HTML5

SOFTWAVES CONSULTANCY BANGALORE PVT LTD

What’s the big deal about HTML5

- Its not one big thing
- You don’t need to throw anything away
- Its easy to get started
- It already works
- Its here to stay

History of HTML

- W3C: HTML 4, XHTML
- W3C: XHTML2.0
- WHATWG: HTML4, HTML5
- W3C + WHATWG

Browser Support

- Not everyone supports everything.
- But we all get there
- There is no Single Winner

Detecting the presence of HTML5

- Check if a certain property exists on a global object: geolocation API
- Create an element, then check if a certain property exists on that element.: <canvas>
- Create an element, check if a certain method exists on that element, then call the method and check the value it returns.: <video>
- Create an element, set a property to a certain value, then check if the property has retained its value.: <input>

#1

```
function supports_local_storage() {
try {
   return 'localStorage' in window &&
     window['localStorage'] !== null;
} catch(e){
   return false;
}
}
```

#2

```
function supports_canvas() {
   return
     !!document.createElement('canvas').getContext;
}
```

#3

```
function supports_h264_baseline_video()
{
   if (!supports_video()) { return false; }
   var v = document.createElement("video");
   return
     v.canPlayType('video/mp4;codecs="avc1.42E01E,mp4a.40.
     2"');
}
```

#4

```
var i =
  document.createElement("input");
i.setAttribute("type", "color");
```

Readymade Detections

- Modernizr Detection library
  - <script src="modernizr.min.js"></script>
  - http://www.modernizr.com/downloads/modernizr-2.5.1.js
- Contains a set of Boolean properties for each feature it can detect
- HTML5 Test
  - http://html5test.com/

HTML5 Gallery

- http://www.weheart.co.uk/
- http://beta.dublinbikes2go.com/#home
- http://www.the-art-of-web.com/css/css-animation/
- http://www.mba-multimedia.com/
- http://www.timer-tab.com/
- http://www.bifter.co.uk/
- http://www.the-art-of-web.com/css/css-animation/

Best Starrers

- Canvas
- Media
- Geolocation
- Local Storage
- Form Elements
- Miscellaneous

DOCTYPE

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

- Microsoft started DOCTYPE
- Quirks mode
- Standards Mode
- Almost Standards Mode
- <!DOCTYPE html>

Some More Changes

```
<html xmlns="http://www.w3.org/1999/xhtml"
lang="en“ xml:lang="en">
```

To

```
<html lang="en">
```

```
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
```

To

```
<meta charset="utf-8" />
```

New Semantic Elements

- <section>
- <nav>
- <article>
- <aside>
- <hgroup>
- <header>
- <footer>
- <time>
- <mark>

<section>

- A section is a section!
- A block of information. A section element represents a generic document or application section, thematic grouping of content, typically with a heading.
- E.g. – Chapters, tabbed pages

<nav>

- An nav element represents a section of a page that links to other pages or to parts within the page: a section with navigation links.
- E.g. – Menu items

<article>

- An article element represents a component of a page that consists of a self-contained composition in a document, page, application, or site and that is intended to be independently distributable or reusable.
- E.g. – Forum Posts, a user-submitted comment, an interactive widget or gadget.

<aside>

- An aside element represents a section of a page that consists of content that is tangentially related to the content around the aside element, and which could be considered separate from that content.
- E.g. – Pull quotes, Advertising bars, Feeds from other Blogs

<hgroup>

- The hgroup element represents the heading of a section. This element is used to group a set of h1–h6 elements when the heading has multiple levels, such as subheadings, alternative titles, or taglines.

<header>

- The header element represents a group of introductory or navigational aids.
- Usually contains the section’s heading (an h1–h6 element or an hgroup element), but this is not required.
- header element can also be used to wrap a section’s table of contents, a search form, or any relevant logos.

<footer>

- the footer element represents a footer for its nearest ancestor sectioning content or sectioning root element. A footer typically contains information about its section such as who wrote it, links to related documents, copyright data, and the like.
- E.g. – Information of a Blog post, author, date, no. of comments

