HTML5

Lesson 1: **Defining HTML**

Windows, Bing, PowerPoint, Internet Explorer, Visual Studio, WebMatrix, DreamSpark, and Silverlight are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

The information contained in this tutorial is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this tutorial.

Created for Microsoft by Avlade: [www.Avlade.com](http://www.Avlade.com)

Copyright 2011 by Microsoft Corporation.

Certain portions excerpted from the *HTML5 Digital Classroom* are copyright 2011 by AGI Training, and are used by permission. For more information about the *HTML5 Digital Classroom* book visit <http://www.digitalclassroombooks.com/Web-Design/HTML5-Digital-Classroom-Book.html>

Lesson 1

Defining HTML5

## Learning objective:

In this lesson, you will discover the features and capabilities that are part of the HTML5 specification and related web technologies. You will also understand the benefits HTML5 provides to web designers, developers and end-users.

## Starting up

You will not need any files for this lesson.

This lesson provides a general overview of HTML, and remaining lessons involve step-by-step exercises.

## Defining HTML5

HTML5 is a combination of new HTML markup tags, CSS3 properties, JavaScript, and several supporting technologies related to, but technically separate from, the HTML5 specification. For this reason, we make a distinction between the HTML5 specification and the HTML5 family.

You can define the HTML5 specification as new markup elements, or syntax, used by designers to build web pages in conjunction with the tags that are currently used. Many of these new elements are familiar to designers who work with traditional HTML tags, such as <p>, <ul>, and <div>. These new tags provide better tools for developers and designers, and translate to better experiences for users.

The HTML5 family includes the new tags and technologies such as CSS3, Geolocation, Web Storage, Web Workers, and Web Sockets. These technologies provide a more powerful upgrade to the toolset, and result in more useful and sophisticated web pages.

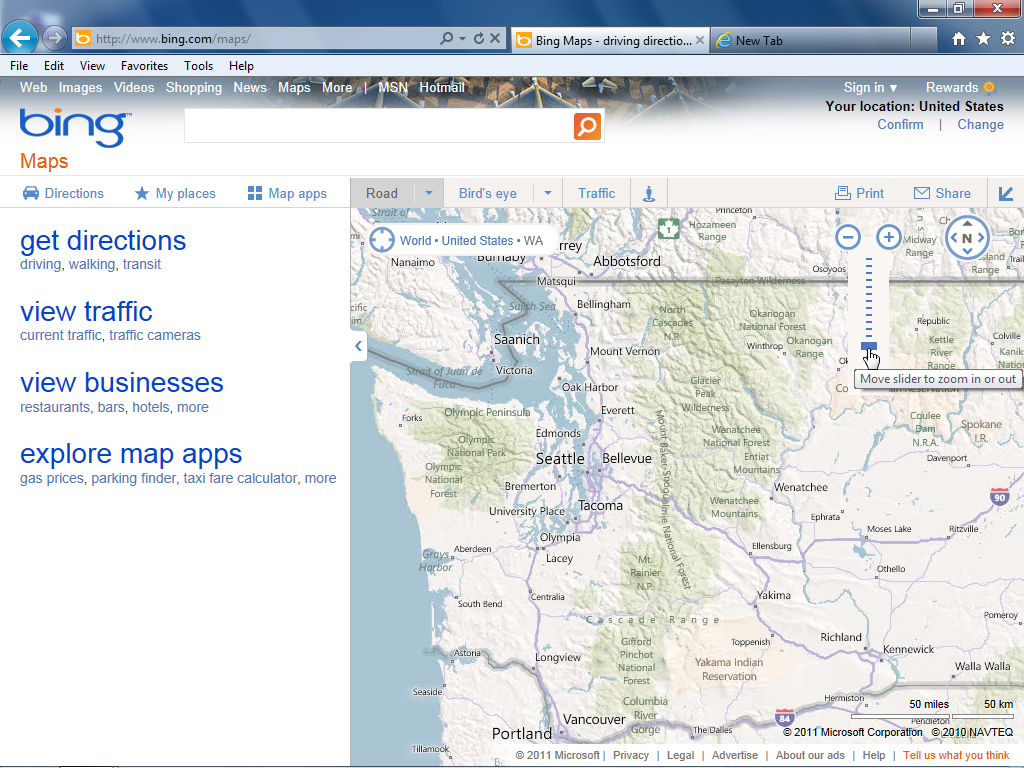
New browsers add features based on consumer expectations and as part of the natural evolution of technology. As web applications become more responsive, speedy, and able to work with complex tasks such as image editing, mapping, spreadsheets, and video, users expect this level of performance from all web applications. There are limitations with the capabilities of current languages and the ease of implementing and adding these features. HTML5 provides new tools and features to help make websites more useful and exciting.

