

<HTML5>

*NEW AND IMPROVED*

*Thanks to Timothy Fisher*

<http://www.slideshare.net/timothyvf/html5-new-and-improved>

- More Semantic HTML tags

- Media Tags

- Geolocation

- Canvas

- Input Types

- Form Validation

- Local Storage

- WebSQL Storage

- Offline Applications

- Draggable

- Cross-Domain Messaging

- Web Sockets

- Web Workers

- History API

- HTML 5 Support



# HTML5 History

- Specification of HTML published by W3C
- W3C HTML5 Spec=> 900+ pages Work
- started on HTML5 in late 2003
- First Working Draft published January 2008
- Present Working Draft 5.1- 2015
- Recommendation - 2022 or later \*

\* Requires 2 100% complete and fully interoperable implementations

# Less Header Code

## Pre HTML5:

```
DOCTYPE HTML PUBLIC "-//W3C//Dtd HTML 4.01 Transitional//EN" "http://  
www.w3.org/tr/html4/loose.dtd">  
html>  
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
  <title>Awesome Stuff</title>  
</head>
```

## With HTML5:

```
DOCTYPE HTML>  
html>  
<head>  
  <meta charset="utf-8">  
  <title>Awesome Stuff</title>
```

# No Need for Type Attribute

## Pre HTML5:

```
<script type="text/javascript" src="script.js"> </script>  
<link type="text/css" href="style.css"></link>
```

## With HTML5:

```
<script src="script.js"> </script>  
<link href="style.css"></link>
```

# MORE Semantic HTML tags

```
<div id="header">
```

```
<div id="nav">
```

```
<div id="sidebar">
```

```
<div id="article">
```

# MORE SEMANTIC HTML TAGS

`<header>`

`<nav>`

`<aside>`

`<section>`  
`<article>`

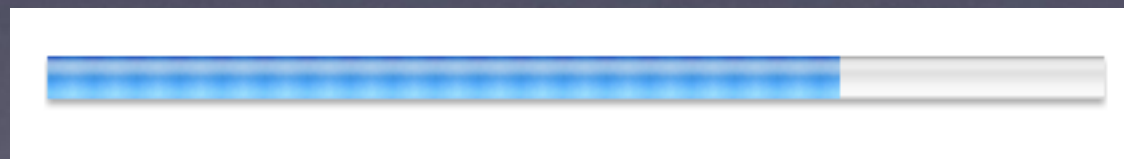
# MORE Semantic HTML tags

## Output

```
<output name="result"></output>
```

## Progress

```
<progress id="p" max=100><span>0</span>%</progress>
```





## Meter

Storage space usage:

```
<meter value=6 max=8>6 blocks used (out of 8 total)</meter>
```

Server turnout:

```
<meter value=0.75></meter>
```

Tickets sold:

```
<meter min="0" max="100" value="75"></meter>
```

- Meter: empty
- Meter: full
- Meter: "a bit"
- Preferred usage
- Too much traffic
- Optimum value



