LECTURE 4

HTML 5

INTRODUCTION TO HTML5

- HTML5 ≈ HTML 5 + CSS 3 + JavaScript
- HTML5 is a suite of tools for:
  - Markup (HTML 5)
  - Presentation (CSS 3)
  - Interaction (DOM, Ajax, APIs)
- Brought on by the evolving use of the web

What is HTML5?

- HTML5 is the newest version of HTML, only recently gaining partial support by the makers of web browsers.
- It incorporates all features from earlier versions of HTML, including the stricter XHTML.
- It adds a diverse set of new tools for the web developer to use.
- It is still a work in progress. No browsers have full HTML5 support. It will be many years – perhaps not until 2018 or later - before being fully defined and supported.

Goals of HTML5

- Support all existing web pages. With HTML5, there is no requirement to go back and revise older websites.
- Reduce the need for external plugins and scripts to show website content.
- Improve the semantic definition (i.e. meaning and purpose) of page elements.
- Make the rendering of web content universal and independent of the device being used.
- Handle web documents errors in a better and more consistent fashion.

A Rough History of Web Standards

91-92	93-94	95-96	97-98	99-00	01-02	03-04	05-06	07-08	09-10	11-12	13-14
HTML 1	HTML 2		HTML 4	XHTML 1					**HTML 5**		
		CSS 1	CSS 2		T-less D	Web 2.0			**CSS3**		
		JS	ECMA, DOM	DOM 2			Ajax		**DOM, APIs**		

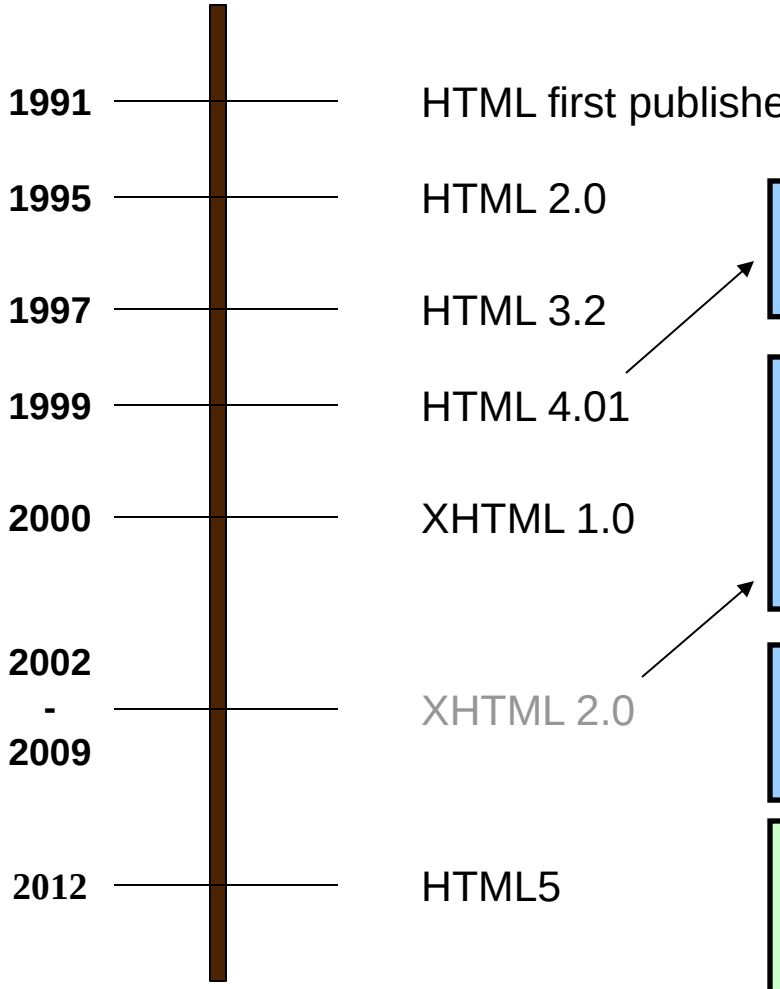
HTML 5

2004	WHATWG started
2008	W3C Working Draft
2012 (2010)	W3C Candidate Rec
2022	W3C Rec

CSS

1996 – CSS 1	W3C Rec
1998 – CSS 2	W3C Rec
1999 – CSS 3	Proposed
2005 – CSS 2.1	W3C Candidate Rec
2001 – CSS 3	W3C Working Draft

History of HTML



After HTML 4.01 was released, focus shifted to XHTML and its stricter standards.

