

HTML5 Unleashed: Tips, Tricks and Techniques

May 7th, 2010 by [Saud Khan](#) in [Articles](#), [Code](#), [Lists](#)

Can we use HTML5 today? What can we do with it? Is it really going to kill Flash? You must have noticed a gradual increase in the frequency of these and similar questions being asked, debated and even answered. In my opinion, you must answer such fundamental questions yourself.



The whole purpose of this article is to help you get started with some basic guidelines and easy to follow code templates. If you are familiar with the basics and want to take things to the next level, you will also find several useful resources providing more tips, tricks and techniques.

HTML5 Features

[Official Specification Document](#) is the best place to look for HTML5 features, but you can also start with simple and easy to follow [HTML5 Tag Reference](#) at W3Schools. We will cover following features in this article:

- Semantic Markup
- Form Enhancements
- Audio / Video
- Canvas
- ContentEditable
- Drag and Drop
- Persistent Data Storage

Check For Browser Support

Before you start experimenting with HTML5, you need to know level of support available for it in major web browsers. These useful resources will help you keep a track:

- [When Can I Use](#)
- [Web Designers' Browser Support Checklist](#)
- [HTML5 Test](#)
- [Comparison of Layout Engines](#)

You can also detect HTML5 feature support on the fly using JavaScript ([Guide to Detecting HTML5 Features With JavaScript](#)). You must also check [Modernizr](#): a really useful JavaScript library that detects native availability of HTML5/CSS3 features. If you prefer using MooTools, you can use [MooModernizr](#) (MooTools port of Modernizr).

You may also want to keep an eye on constantly changing "[Browser Market Share](#)" – this information will be helpful when you need to determine if any workarounds or fallbacks are necessary.

Note Changes

In addition to the new features, you should also take a note of following important changes:

- **Simple DOCTYPE**
HTML5 requires very simple and easy to remember DOCTYPE: `<!DOCTYPE html>`. It intentionally contain no version, so the DOCTYPE will remain usable for all future revisions of HTML.
- **Easy to Remember Attribute For Language**
You are not required to use `xmlns` or `xml:lang` attributes in the `<html>` tag. `<html lang="en">` will now do the trick with HTML5.
- **Easy to Remember Attribute For Character Set**
You can now define character encoding, using new "charset" attribute in the meta tag:
`<meta charset="utf-8" />`
- **Trailing Slashes Not Required**
Void elements in HTML5 (e.g. the `br`, `img` and `input` elements) don't require a trailing slash.
- **Elements Thrown Away**
Following elements are not supported in HTML5: `<acronym>`, `<applet>`, `<basefont>`, `<big>`, `<center>`, `<dir>`, ``, `<frame>`, `<frameset>`, `<noframes>`, `<s>`, `<strike>`, `<tt>`, `<u>` and `<xmp>`

Sample Code:

[?](#)

```
01 <!doctype html>
02 <html lang="en">
03   <head>
04     <meta charset="utf-8" />
05     <title>HTML5 Document</title>
06   </head>
07   <body>
```

```
07
08 </body>
09 </html>
10
```

You can use [HTML5 Validator](#) or [W3C Markup Validation Service](#) to test your HTML5 documents.

Semantic Markup

Several new elements proposed in HTML5 are meant to only add more semantic markup, and will do nothing except provide a more meaningful alternative to `<div>`. These new elements include: [<article>](#), [<section>](#), [<aside>](#), [<hgroup>](#), [<header>](#), [<footer>](#), [<nav>](#), [<time>](#), [<mark>](#), [<figure>](#), and [<figcaption>](#).

These elements are available in all modern browsers (Firefox 3+, Safari 3.1+, Chrome 2+, and Opera 9.6+), except for Internet Explorer. JavaScript offers an easy solution with

"`document.createElement(tagName)`"; that you can use to create new HTML5 elements in IE.

Instead of creating these elements yourself, you can also use [HTML5 Enabling Script](#) or [IE Print Protector](#) – these scripts will also help IE render HTML5 elements correctly in print.

You may also want to apply CSS reset to these new elements. Here are few CSS resets that you can use in your HTML5 based projects:

- [HTML5 Reset CSS](#)
- [Reset5](#)

Sample Code: HTML5 Page Layout With IE Support.