## Details and Summary

etails>

<summary>

American League Central Division

</summary>

Detroit Tigers<br/>

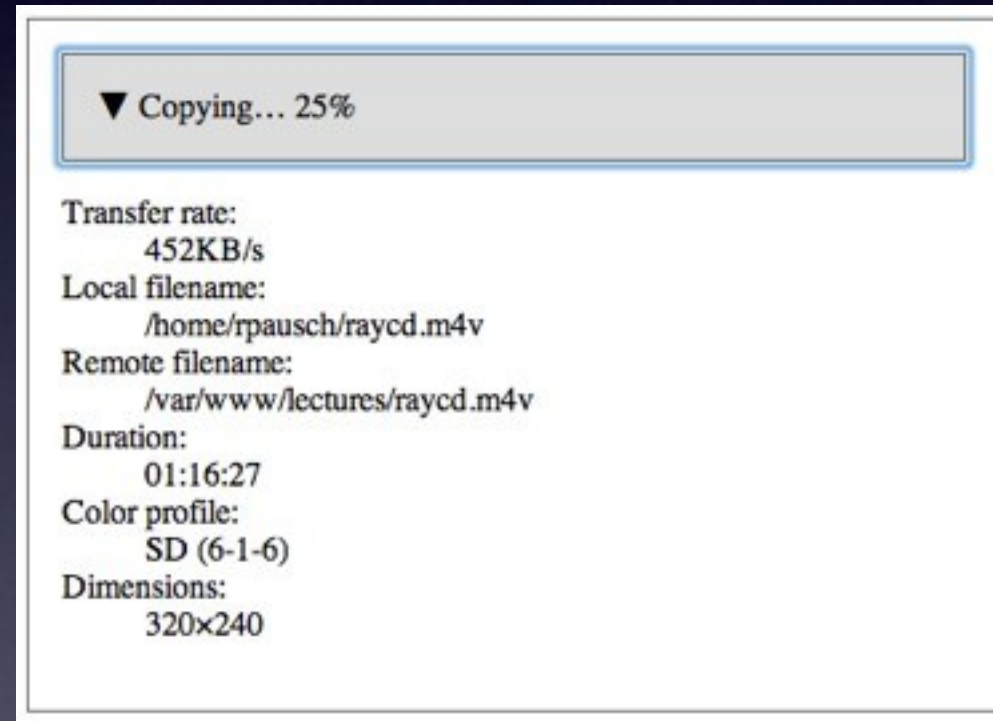
Minnesota Twins<br/>

Chicago White Sox<br/>

Cleveland Indians<br/>

Kansas City Royals<br/>

etails>



Use to create an expanding and contracting element that you can use to hide

# More Essential HTML tags

## Address

```
address>
```

```
  Written by:<br/>
```

```
  <a href="/people/show/23">Timothy Fisher</a>, <br/>
```

```
  Address: 25296 Hunter Lane, Flat Rock, MI48134 <br/>
```

```
  Phone: 555-1212
```

```
</address>
```

**address** applies to the nearest **Article** or **Body** tag.

Prior to HTML5 the Address element applied to the document/body as a whole.

# More HTML tags

## Data Attributes

```
div class="car" data-brand="ford" data-model="mustang">  
<button class="fire">  
div>
```

### *using DOM's `getAttribute()` property*

```
brand=mydiv.getAttribute("data-brand") //returns "ford"  
mydiv.setAttribute("data-brand", "mazda") //changes "data-brand" to "mazda"  
mydiv.removeAttribute("data-brand") //removes "data-brand" attribute entirely
```

### *using JavaScript's `dataset` property*

```
brand=mydiv.dataset.brand //returns "ford"  
mydiv.dataset.brand='mazda' //changes "data-brand" to "mazda"  
mydiv.dataset.brand=null //removes "data-brand" attribute
```

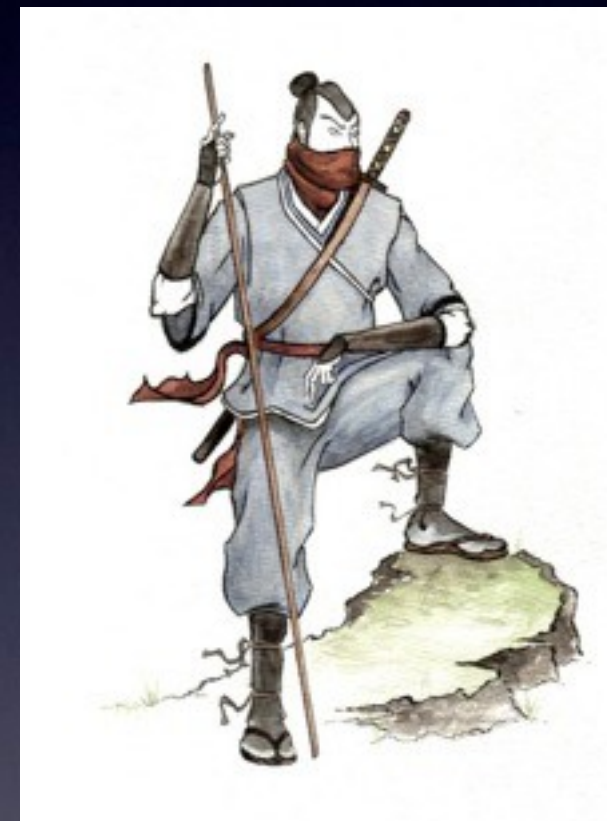
# More XHTML 1.1ML tags

## ction, hgroup, Article

```
</article>
</hgroup>
  <h1>Mobile Phones</h1>
  <h2>Different Smart Phones</h2>
</hgroup>
<p>Some of the more popular mobile smart phones</p>
<section>
  <h1>Apple iPhone</h1>
  <p>A popular smart phone from Apple.</p>
</section>
<section>
  <h1>Android-based Phones</h1>
  <p>A series of smart phones that use the Google Android operating system.</p>
</section>
</article>
```

## Figure and Figure Caption

```
figure>  
    
  <figcaption>Cool Ninja Guy</figcaption>  
</figure>
```



Cool Ninja Guy

## Menu and Command

```
<menu label="Hero List">  
<command type="radio" radiogroup="herolist" label="Spiderman">  
<command type="radio" radiogroup="herolist" label="Superman">  
<command type="radio" radiogroup="herolist" label="Batman">  
</menu>
```

a simple radiobutton group

## Menu (continued)

```
menu type="toolbar">
```

```
li>
```

```
<menu label="File">
```

```
<button type="button" onclick="file_new()">New...</button>
```

```
<button type="button" onclick="file_open()">Open...</button>
```

```
<button type="button" onclick="file_save()">Save...</button>
```

```
<button type="button" onclick="file_saveas()">Save As...</button>
```

```
</menu>
```

```
/li>
```

```
li>
```

```
<menu label="Edit">
```

```
<button type="button" onclick="edit_copy()">Copy...</button>
```

```
<button type="button" onclick="edit_cut()">Cut...</button>
```

```
<button type="button" onclick="edit_paste()">Paste...</button>
```

```
</menu>
```