XHTML 2.0 had even stricter standards than 1.0, rejecting web pages that did not comply. It fell out of favor gradually and was abandoned completely in 2009.

HTML5 is much more tolerant and can handle markup from all the prior versions.

Though HTML5 was published officially in 2012, it has been in development since 2004.

HTML4 vs HTML5

HTML5	HTML4
HTML5 uses new structures such as drag, drop and much more.	HTML 4 uses common structures like headers , footers.
Embedded video and audio without using flash player.	HTML 4 cannot embed video or audio directly and makes use of flash player for it.
HTML 5 is capable of handling inaccurate syntax	HTML 4 cannot handle inaccurate syntax
HTML 5 introduced many new API's which facilitate flexibility of web pages.	HTML 4 has traditional API's which does not include canvas and content editable API's.
In HTML 5, new tags and new features like local storage are enhanced.	In HTML 4, local storage is not possible and tags that can handle only one dimension are present

HTML vs XHTML vs HTML5

- Document Type Declaration:
  - Doctype for HTML 4.01:

```
<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.01/EN”
“http://www.w3.org/TR/html4/strict.dtd”>
```

  - Doctype for XHTML:

```
<!DOCTYPE HTML PUBLIC “-//W3C//DTD XHTML 1.0
Strict//EN” “http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd”>
```

  - Doctype for HTML5:

```
<!DOCTYPE html>
```

HTML vs XHTML vs HTML5 (cont.)

- The structural elements:

```
<html>
   <head>
      <!--head content-->
   </head>
   <body>
      <!--body content-->
   </body>
</html>
```

HTML

```
<html>
   <head>
      <!--head content-->
   </head>
   <body>
      <div id="header">
         <!--content-->
      </div>
      <div id="nav">
         <!--content-->
      </div>
      <div id="section">
         <!--content-->
      </div>
      <div id="aside">
         <!--content-->
      </div>
      <div id="footer">
         <!--content-->
      </div>
   </body>
</html>
```

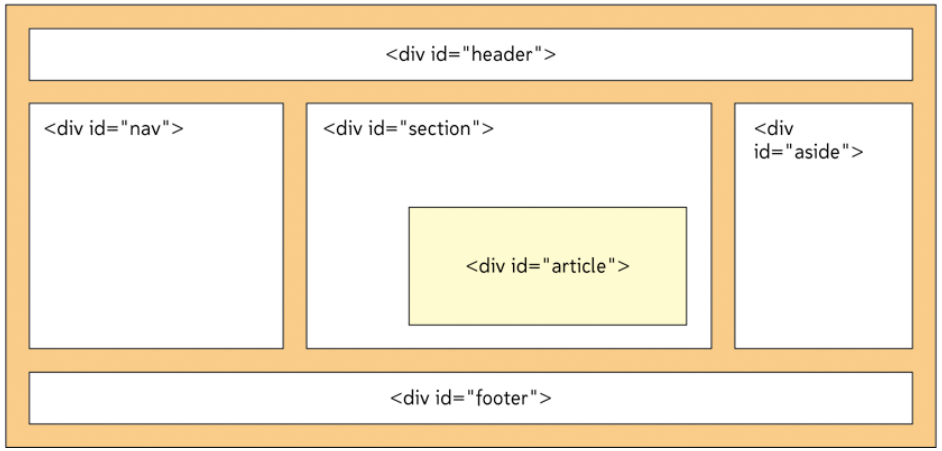
HTML 4.01 / XHTML

```
<html>
   <head>
      <!--head content-->
   </head>
   <body>
      <header>
         <!--content-->
      </header >
      <nav>
         <!--content-->
      </nav>
      <section>
         <!--content-->
      </section>
      <aside>
         <!--content-->
      </aside>
      <footer>
         <!--content-->
      </footer>
   </body>
</html>
```