| The history of HTML5 |
| --- |
| HTML4, which was nearing completion in 1998, is the markup language that forms the basis for much of the underlying framework of web pages today. Web designers and developers have been using the specification HTML 4.01 for years quite successfully, in combination with CSS to define style and JavaScript to add interactivity.  After HTML 4.0.1 was finalized, the W3C continued the evolution of the web by ending the development of HTML 4.0.1 and starting a language called XHTML 1.0. There is little difference between HTML 4.0.1 and XHTML 1.0 (XHTML is a more strict language; for example, all tags need to be lowercase). One of the goals of XHTML 1.0 was to create a markup language that would expand and meet the demands of future technology, such as mobile devices. Many websites have been built using XHTML 1.0 as a framework, and many designers and developers appreciated the more strict rules of syntax.  As XHTML 1.0 was adopted and used, the development of another language, XHTML 2.0, began. XHTML 2.0 added several new powerful features to web pages and borrowed heavily from XML. However, there were several technical issues with the new language, and it didn’t reflect the actual needs of web developers.  In 2004, a group of representatives from browser manufacturers, and a group of working web developers formed an independent group called the WHATWG (Web Hypertext Application Technology Working Group). Their mission was to create a better HTML markup specification, designed to build the new type of web application without breaking backwards compatibility with existing browsers.  The result was the Web Applications 1.0 specification, which identified the features that existing browsers shared, and proposed new features, such as the APIs that form the web family. As a result, the development of XHTML 2.0 ceased and HTML 4.0 was recreated as HTML5. However, remembering that HTML5 started as Web Applications 1.0 helps to understand what HTML5 is designed to do. |

For more information about the WHATWG, visit www.whatwg.org.

## HTML5 expands the definition of what a web page can do

Currently, HTML cannot play multimedia such as audio and video without a browser plug-in like Adobe® Flash® or Microsoft® Silverlight®. HTML also has no capability to store data on a user’s computer; this is currently done with a scripting language or another technology. There is no native drawing format in HTML: graphics and animations are currently supplied as image files or through browser plug-ins such as Adobe Flash, Java, or Microsoft Silverlight. In general, as more and more people rely on the web and web applications, the expectations of what a web page can efficiently do is constantly growing.

This user demand for higher performance and more fully featured websites is limited by the current HTML language.

A site such as Bing™ Maps is a high-performance web application that could benefit from the upcoming features in HTML5.

The following sections provide a brief explanation of some of the most important aspects of HTML5.

## HTML5 markup

HTML5 markup introduces several new tags designed to make the structure of a web page more logical and functional. Before HTML5, the page structure relied heavily on the <div> tag, often paired with a CSS class or ID. For example, in HTML 4.0 it is standard practice to define the header of a web page as follows:

<div id="header" > This is my header </div>

In this example, the code highlighted in red refers to the CSS ID that defines the width and height of the header, as well as background color. In the CSS code, this would appear as follows:

#header {

  width:960px;

  height:100px;

  background-color:gray;

}

The ID name header is arbitrary. Some designers might use the name masthead, topsection, or box. In the HTML5 specification, there is a unique tag called <header>, which replaces the use of the <div> element so the syntax is much more logical and consistent:

<header> This is my header </header>

In this example, you can directly add style properties (width, height, background color, and more) to the header rule in CSS:

header {

  width:960px;

  height:100px;

  background-color:gray;

}

The difference between the two code examples above may be difficult to understand if you aren’t familiar with CSS. The first example uses a selector (#header) that is referring to a CSS ID attribute. The second example uses a selector (header) that is new to HTML5 and allows you to style the element directly. HTML5 has a number of other new elements such as <footer>, <nav>, <section>, <aside>, and <article>. These names are based on the most common ones used for layout sections in today’s web pages (div id="footer", div id="nav", and so on). The goal of the new HTML5 elements is to reduce the current reliance on <div> tags and replace them with a more consistent and readable page structure. Note that HTML5 doesn’t replace any HTML syntax; it adds new vocabulary to the existing list. In other words, you can still use the <div> tag, but it no longer supports the entire load of a page layout.



The structure of a website with HTML5 elements.  
A. Header. B. Nav. C. Section. D. Article. E. Aside. F. Footer.

## A tour of the key HTML5 elements

This book explains many of the new HTML5 elements in depth, beginning with Lesson 4. But first, Lesson 1 will provide an overview of the key additions to the HTML5 syntax.

## The **<video>**, **<audio>**, and **<canvas>** elements

The HTML5 specification includes tags that let you incorporate multimedia without browser plug-ins. The <video> and <audio> tags let you embed video and audio into your pages as you currently do with images using the <img> tag. The <canvas> tag supplies HTML with a native drawing and animation format. This tag could also provide an alternative platform for the type of graphics and animation found in Flash movies. However, there are significant issues that need to be addressed.