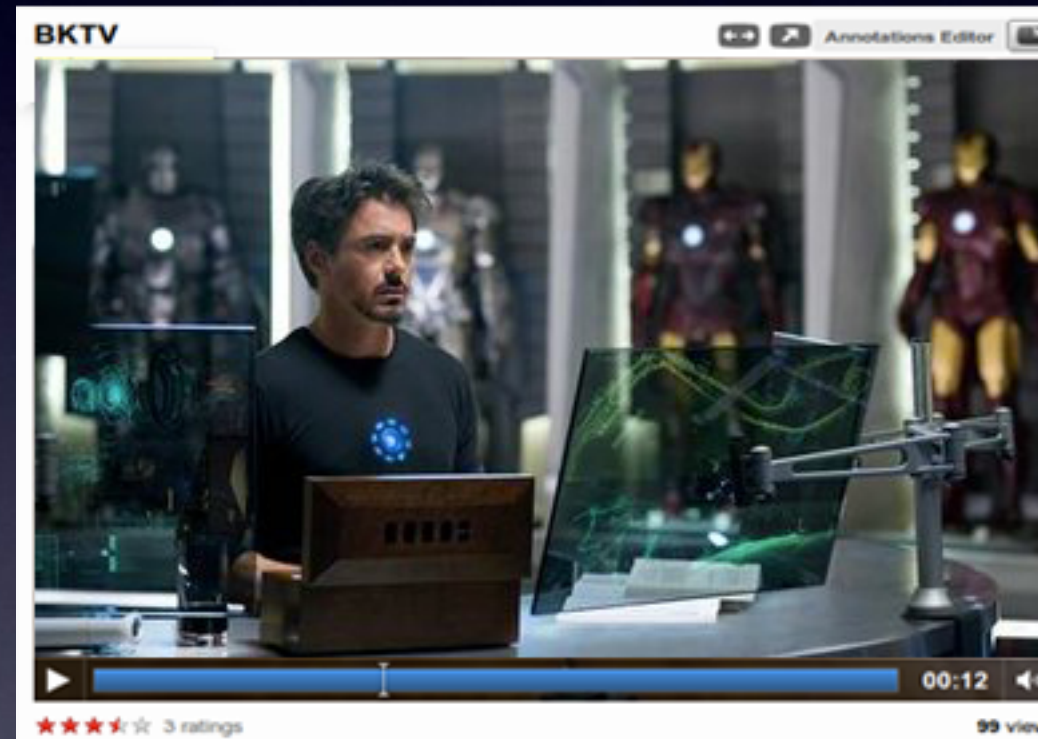


# Media Tags

```
<video src="ironman.ogg" />
```

**Automatically show native controls**

```
<video src="ironman.ogg" controls />
```



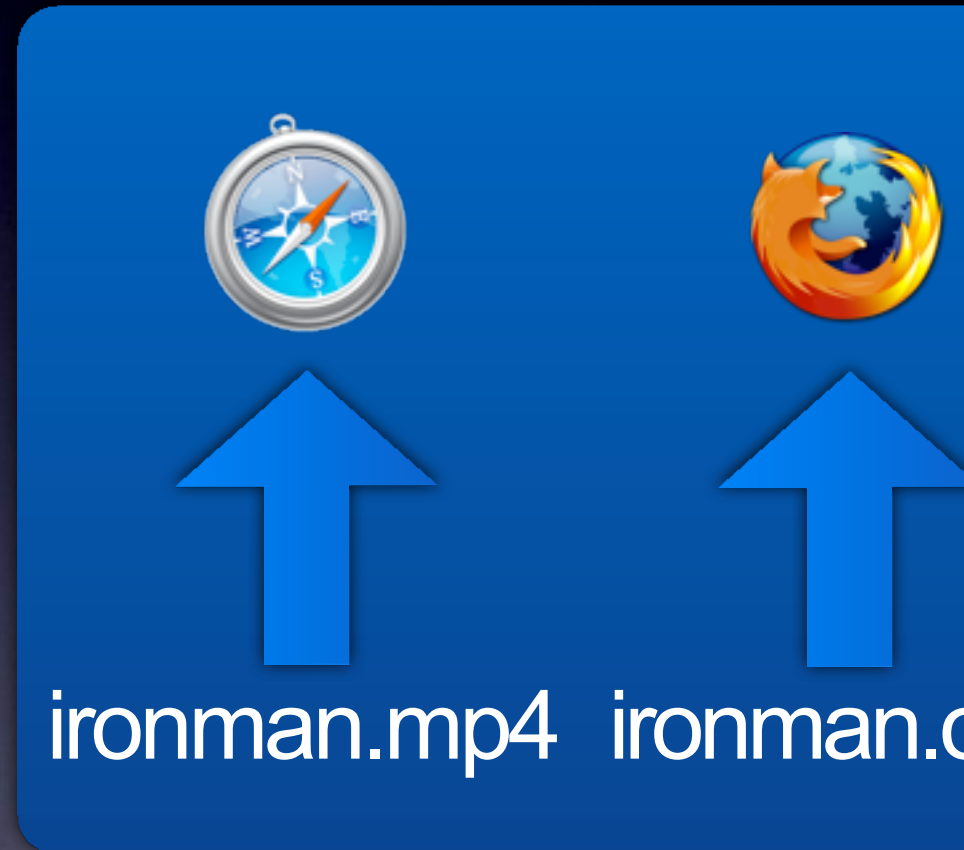
# Media Tags

```
video controls/>
```

```
<source src="ironman.mp4" />
```

```
<source src="ironman.ogg" />
```

```
video>
```



