HTML5 and CSS3
Web Browser

→ Fetches/displays documents from web servers

→ Mosaic 1993

→ Firefox, IE, Chrome, Safari, Opera, Lynx, Mosaic, Konqueror

 ° There are standards, but wide variation in features
# Desktop Browser Market Share

<table>
<thead>
<tr>
<th></th>
<th>Chrome</th>
<th>IE</th>
<th>Firefox</th>
<th>Safari</th>
<th>Opera</th>
</tr>
</thead>
<tbody>
<tr>
<td>April</td>
<td>63.9 %</td>
<td>8.0 %</td>
<td>21.6 %</td>
<td>3.8 %</td>
<td>1.5 %</td>
</tr>
<tr>
<td>March</td>
<td>63.7 %</td>
<td>7.7 %</td>
<td>22.1 %</td>
<td>3.9 %</td>
<td>1.5 %</td>
</tr>
<tr>
<td>February</td>
<td>62.5 %</td>
<td>8.0 %</td>
<td>22.9 %</td>
<td>3.9 %</td>
<td>1.5 %</td>
</tr>
<tr>
<td>January</td>
<td>61.9 %</td>
<td>7.8 %</td>
<td>23.4 %</td>
<td>3.8 %</td>
<td>1.6 %</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Chrome</th>
<th>IE</th>
<th>Firefox</th>
<th>Safari</th>
<th>Opera</th>
</tr>
</thead>
<tbody>
<tr>
<td>December</td>
<td>61.6 %</td>
<td>8.0 %</td>
<td>23.6 %</td>
<td>3.7 %</td>
<td>1.6 %</td>
</tr>
<tr>
<td>November</td>
<td>60.1 %</td>
<td>9.8 %</td>
<td>23.4 %</td>
<td>3.7 %</td>
<td>1.6 %</td>
</tr>
<tr>
<td>October</td>
<td>60.4 %</td>
<td>9.5 %</td>
<td>23.4 %</td>
<td>3.9 %</td>
<td>1.6 %</td>
</tr>
<tr>
<td>September</td>
<td>59.6 %</td>
<td>9.9 %</td>
<td>24.0 %</td>
<td>3.6 %</td>
<td>1.6 %</td>
</tr>
<tr>
<td>August</td>
<td>60.1 %</td>
<td>8.3 %</td>
<td>24.7 %</td>
<td>3.7 %</td>
<td>1.8 %</td>
</tr>
</tbody>
</table>

http://www.w3schools.com/browsers/browsers_stats.asp
HTML5: New Features

- Semantic elements and Markups
- Audio and video support
- Canvas
- Drag and drop
- Local data storage
- Offline applications
- Server events
- Geolocation
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 `<img>` - Clearly defines its content.
Semantic elements: some other

- `<header>`
- `<section>`
- `<aside>`
- `<figcaption>`
- `<details>`
- `<mark>`
- `<nav>`
- `<article>`
- `<figure>`
- `<footer>`
- `<summary>`
- `<time>`
Semantic Element example & Markups

http://slides.html5rocks.com/#semantic-tags-1
Audio
<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>

Video
<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>

Output: <http://jsfiddle.net/mashiyat/g4hMX/>
<canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript).

→ only a container for graphics. You must use a script to actually draw the graphics.
Drag and drop means when you "grab" an object and drag it to a different location.

→ In HTML5, drag and drop is part of the standard, and any element can be draggable.

Demo: http://slides.html5rocks.com/#native-drag-and-drop
Using HTML5 we can store data locally within the user's browser.

→ Earlier, this was done with cookies. However, Web Storage is more secure and faster.

→ The data is not included with every server request, but used ONLY when asked for. It is also possible to store large amounts of data, without affecting the website's performance.

→ The data is stored in name/value pairs, and a web page can only access data stored by itself.

→ Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred.

Demo: [http://slides.html5rocks.com/#web-storage](http://slides.html5rocks.com/#web-storage)
Web Storage for local data storage & Offline app

Use web store wisely!