## The **<video>** and **<audio>** elements embed media files into your pages

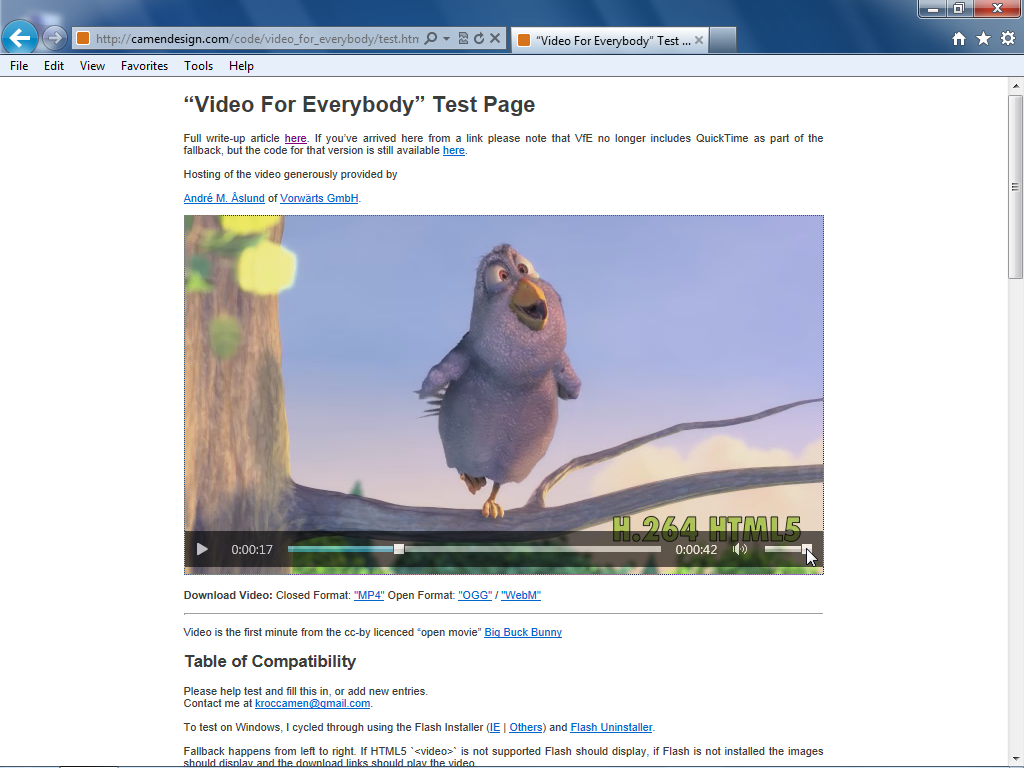
The structure for embedding video and audio is simple, as the following example to adding video to a web page shows:

<video src="catz.mp4" width="400" height="300"></video>

Embedding an mp3 audio file onto your page is similar; for video and audio, you can add built-in player controls and preloading capabilities with the controls, preload, and autobuffer attributes:

<audio src="high\_seas\_rip.mp3" controls preload="auto" autobuffer></audio>

This feature of HTML5 is helpful when compared with current available methods; for example, embedding video using Adobe Flash or Microsoft Silverlight requires more code to implement. Additionally, there is currently a large category of mobile devices such as the Apple® iPhone® or Windows® Mobile Phone 7 that do not support Flash or Silverlight, and in these cases the HTML5 video element is the primary alternative. Although the simplicity of having <video> and <audio> tags is welcome, it does raise the question of how older browsers that do not recognize these tags are supposed to handle them.

This is a question you will explore in Lesson 4 “*Using HTML5 Markup*”.

An HTML5 video player with controls is embedded into a web page; no browser plug-in needed.

### The **<canvas>** element provides drawing and animation features

The canvas element works as a drawing surface on your page. Within this drawing surface, you can create shapes with color, gradient, and pattern fills. You can interactively manipulate pixels onscreen, render text, and export these contents to a static image file, such as a .png. You can also add JavaScript or the new CSS3 animation features to make the objects you create move, fade, scale, and so on. Adding a canvas element to a page is simple:

<canvas id="myCanvas"></canvas>

JavaScript handles all of the work, and it provides a context for the object you create. For example, to create one of the simplest canvas objects, a black rectangle, the code might appear as follows:

<script>

  var canvas = document.getElementById ("myCanvas"),   
context = canvas.getContext("2d");

  // x = 10, y = 20, width = 200, height = 100

  context.fillRect(10, 20, 200, 100);

</script>

This creates a simple rectangle filled with the color black. The code might seem longer than necessary for a simple result, but the canvas tag simply exposes the code within an HTML document and this model provides a number of benefits. Future uses include user interface elements such as player controls, illustration elements that can be animated, and data visualization uses, such as charts and graphs.

You will learn more about creating content with the <canvas> tag in Lesson 5, Working with Canvas.

You can see examples of <canvas> (including animated canvas elements) and other HTML5 features at www.nevermindthebullets.com.

## Web forms

The new form elements in HTML, when implemented, will make working with forms easier than at present. For example, many web designers need to create forms that require validation before the data is submitted. In order to do so, a user must enter a value in the email field of a form submission. Currently, most solutions to this problem require JavaScript or another scripting language, but HTML5 just adds the required attribute to the list of form input types, as shown below:

<input type="email" required>

There are many new form input types, such as email for email addresses, search to designate form fields used with search terms, url to specify a form field that uses a web address, and many more. These new web form elements will take time to become part of the official specification, but they are designed to regress to generic input forms. In other words, you can begin to use these new input types, and if a browser does not support the new form element, it will use a generic (supported) element.