Specify multiple source elements to support more browsers

(i.e. mp4 will work in Safari ogg will work in Firefox)

# Media Tags

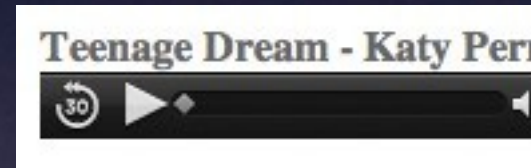
```
<audio src="teenage_dream.mp3"></audio>
```

```
<audio controls>
```

```
<source src="teenage_dream.mp3"/>
```

```
<source src="teenage_dream.ogg"/>
```

```
</audio>
```



**Provides a download link for non-supporting browsers:**

```
<audio src="teenage_dream.ogg" autoplay controls loop>
```

```
<a href="teenage_dream.ogg">download</a>
```

```
</audio>
```

# Native Geolocation

Build location-aware apps without access to native mobile a

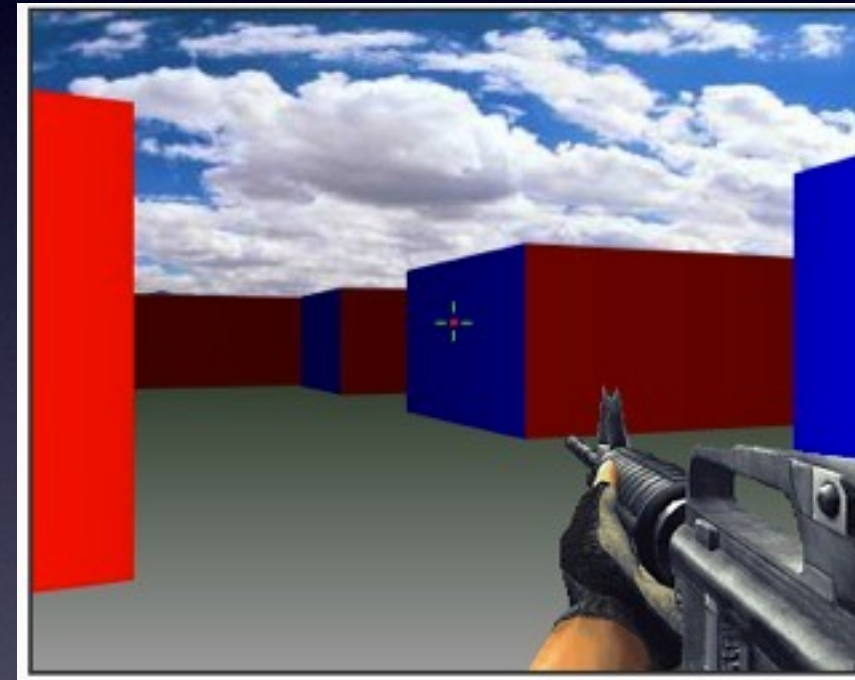
```
navigator.geolocation.getCurrentPosition(  
  function(position) {  
    // display position  
  }  
);
```



## A complete drawing and animation API

```
canvas id="square">  
  fallback content  
canvas>
```

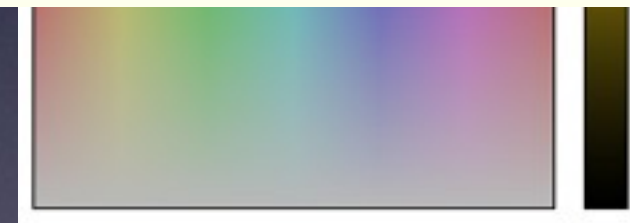
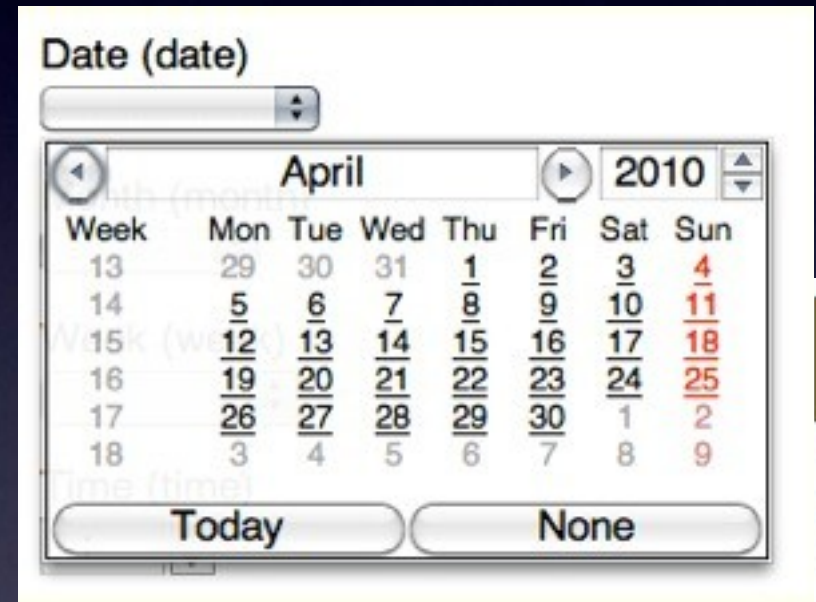
```
script>  
  // create basic filled square  
  canvas = canvas.getElementById('square');  
  context = canvas.getContext('2d');  
  context.fillStyle = "#000000";  
  context.fillRect(0, 0, 100, 100);  
script>
```



