



## What's coming in HTML 5

# New Start

Back in 1999 when the HTML 4.01 standard first appeared, virtually nobody envisioned video blogs, social networking sites, or Internet office tools. The upcoming HTML 5 standard will remake the web for the new generation of technologies and services.

By Peter Kreussel

**B**y 2004, social network architects and web application developers knew that the HTML 4.01 standard was long out of date. The Document Object Model (DOM 2) standard for language-independent interaction with HTML objects was even worse. In the five years since these standards had been ratified, a generational shift had occurred in the web development industry, giving rise to new techniques for dynamic content, multimedia, and semantic processing. Representatives from Mozilla, Opera, and Apple, tired of waiting for progress, founded the Web Hypertext Application Technology Working Group (WHATWG) to explore the possibility of an updated web standard.

The group created Web Applications 1.0 as a quasi-standard that reflected the more complex requirements of modern web applications. By the end of 2007, the Browser alliance had finally convinced the World Wide Web Consortium (W3C), and the alliance's proposals now serve as the basis for the next version of the HTML standard, which is known as HTML 5 [1].

HTML 5 provides new tools and capabilities to address the problems commonly faced by web developers on today's more interactive Internet (Table 1). Many experts are already predicting the multimedia features of HTML 5 will end the dominance of Adobe's powerful but proprietary Flash technology, which runs in browser plugins around the world. Steve Jobs' recent announcement that the iPhone team is abandoning Flash in favor of HTML 5 and its surrounding technologies has only increased this anticipation. The many additional benefits of HTML 5 will one day lead to improved forms, faster development, and easier scripting.

According to HTML5 editor Ian Hickson, the HTML 5 standard should reach the Candidate Recommendation stage sometime during 2012, but many features will be implemented in web browsers (and web pages) well before the final standard.

## Flashback

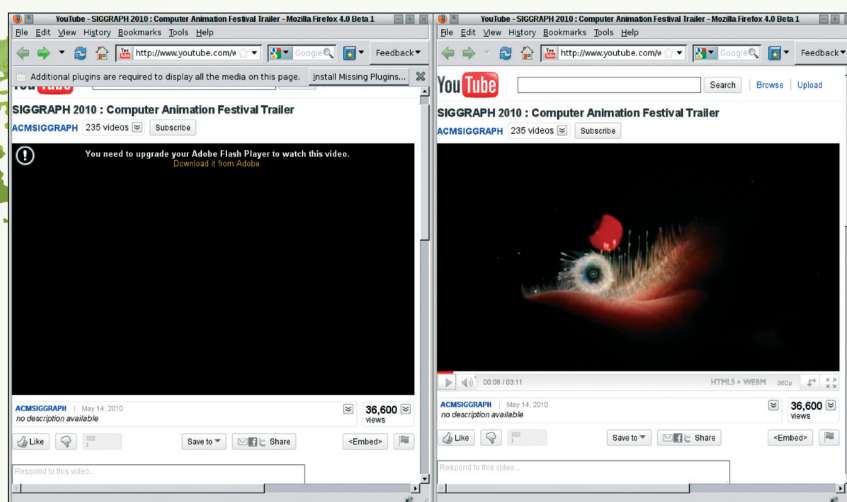
The multimedia capabilities of HTML 5 have received the most public attention. At the code level, HTML 5 multimedia (like all HTML) comes down to tags. The standard defines two new tags for multimedia content: `<video>` and `<audio>`. Multimedia tags can specify multiple, alternative sources for built-in elements. The browser selects the first file from a list it parses. If the `controls` attribute is set, the browser displays Play, Pause, Fast Forward, and Reverse buttons. An alternative player is a JavaScript control API. An HTML 5-based YouTube beta [11] builds a user interface around the JavaScript control API (Figure 1) that is hard to distinguish from the Flash Player (Figure 2). Before you get too excited about this promising power to trigger multimedia events from within HTML, keep in mind that the multimedia system is still not necessarily seamless. HTML 5 doesn't recommend any codecs for video and audio compression, so the standard offers no guarantee for running standards-compliant media data on popular browsers. The presence of the necessary codecs is the responsibility of the individual browser or operating system distribution.

For patenting and licensing reasons, the free, stable Firefox v3.6 only provides the free Theora codec. In May 2010, Google published the WebM video format [12], a

## OTHER DOCS

By now, W3C has spun off many of the technologies covered in the original 2004 proposal into separate documents. These additional resources cover innovative technologies such as Server Sent Events, which helps Ajax applications respond more efficiently to server events, or IndexedDB, a database built into the browser.