?

```
01 <!doctype html>
02 <html lang="en">
03   <head>
04     <meta charset="utf-8" />
05     <title>HTML5 Semantic Markup Demo: Cross Browser</title>
06     <link rel="stylesheet" href="html5reset.css" type="text/css" />
07     <link rel="stylesheet" href="html5semanticmarkup.css" type="text/css" />
08     <!--[if lt IE 9]>
09       <script src="html5.js"></script>
10     <![endif]-->
11   </head>
12   <body>
13     <header>
14       <hgroup>
15         <h1>Page Header</h1>
16         <h2>Page Sub Heading</h2>
17       </hgroup>
18     </header>
19
20     <nav>
21       <ul>
22         <li><a href="#">Home</a></li>
23         <li><a href="#">Projects</a></li>
24         <li><a href="#">Portfolio</a></li>
25         <li><a href="#">Profile</a></li>
26         <li><a href="#">Contact</a></li>
27       </ul>
28     </nav>
```

```

24
25     <article>
26         <header>
27             <h1>Article Heading</h1>
28             <time datetime="2010-05-05" pubdate>May 5th, 2010</time>
29         </header>
30         <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
31         <section>
32             <header>
33                 <h1>Section Heading</h1>
34             </header>
35             <p>Ut sapien enim, porttitor id feugiat non, ultrices non odio.</p>
36             <footer>
37                 <p>Section Footer: Pellentesque volutpat, leo nec auctor
38                 euismod</p>
39             </footer>
40         </section>
41         <section>
42             <header>
43                 <h1>Section Heading</h1>
44             </header>
45             <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
46             <figure>
47                 
48                 
49                 
50                 
51                 <figcaption>FigCaption: Club, Heart, Spade and
52                 Diamond</figcaption>
53             </figure>
54             <p>Ut sapien enim, porttitor id feugiat non, ultrices non odio</p>
55             <footer>
56                 <p>Section Footer: Pellentesque volutpat, leo nec auctor euismod
57                 est.</p>
58             </footer>
59         </section>
60         <footer>
61             Article Footer
62         </footer>
63     </article>
64
65     <aside>
66         <header>
67             <h1>Sidebar Heading</h1>
68         </header>
69         <p>Ut sapien enim, porttitor id feugiat non, ultrices non odio.</p>
70
71     </aside>
72
73     <footer>
74         Page Footer
75     </footer>
76
77 </body>
78 </html>
79

```

Note: Not a single “div” element, “id” or “class” attribute is used – Clean, Lean and more Meaningful Markup ([View Demo](#)). You should still be able to validate your document using [HTML5 Validator](#) or [W3C Markup Validation Service](#).

PAGE HEADER

Page Sub Heading

Home Projects Portfolio Profile Contact

Article Heading

May 5th, 2010

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ornare faucibus sem non porta. Nullam tempor dolor vitae quam placerat porttitor. Sed consequat dignissim lectus. Maecenas mi eros, venenatis auctor et, faucibus varius lectus. In elementum sem vitae velit euismod a facilisis elit pulvinar. Nunc eget magna elit. Nam et quam erat, a luctus urna. Pellentesque volutpat, leo nec auctor euismod, lorem arcu dapibus augue, ornare lacinia purus mi cursus est. Praesent vitae lacus id lorem vulputate lobortis vel vitae quam. Integer quis iaculis sapien. Donec sit amet massa nunc, rhoncus condimentum turpis. Duis hendrerit nisl et lorem elementum accumsan accumsan vitae dolor. Donec et sem a erat facilisis dignissim. Aenean a tempor metus.

Section Heading

Ut sapien enim, porttitor id feugiat non, ultrices non odio. Donec sit amet augue metus, nec sollicitudin est. Donec iaculis porttitor viverra. Sed sed magna ac lectus pharetra iaculis at quis tellus. Nunc quis lorem sit amet nulla viverra mattis. Aenean suscipit libero accumsan sapien fermentum non aliquet lorem rutrum.

Section Footer: Pellentesque volutpat, leo nec auctor euismod, lorem arcu dapibus augue, ornare lacinia purus mi cursus est.

Section Heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ornare faucibus sem non porta. Nullam tempor dolor vitae quam placerat porttitor. Sed consequat dignissim lectus. Maecenas mi eros, venenatis auctor et, faucibus varius lectus. In elementum sem vitae velit euismod a facilisis elit pulvinar. Nunc eget magna elit. Nam et quam erat, a luctus urna. Pellentesque volutpat, leo nec auctor euismod, lorem arcu dapibus augue, ornare lacinia purus mi cursus est. Praesent vitae lacus id lorem vulputate lobortis vel vitae quam. Integer quis iaculis sapien. Donec sit amet massa nunc, rhoncus condimentum turpis. Duis hendrerit nisl et lorem elementum accumsan accumsan vitae dolor. Donec et sem a erat facilisis dignissim. Aenean a tempor metus.



FigCaption: Club, Heart, Spade and Diamond

Ut sapien enim, porttitor id feugiat non, ultrices non odio. Donec sit amet augue metus, nec sollicitudin est. Donec iaculis porttitor viverra. Sed sed magna ac lectus pharetra iaculis at quis tellus. Nunc quis lorem sit amet nulla viverra mattis. Aenean suscipit libero accumsan sapien fermentum non aliquet lorem rutrum.

Section Footer: Pellentesque volutpat, leo nec auctor euismod, lorem arcu dapibus augue, ornare lacinia purus mi cursus est.

Article Footer

Page Footer

Sidebar Heading

Ut sapien enim, porttitor id feugiat non, ultrices non odio. Donec sit amet augue metus, nec sollicitudin est. Donec iaculis porttitor viverra. Sed sed magna ac lectus pharetra iaculis at quis tellus. Nunc quis lorem sit amet nulla viverra mattis. Aenean suscipit libero accumsan sapien fermentum non aliquet lorem rutrum.

Note: Our demo will not render correctly in IE6. This is only because I am using CSS child combinators to avoid use of additional classes. You can render HTML5 markup in IE6 like in any other browsers, using your own CSS.

