HTML5

A SEMANTIC WEB
MARKUP

A BRIEF HISTORY

(hopefully)
We never had HTML 1. Instead HTML 2 was the first official spec, published by the Internet Engineering Task Force in 1995.
In 1997, HTML 3.2 was published by the W3C. It was the first version to be developed and standardized exclusively by the W3C.
Later in 1997, HTML 4.0 was published. It was the first version to offer variations in its standards mode.

- **STRICT**: No deprecated elements
- **TRANSITIONAL**: Allows deprecated elements
- **FRAMESET**: For frame related elements
HTML 4.01

In 1999, HTML 4.01 was published. It offered the same options as 4.0; just with some minor revisions to the spec.
After 4.01, we shifted to XHTML. It was a merger of the HTML 4.0 spec with the semantics of XML. The X stood for Extensible, and markup became much more restrictive. XHTML 1 was officially adopted in 2000. XHTML 1.1 was adopted in 2001.
The Web Hypertext Application Technology Working Group was formed out of dissatisfaction with the W3C’s work on XHTML 2.0. The W3C seemed to be only concerned with pure standards, and not practical solutions to meet the needs of developers.
Ian Hickson proposed that the W3C move to extend HTML to support the creation of Web Applications. His proposal was rejected. The WHATWG was born.
THE WAR

The WHATWG began its work on HTML5 independently of the W3C. The W3C continued to work on XHTML 2.0. Soon, lines were drawn between the two groups, and supporters rallied to both sides.
In October of 2006, Tim Berners-Lee wrote a blog post, admitting that the W3C’s efforts to move the web further towards XML simply wasn’t working.

Several months later, the W3C issued a new charter for an HTML Working Group. The joined up with the WHATWG
HTML5

(Finally!)
HTML5

HTML5 is now being worked on by both the W3C and the WHATWG. The two groups are collaborating on developing the spec.

We should see a candidate recommendation in 2012, and an official spec in 2022.
<html lang="en">

</html>
One of the biggest things about HTML5 is the new elements that it introduces, all in the hope of creating more semantic markup.
 Semester: These elements include:

- `<header>`
- `<footer>`
- `<nav>`
- `<section>`
- `<article>`
- `<aside>`
- `<hgroup>`
- `<canvas>`
- `<video>`
- `<time>`
- `<mark>`
SEMANTICS

Markup would look something like this

<header>
<nav>
<article>
<section>
<aside>
<footer>

image courtesy of Lachlan Hunt
VIDEO

The video element is the first standardized way of embedding video into markup. Prior to this, all video required a third party plugin such as Flash, Quicktime, or RealPlayer.

Currently supported by IE9, FireFox 3.5, Safari 4, Safari 3, Chrome and Opera.
VIDEO

Attributes of the video element include:

- Autoplay
- Controls
- Height
- Width
- Loop
- Preload
- Source
Example Markup:

<video src="movie.ogv" controls="controls">
  your browser does not support the video tag
</video>
HTML5 defines the canvas element as a "resolution-dependent bitmap canvas which can be used for rendering graphs, game graphics, or other visual images on the fly."

It is currently supported by IE 7, Firefox 3.0+, Safari 3.0+, Chrome 3.0+, Opera 10+, iPhone 1.0+, and Android 1.0+
Example Markup:

```html
<canvas width="800" height="600"></canvas>
```

You can then create functions that allow people to interact with the canvas.
OTHER FEATURES

Some other new features of HTML5 include

- Local Storage
- Geolocation
- Offline Web Applications
LOCAL STORAGE

Local Storage allows you to store key/value pairs locally within the client’s web browser. This data persists much like cookies, even when one navigates away, closes a tab, or even shuts down the browser.

Currently supported in IE 8+, Firefox 3.5+, Safari 3.5+, Chrome 4.0+, Opera 10.5+, iPhone 2.0+, Android 2.0+
Geolocation API allows your web applications to interact with users based on their location. The API is always an opt-in choice for the user.

Currently supported by Firefox 3.5+, Chrome 5.0+, iPhone 3.0+ and Android 2.0+
Offline web applications allow you to create an application that will function even if there is a network interruption. It prompts the browser to download a cache a list of files so that even if network access is lost, the user can still work in the application.