Before using web storage, check browser support for localStorage and sessionStorage:

```javascript
if(typeof(Storage) !== "undefined") {
    // Code for localStorage/sessionStorage.
} else {
    // Sorry! No Web Storage support..
}
```
A server-sent event is when a web page automatically gets updates from a server.

Before, the web page would have to ask if any updates were available. With server-sent events, the updates come automatically.

Examples: Facebook/Twitter updates, stock price updates, news feeds, sport results, etc.
The HTML5 Geolocation API is used to get the geographical position of a user. Since this can compromise user privacy, the position is not available unless the user approves it.

Demo: [http://slides.html5rocks.com/#geolocation](http://slides.html5rocks.com/#geolocation)
http://www.w3schools.com/css/css3_intro.asp

http://slides.html5rocks.com/#css3-title
The CSS browser prefixes are:

Android: -webkit-
Chrome: -webkit-
Firefox: -moz-
Internet Explorer: -ms-
iOS: -webkit-
Opera: -o-
Safari: -webkit-
Just a few years ago, to set a rounded corner on a box you had to write:

```
-moz-border-radius: 10px 5px;
-webkit-border-top-left-radius: 10px;
-webkit-border-top-right-radius: 5px;
-webkit-border-bottom-right-radius: 10px;
-webkit-border-bottom-left-radius: 5px;
border-radius: 10px 5px;
```

But now you really only need the standards version:

```
border-radius: 10px 5px;
```
References

1. http://www.w3schools.com/
Social Coding
The Cathedral and the Bazaar

Eric S. Raymond

Cathedral model:

Source code is available with each software release, but code developed between releases is restricted to the code developers.
The Cathedral and the Bazaar

Eric S. Raymond

Bazaar model:
Code is developed over the Internet in view of the public.
The Bazaar

- Release early. Release often
- More transparent, users as co-developers
- Greater visibility of bugs, lots of testers
- Recognize good ideas (from others)
- More scrutiny, and experimentation possible.
Development is Collaborative
Development is Collaborative
Development is Collaborative
Development is Collaborative
Version Control System (VCS)

CVS, SVN, Git, Mercurial, Dropbox(!), etc.

- Keeps multiple (older and newer) versions of everything (not just source code).
- Requests comments regarding every change.
- Typically synchronize through “check in” and “check out”.
- Displays differences between versions.
Version Control System (VCS)

**Local** Version Control saves changes to files in a database.

**Centralized** Version Control saves changes to a shared server.

**Distributed** Version Control allows for easier sharing of files than LVC and also eliminates problems that could occur if access to the server is lost under a CVC system.
DVC clients have a complete backup of the files on their computer. If the server is lost the client just waits to regain contact and then uploads changes.

-- Each client has a complete history of all changes stored locally.
-- The client can also access all changes made to the files historically with a simple command.
-- Git monitors all file change, constantly.
Distributed systems like Mercurial and Git are newer and are gradually replacing centralized systems like CVS and Subversion.
Git

- Created in 2005
- Designed for speed
- Support for many parallel branches
- Distributed, promotes local work
- Able to handle large projects
Git (Create a new repository)

touch README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/mashiyat/CSC309.git

git push -u origin master
Git (pushing existing repository)

```
  git remote add origin https://github.com/mashiyat/CSC309.git
  git push -u origin hotfix
```
File status lifecycle in Git

- untracked
- unmodified
- modified
- staged

- edit the file
- stage the file
- commit

- add the file
- remove the file
Git

For free private repo

https://bitbucket.org/
Social Coding (Real Time)

For real time collaboration

https://c9.io/
Task Tracking System

GitHub

https://github.com/jquery/jquery-mobile/issues

Other Free alternative

https://Freedcamp.com
Organizing your tasks and thoughts

Trello (https://trello.com)

https://www.youtube.com/watch?v=aaDf1RqeLfo#t=15

Online Task Manager

https://www.online-task-manager.com/

-- Trello is weak in tracking time estimations.
References

http://en.wikipedia.org/wiki/The_Cathedral_and_the_Bazaar
http://www.newthinktank.com/2014/04/git-video-tutorial/