## Many more new HTML5 elements

In addition to the new <video>, <audio>, <canvas>, and form elements, there are several more new elements for use in HTML5. Some examples are the <figure> and <figurecaption> elements used to label images on your pages, the <hgroup> element to group a set of heading elements into a logical section, and more. HTML5 also addresses existing tags found in HTML 4.0.1 that were outdated or needed refinement, such as <i>, <b>, <small>, <strong>, and <abbr>, which have new meaning and new uses in HTML5.

An overview of HTML5 APIs and supporting technologies

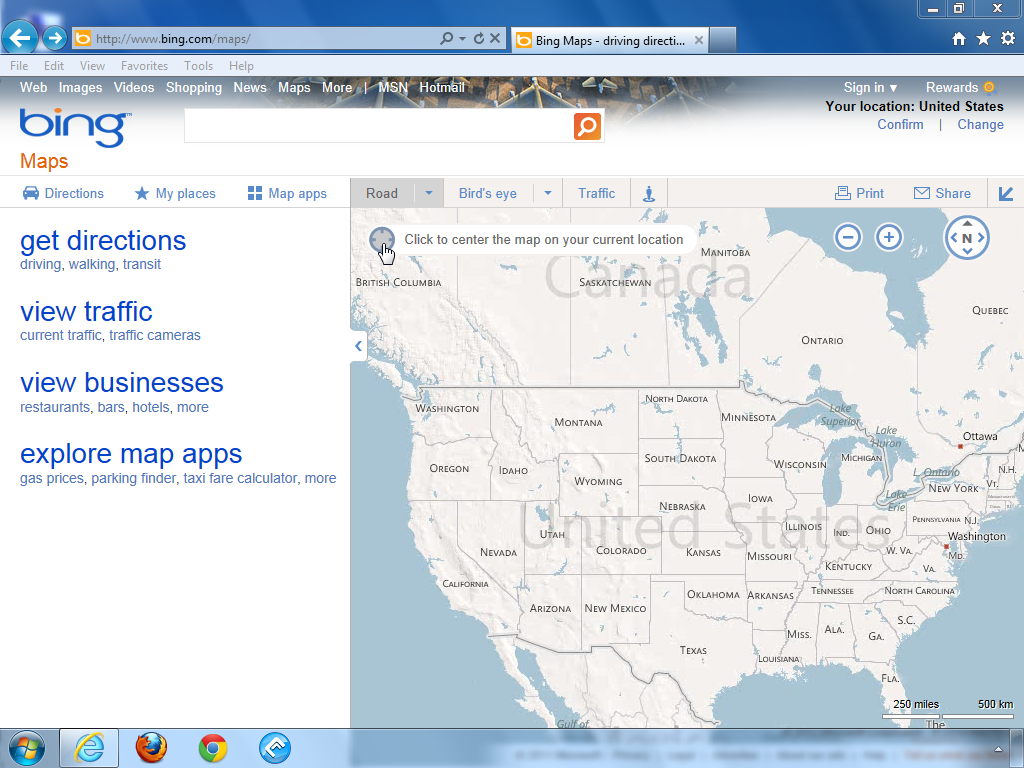
There are other developing web technologies related to the HTML5 specification on syntax. These techniques will help you perform tasks in the web browser and supported mobile devices that were not possible in previous versions of HTML.

| What is an API? |
| --- |
| APIs (Application Programming Interfaces) are a way to create applications using pre-built components and are not unique to web development, or even to scripting languages. Websites such as Twitter, YouTube and others provide APIs to the public so designers and developers can integrate features into their own websites (or for other purposes such as mobile or desktop applications). One of the main goals of an API is to standardize how certain mechanics work and to simplify otherwise complicated programming tasks. APIs are very significant in the world of HTML5 and there a number of them to explore, including Drag and Drop, Web Storage, Microdata, and Geolocation, among others.  Knowing that the official documentation for these APIs is separate from the official documentation of HTML5 is the first step toward using them. Keep in mind that unless you are comfortable with the more technical side of web development, this documentation will not be intuitive at first. However, there are code examples here for you to examine.  Drag and Drop http://developers.whatwg.org/dnd.html#dnd  Web Storage http://dev.w3.org/html5/webstorage/#introduction  Microdata http://developers.whatwg.org/links.html#microdata  Geolocation http://dev.w3.org/geo/api/spec-source.html#introduction |

