**New Tags in HTML5 and their status/usefulness**

A list of the new elements may be found at <http://w3schools.com/html5/html5_new_elements.asp>

Since some of them are not yet implemented in Firefox, and most are not implemented in IE, we will discuss only those which are currently/soon to be useful.

Almost all tags may have attributes. There are two useful sets of attributes – those linked to events (broken down as window events, forms events, etc, - See <http://w3schools.com/html5/html5_ref_eventattributes.asp> ) and a so-called standard set. See <http://w3schools.com/html5/html5_ref_standardattributes.asp>

A complete summary of changes from HTML4.01 may be found at <http://dev.w3.org/html5/html4-differences/>

An introduction to what is new in HTML5 may be found at <http://w3schools.com/html5/html5_new_elements.asp>

I have summarized what I think is most important below.

**Things you can no longer do:**

* **align** – this attribute is not supported and you must use CSS for this
* **doctypes** –you may no longer use the various doctypes from HTML4.01
You must use <!DOCTYPE HTML> (No closing tag; case insensitive!)
* **<b>** now represents a different style – not necessarily bold; use CSS
* **<i>** now represents a different presentation – not necessarily italics; use CSS
* **<img>** may be referenced with the **id** attribute; the **name** attribute will not work.. Also the **border** attribute must be included, even if set to 0.
* **<a>** may be referenced with the **id** attribute; the **name** attribute will not work. Also an <a> without an href attribute will be viewed as a placeholder for a link to be added, rather than as an anchor. There is also an anchor object and an anchors[] array which may be used to reference them.
* **frames** are gone (hence no frames or framesets elements). See <http://dev.w3.org/html5/spec/rendering.html#frames-and-framesets> for current implementation.
* **longdesc** is no longer an attribute for <img> - which may pose an accessibility problem.
* **Many attributes used for table and page layout** now must be handled with CSS. These include (per w3c reference above)
	+ **align** attribute on caption, iframe, img, input, object, legend, table, hr, div, h1, h2, h3, h4, h5, h6, p, col, colgroup, tbody, td, tfoot, th, thead and tr.
	+ **alink, link, text** and **vlink** attributes on body.
	+ **background** attribute on body.
	+ **bgcolor** attribute on table, tr, td, th and body.
	+ **border** attribute on table and object.
	+ **cellpadding** and **cellspacing** attributes on table.
	+ **char** and **charoff** attributes on col, colgroup, tbody, td, tfoot, th, thead and tr.
	+ **clear** attribute on br.
	+ **compact** attribute on dl, menu, ol and ul.
	+ **frame** attribute on table.
	+ **frameborder** attribute on iframe.
	+ **height** attribute on td and th.
	+ **hspace** and **vspace** attributes on img and object.
	+ **marginheight** and **marginwidth** attributes on iframe.
	+ **noshade** attribute on hr.
	+ **nowrap** attribute on td and th.
	+ **rules** attribute on table.
	+ **scrolling** attribute on iframe.
	+ **size** attribute on hr.
	+ **type** attribute on li, ol and ul.
	+ **valign** attribute on col, colgroup, tbody, td, tfoot, th, thead and tr.
	+ **width** attribute on hr, table, td, th, col, colgroup and pre.

**New and Not Yet Useful, but you should know about:**

* **<input>** has many new possible value for the type attribute: email, url, date etc. See <http://w3schools.com/html5/tag_input.asp> for a good summary and Section 10.4 of <http://dev.w3.org/html5/spec/> for all the details. You can also check at <http://w3schools.com/html5/html5_form_input_types.asp> to see which types have been implemented. As of this writing (Aug. 2010) only Opera has implemented the new types, so this falls under the heading of “coming attractions”.
Alissa Miller, Manager of our Communication Department’s Lab, points out that:

“ Many of the new input types are already being implemented by several browsers. The more basic ones such as email, number, search, tel and password can be used now as is.  Safari (mobile and desktop) and Chrome are ahead of the curb. Firefox 4.0 will introduce support for several new form types.

Here's some links(about what is implemented in various browsers):

Chrome: http://dev.chromium.org/developers/web-platform-status/forms
Firefox (4): https://developer.mozilla.org/en/HTML/HTML5/Forms\_in\_HTML5
Safari: http://developer.apple.com/safari/library/documentation/AppleApplications/Reference/SafariHTMLRef/Articles/InputTypes.html#//apple\_ref/doc/uid/TP40008055-InputTypeValues

The best part about these new input types is that if the browser does not recognize them, it will simply render the input as type text. So, there's no reason not to use them. They can and should be used now. The experience will be enhanced in browsers with support and the fallback is there for ones that don't support it.”
* **New form elements,** including one to authenticate users, are available at <http://w3schools.com/html5/html5_form_elements.asp> . Again, only Opera has implemented them so they are “coming attractions.”
* **New form attributes** Other than height, width and autocomplete, none of these is widely implemented. IE8 hasn’t even implemented multiple! (although presumably IE9 will do so.) You can track the status at <http://w3schools.com/html5/html5_form_attributes.asp>