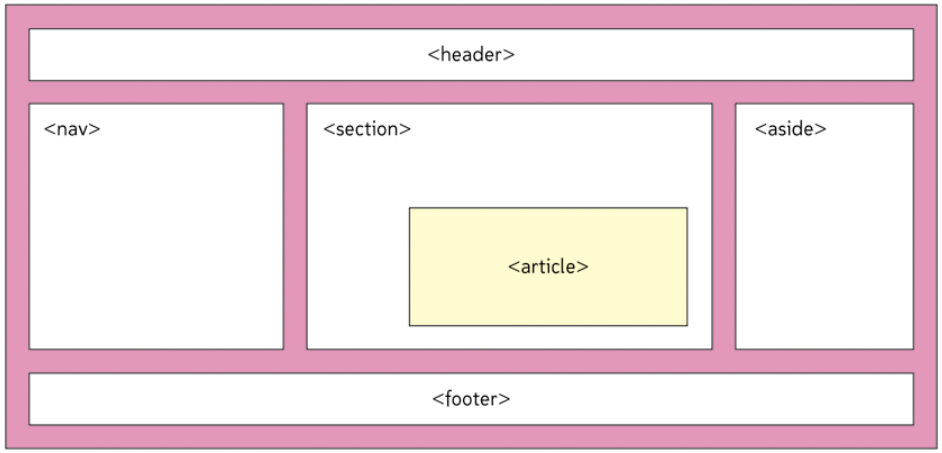
HTML5

HTML vs XHTML vs HTML5 (cont.)

- The structural elements diagram:
  - HTML 4.01



  - HTML5



New Elements in HTML5

<article>
<aside>
<audio>
<canvas>
<datalist>
<figure>
<figcaption>
<footer>
<header>
<hgroup>
<mark>
<nav>
<progress>
<section>
<source>
<svg>
<time>
<video>

These are just some of the new elements introduced in HTML5. We will be exploring each of these during this course.

Other New Features in HTML5

- Built-in audio and video support (without plugins)
- Enhanced form controls and attributes
- The Canvas (a way to draw directly on a web page)
- Drag and Drop functionality
- Support for CSS3 (the newer and more powerful version of CSS)
- More advanced features for web developers, such as data storage and offline applications.

5 HTML Enhancements

- HTML
- Forms
- CSS
- Offline applications
- Local storage

HTML Extended

- Document Flow: div, section, article, nav, aside, header, footer
- Audio, Video and Embed
- Canvas: paths, gradients, image manipulation, events
- Microdata for semantics and enhanced search engine results (Google Rich Snippets)

HTML5 Media Elements

Tag	Description

<audio>	Defines sound or music content
	`<audio controls>`
	`  <source src="horse.ogg" type="audio/ogg">`
	`  <source src="horse.mp3" type="audio/mpeg">`
	`  Your browser does not support the audio element.`
	`</audio>`

<embed>	Defines containers for external applications (like plug-ins)
	`<object width="400" height="50" data="bookmark.swf"></object>`
	`<object data="audi.jpeg"></object>`

<source>	Defines sources for <video> and <audio> `<source src="horse.mp3" type="audio/mpeg">`
<track>	Defines tracks for <video> and <audio>

<video>	Defines video or movie content
	`<video width="320" height="240" controls>`
	`  <source src="movie.mp4" type="video/mp4">`
	`  <source src="movie.ogg" type="video/ogg">`
	`  Your browser does not support the video tag.`
	`</video>`

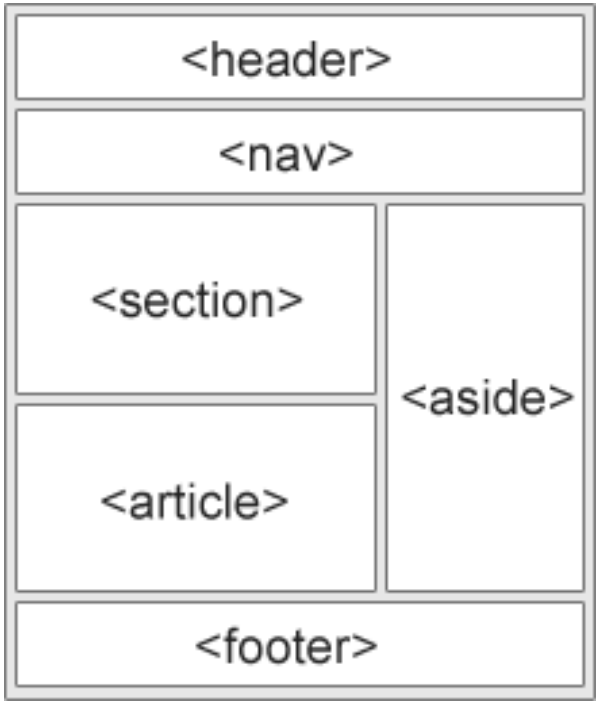
HTML5 Semantic Elements

What are Semantic Elements?