### Geolocation in action

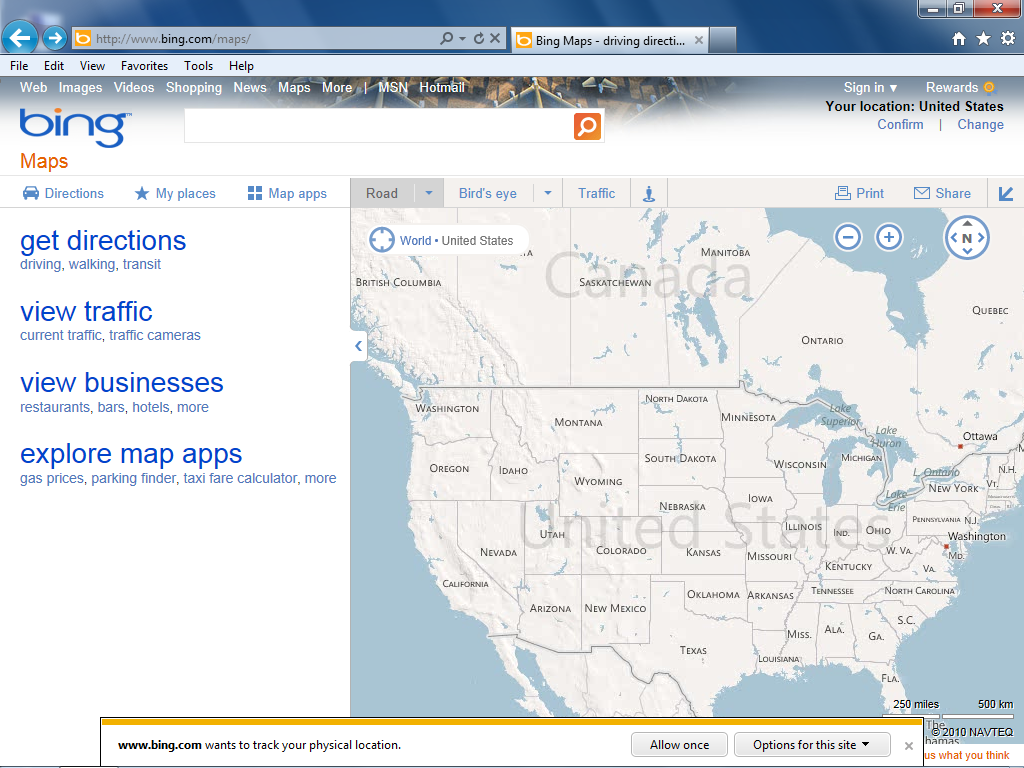
Geolocation is an API that helps identify the web browser’s geographic location. This information is then used to send you relevant data based on your position. Version 3.5 of Firefox was one of the first browsers to use geolocation. An example of the usefulness of geolocation is to connect a web search with map data to help you locate restaurants within walking distance of your hotel. Instead of manually entering your address, a geolocation-enabled browser could deliver local results based on your location.

Geolocation is currently enabled in some modern browsers. For an example of how this feature works, follow the instructions in the next procedure to visit Bing™ Maps, Microsoft’s mapping site. (You need a browser that supports geolocation in this case we are using Internet Explorer 9)

1. Open your geolocation-enabled browser and type the following URL: bing.com/maps. This page displays a map of the United States and Canada by default.
2. In the top left corner of the map, place your cursor over the viewfinder icon and it will toggle to the option “Click to center the map on your current location”.

Click on the viewfinder icon to start the geolocation process.

1. Click on the viewfinder and you will be prompted to "Allow bing.com to track your physical location." All geolocation-enabled browsers prompt users and give them a chance to accept or deny the geolocation tracking feature

In Internet Explorer 9 you are prompted to allow tracking once or choose other options.

1. To see geolocation in action, click the “Allow once” button and then click on the Viewfinder again. The map will center on your current geographical location.

For the last few years, smartphones have provided the ability of connecting geographical or GPS data with the device’s browser or a native application. When the geolocation feature becomes available to more web browsers, these benefits can be part of the desktop experience as well as on smartphone devices that support HTML5 geolocation.

## Web Workers

Web Workers is another API often grouped into the HTML5 family. Web Workers is a framework that addresses browser performance. When you access advanced web applications, such as mapping or applications that instantly generate charts and graphs, there are several processor-intensive computations that can slow down the performance of your application. Much of the slowdown occurs because there is competition between user interaction (clicking, dragging, and so on) and the need for the application to access resources (graphics, data, and more).

Web Workers are scripts that run in a separate thread. This separation means that processes, such as obtaining data from a database, occur independently from user behavior, which creates a more seamless environment for the user. Web Workers are in an early phase of development; at the time of this writing, there has been relatively little browser support.

### Web storage

Web storage is an example of using pre-existing models of web technology in more powerful ways. Web storage improves browser cookies. Currently, browser cookies let websites store small pieces of data on your system so the site can save information for later use; for example, sites that recognize your login information when you return.

Cookies are a limited technology, and are not particularly easy for designers to use. Web Storage updates the model to allow for greater storage space for modern Web Applications and makes the addition of storage features more accessible.