Additional Resources

- [HTML5 "articles" and "sections": What's The Difference?](#)
- [Structural Tags in HTML5](#)
- [The Figure & Figcaption Elements](#)

- [HTML5: <nav> Ambiguity Resolved](#)
- [<section> is not just a "semantic <div>"](#)
- [Bulletproof HTML5 <details> fallback using jQuery](#)
- [Styling HTML5 markup in IE without script](#)

Form Enhancements

HTML5 provides several new attributes, input types and elements for [web forms](#). As of today, only Opera provides decent support for HTML5 forms. You must therefore download Opera to see how most of the new features will work.

- **New Input Types**
[color](#), [email](#), [date](#), [month](#), [week](#), [time](#), [datetime](#), [datetime-local](#), [number](#), [range](#), [search](#), [tel](#), and [url](#)
- **New Attributes**
[required](#), [autofocus](#), [pattern](#), [list](#), [autocomplete](#) and [placeholder](#)
- **New Elements**
[<keygen>](#), [<datalist>](#), [<output>](#), [<meter>](#) and [<progress>](#)

The good news is that even with very limited support, using some of these features may not be a bad idea. This is because new input types will degrade nicely to regular text fields. And remember you can always fallback on JavaScript based form controls that you use today (the trick is to detect for native support first and then use a fallback).

Sample Code: Showcasing some new features that you can test today.

?

```
01 <form>
02   <fieldset>
03     <legend>New Attributes</legend>
04     <p>
05       <label>Required:</label>
06       <input type="text" name="html5required" required="true">
07       <small>Works in Opera & Chrome</small>
08     </p>
09     <p>
10       <label>AutoFocus:</label>
11       <input type="text" name="html5autofocus" autofocus="true">
12       <small>Works in Opera, Chrome & Safari</small>
13     </p>
14     <p>
15       <label>PlaceHolder:</label>
16       <input type="text" name="html5placeholder" placeholder="This Will Show in">
17       <small>Works in Chrome & Safari</small>
18     </p>
19     <p>
20       <label>Input Pattern:</label>
21       <input type="text" pattern="[0-9]{3}" name="html5pattern" required t
22       uppercase letters"/>
23       <small>Works in Opera & Chrome</small>
24     </p>
25     <p>
26       <label>Multiple Files:</label>
27       <input type="file" name="html5multiplefileupload" multiple>
```

```

23         <small>Works in Chrome, Safari & Firefox</small>
24     </p>
25     <p>
26         <label>List:</label>
27         <input type="text" name="html5textwithdatalist" list="colors">
28         <datalist id="colors">
29             <option value="Red">
30             <option value="Green">
31             <option value="Blue">
32         </datalist>
33         <small>Works in Opera</small>
34     </p>
35 </fieldset>
36
37 <fieldset>
38     <legend>New Input Types</legend>
39     <p>
40         <label>Email:</label>
41         <input type="email" name="html5email">
42         <small>Works in Opera</small>
43     </p>
44     <p>
45         <label>URL:</label>
46         <input type="url" name="html5url">
47         <small>Works in Opera</small>
48     </p>
49     <p>
50         <label>Number:</label>
51         <input type="number" name="html5number" min="1" max="10" step="1" value="1">
52         <small>Works in Opera</small>
53     </p>
54     <p>
55         <label>Range:</label>
56         <input type="range" name="html5range" min="-100" max="100" value="0" step="1">
57         <small>Works in Opera, Chrome & Safari</small>
58     </p>
59     <p>
60         <label>Time:</label>
61         <input type="time" step="900" name="html5time">
62         <small>Works in Opera</small>
63     </p>
64     <p>
65         <label>Date:</label>
66         <input type="date" name="html5date">
67         <small>Works in Opera</small>
68     </p>
69     <p>
70         <label>Month:</label>
71         <input type="month" name="html5month">
72         <small>Works in Opera</small>
73     </p>
74     <p>
75         <label>Week:</label>
76         <input type="week" name="html5week">
77         <small>Works in Opera</small>
78     </p>
79     <p>
80         <label>DateTime:</label>
81         <input type="datetime" name="html5datetime">
82         <small>Works in Opera</small>
83     </p>
84 </fieldset>

```

```
73     </fieldset>
74
75     <div><button>Submit</button></div>
76 </form>
77
```


New Attributes

Required:

Works in Opera & Chrome

AutoFocus:

Works in Opera, Chrome & Safari

Placeholder:

Works in Chrome & Safari

Input Pattern:

Works in Opera & Chrome

Multiple Files:

Works in Chrome, Safari & Firefox

List:

Works in Opera

New Input Types

Email:

Works in Opera

URL:

Works in Opera

Number:

Works in Opera

Range:

Works in Opera, Chrome & Safari

Time:

Works in Opera

Date:

Month:

		May 2010						
Week	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
17	26	27	28	29	30	1	2	
18	3	4	5	6	7	8	9	
19	10	11	12	13	14	15	16	
20	17	18	19	20	21	22	23	
21	24	25	26	27	28	29	30	
22	31	1	2	3	4	5	6	
Today				None				

Week:

Date Time:

Most of the features showcased in our demo will work in Opera, but you will also need to use Chrome or Safari to see few remaining features in action ([View Demo](#)).