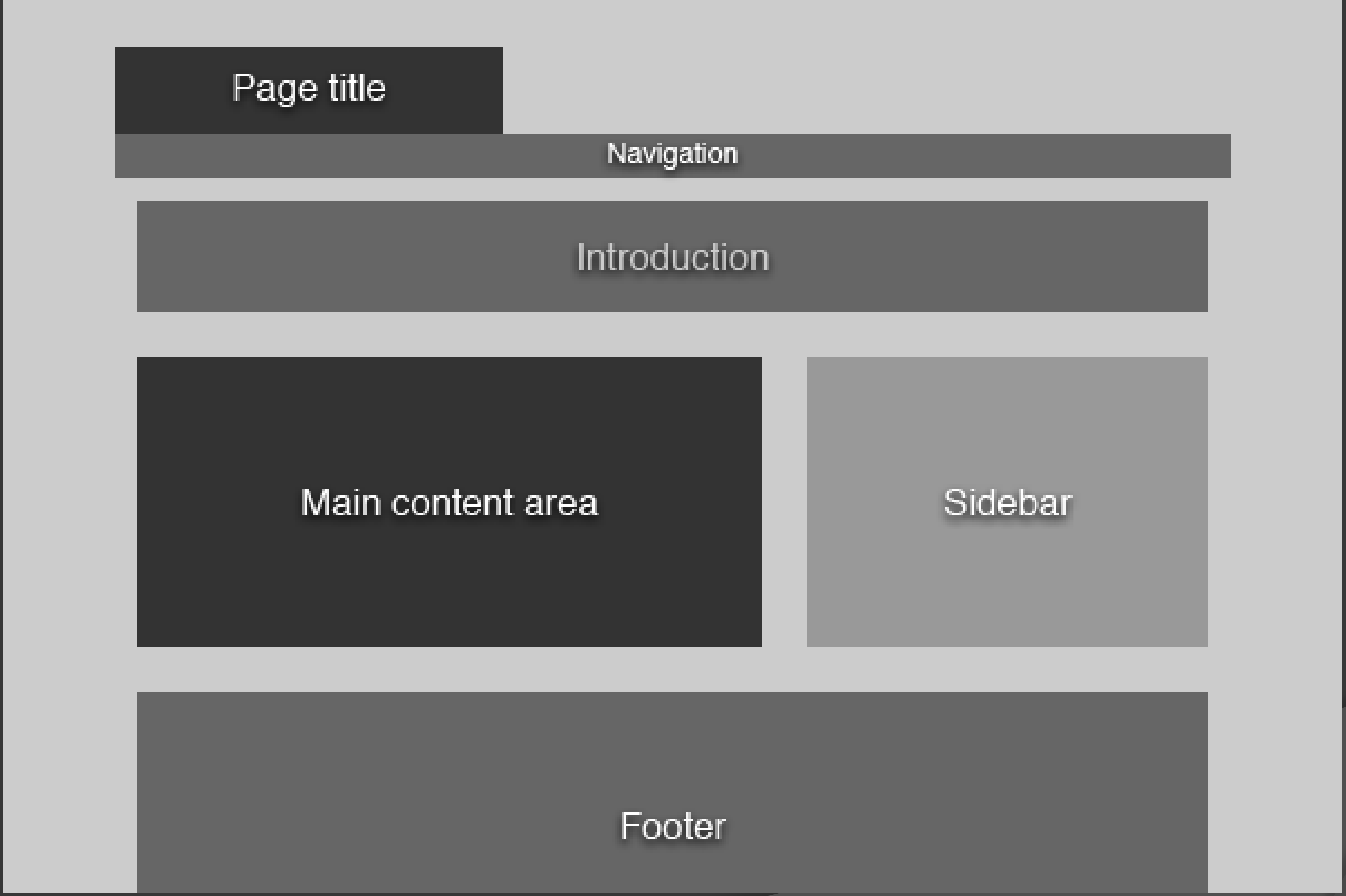
<time>

- The time element represents either a time on a 24 hour clock, or a precise date in the proleptic Gregorian calendar, optionally with a time and a timezone offset.

<mark>

- The mark element represents a run of text in one document marked or highlighted for reference purposes.

Create your own Blog



Skeleton

```
<!doctype html>
<html>
<head>
    <title>Page title</title>
</head>
<body>
    <header>
        <h1>Page title</h1>
    </header>
    <nav>
        <!-- Navigation -->
    </nav>
    <section id="intro">
        <!-- Introduction -->
    </section>
    <section>
        <!-- Main content area -->
    </section>
    <aside>
        <!-- Sidebar -->
    </aside>
    <footer>
        <!-- Footer -->
    </footer>
  </body>
</html>
```

Marking up Navigation

```
<nav>
    <ul>
        <li><a href="#">Blog</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Archives</a></li>
        <li><a href="#">Contact</a></li>
        <li class="subscribe"><a href="#">Subscribe via. RSS</a></li
    >
    </ul>
</nav>
```