A semantic element clearly describes its meaning to both the browser and the developer.
Examples of **non-semantic** elements: `<div>` and `<span>` - Tells nothing about its content.
Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

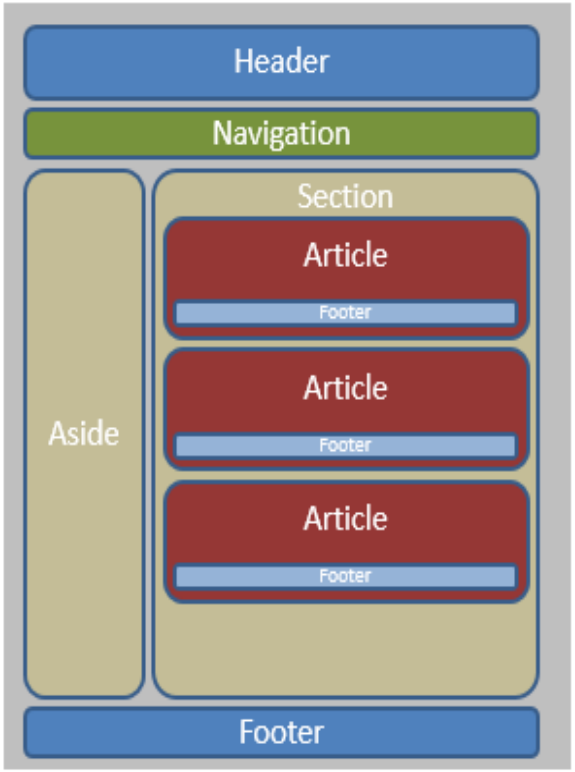
HTML5 offers new semantic elements to define different parts of a web page:

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



HTML5 Semantic Elements

- <header>
- <nav>
- <section>
- <article>
- <aside>
- <figcaption>
- <figure>
- <footer>



HTML5 Media

- Multimedia on the web is sound, music, videos, movies, and animations.
- HTML 5 VIDEO
- HTML 5 AUDIO
- HTML Helpers (Plug-ins)
- HTML YouTube Videos

HTML Helpers (Plug-ins)

The purpose of a plug-in is to extend the functionality of a web browser.

Helper applications (plug-ins) are computer programs that extend the standard functionality of a web browser.
Examples of well-known plug-ins are Java applets.
Plug-ins can be added to web pages with the `<object>` tag or the `<embed>` tag.
Plug-ins can be used for many purposes: display maps, scan for viruses, verify your bank id, etc.

HTML YouTube Videos

- To play your video on a web page, do the following:
- Upload the video to YouTube
- Take a note of the video id
- Define an <iframe> element in your web page
- Let the src attribute point to the video URL
- Use the width and height attributes to specify the dimension of the player

Canvas VS SVG

What is HTML Canvas?

The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.
The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.
Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

What is SVG?

SVG stands for Scalable Vector Graphics
SVG is used to define graphics for the Web
SVG is a W3C recommendation
SVG has several methods for drawing paths, boxes, circles, text, and graphic images.
The HTML <svg> element is a container for SVG graphics.