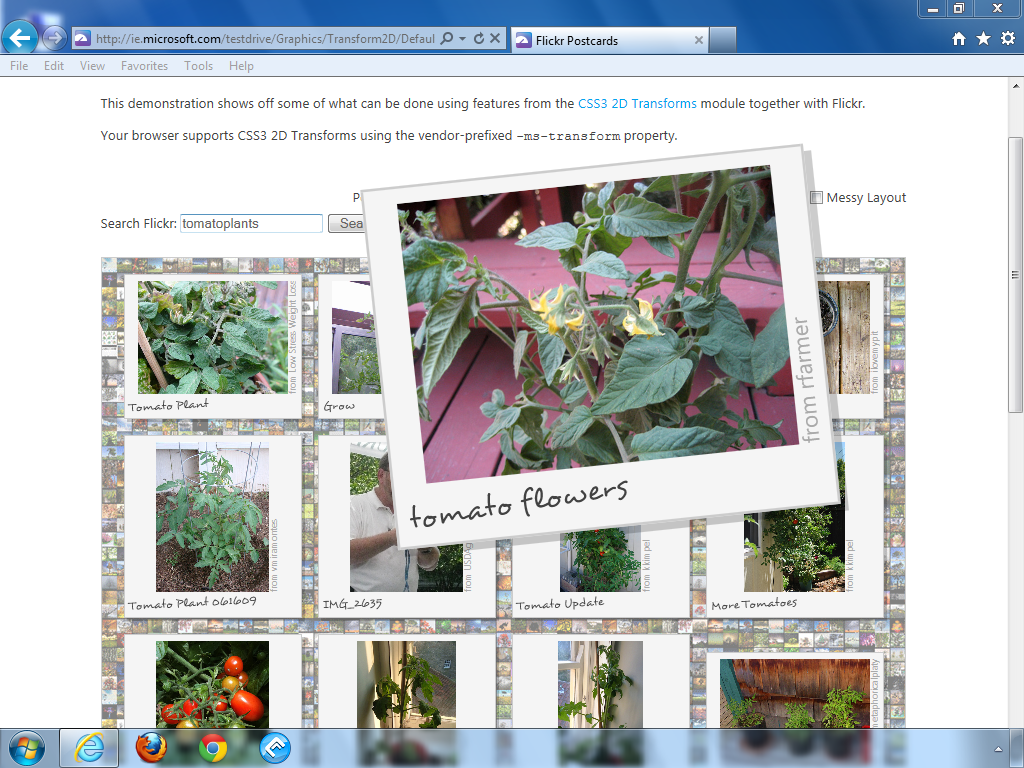
HTML5 provides two ways to store data: localStorage and sessionStorage. Data saved as localStorage is available to the browser at any point, even when the browser closes or the system is restarted. Data saved as sessionStorage is lost when the browser is closed.

An example of how Web Storage is in use today is for the storage of data in some smartphone mobile devices. Mobile browsers, for example, could write data in one browsing session and allow the user to read and/or modify this data even if they do not have a network connection.

## CSS3 is not part of HTML5, but is closely related

As noted earlier, many of the examples labeled as HTML5 are a combination of the HTML5 technologies described above as well as JavaScript or CSS. CSS (Cascading Style Sheets) is an evolving language identified by version numbers, and the latest version of CSS (3.0) has evolved alongside the HTML5 specification. Some components of CSS3 are frequently mistaken as components of HTML5, such as animation or transitions. The confusion is understandable but it is important for professional designers and developers to understand the distinction. Below is a brief description of some of the most relevant CSS3 features.

## CSS 2D- and 3D-transformations

The CSS transform property allows you to rotate, scale, or skew an element on your page. An example is the ability to rotate an image on the page slightly to one side for aesthetic effect. You can also animate transforms; for example, animating the scale property lets you create an enlarging or minimizing effect on an image or other element. You can also add the perspective property to the transformation effect to simulate an object positioned or animated in 3D space.

An example of a 2D-CSS transformation located on ie.microsoft.com/testdrive.

## CSS3 backgrounds, borders, RGBa colors, gradients, drop shadows, and rounded corners

There are several enhancements to the visual style of the page that are now possible with CSS3. A simple example is the border-radius property which allows you to add rounded corners to your boxes, but there are many other new effects you can create, such as native gradients and drop shadows. Traditional effects, such as the background-image and the border property, are improved in CSS3. For example, you can use the border-image property to apply images to style a border, or add multiple background images to a single container; this removes the current limitation of a single background-image. RGBa colors are another new CSS3 feature, the “a” stands for alpha or transparency. With RGBa colors, you now have the ability to create transparent effects for any element on your page.



Creating rounded corners with the border-radius property is just one of many new styling options in CSS3.

## **@font-face** Web fonts

There is increasing support for the ability to add custom fonts to page designs using the @font-face property, which lets the designer specify a particular font and provide a source link for the font to allow the browser to download it. This feature could dramatically transform the appearance of web pages worldwide, but it has many of the same browser support issues as the various HTML5 features.

See the potential of how web fonts can be used at www.lostworldsfairs.com/atlantis.