Learn More:

- [A Form of Madness](#)
- [The Future of The Web: How We'll Create Form in HTML5](#)
- [HTML5 Input Types](#)
- [Have a Field Day with HTML5 Forms](#)
- [Rethinking Forms in HTML5](#)
- [HTML5 Forms Are Coming](#)

Tools & Resources:

- [Cross Browser HTML5 Placeholder Using jQuery and Modernizr](#)
- [Cross Browser HTML5 Date Input Type, Range Input Type and Validations Using jQuery Tools](#)
- [jQuery HTML5 File Upload: Asynchronously Uploads Multiple Files](#)
- [jQuery Plugin to Enable Web Forms 2.0](#)
- [Web Forms 2.0 Cross-Browser Implementation](#)

Audio & Video Without Plugins

HTML5 specification in regards to audio and video is one of its most talked about feature. In addition to the obvious benefits of having native audio and video support, this debate is mainly focused on differences in audio / video formats preferred by browser vendors. If you are going to use HTML5 `<audio>` and `<video>`, it is important that you get yourself familiar with following audio/video codecs and know which browser will support what:

- **Audio:** [ogg \(ogg, oga\)](#), [mp3](#), [wav](#), [AAC](#)
- **Video:** [ogg \(ogv\)](#), [H.264 \(mp4\)](#)

You should also keep an eye on Google's [VP8](#) video codec, which it may offer as open source to end the controversy. HTML5 offers a solution by allowing you to specify more than one source file (in different formats), so that user's browser will play the file it understands. For Internet Explorer and older browsers you will need to fallback on current solutions.

When you first start experimenting with HTML5 audio / video, you may want to note couple of things that may be helpful:

- Your web server must support audio / video MIME types for the file formats that you intend to serve. You will specially need to check this if you are testing HTML5 audio/video on your local web server.
- If you are using Safari to test HTML5 audio / video, you must install QuickTime player as well. Safari can not play anything without QuickTime.

Sample Code: Audio Markup ([View Demo](#))

?

```
1 <audio controls>
2   <source src="demo-audio.ogg" />
3   <source src="demo-audio.mp3" />
4 </audio>
```



Sample Code: Video Markup ([View Demo](#))

?

```

1 <video width="320" height="240" controls preload="none" poster="videoframe.jpg">
2   <source src="demo-video.mp4" type="video/mp4" />
3   <source src="demo-video.ogv" type="video/ogg" />
4 </video>

```



Although a big step from having to depend on third party plugins, the above examples are just a starting point. Since each browser provide its own look and feel for controls, you may want to change them to match your design . Following is a simple example of how to use DOM API to create your own custom controls.

Sample Code: Video With Custom Controls ([View Demo](#))

?

```

01
02 <video width="320" height="240" preload="none" poster="videoframe.jpg">
03   <source src="demo-video.mp4" type="video/mp4" />
04   <source src="demo-video.ogv" type="video/ogg" />
05 </video>
06
07 <script>
08   var video = document.getElementsByTagName('video')[0];
09   function toggleMute() {
10     video.muted = !video.muted;
11   }
12 </script>
13 <p>
14   <a href="JavaScript:video.play();">Play</a> |
15   <a href="JavaScript:video.pause();">Pause</a> |
16   <a href="JavaScript:toggleMute();">Sound On/Off</a>
17 </p>

```



[Play](#) | [Pause](#) | [Sound On/Off](#)

You can still do a lot more with DOM API for audio and video. But you can take things to an entirely different level if you are willing to mix things a little bit. For example, you can place your video on HTML5 canvas element. This will allow you to read pixel data, manipulate or process frames from a video and do some awesome stuff (See for yourself: [Blowing up HTML5 Video](#), [Ambilight for HTML5 Video](#)).

Learn More:

- [Everything You Need to Know About HTML5 Video and Audio](#)
- [Embed Audio and Video in HTML5 Pages](#)
- [Native Audio in the browser](#)
- [Video in a Flash \(Without That Other Thing\)](#)
- [HTML5Video.org](#)
- [Video Type Parameters](#)

Cross Browser Solutions:

- [SublimeVideo](#)
- [Kaltura HTML5 Media Library](#)
- [Video for Everybody](#)
- [jPlayer: jQuery Audio Player](#)
- [Projekktor](#)
- [YUI HTML5 Player](#)
- [OIPlayer jQuery Plugin](#)
- [Degradable HTML5 audio and video Plugin](#)
- [Xiph QuickTime Component to make Ogg work in Safari](#)

JavaScript Interfaces

- [CwVideo: MooTools plugin to handle the HTML5 video](#)

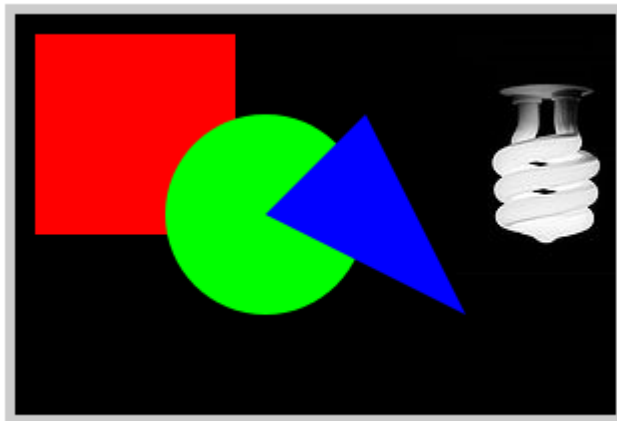
Encoding/Conversion Tools:

- [Miro Video Converter](#)
- [SUPER](#)
- [MiniCoder](#)
- [HandBrake](#)
- [ffmpeg2theora](#)

- [Firefogg](#)

2D Drawing With JavaScript

One of the most exciting feature of HTML5 is the [<canvas>](#) element – which can be used to draw graphics. Inserting a canvas element in your page is as simple as adding any other markup, but you will need to have some programming experience to create basic shapes, graphs, animations, games, image composition, etc.