## HTML 5



**Figure 1:** No Flash? No problem. The YouTube Beta, Firefox 4 Beta, Opera 10.60, and Chrome 6 Beta support the HTML 5 video tag and Google's WebM standard with the free VP8 video codec.

package using a Matroska container, Ogg media audio compression, and Google's own VP8 video codec under a free license [13]. Firefox 4 Beta, Opera 10.60, and the current unstable Chrome Version 6 integrate this WebM out of the box. Although Internet Explorer supports the `<video>` and `<audio>` tags, Version 9 Preview 3 still doesn't include built-in codecs.

## Master Painter

The new `<canvas>` element lets the developer draw lines, arcs, and Bezier curves with JavaScript – in real time and without contacting the server. This feature means that HTML 5 is no longer restricted to displaying text, rectangles, or predefined graphics. Also included is a command for filling curves. With a number of optical effects, `<canvas>` can integrate, rotate, scale, and manipulate bitmaps.

The `<canvas>` tag is a practical addition to the toolbox whenever a web application needs to draw live diagrams, display or hide bitmaps, and scale or rotate bitmaps (Figure 3). The 3D version is even more impressive (Figure 4); the developers did not use a simple character-based language to program the 3D component (in contrast to the 2D Canvas), but instead chose OpenGL ES 2.0 [14], an OpenGL variant originally designed for mobile devices. With the exception of Internet Explorer, the current crop of browsers supports the `<canvas>` element fairly well. Explorer Canvas [15], JavaScript software developed by Google and released under the Apache license, adds `<canvas>` support to the Microsoft browser.

## Semantics

The term *semantic web* describes a collection of technologies (some available and others yet unrealized) to embedded clues

**TABLE 1: Major Changes in HTML 5**

Feature	Function	Support in current browsers*
<code>&lt;audio&gt;</code> , <code>&lt;video&gt;</code>	Media player, including play buttons and a JavaScript API [2]	Firefox (WebM Version 4 Beta), Opera, Chrome (WebM Version 6 Beta), Internet Explorer 9 Preview 3 (no built-in codec)
Tags for page structure ( <code>&lt;section&gt;</code> , <code>&lt;article&gt;</code> , <code>&lt;figure&gt;</code> , etc.)	Highlight contiguous text snippets in search engines and screen readers [3]	Firefox 4 Beta, Chrome (up to <code>&lt;figure&gt;</code> )
Improved forms	Default text, autocomplete, autofocus, new data types such as phone number or date, regex-based validity checks [4]	Firefox 4 Beta (20%), Opera (80%), Chrome (85%)
<code>&lt;meter&gt;</code> , <code>&lt;progress&gt;</code>	Display dimensions and degree of completeness graphically/machine readable [5]	Chrome 6 Beta
Menus	Declarative only menus without JavaScript [6]	No support
Drag and Drop	Cross-document drag-and-drop function [7]	Firefox, Chrome
Offline applications	API for caching code or media [8]	Firefox, Opera, Chrome
Canvas	Drawing area programmable in JavaScript [9]	Firefox, Opera, Chrome, Internet Explorer (text only)
Direct integration of SVG and MathML	SVG and MathML tags directly in HTML page code [10]	Firefox 4 Beta, Chrome, Internet Explorer 9 Preview 3 (SVG only)

\* Firefox 3.6.8, Opera 10.60, Chrome 5.0.375, IE 8, or specified beta version

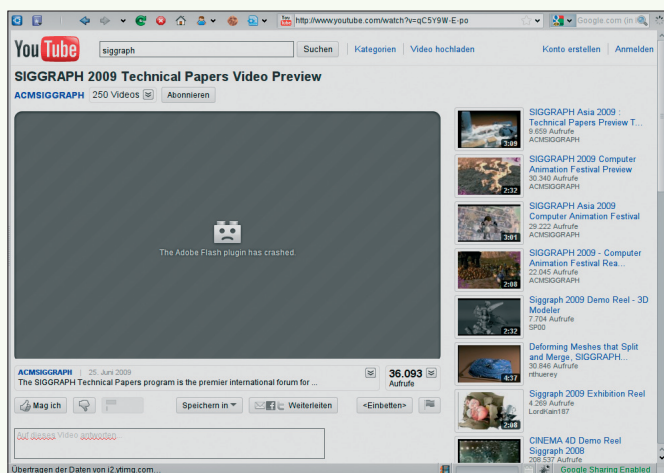


Figure 2: The frowning version of the Flash player, frequently seen on 64-bit systems, might become a thing of the past on the HTML 5 web.

about the meaning and structure of web information. The ultimate goal of these semantic techniques is to allow the development of smarter tools that will see web data as more than just strings of letters and words. HTML 5 provides several new semantic features with the integration of new tags on the basis of the DocBook [17] semantic markup language. These new tags let the developer build a topical structure explicitly into the HTML code (Table 2). Figure 5 shows the possible structure of a blog page. Many of the structuring tags, such as `<section>`, `<header>`, or `<footer>`, can be nested.