## CSS Media Queries

The way users interact with the Internet largely takes place through screens and the size of screens on the marketplace is dramatically different. A large computer monitor might have a screen resolution 2000 pixels wide and a smartphone might have a screen 320 pixels wide. The challenge of how to present a similar experience on these two screens can be handled by CSS3 media queries. The basic idea is that a media query can detect the type of screen a user has and then send them a specific style optimized for that screen. For example, the 2000 pixel website might support a four or even five column website whereas the 320 pixel screen would send the user styles that only use a single column.

## CSS animations

Technologies such as Flash or Silverlight have traditionally been used to animate objects in the browser. You can now create some of the same functionality using CSS rules and properties. In the future, the HTML5 Canvas element and CSS3 transitions can help designers create interactive and animated elements on the page. CSS animations and the next topic, CSS transitions, will likely be supported in most future web browsers. At the time of this writing, they are only supported in Webkit-based browsers. Similarly, there are some future HTML5 technologies not yet supported in Webkit browsers because these technologies are still emerging.

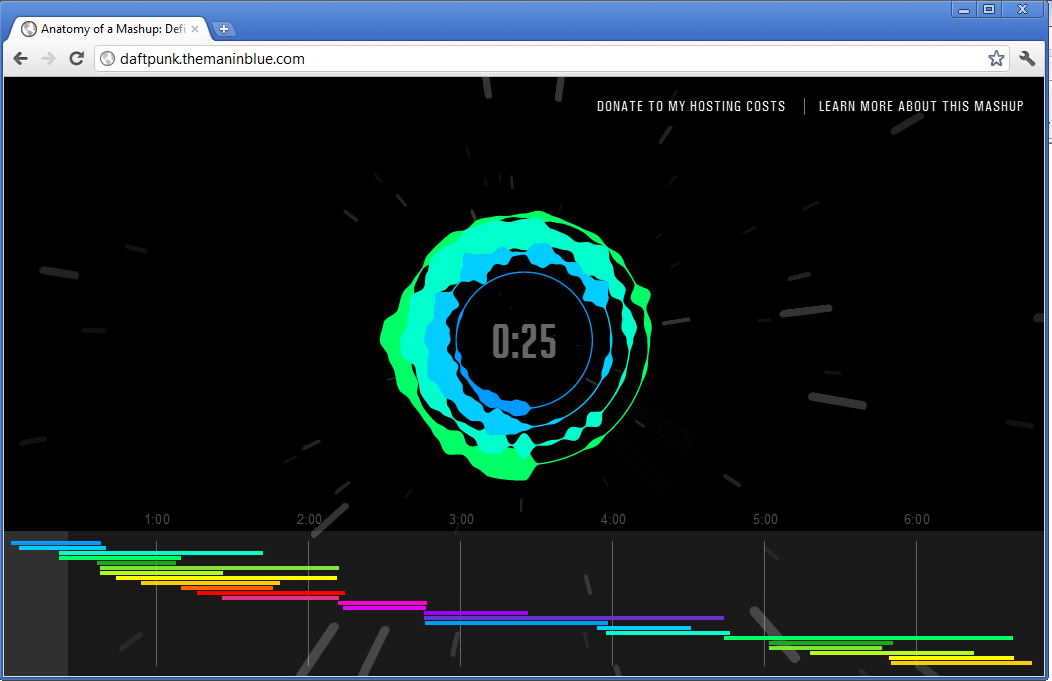


Visit http://animatable.com/demos/madmanimation/ in a Webkit browser, such as Chrome or Safari, to see CSS animation in action.

## CSS transitions

CSS transitions are closely related to animation, but fundamentally different. A transition allows property changes in CSS values to occur smoothly over a specified duration. For example, a button that has a background color of green can smoothly animate to a different background color when the user rolls over the button.

You can currently perform this type of animation with JavaScript and Flash, but as with much of CSS3, transitions give designers a tool to use without becoming a scripting expert.



Visit http://daftpunk.themaninblue.com/ to see an example of CSS transitions in action.

## HTML5 is in a state of transition

The technologies behind HTML5 are in transition, so you need to determine when you can use them and when you should not. Throughout this book, we will guide you and offer a perspective on the kind of support you can expect in web browsers, and provide scenarios where HTML5 might be more appropriate to use than other languages.

Each of the major browsers in use today (Microsoft Internet Explorer, Mozilla Firefox, Apple Safari, Google Chrome, and Opera) have different support for HTML5 features in the syntax and the supporting family. In some cases, a page that has new functionality or appearance in one browser might not appear at all in another, or features might be missing, but the page continues to be functional. These scenarios might change in the future, but desktop web browsers evolve very slowly, so there will continue to be inconsistent browser support in the near future.

| The timeline for browser support |
| --- |
| The timeline for full HTML5 browser support cannot be accurately predicted, but the W3C has targeted the second half of the year 2014 as the date when the HTML5 specification will be finalized. Conservative web designers might choose to wait until then to use HTML5 in production environments, but the standard is separate from browser support. Most current web browsers support some HTML5 features. Some of these features are relatively well-developed and safe to use, others are still under development and designers can use them to experiment with the new features, or with the understanding that they might be exchanging reliability for innovation. |

## Who is using HTML5 today?

This is a difficult question to answer because as you have seen, there is no single definition for “HTML5”. There are many demos for HTML5 that would not be implemented in a production environment due to lack of consistent browser support. If you include the styling features of CSS3, there are many business and personal sites that use the more decorative aspects such as border-radius or the transform features. In these situations, designers will use these features and take extra care to ensure that their pages are not substantially different in non-supported browsers.