Canvas is available in all modern browsers (Firefox 3, Safari 3.1, Chrome 2, and Opera 9.6), except for Internet Explorer. You can use [ExplorerCanvas](#), a JavaScript solution to add this feature in Internet Explorer.

Sample Code: Command Based 2D Drawing on HTML5 Canvas ([View Demo](#)).

?

```
01 <!doctype html>
02 <html lang="en">
03 <head>
04     <meta charset="utf-8" />
05     <title>HTML5 Canvas Demo</title>
06     <link rel="stylesheet" href="html5reset.css" type="text/css" />
07     <link rel="stylesheet" href="html5simple.css" type="text/css" />
08     <!--[if lt IE 9]>
09         <script src="html5.js"></script>
10         <script src="excanvas.js"></script>
11     <![endif]-->
12     <script type="text/javascript">
13         function draw() {
14             var canvas = document.getElementById('mycanvas');
15             if (canvas.getContext){
16                 var ctx = canvas.getContext('2d');
17
18                 // Draw Rectangle
19                 ctx.fillStyle = "rgb(255,0,0)";
20                 ctx.fillRect (10, 10, 100, 100);
21
22                 // Draw Circle
23                 ctx.fillStyle = "rgb(0,255,0)";
24                 ctx.beginPath();
```

```

21         ctx.arc(125,100,50,0,Math.PI*2,true);
22         ctx.fill();
23         ctx.closePath();
24
25         // Draw Custom Shape With Lines
26         ctx.fillStyle = "rgb(0,0,255)";
27         ctx.beginPath();
28         ctx.moveTo(125,100);
29         ctx.lineTo(175,50);
30         ctx.lineTo(225,150);
31         ctx.fill();
32         ctx.closePath();
33
34         // Draw Image From External File
35         var myImage = new Image();
36         myImage.onload = function(){
37             ctx.drawImage(myImage, 220, 10);
38         }
39     }
40     </script>
41     <style type="text/css">
42         canvas {
43             border: 5px solid #ccc;
44             background: #000;
45         }
46     </style>
47 </head>
48 <body onload="draw();" >
49     <header>
50         <h1>HTML5 Canvas Demo</h1>
51     </header>
52
53     <figure>
54         <canvas id="mycanvas" width="300" height="200">Fallback content, in case the
55 Canvas.</canvas>
56         <figcaption>Works in Firefox 3+, Safari 3.1+, Chrome 2+ and Opera 9.6+</fi
57     </figure>
58 </body>
59 </html>

```

Learn More:

- [HTML5 Canvas – The Future of Graphics on the Web](#)
- [Canvas Tutorial at Mozilla Developer Center](#)
- [How to Draw with HTML 5 Canvas](#)
- [HTML5 Canvas The Basics \(Opera Developer Community\)](#)
- [Let's Call It a Drawing Surface](#)

Tools & Resources:

- [HTML5 Canvas Cheat Sheet](#)
- [jQuery Visualize Plugin: Accessible Charts & Graphs from Table Elements](#)
- [Doodle.js: Sprite Library for Canvas](#)
- [Processing.js: Port of the Processing Visualization Language](#)

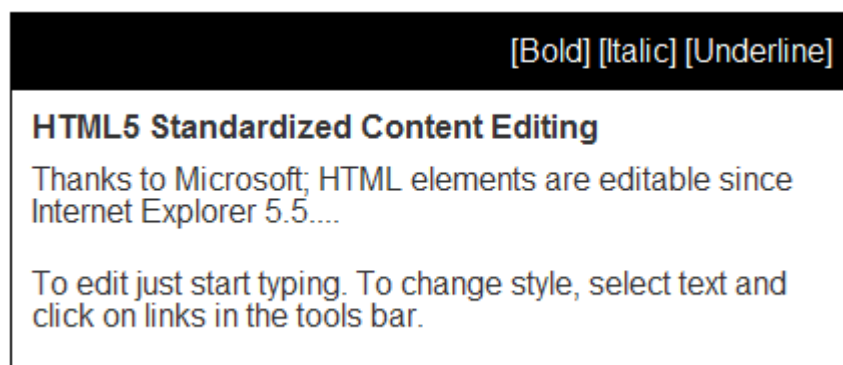
- [Cartagen: Framework For Rendering Maps in HTML5](#)
- [RGraph: HTML5 Canvas Graph Library](#)

Examples & Applications:

- [Chrome Experiments](#)
- [Harmony: Procedural Drawing Tool](#)
- [Sketchpad: Online Paint/Drawing Application](#)
- [CanvasPaint: Copy of Microsoft Paint](#)

Making Content Editable

If you want to allow users to edit content some where in your page, all you need to do is set `contenteditable` attribute of the container element. You have probably seen it in action in web based WYSIWYG editors. This attribute is part of HTML5 specifications, and is supported by all major browsers (Internet Explorer 5.5+, Firefox 3+, Safari 3.1+, Chrome 2+, and Opera 9.6+). Note that setting `contenteditable="true"` will allow users to edit, delete and insert content, but it will not automatically provide any controls like you see in WYSIWYG editors to save changes or apply styles. You will need to use JavaScript to add these controls yourself.