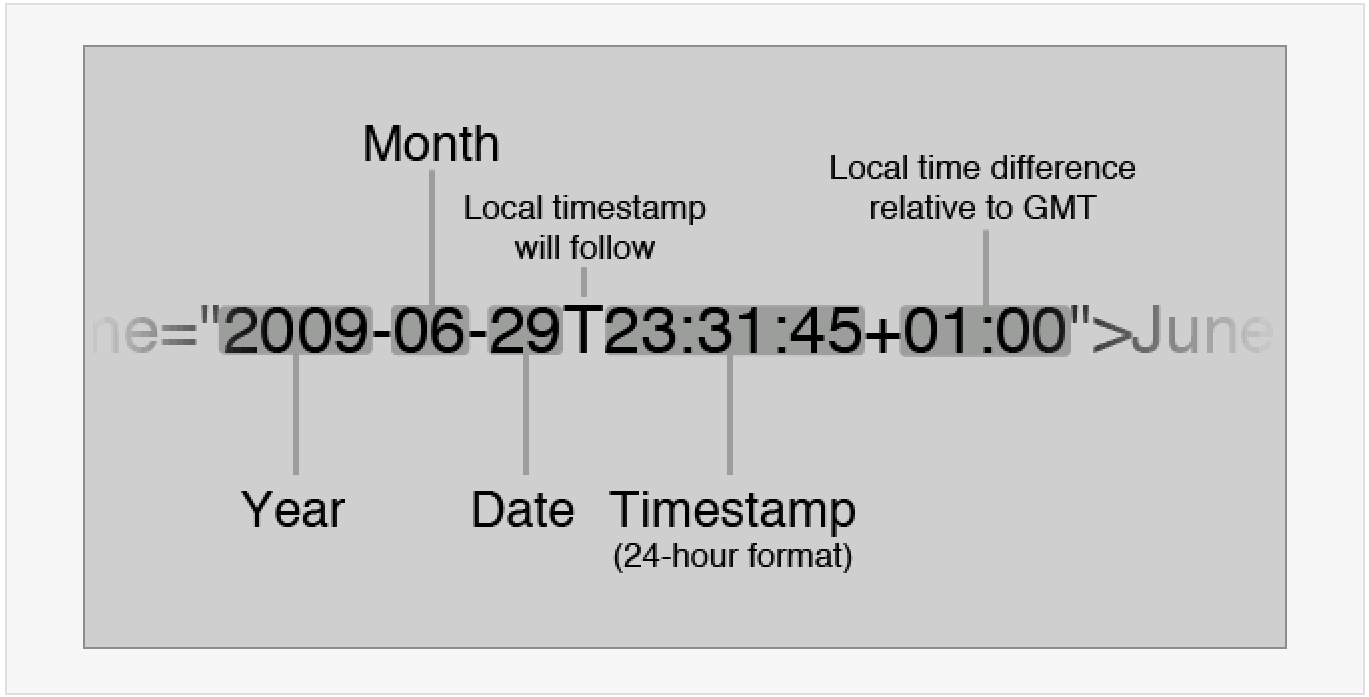
Marking up Introduction

```
<section id="intro">
    <header>
        <h2>Do you love flowers as much as we do?</h2>
    </header>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do e
    iusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
     minim veniam, quis nostrud exercitation ullamco laboris nisi ut.</p>

</section>
```

Marking up the Main content area

```
<section>
    <article class="blogPost">
        <header>
            <h2>This is the title of a blog post</h2>
            <p>Posted on
                <time datetime="2009-06-29T23:31:45+01:00">
                        June 29th 2009</time>
                by <a href="#">Mads Kjaer</a>
                <a href="#comments">3 comments</a>
            </p>
        </header>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing e
   lit. Proin euismod tellus eu orci imperdiet nec rutrum lacus
    blandit. Cras enim nibh, sodales ultricies elementum vel, f
   ermentum id tellus. Proin metus odio, ultricies eu pharetra
   dictum, laoreet id odio
        </p>
    </article>

</section>
```

Wait a minute…


Month
Local timestamp will follow
Local time difference relative to GMT
ne="2009-06-29T23:31:45+01:00">June
Year
Date
Timestamp
(24-hour format)

Marking up the Comments

```
<section id="comments">
    <header>
        <h3>Comments</h3>
    </header>
    <article>
        <header>
            <a href="#">George Washington</a> on <time datetime="2009-06-
    29T23:35:20+01:00">June 29th 2009 at 23:35</time>
        </header>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do ei
    usmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
     veniam, quis nostrud exercitation ullamco laboris nisi ut.</p>
    </article>
    <article>
        <header>
            <a href="#">Benjamin Franklin</a> on <time datetime="2009-06-
    29T23:40:09+01:00">June 29th 2009 at 23:40</time>
        </header>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do ei
    usmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
     veniam, quis nostrud exercitation ullamco laboris nisi ut.</p>
    </article>
</section>
```