Comparison of Canvas and SVG

The table below shows some important differences between Canvas and SVG:

Canvas	SVG
•Resolution dependent •No support for event handlers •Poor text rendering capabilities •You can save the resulting image as .png or .jpg •Well suited for graphic-intensive games	•Resolution independent •Support for event handlers •Best suited for applications with large rendering areas (Google Maps) •Slow rendering if complex (anything that uses the DOM a lot will be slow) •Not suited for game applications

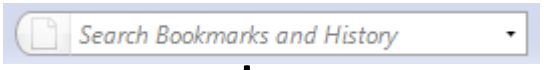
Canvas

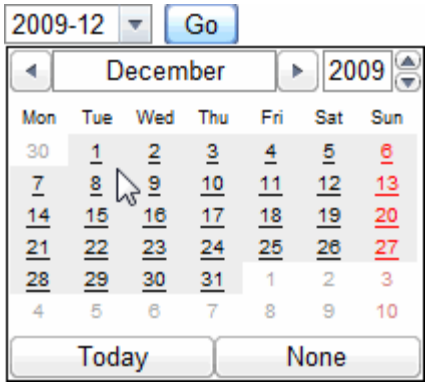
- The HTML <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript).
- The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.

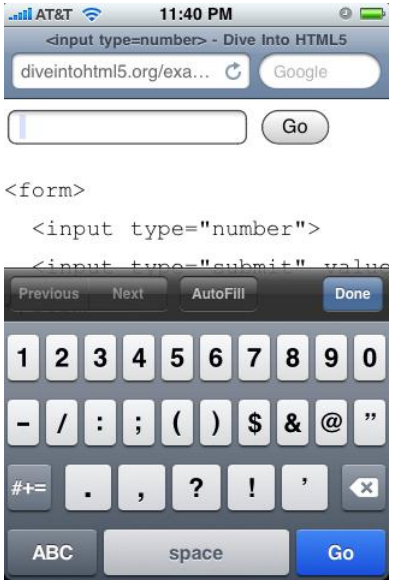
```
<canvas id="canvas" width="150" height="150">
</canvas>

function draw() {
    var canvas =
    document.getElementById("canvas");
    if (canvas.getContext) {
          var ctx = canvas.getContext("2d");
          ctx.fillStyle = "rgb(200,0,0)";
          ctx.fillRect (10,10,55,50);

          ctx.fillStyle = "rgb(0,0,200)";
          ctx.fillRect (30,30,55,50);
    }
}
```

Form Enhancements

- Placeholder text
- Specific text input: email, URL, number, search
- Slider
- Date picker
- User Agent validation







CSS Effects

- Rounded corners
- Gradients
- Box and text shadows
- Fonts
- Transparencies
- Multiple background images and border images
- Multiple columns and grid layout
- Box sizing
- Stroke and outlines
- Animation, movement and rotation
- Improved selectors

CSS Effect Example

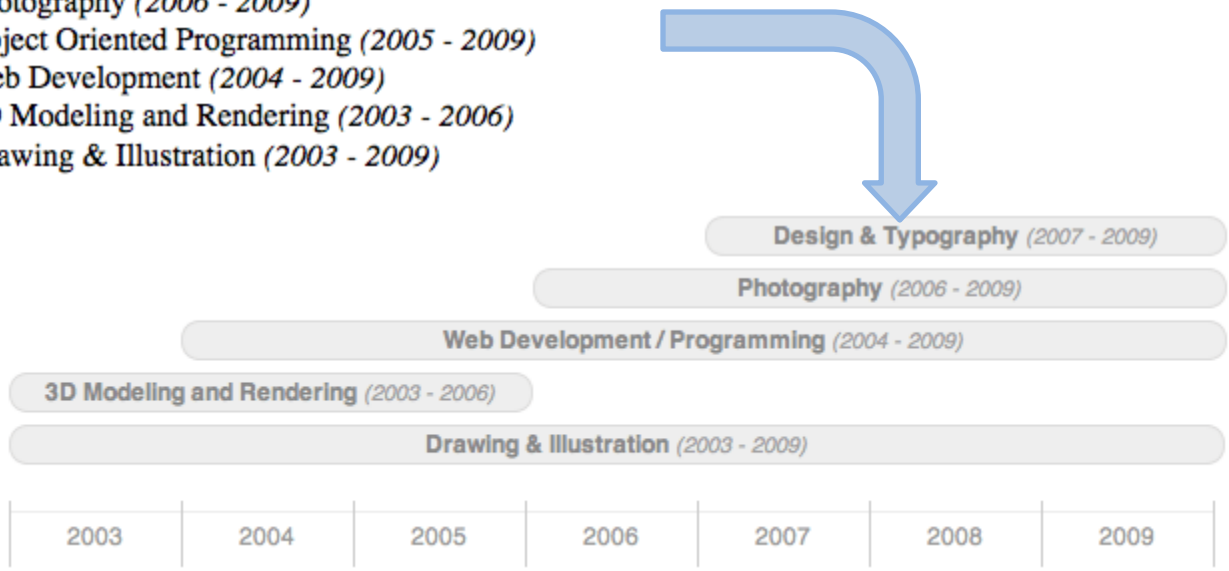
Amazing CSS Effects

```
.amazing {
border: 1px solid blue;
color: red;
background-color: gold;

-webkit-border-radius: 40px;
-moz-border-radius: 40px;
border-radius: 40px;

-webkit-box-shadow: 8px 8px 6px
   #474747;
-moz-box-shadow: 8px 8px 6px
   #474747;
box-shadow: 8px 8px 6px #474747;

text-shadow: 8px 8px 2px #595959;
filter: dropshadow(color=#595959,
   offx=8, offy=8);
}
```