Sample Code: Basic content editor with bold, italic and underline controls ([View Demo](#)).

?

```

01 <!doctype html>
02 <html lang="en">
03   <head>
04     <meta charset="utf-8" />
05     <title>HTML5 ContentEditable Demo</title>
06     <link rel="stylesheet" href="html5reset.css" type="text/css" />
07     <link rel="stylesheet" href="html5simple.css" type="text/css" />
08     <!--[if lt IE 9]>
09       <script src="html5.js"></script>
10     <![endif]-->
11     <script language="javascript">
12       function editStyle(cmd) {
13         document.execCommand(cmd, null, null);
14       }
15     </script>
16   </head>
17   <body onload="draw();" >

```

```

16     <header>
17         <h1>HTML5 ContentEditable Demo</h1>
18     </header>
19
20     <div id="democontainer">
21         <div id="editingcontrols">
22             <a href="#" onclick="editStyle('bold');">[Bold]</a>
23             <a href="#" onclick="editStyle('italic');">[Italic]</a>
24             <a href="#" onclick="editStyle('underline');">[Underline]</a>
25         </div>
26         <div id="editor" contenteditable="true">
27             <h2>HTML5 Standardized Content Editing</h2>
28             <p>Thanks to Microsoft; HTML elements are editable since Internet Explorer 11.
29             <p>To edit just start typing. To change style, select text and click on
30             tools bar.</p>
31         </div>
32     </div>
33
34 </body>
35 </html>

```

Learn More:

- [The Road to HTML5: contentEditable](#)
- [How to build a desktop WYSIWYG editor with WebKit and HTML5](#)

Tools & Resources:

- [QuirksMode: execCommand Compatibility Table](#)
- [Internet Explorer commands](#)
- [Mozilla Firefox commands](#)
- [List of Free Web Based HTML Editors](#)

Drag and Drop

HTML5 “[Drag and Drop](#)” will take user interaction to a whole new level and will have major impact on how you design user interfaces. As of today [Firefox 3.5+](#) provides best support for this feature, and other browsers are cautiously following with only partial support (Opera provides no support at all). Unfortunately due to differences in vendor implementation at this time, you will need to heavily rely on JavaScript and CSS to make it work cross browser.

Note:

- Image and Hyperlink elements can be dragged by default. For all other elements, HTML5 introduced a new attribute “**draggable**”; which needs to be set in order to make the element draggable.
- Following events are associated with Drag and Drop: **dragstart**, **drag**, **dragenter**, **dragover**, **dragleave**, **drop** and **dragend**. You will need to write function (event handlers) that will take care of what you need to do when these events are fired. You may attach event listeners to the elements and/or do event delegation to simplify your code.
- Drag and Drop events also let you transfer data along with the dragged element, using **getData** and **setData** methods of event’s “**dataTransfer**” property.

- You can perform drag and drop between different browser windows, and even other applications.

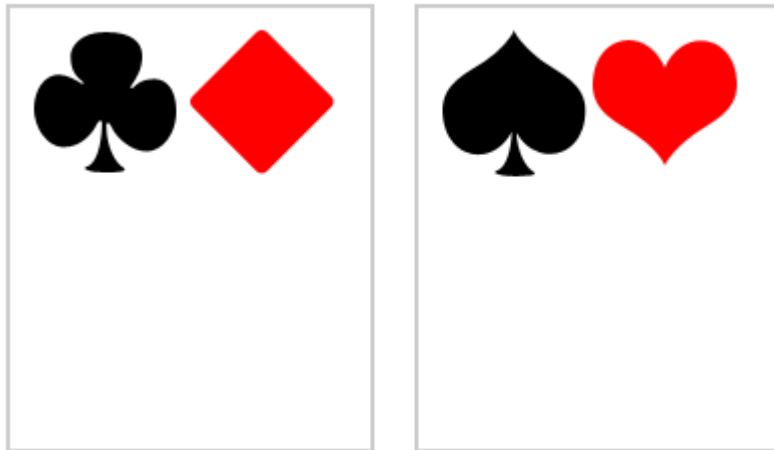
Sample Code: Drag and Drop images from one container to another ([View Demo](#)).