The `<aside>` tag (Table 2) could save time in searches and suppresses matches for search keys in areas not directly related to the core topic. (Some question whether webmasters will want to implement this tag because it is likely to reduce the number of page views.) Page structure tags will not cost browsers much overhead, but to allow them to be formatted with CSS, browsers do need to avoid discarding them as unknown. (The only browsers that process these tags correctly right now are Firefox 4 Beta and Chrome.)

In HTML 4, `<i>` (italic) and `<b>` (bold) tags might seem convenient, but they are a departure from the common principle of separating mark up (HTML) and rendering (CSS). HTML 5 no

TABLE 2: Tags for Semantic Page Structuring

Tag	Function
<code>&lt;section&gt;</code>	General document or application unit
<code>&lt;article&gt;</code>	Independent text block (e.g., a blog entry, magazine article)
<code>&lt;aside&gt;</code>	Content only loosely related to the topic
<code>&lt;hgroup&gt;</code>	Groups multiple, successive tags <code>h1</code> , <code>h2</code> , ...
<code>&lt;header&gt;</code>	Header with navigation or introductory notes
<code>&lt;footer&gt;</code>	Footer, which can contain information on the author or the like
<code>&lt;nav&gt;</code>	Navigation area
<code>&lt;figure&gt;</code>	Content block referenced as a single unit (e.g., "Figure 1" or "Diagram 1")
<code>&lt;figcaption&gt;</code>	caption or descriptive text for figure

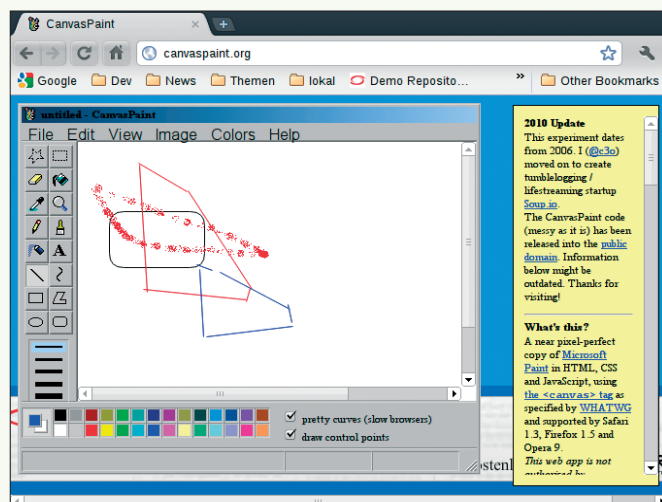


Figure 3: The canvas element supports complex, interactive, graphical applications, such as Canvas Paint [16] which has been around since 2006.

longer defines these tags typographically, but instead gives them a semantic function (Table 3): `<i>` will refer to a technical term, or an expression in a foreign language, and `<b>` is a key word. The standard leaves the visuals to the browser, unless the presentation is handled by a CSS-style assignment.

Neither `<b>` nor `<i>` are used in HTML 5 to emphasize individual words. The standard defines three tags for adding emphasis, including two old friends that assume different roles: The emphasis element `<em>` indicates stress as expressed in the intonation of a sentence; `<strong>` emphasizes the factual meaning of the enclosed text; `<mark>` is for retrospective highlighting in quoted text; `<small>` reduces the meaning of the highlighted text, which could be expressed through style options, such as a smaller font in display.

Depending on your point of view, this triple subdivision of stress can be seen as granular or finicky. On the one hand, semantic markup is really valuable for targeted research. On the other, their value will depend on how quickly they establish themselves in the mind of an average web designer. The only element to have made its way into the browsers thus far is `<mark>`, and this is only in Firefox 4 Beta and Chrome 6 Beta.