http://css3generator.com/

CSS Timelines

- Design & Typography *(2007 - 2009)*
- Photography *(2006 - 2009)*
- Object Oriented Programming *(2005 - 2009)*
- Web Development *(2004 - 2009)*
- 3D Modeling and Rendering *(2003 - 2006)*
- Drawing & Illustration *(2003 - 2009)*



http://mattbango.com/notebook/web-development/pure-css-timeline/

Offline Applications

```
<html mainfest="http://m.health.unm.edu/someapp.manifest">
…
</html>
```


```
CACHE MANIFEST
#v1.01

#Explicitly cached files
CACHE:
index.html
Stylesheet.css
Images/logo.png

NETWORK:
Search.cfm
Login.cfm
/dynamicpages

FALLBACK:
/dynamicpage.cfm /static.html
```

http://www.whatwg.org/specs/web-apps/current-work/multipage/offline.html

Web storage

The Web Storage API defines a standard for how we can save simple data locally on a user's computer or device. Before the emergence of the Web Storage standard, web developers often stored user information in cookies, or by using plugins. With Web Storage, we now have a standardized definition for how to store up to 5MB of simple data created by our websites or web applications.

Web Storage is a great complement to Offline Web Applications, because you need somewhere to store all that user data while you're working offline, and Web Storage provides it.

Two kinds of storage

session Storage

Session storage lets us keep track of data specific to one window or tab. It allows us to isolate information in each window. Even if the user is visiting the same site in two windows, each window will have its own individual session storage object and thus have separate, distinct data.

Session storage is not persistent—it only lasts for the duration of a user's session on a specific site (in other words, for the time that a browser window or tab is open and viewing that site).

Local Storage

Unlike session storage, local storage allows us to save persistent data to the user's computer, via the browser. When a user revisits a site at a later date, any data saved to local storage can be retrieved.

Local Storage

- Beyond cookies- local storage
  - Manipulated by JavaScript
  - Persistent
  - 5MB storage per “origin”
  - Secure (no communication out of the browser)
- Session storage
  - Lasts as long as the browser is open
  - Each page and tab is a new session
- Browser based SQLite or IndexedDB

HTML Local Storage Objects

- HTML local storage provides two objects for storing data on the client:
  - window.localStorage - stores data with no expiration date
  - window.sessionStorage - stores data for one session (data is lost when the browser tab is closed)
- Before using local storage, check browser support for localStorage and sessionStorage.

```
<script>
// Check browser support
if (typeof(Storage) !== "undefined") {
    // Store
    localStorage.setItem("lastname", "Smith");
    // Retrieve
    document.getElementById("result").innerHTML =
localStorage.getItem("lastname");
} else {
    document.getElementById("result").innerHTML = "Sorry,
your browser does not support Web Storage...";
}
</script>
```

Local Storage

- Web storage

```
window.localStorage['value'] = 'Save this!';
```

- Session storage

```
sessionStorage.useLater('fullname', 'Garth Colasurdo');
alert("Hello " + sessionStorage.fullname);
```

- Database storage

```
var database = openDatabase("Database Name", "Database Version");
database.executeSql("SELECT * FROM test", function(result1) {
  …
});
```

http://dev.w3.org/html5/webstorage/

Geolocation

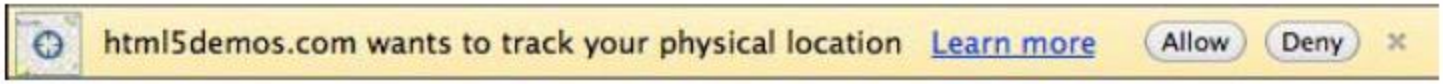
The first new API we'll cover is geolocation. Geolocation allows your visitors to share their current location.