?

```

01
02 <!doctype html>
03 <html lang="en">
04 <head>
05     <meta charset="utf-8" />
06     <title>HTML5 Drag & Drop Demo</title>
07     <link rel="stylesheet" href="html5reset.css" type="text/css" />
08     <link rel="stylesheet" href="html5simple.css" type="text/css" />
09     <!--[if lt IE 9]>
10         <script src="html5.js"></script>
11     <![endif]-->
12
13     <script>
14         function DragHandler(target, e) {
15             e.dataTransfer.setData('Text', target.id);
16         }
17
18         function DropHandler(target, e) {
19             var id = e.dataTransfer.getData('Text');
20             target.appendChild(document.getElementById(id));
21             e.preventDefault();
22         }
23     </script>
24 </head>
25 <body>
26     <header>
27         <h1>HTML5 Drag & Drop Demo</h1>
28     </header>
29     <div id="dndcontainer">
30         <div ondrop="DropHandler(this, event)" ondragenter="return false" ondragover="return false" style="border: 1px solid black; padding: 10px; min-height: 100px;">
31             
32             
33             
34             
35         </div>
36
37         <div ondrop="DropHandler(this, event)" ondragenter="return false" ondragover="return false" class="dndbox"></div>
38
39         <div id="demonotes">Drag and drop images from one container to another. Works in
40         Opera.</div>
41     </div>
42 </body>
43 </html>

```



Learn More:

- [Cross Browser HTML5 Drag and Drop](#)
- [HTML5 Drag and Drop in Firefox 3.5](#)
- [Mozilla Developer Center: Drag and Drop](#)
- [Native Drag and Drop](#)
- [HTML5 Drag and Drop + Microformats = a Whole World of Possibilities](#)
- [The HTML5 Drag and Drop Disaster](#)

JavaScript Solutions

- [jQuery Plugin for HTML5 Native Drag and Drop](#)
- [Drag and Drop With Simple UI Kit](#)

Persistent Data Storage

[Web Storage](#) specifications provides two new attributes for persistent data storage similar to HTTP session cookies. These are "sessionStorage" and "localStorage".

- **sessionStorage** is used to save data during lifetime of top-level browsing context (i.e., as long as its browser tab or window remains open).
- **localStorage** is used to store data for longer period of time, over multiple pages and browser sessions (i.e., will last even when you restart your web browser and / or computer).

Both local and session storage features are supported in following web browsers: IE8+, Firefox 3.5+, Safari 4.0+, Chrome 2.0+ and Opera 10.5+.

Sample Code: Using localStorage to create a page counter ([View Demo](#)).

?

```
01 <p>You Have Viewed This Page <b>
02 <script>
03     if (!localStorage.pageCounter)
04         localStorage.setItem('pageCounter',0);
05     localStorage.setItem('pageCounter',parseInt(localStorage.pageCounter)+1);
06     document.write(localStorage.pageCounter);
07 </script>
08 </b> Time (s) .</p>
```

```

07
08 <p><input value="Clear localStorage" type="button"
09   onClick="localStorage.clear();location.reload(true);" />
10 <input value="Reload Page" type="button" onClick="location.reload(true);" /></p>
11

```

You can also test sessionStorage by replacing all instances of "localStorage" with "sessionStorage" in the above code. Your counter will now work as long as its browser tab or window remains open. If you restart your browser, counter will start from the beginning ([View Demo](#)).

Learn More:

- [Learning About HTML5 Local Storage](#)

JavaScript Solutions:

- [realstorage: JavaScript compatibility wrapper for localStorage – provides fallback to Gears](#)
- [sessionstorage: HTML5 sessionStorage for "every" Browsers](#)
- [jQuery localStorage Plugin](#)
- [jStorage: Uses HTML5 local storage where available and userData behavior in older versions of Internet Explorer](#)

There's More

In addition to things already discussed, there are many more features and related technologies that you may want to keep track of. Here is brief list of resources to help you out:

Offline Web Applications

HTML5 [specifications](#) also define how to enable users to continue interacting with Web applications and documents, even when their network connection is down. This feature is currently supported by Firefox 3.5+, Chrome 4.0+, Safari 4.0+ and Mobile Safari 3.1+.

You can create your offline application by providing a manifest file that defines which files needs caching and which files needs to be replaced by fallbacks in offline mode. When a user visits this page, supporting web browser will fetch a copy of the manifest. It will download and cache all the resources it mentions, and if the manifest has changed since user's previous visit, it will download and cache all the resources again.

- [Safari Client-Side Storage and Offline Applications Programming Guide](#)
- [Using Offline Web Applications to offer offline capabilities](#)
- [Mozilla Hacks: Offline Web Applications](#)
- [How to Create offline Webapps on the iPhone](#)

Cross Document Messaging

New [specification](#) also provide "postMessage" method that we can use to send information from pages on different domains to each other. This feature is available in all modern browsers (IE8+, Firefox3+, Safari 4+, Chrome 2+ and Safari 9.6+).

- [Safari Reference Library: Cross-Document Messaging](#)
- [Mozilla Developer Center: postMessage](#)

Access DOM Elements By Class Name

One of the most common things that we do in JavaScript is select DOM elements and do something dynamically. Most of us now use "getElementById" method to do this in pure

JavaScript. HTML5 specifications have added "[getElementsByClassName](#)" method, that will provide major boost to performance of our scripts when trying to access elements by class name. This feature is available in all major browsers (Firefox3+, Safari 3.2+, Chrome 2+ and Safari 9.6+) except Internet Explorer.

