the content that is between them. See the following example:

```html
<canvas id="myCanvas" width="200" height="100">
  Your browser does not support the Canvas element.
</canvas>
```

Using the preceding code, an older browser will display the text, “Your browser does not support the Canvas element.” All compatible browsers will display the Canvas element styled with a solid 1-pixel black border.

**CIW Online Resources – Course Mastery**

Visit CIW Online at [http://education.Certification-Partners.com/CIW](http://education.Certification-Partners.com/CIW) to take the Course Mastery review of this lesson or lesson segment.

*Course Mastery Lesson 1*
Case Study

Google This

Microsoft created the Windows Phone 8 operating system for smartphones. The company is also creating apps as quickly as possible to support the phone, and to compete with Apple and Google.

One of the apps under development is a YouTube app, which allows users to access YouTube videos from their Windows Phone 8 devices (even though YouTube is owned by Google). Microsoft developed the app and uploaded it to the Windows Store.

Windows Phone 8 users were surprised to be denied access to YouTube from their phones when using this app. Google disabled all usage from the Microsoft app.

- Why would Google block access? Wouldn’t Google encourage more traffic to its site, even if it was from a competitor’s operating system?

Surprisingly, Google blocked the Microsoft app because it was not developed in HTML5.

A Google spokesman said, “We’re committed to providing users and creators with a great and consistent YouTube experience across devices, and we’ve been working with Microsoft to build a fully featured YouTube For Windows Phone app, based on HTML5.”

Although there are certainly politics involved (Apple and Android YouTube apps are not developed in HTML5), this highlights the importance of HTML5 as a mobile app development tool.

* * *

Consider this scenario and answer the following questions:

- What specific advantages did Google’s representative cite as the reason for requiring an HTML5 app?

- From what you have learned so far, what disadvantages do you think users of the Microsoft app would have suffered with the app that was developed without HTML5?

- Which HTML5 elements might this app use?
Lesson 1: HTML5 Essentials

Lesson Summary

Application project

HTML5 implementations are spreading quickly as organizations and users adopt mobile devices for work and play.

Think of five Web sites that you visit frequently. Are they HTML5 sites? To find out, view the source code of the page and view the DOCTYPE declaration. Is the DOCTYPE declaration written simply as follows?

```xml
<!DOCTYPE html>
```

If so, then you are looking at an HTML5 page.

Note that the DOCTYPE declaration is not required for HTML5, but it is good practice to include it for older HTML standards to tell the browser which version of HTML to expect. It is almost always written in uppercase letters by Web developers because older versions of HTML require this case-specificity.

Skills review

In this lesson, you learned about the flexibility of HTML5 for both traditional and mobile device environments. You were introduced to the Web development trifecta, which consists of HTML5, CSS, and JavaScript. You reviewed previous versions of HTML, such as HTML 4.01, and you learned the importance of validating HTML5 code, a step that should be included with all Web development. You also learned about the HTML5 structure elements, the video and audio elements, and the canvas API.

Now that you have completed this lesson, you should be able to:

- 1.1: Consider HTML5 development skills you can apply to both traditional and non-traditional delivery platforms, including mobile apps.
- 1.2: Identify the Web development trifecta and explain its significance.
- 1.3: Explain the evolution of HTML and its relevance to modern design techniques.
- 1.4: Create and deploy HTML5 structure tags.
- 1.5: Perform HTML5 code validation, which includes explaining the practical value of code validation and listing the steps to validate HTML5 code.
- 1.6: Explain the importance of universal markup creation.
- 1.7: Apply the HTML5 `<video>` element appropriately.
- 1.8: Use the HTML5 `<audio>` element.
- 1.9: Define common elements of HTML5 application programming interfaces (APIs).
Lesson 1 Review

1. What are three reasons to justify the use of HTML5?

2. What is the primary benefit of the HTML5 <video> element?

3. How can an external style sheet save development and maintenance time, as well as make a more consistent, accessible interface?

4. How is the HTML5 <article> structure element used in a Web page?

5. Why should you always validate your code before publishing your Web pages?