# Input Types

`input type="email" />`

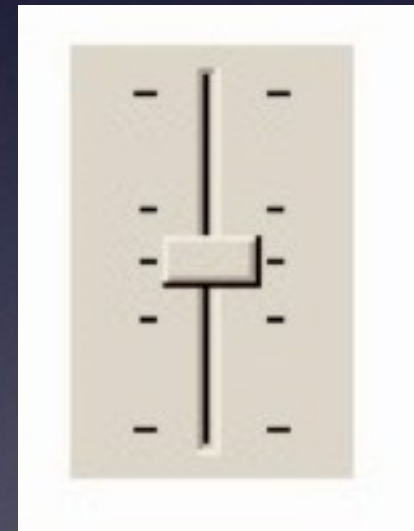
el                    datetime  
earch                date range  
email url            color



Unsupported browsers default to text type

## Input Type Range + Datalist

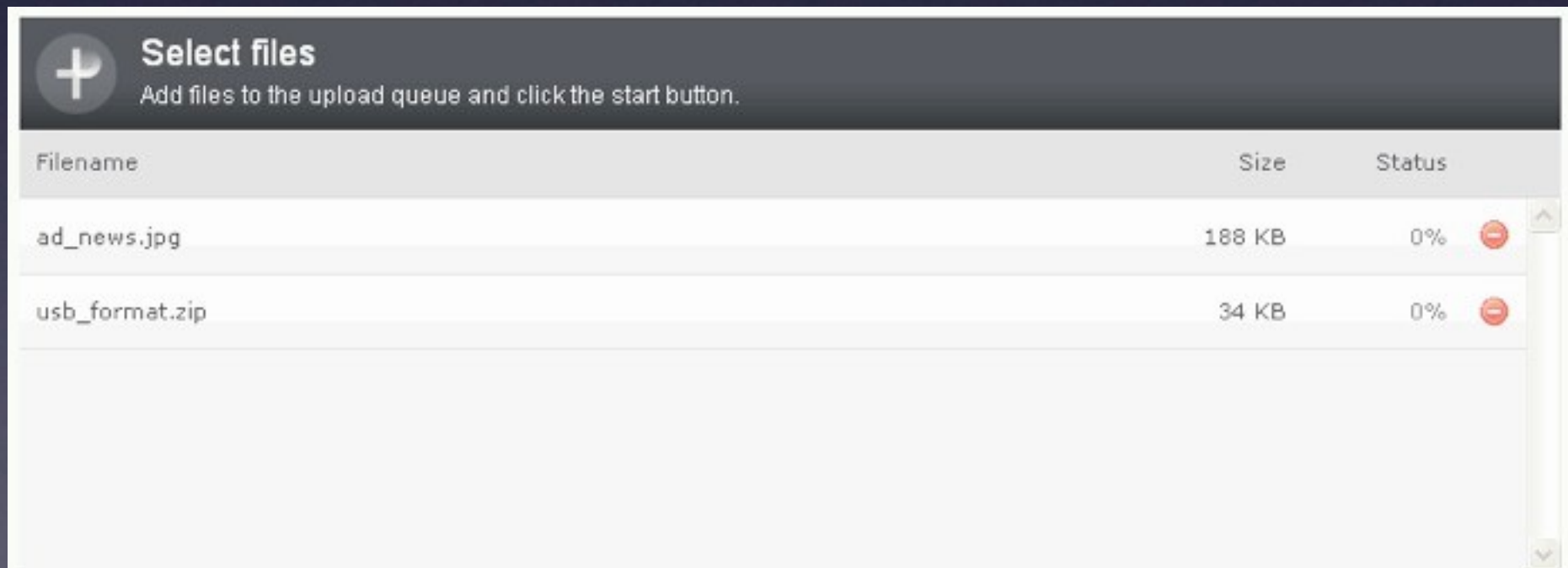
```
<input type="range" min="-100" max="100" value="0"
  step="10" name="power" list="powers">
<datalist id="powers">
  <option value="0">
  <option value="-30">
  <option value="30">
  <option value="+50">
</datalist>
```