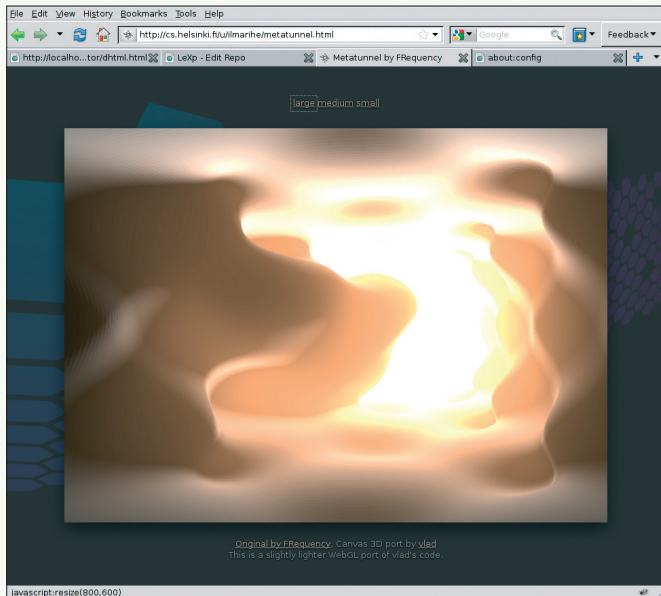
## Forms

HTML 5 introduces major improvements for input forms, an important part of many web applications. The new standard includes fields for email addresses, phone numbers, URLs, times,

TABLE 3: Semantics at Word Level

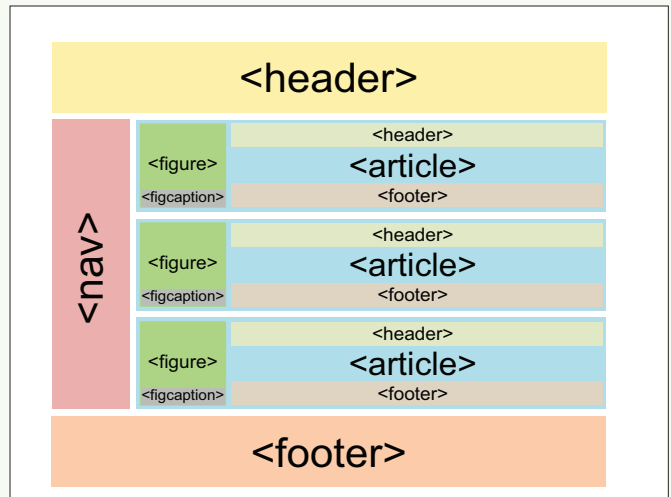
Element	Function
<code>&lt;i&gt;</code>	Technical term, foreign language term
<code>&lt;b&gt;</code>	Key word without special emphasis
<code>&lt;em&gt;</code>	Stress at language level
<code>&lt;strong&gt;</code>	Stress of factual significance
<code>&lt;mark&gt;</code>	Retrospective markup
<code>&lt;small&gt;</code>	"Small print"
<code>&lt;time&gt;</code>	Date and/or time





**Figure 4:** Thanks to WebGL, OpenGL code will run without any major reworking.

dates plus times, and color values (Table 4). The browser validates the input against the specified data type before sending the form. Numeric input fields with increment and decrement buttons validate the data against definable maximum and minimum values.



**Figure 5:** DocBook-style Tags open up the structure of an HTML 5 web page to search engines.

HTML5 lets you validate standard data types, such as email addresses or URLs, without a script, and you can set the required attribute for mandatory entries. With the pattern attribute, you can specify a regular expression for the value check. If the placeholder attribute is set, the browser displays placeholder text, which disappears when the user makes an entry. Firefox 4 already supports some elements from the new form standard; Chrome and Opera support far more (Table 4).

## LINUX PROS READ **LINUX PRO**

Enjoy a rich blend of tutorials, reviews, international news, and practical solutions for the technical reader.

Subscribe now  
to receive:

3 issues  
+ 3 DVDs  
for only

**\$3.00!**

[www.linuxpromagazine.com/trial](http://www.linuxpromagazine.com/trial)



**PREMIUM  
BLEND**

The new `<time>` tag is also worth mentioning: It includes formulations such as `<time datetime="2010-7-31T21:18:12-02:00">late in the evening on the same day</time>` and assigns a machine-readable value, including a time zone [18].

## Other Features

The DOM programming interface is regarded as one of the greatest restrictions to web development. DOM use JavaScript to add new elements or access existing elements on an HTML page. W3C has cut out some of the dead wood and integrated some useful Microsoft techniques. For example, the `innerHTML` method, which popular browsers started to integrate years ago and which web developers would find hard to live without, has finally become part of the standard. Developers no longer need to write their own JavaScript code and navigate the entire DOM tree to select all the elements in a CSS class – something that nearly every web application does. `getElementsByClassName()` does this more easily and quickly. Today's popular browsers already support this feature, and Internet Explorer will introduce it in Version 9. `outerHTML()` is another new method that overwrites a whole element with HTML code, rather than just its content (as is the case with `innerHTML()`). Chrome, Opera, and Internet Explorer 9 support this method.