Depending on how they're visiting your site, their location may be determined by any of the following:

- IP address
- wireless network connection
- cell tower
- GPS hardware on the device

Privacy Concerns

Not everyone will want to share their location with you, as there are privacy concerns inherent to this information. Thus, your visitors must opt in to share their location. Nothing will be passed along to your site or web application unless the user agrees. The decision is made via a **prompt** at the top of the browser. Figure 10.1 shows what this prompt looks like in Chrome.



With geolocation, you can determine the user's current position. You can also be notified of changes to their position, which could be used, for example, in a web application that provided real-time driving directions.

Geolocation: methods

These different tasks are controlled through the three methods currently available in the Geolocation API:

- getCurrentPosition
- watchPosition
- clearPosition

Web Worker

- When executing scripts in an HTML page, the page becomes unresponsive until the script is finished.
- A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.

Markup Elements

- New elements in HTML 5 :

Tag	Description
<article>	For external content, like text from a news-article, blog, forum, or any other content from an external source
<aside>	For content aside from the content it is placed in. The aside content should be related to the surrounding content
<command>	A button, or a radiobutton, or a checkbox
<details>	For describing details about a document, or parts of a document
<summary>	A caption, or summary, inside the details element
<figure>	For grouping a section of stand-alone content, could be a video
<figcaption>	The caption of the figure section

Markup Elements (cont.)

Tag	Description
<footer>	For a footer of a document or section, could include the name of the author, the date of the document, contact information, or copyright information
<header>	For an introduction of a document or section, could include navigation
<hgroup>	For a section of headings, using <h1> to <h6>, where the largest is the main heading of the section, and the others are sub-headings
<mark>	For text that should be highlighted
<meter>	For a measurement, used only if the maximum and minimum values are known
<nav>	For a section of navigation
<progress>	The state of a work in progress

Markup Elements (cont.)

- New media elements in HTML 5 :

Tag	Description
<audio>	For multimedia content, sounds, music or other audio streams
<video>	For video content, such as a movie clip or other video streams
<source>	For media resources for media elements, defined inside video or audio elements

- Canvas element in HTML 5 :

Tag	Description
<canvas>	For making graphics with a script

Markup Elements (cont.)

- New form elements in HTML 5 :

Tag	Description
<datalist>	A list of options for input values
<keygen>	Generate keys to authenticate users
<output>	For different types of output, such as output written by a script

MARKUP ELEMENTS (cont.)

- New input type attribute values in HTML 5 :

Tag	Description
tel	The input value is of type telephone number
search	The input field is a search field
url	The input value is a URL
email	The input value is one or more email addresses
datetime	The input value is a date and/or time
date	The input value is a date
month	The input value is a month
week	The input value is a week
time	The input value is of type time
datetime-local	The input value is a local date/time
number	The input value is a number
range	The input value is a number in a given range
color	The input value is a hexadecimal color, like #FF8800

Why & When?

Why We Should Start Using CSS3 and HTML5 Today?

For Our Clients

- Conceding to the idea that the project will not be able to look the same across various browsers,
- This means more developed and unfettered imaginative designs for our clients,
- This could lead to increased costs for clients as well, but with higher levels of innovation and
- Client’s visions for what they want will be less hindered by these limitations.

For the Users

- Potentially less disruptions of experience from one device to another and
- An overall improved user experience.

For Designers/Developers

- Conceding to the idea that the project will not be able to look the same across various browsers,
- A more open playing field for designers and developers all around; less restricted by this holding pattern,
- More exciting and innovative landscape to attract new clientele,
- Division of project audience into separate presentational approaches and
- Probably less work involved because we don’t need the many hacks and workarounds we’ve used before.