# Input Types

## File Upload Multiple

`<input type=file multiple>`

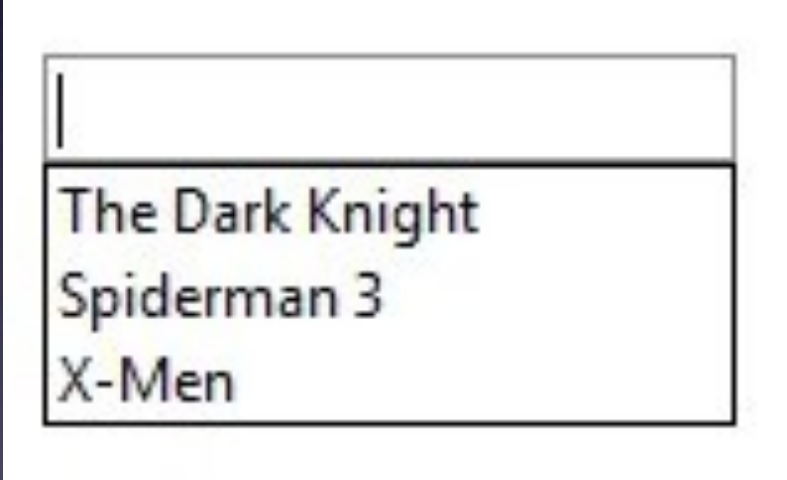




# Input Types

## datalist

```
<input list="movies" />  
<datalist id="movies">  
  <option>The Dark Knight</option>  
  <option>Spiderman 3</option>  
  <option>X-Men</option>  
</datalist>
```



The image shows a browser window with a text input field. A dropdown menu is open below the input field, displaying three options: "The Dark Knight", "Spiderman 3", and "X-Men".

Used to provide Auto Complete feature

# Form Validation

```
input name="custname" required>
```

```
script>
```

```
form.checkValidity();
```

```
script>
```

A screenshot of a web form with several fields and a submit button. The form is titled "Form Validation" and contains the following elements:

- Username:** A text input field containing "blogfreakz". A red error message bubble points to it, saying "This field is required".
- Password:** An empty text input field. A red error message bubble points to it, saying "This field is required".
- Password Confirmation:** An empty text input field. A red error message bubble points to it, saying "Select between 1 and 3 checkboxes".
- Speciality:** A group of checkboxes with the following labels and states:  jQuery,  JavaScript,  Rails,  PHP,  Wordpress,  Other.
- About you:** A text input field containing "CakePHP lover jQuery fans".
- Submit Button:** A button labeled "(Try to) Submit".

## Custom Validation

```
<label>Gender: </label>
```

```
<input name="gender" type="text" oninput="check(this)">
```

```
<script>
```

```
function check(input) {
```

```
  if (input.value != "male" && input.value != "female")
```

```
    { input.setCustomValidity("'" + input.value + "' is not a  
    gender.');
```

```
  }
```

```
  else {
```

```
    // input is good - reset error message
```

```
    input.setCustomValidity("");
```

```
  }
```

# Local / Session Storage

```
sessionStorage.setItem(key, value);  
sessionStorage.getItem(key);
```

```
localStorage.setItem(key, value);  
localStorage.getItem(key);
```



Save key/value pairs to a client-side data store implemented by browser

# WebSQL Storage

Set of APIs to manipulate client-side databases using SQL

*Open/create a database*

```
var db = openDatabase(db_name, version, db_desc, est_size);
```

*Create a table and insert some data*

```
transaction(function (tx) {  
  tx.executeSql('CREATE TABLE foo (id unique, text)');  
  tx.executeSql('INSERT INTO foo (id, text) VALUES (1, "synergies")');
```

*Select data and display it*

```
executeSql('SELECT * FROM foo', [], function (tx, results)  
{  
  var len = results.rows.length, i;  
  for (i = 0; i < len; i++)  
    { alert(results.rows.item(i).text);
```



# Offline Applications

## Offline Applications using manifest

```
<html manifest="cache.manifest">
```

Provide a cache.manifest file:

```
CACHE MANIFEST  
clock.html  
clock.css  
clock.js
```

uses **MIME type:**  
text/cache-manifest



Run a web application in offline mode, disconnected from Internet

# Online Applications

## Detect Online or Offline

```
window.addEventListener("online", function()  
    { do_something();  
}, true);
```

```
window.addEventListener("offline", function()  
    { do_something();  
}, true);
```