**New and getting a big play: <canvas>, <audio>, <video>, <embed> and <source>**

* **Current status:** At [http://en.wikipedia.org/wiki/Comparison\_of\_layout\_engines\_(HTML5)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_%28HTML5%29) you can find which engines have implemented which layout tags. (WebKit is Safari and Gecko is Firefox & SeaMonkey; Presto is Opera and Trident is IE.)
* **<canvas>** As of this writing Firefox, Opera, and Safari have implemented the 2D graphics and have partially implemented the 3D graphics (WebGL).IE8 has implemented essentially none of this, but promises to do so in IE9. (Note: This does not mean your user will have downloaded IE9.) **canvas is used for drawing images**, so you may need to offer a gif file for browsers which don’t yet support this element. Current status is available at [http://en.wikipedia.org/wiki/Comparison\_of\_layout\_engines\_(HTML5\_Canvas)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_%28HTML5_Canvas%29)
* **< audio>, <video>, <embed> and <source>** are all related to media. Audio and video are obvious, embed defines the plug-in and source provides the link to the media resource.
[http://en.wikipedia.org/wiki/Comparison\_of\_layout\_engines\_(HTML5\_Media)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_%28HTML5_Media%29) has a general status update on these elements’ attributes, but the table at [http://en.wikipedia.org/wiki/Comparison\_of\_layout\_engines\_(HTML5\_Media)#Video\_format\_support](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_%28HTML5_Media%29#Video_format_support) tells you which media formats are supported in which browsers. Again, Microsoft has promised further implementation in IE9 (Trident). Fortunately, HTML5 allows you to list several different formats and have the browser use the first one it can on the list.
[**http://diveintohtml5.org/video.html**](http://diveintohtml5.org/video.html)tells you probably more than you want to know about various codecs (algorithms for streaming video). The complete status as of August 2010 is
	+ Mozilla Firefox (3.5 and later) supports Theora video and Vorbis audio in an Ogg container.
	+ Opera (10.5 and later) supports Theora video and Vorbis audio in an Ogg container.
	+ Google Chrome (3.0 and later) supports Theora video and Vorbis audio in an Ogg container. It also supports H.264 video (all profiles) and AAC audio (all profiles) in an MP4 container.
	+ As of this writing (June 9, 2010), [the “dev channel” of Google Chrome](http://www.chromium.org/getting-involved/dev-channel), [nightly builds of Chromium](http://build.chromium.org/buildbot/snapshots/), [nightly builds of Mozilla Firefox](http://nightly.mozilla.org/), and [experimental builds of Opera](http://labs.opera.com/news/2010/05/19/) all support VP8 video and Vorbis audio in a WebM container. (Visit [webmproject.org](http://www.webmproject.org/users/) for more up-to-date information and download links for WebM-compatible browsers.)
	+ Safari on Macs and Windows PCs (3.0 and later) will support anything that QuickTime supports. In theory, you could require your users to install third-party QuickTime plugins. In practice, few users are going to do that. So you’re left with the formats that QuickTime supports “out of the box.” This is a long list, but it does not include Theora video, Vorbis audio, or the Ogg container. However, QuickTime does support H.264 video (main profile) and AAC audio in an MP4 container.
	+ Mobile phones like Apple’s iPhone and Google Android phones support H.264 video (baseline profile) and AAC audio (“low complexity” profile) in an MP4 container.
	+ Adobe Flash (9.0.60.184 and later) supports H.264 video (all profiles) and AAC audio (all profiles) in an MP4 container.
	+ Internet Explorer 9 will support some as-yet-unspecified profiles of H.264 video and AAC audio in an MP4 container.
	+ Internet Explorer 8 has no HTML5 video support at all, but virtually all Internet Explorer users will have the Adobe Flash plugin. Later in this chapter, I’ll show you how you can use HTML5 video but gracefully fall back to Flash.

Not surprisingly, Microsoft will be using a different codec from Firefox, Opera, Safari, and Chrome. In order to accommodate all the major browsers (including for the iPhone and Droid) you will need to encode your video THREE times!
<http://diveintohtml5.org/video.html> gives detailed information about how to encode in all these formats (as well as licensing information for one which requires it.) and I recommend the article.
A more casual description at <http://www.computerworld.com/s/article/9179166/Crash_course_HTML_5_video_?taxonomyId=167&pageNumber=1> has enough information to get you going, and a nice description of the pros and cons of various codecs. They recommend one or more encodings plus Flash as a fallback.
To make this a little clearer, they offer the template, which references the three major codecs and omits Flash (for those of us who believe in standards):

<video width="640" height="480" controls>
    <source src="/video/video.mp4" type='video/mp4;
 codecs="avc1.42E01E, mp4a.40.2"'>
    <source src="/video/video.ogv" type='video/ogg;
 codecs="theora, vorbis"'>
    <source src="/video/video.Webm" type='video/Webm;
 codecs="vp8, vorbis"'>
</video>

The <video …> tag sets the size of the element and the controls attribute allows the user to pause and restart the video.

The three embedded <source …> tags refer to the three encodings. All the files have the same name (your choice), but different extensions, depending on their type of codec. The type and codecs attribute specify these.

Again, a browser which supports the <video> and <source> elements of HTML5 will use the first source file in the list which the browser supports. According to this article, Flash is embedded in the <video> tag, so that if none of these codecs is available (think IE8) then your Flash encoding will take over.
Going one better, CarmenDesign.com has provided a template which also degrades to HTML4 and Flash if the browser doesn’t support HTML5. The code, fully commented, is (thank you!) at <http://camendesign.com/code/video_for_everybody#video-code> with further info at <http://camendesign.com/code/video_for_everybody>

HOT UPDATE: On 8/26/2010 MPEG LA announced (<http://www.businesswire.com/news/home/20100825006629/en>) that they would not charge any royalties to end-users for use of the H.264 codec. It is expected that Firefox and other browsers which had stayed away from this codec will implement it in their browsers. At that point, all videos will be able to be seen by using only two formats – H.264 and Flash (for IE8 and earlier browsers).