- [getElementsByClassName Speed Comparison](#)
- [Some thoughts on HTML5's getElementsByClassName](#)

More Features

Links to Official Specifications:

- [Microdata](#)
- [Microdata Vocabularies](#)
- [Channel Messaging](#)

Related Technologies

Links to Official Specifications:

- [Web Workers](#)
- [WebSocket API](#)
- [WebSocket Protocol](#)
- [Server Sent Events](#)
- [Web SQL Database](#)
- [Geolocation API](#)
- [SVG](#)
- [MathML](#)
- [XMLHttpRequest](#)

Additional Resources

If you are still craving for more information, you should checkout these resources.

Essential Bookmarks

- [HTML5 Draft](#)
Draft Proposal by WHATWG
- [The WHATWG Blog](#)
Official blog of the WHATWG – group responsible for HTML5 specification.
- [WHATWG Wiki](#)
Place for WHATWG contributors to post and compile their own proposals and ideas regarding WHATWG specifications.
- [Dive Into HTML5](#)
Mark Pilgrim elaborates on a hand-picked selection of features from the HTML5 specification. This is a work in progress that will finally be published on paper by O'Reilly.
- [HTML5 Validator](#)
Online tool for testing your HTML5 documents.
- [Planet HTML5](#)

- [HTML5 Revision Tracker](#)
Provides an online interface for selecting and comparing revisions of the specifications.
- [HTML5 Doctor](#)
Online community focused on helping you implement HTML5 today.

Demos / Showcases

- [HTML5 Demos and Examples](#)
Collection of HTML5 experiments created by Remy Sharp.
- [HTML5 Presentation](#)
Developed by Marcin Wichary and modified by Ernest Delgado; this presentation shows HTML5 features for modern desktop and mobile browsers.
- [Information and Samples for HTML5 and related APIs](#)
Collection of HTML5 demos and samples created by Robert Nyman.
- [HTML5 Gallery](#)
A showcase of sites using HTML5 markup.

Cheat Sheets

- [HTML5 Quick Reference Guide](#)
- [HTML5 Cheat Sheet](#)
- [HTML5 Visual Cheat Sheet](#)
- [HTML5 Security Cheatsheet](#)

Frameworks

- [Less](#)
An HTML5 powered CSS framework for building flexible multi column website layouts for varying screen widths.
- [52Framework](#)
A web development framework that is aimed at enabling you to implement HTML5 and CSS3 in your projects today.
- [SproutCore](#)
A JavaScript HTML5 application framework developed as a Ruby gem.
- [Frame](#)
Frame is a modular CSS framework that supports: Layout, Typography, Forms, Code, Table, Reset, and Print styling. It is HTML5 compatible, and provides default styles and support for HTML5 elements.

General Articles & Tutorials

- [Creating Mobile Web Applications With HTML5](#)
By Michael Galpin (IBM developerWorks – May 6th, 2010)
- [Touch The Future: Create an Elegant Website With HTML5 And CSS3](#)
By Piervincenzo Madeo (PV.M Garage – April 23rd, 2010)
- [How to use HTML5 in Your Client Work Right Now](#)
By Richard Clark (HTML5 Doctor – March 30th, 2010)
- [Build Web Applications With HTML5](#)
By Michael Galpin (IBM developerWorks – March 30th, 2010)
- [How to Make an HTML5 iPhone App](#)
By Alex Kessinger (Six Revisions – March 26th, 2010)

- [Why HTML5 is Worth Your Time](#)
By Mac Slocum (O'Reilly – March 15th, 2010)
- [HTML5 Rocks My Socks Off](#)
By Shane Jeffers (Three Styles – March 11th, 2010)
- [Create Modern Web Sites using HTML5 and CSS3](#)
By Joe Lennon (IBM developerWorks – March 2nd, 2010)
- [Code a Backwards Compatible, One Page Portfolio with HTML5 and CSS3](#)
By Tom Kenny (Inspect Element – January 25th, 2010)
- [How HTML5 Will Change the Way You Use the Web](#)
By Kevin Purdy (Lifehacker – December 1st, 2009)
- [Coding A HTML 5 Layout From Scratch](#)
By Enrique Ramirez (Smashing Magazine – August 4th, 2009)
- [The Power of HTML5 and CSS3](#)
By Jeff Starr (Perishable Press – July 19th, 2009)
- [Yes, You Can Use HTML5 Today!](#)
By Bruce Lawson (SitePoint – July 1st, 2009)
- [New Elements in HTML5](#)
By Elliotte Rusty Harold (IBM developerWorks – August 7th, 2007)

Demos & Download

You can download all demos in a single ZIP file. Note: Audio / video files are not included in the download.



I hope you have enjoyed this post and found it useful. Please feel free to suggest any addition, correction or updates in your comments.

Stay Updated

If you found this post useful, consider subscribing to our [RSS Feed!](#) You can also stay updated by following us on [Twitter](#) and [Facebook](#) or by subscribing to our [FriendFeed](#).

Bookmark / Share



Read More

SIMILAR/RELATED POSTS:

- [How To Get Digg, Delicious and Tweet Counts Using jQuery](#)
- [List of Really Useful Plugins For jQuery Developers](#)
- [CSS3 Unleashed: Tips, Tricks and Techniques](#)
- [List of Really Useful JavaScript Libraries](#)
- [List of Free Web Based HTML Editors For Your CMS Project](#)