# Draggable

```
<div draggable="true"></div>
```

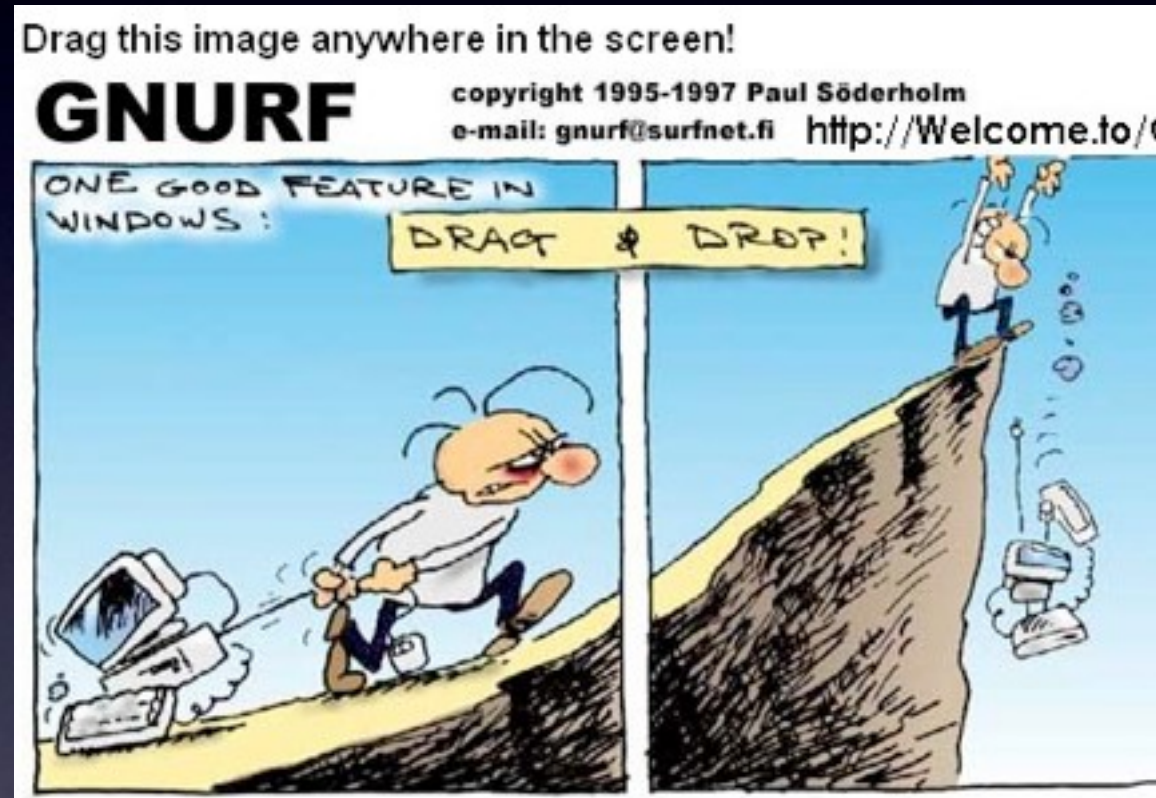
*Set data to access at target*

```
Event(div, "dragstart", function(e)  
  e.dataTransfer.setData('foo',  
  'bar');  
  e);
```

*Access data from dragged object*

```
Event(div, 'dragend', function(e)  
  e.dataTransfer.getData('foo');  
  e);
```

HTML5 drag and drop should work across frames, and across browser windows.





# Cross-Domain Messaging



*Sender*

```
o = document.getElementsByTagName('iframe')[0];
```

```
o.contentWindow.postMessage('Hello world', 'http://b.example.org/');
```

*Recipient*

```
addEventListener(window, "message", function(e)
```

```
document.getElementById("rcvdr_message").innerHTML += e.origin + " said: " +
```

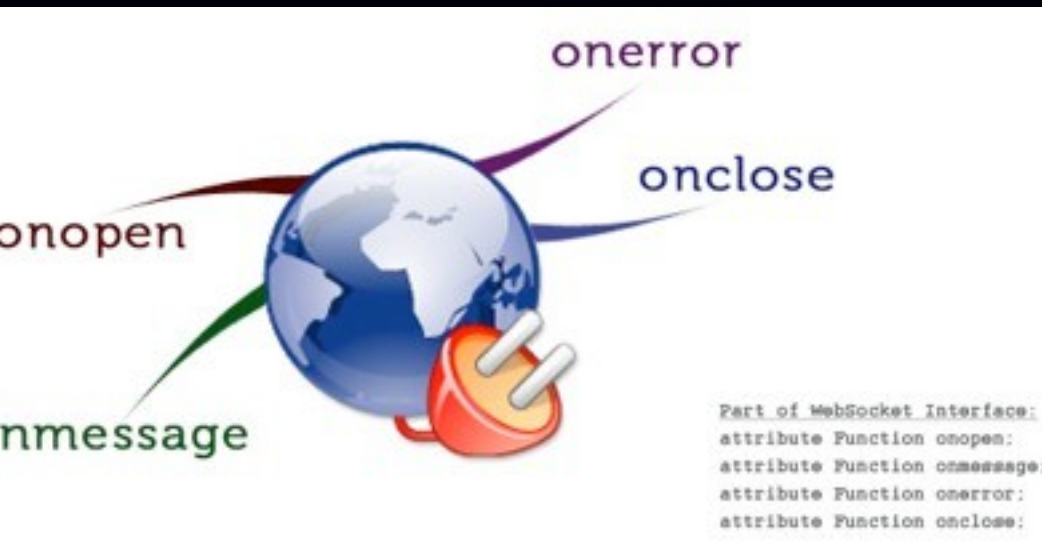
# Editable Content



Turn any element into an editable area

```
ipt>  
document.getElementById('notepad').contentEditable = true;  
ript>
```