One area where the use of HTML5 features is active is Mobile devices and smartphones where you might find web applications that take advantage of web storage or CSS animations. Devices such as the iPhone and the iPad can do this because there are distinctive hardware profiles including screen size and memory capabilities as well as a consistent software profile (only one operating system needs to be supported). Within the controlled environment of a smartphone or mobile device, it is much easier to add advanced features when you know definitively they will be supported by the default browser. In contrast, within the landscape of desktop computing, there are far too many variables to support easily, including multiple browsers, old browsers, different monitor resolutions and more.

All of the major web browsers are committed to HTML5, and by learning these skills today, you can be a part of the exhilarating (and challenging) evolution of the web.

## Future features of HTML5

As noted earlier, HTML5 features are in different levels of completion and we have touched on just a few here. The following is a list of features that are part of the HTML5 specification or related technologies that will currently be added (or in some cases have been added) to future browsers.

* **Drag and Drop**Drag-and-drop operations let users visually position elements on the screen instead of positioning elements by clicking a button. The HTML5 Drag and Drop API is based on the original Internet Explorer implementation. Some details have changed and certain browsers have an alternate syntax, but a standardized API is close to being finalized. At the time of this writing, support for certain drag-and-drop features are not shared between browsers. For example, some browsers allow selections or files from other windows or applications to bedropped into the browser, others do not.
* **The File API**The File API allows developers to access local files on client machines without the need for extensions or plug-ins. Ultimately this will provide a unified way for web applications to be able work with files that a user uploads. A web based photo-editing application for example, could be standardized across all compatible browsers using this API.
* **Flexbox layouts**CSS3 introduces a different way to create layouts called Flexible Box Layout. In this system, fluid layouts can be created without the traditional reliance on CSS floats and clears. The goal of this feature is to provide designers and developers more sophisticated and reliable tools for creating complex layouts for the web and mobile devices.
* **Multi-column and Grid layouts**Also part of CSS3, Multi-column and Grid layouts are two more enhancements to layouts on the web. Multi-column CSS allows you to easily split any existing content (a block of text for example) into two or more columns, providing an easy way to make text more readable without resorting to complicated layout tricks. Grid Layouts are somewhat related to Flexbox layouts in that this feature is designed to create more complex layouts for web pages and web applications. In the Grid layout system, rows and columns are introduced as well as ways to stack and align objects within these grids.

Although there are different levels of browser support for these and some of the other features listed in this lesson, you can still familiarize yourself further with these concepts, and depending on the browser can even build pages using the features. The book Digital Classroom HTML covers concepts and contains hands-on exercises that are not included in this book including Geolocation, CSS3 Media Queries, Offline Storage, Flexbox layouts and more.

## Identifying HTML5 Sites

In January 2010, the W3C introduced an HTML5 logo for public use to promote the new capabilities of HTML5 and related technologies.



The HTML5 logo is free   
to download and use.

The logo is available as a graphic to display on websites and other media to indicate the use of this technology. Note that the W3C uses the term HTML5 in a broad sense and includes other technologies. In addition, HTML5 will become an official standard in 2014, but web developers and designers are encouraged to start using the specification today. For more information about HTML5 and the logo as discussed by the W3C, visit: www.w3.org/html/logo/

## Self study

Browser compatibility is a major factor with HTML5. When you include the HTML5 specification, the various related APIs and CSS3, there are a tremendous number of features to consider and widely varying support between browsers. To get a better sense of this visit the following site:

<http://caniuse.com>

This site provides data tables listing the levels of browser support for specific features of HTML5. Not only is current browser support listed, but past support and projected (or planned) support for all of the major browser manufacturers. Type in each of the following HTML5 features into the Search field to get a sense of browser support for some of the features mentioned in this lesson.

* New semantic elements
* Vdeo element
* Geolocation
* Canvas
* CSS3 border-radius

Try typing in other features mentioned in this lesson to see the level of browser support.

## Review

### Questions

1. Name three components of HTML5 that provide alternatives to browser plug-ins such as Flash and Silverlight.
2. What is the difference between the HTML5 specification and the HTML5 family as defined in this lesson?
3. What is Geo-location?

### Answers

1. The <video> and <audio> tags let you embed multimedia directly in HTML without the need for a browser plug-in. The <canvas> element lets you add a drawing surface to your page and create shapes, fills, gradients, and with the help of CSS, it can animate these objects.
2. The HTML5 specification features a number of new elements (or tags) that you can use to add new layout structure or functionality to web pages. The HTML5 family is composed of several related, but independent, technologies such as Geolocation, Web Storage, and CSS3.
3. Geo-location is a web technology that can identify the physical location of the user and provide useful data such as their location on a map.