`insertAdjacentHTML()` inserts new code immediately in front of and behind an element and as its first or last child. This technique now works in Opera, Chrome, and Internet Explorer.

The drag-and-drop API is also important for web applications. Unfortunately, the specification leans too closely on the definition that Microsoft threw together at the height of the browser war, and it is often criticized. The controversial API is implemented by Internet Explorer, Firefox, and Chrome.

The `contenteditable` attribute and the matching JavaScript API standardize a technique introduced to Internet Explorer in

1999, Mozilla in 2003, and Opera later, on which rich text editors such as the CK editor (formerly FCK Editor) [19] are based. The built-in editor has made its way into the Firefox, Opera Chrome, and Internet Explorer browsers.

## In and Out

Without a doubt, HTML 5 gives new life to the HTML standard, which has lain dormant for many years. However, W3C has pushed the most powerful new technologies for high-performance web applications out of the HTML standard. The surgically removed parts include web sockets, which servers use to send messages to the browser. This removes the need for chat systems to check for messages every few seconds or to keep Ajax connections open permanently, tying up an Apache thread and a script interpreter in memory for each user.

The Web Storage API, which is similar to a cookie (apart from the data volume it supports), lets a CMS save data input at regular intervals – preferably in a worker thread executed in the background to avoid blocking the browser. If the browser crashes before sending the data, the information is still available client-side. The Query Selector API shifts convenient CSS selectors from libraries like jQuery into the standard.

Other web technologies, such as the Microdata specification, which could give a considerable boost to the vision of a semantic web that only exists in the minds of the W3C members right now, are far better off in their own specifications. On the other hand, removing these more advanced features from the baseline document does deprive them of the tangible momentum that HTML 5 represents. ■■■

## INFO

- [1] HTML 5 <http://dev.w3.org/html5/spec/Overview.html>
- [2] `<video>` and `<audio>` standards: <http://www.w3.org/TR/html5/video.html>
- [3] Standard for page structure tags: <http://www.w3.org/TR/html5/sections.html>
- [4] HTML 5 forms: <http://www.w3.org/TR/html5/ states-of-the-type-attribute.html> <http://www.w3.org/TR/html5/common-input-element-attributes.html>
- [5] `<meter>` and `<progress>`: <http://www.w3.org/TR/html5/the-button-element.html>
- [6] HTML 5 menus: <http://www.w3.org/TR/html5/ interactive-elements.html>
- [7] Drag and drop: <http://www.w3.org/TR/html5/dnd.html>
- [8] Offline API: <http://www.w3.org/TR/html5/offline.html>
- [9] Canvas: <http://www.w3.org/TR/html5/the-canvas-element.html>
- [10] Embedded SVG code: <http://www.w3.org/TR/html5/the-map-element.html>
- [11] HTML 5 version of YouTube: <http://www.youtube.com/html5>
- [12] WebM: <http://www.webmproject.org>
- [13] WebM license: <http://www.webmproject.org/license/>
- [14] OpenGL ES: <http://www.khronos.org/opengles/>
- [15] ExplorerCanvas: <http://code.google.com/p/explorercanvas/>
- [16] Canvas Paint: <http://canvaspaint.org>
- [17] DocBook.org: <http://www.Docbook.org>
- [18] `<time>`: <http://www.w3.org/TR/html5/ common-microsyntaxes.html#parse-a-date-or-time-string>
- [19] CKEditor: <http://ckeditor.com>

**TABLE 4: Advanced Form Elements**

HTML Code	Function
<code>&lt;input type="date" /&gt;</code> , <code>type="month"</code> , <code>type="week"</code> , <code>type="time"</code> , <code>type="datetime"</code> , <code>type="tel"</code> , <code>type="url"</code>	Validate time and date, telephone numbers, email, and URLs (auto-completion, as in the address bar) calendar widget
<code>&lt;input type="number" min="0" max="10" step="2" value="3" /&gt;</code>	Validation of minimum and maximum, arrow buttons to step increment/decrement values
<code>&lt;input type="range" /&gt;</code>	Slide for numeric values
<code>&lt;input type="color"&gt;</code>	Color chooser widget
Attribute required	Input required
Attribute pattern	Validity check against regular expression
Attribute placeholder	Placeholder text
Attribute autocomplete	Enable/disable browser's auto-complete function
Attribute list	Autocomplete suggestions based on values stored on the page
Attribute autofocus	Focus after loading the page
Attribute multiple	Accept multiple values of type