# Web Sockets



- Opens a persistent connection to the server
- Can be used for server to browser push
- Restricted communication to origin server
- Eliminates need to poll for data

```
var ws = new WebSocket("ws://friendfeed.com/websocket");  
ws.onopen = function() {  
    ws.send("This is a message from the browser to the server");  
};  
ws.onmessage = function(event) {  
    alert("The server sent a message: " + event.data);  
};
```

# WEB WORKERS

- Provide “threads” for JavaScript execution
- Don't have access to DOM or page.
- Have to communicate through postMessage API



# WEB WORKERS

## In the Browser

*// Create a Web Worker*

```
var worker = new Worker("worker.js");
```

*// Post a message to the Web Worker*

```
worker.postMessage(0);
```

*// Triggered by postMessage in the Web Worker*

```
worker.onmessage = function(evt) {  
  // evt.data is the values from the Web Worker  
  alert(evt.data);  
};
```

*// Catch Web Worker error*

```
worker.onerror = function(evt)  
{ alert(evt.data);
```



## In the Web Worker

```
// Triggered by postMessage in the page  
onmessage = function(evt) {  
  // evt.data will be 0 here  
  for (var i=evt.data, k=1000001;i<k; i++) {  
    // Continually sends data back  
    postMessage(i);  
  };  
};
```



# History API

JavaScript API for moving through browser history

```
window.history.back();
```

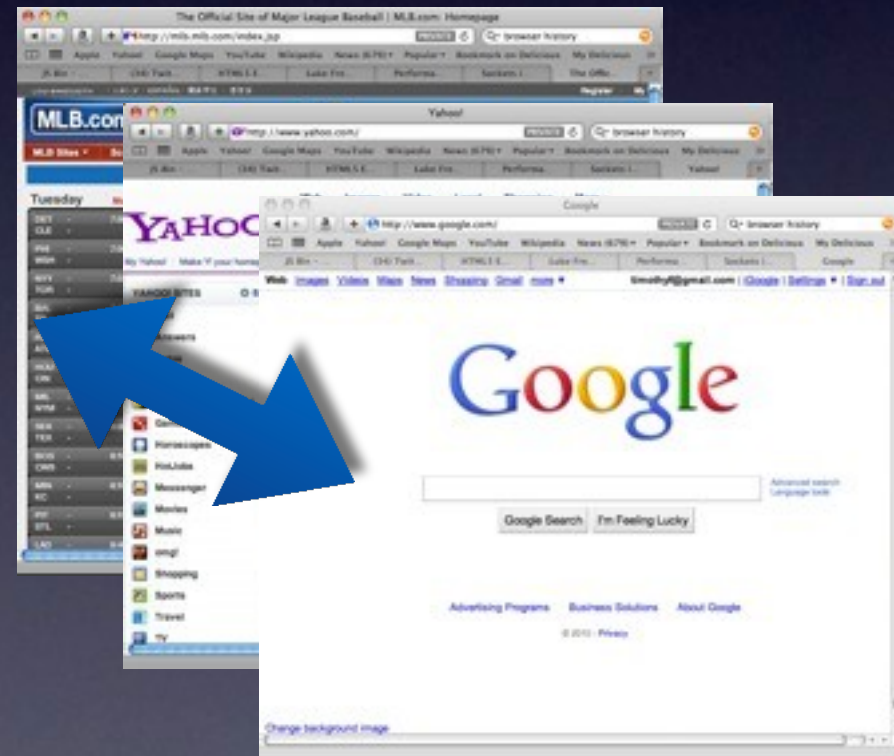
```
window.history.forward();
```

```
window.history.go(2);
```

```
window.history.length;
```

```
window.history.pushState(data, title, url);
```

```
window.history.replaceState(data, title, url);
```



# HTML5 Compatibility

HTML5Test.com scores browsers according to the following criteria:

- Parsing Rules
- Canvas
- Video
- Audio
- Local Devices
- Elements
- Forms
- User Interaction
- Microdata
- Web Applications
- Geo Location
- WebGL
- Communication
- File Storage
- Workers



# HTML5 Compatibility

<http://www.HTML5test.com/>

# Progressive Enhancement

Use HTML5 when available

Fallback to a different mechanism when not available

Supported by many libraries including jQuery.

HTML5 video is not supported, flash video will load

```
video controls width="500">
```

```
<source src="video.ogv" />
```

```
<source src="video.mp4" />
```

```
<embed src="http://blip.tv/play/gcMV" type="application/x-shockwave-flash"  
width="1024" height="798" allowscriptaccess="always"
```

# DON'T FORGET CSS3

- Rounded corners
- Box shadows
- Transitions
- Rotate
- Gradients
- Text shadow
- Web fonts



# Recommended Sites

<http://www.HTML5test.com>

<http://html5demos.com>

<http://caniuse.com>