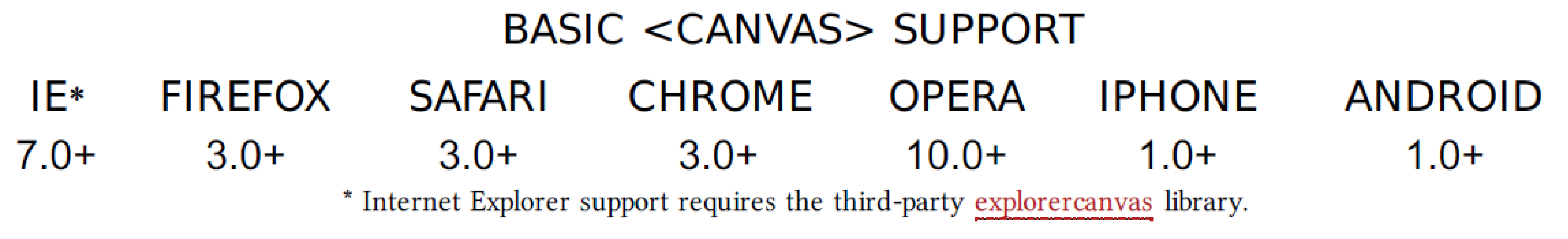
Now for some real stuff!

- Canvas
- Media
- Geolocation
- Local Storage

Canvas

- “The” Definition…

A resolution-dependent bitmap canvas which can be used for rendering graphs, game graphics, or other visual images on the fly



BASIC <CANVAS> SUPPORT

IE*	FIREFOX	SAFARI	CHROME	OPERA	IPHONE	ANDROID
7.0+	3.0+	3.0+	3.0+	10.0+	1.0+	1.0+

* Internet Explorer support requires the third-party explorercanvas library.

Basic tag

- `<canvas id=“b" width="300” height="225"></canvas>`
- Try this and check what happens!

Now try this…

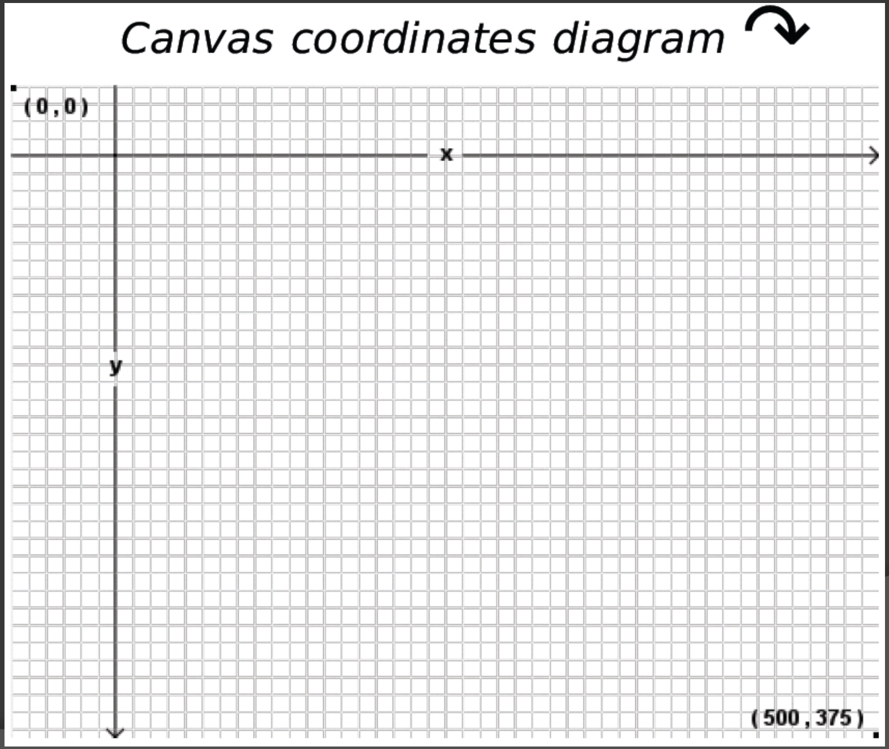
```
var b_canvas =
  document.getElementById("b");
var b_context =
  b_canvas.getContext("2d");
b_context.fillRect(50, 25, 150, 100);
```

Drawing Rectangles

- `fillStyle`
- `fillRect(x, y, width, height)`
- `strokeStyle`
- `strokeRect(x, y, width, height)`
- `clearRect(x, y, width, height)`

Drawing Paths

- Point to be noted – (0,0) always starts at the upper left corner



Drawing Paths

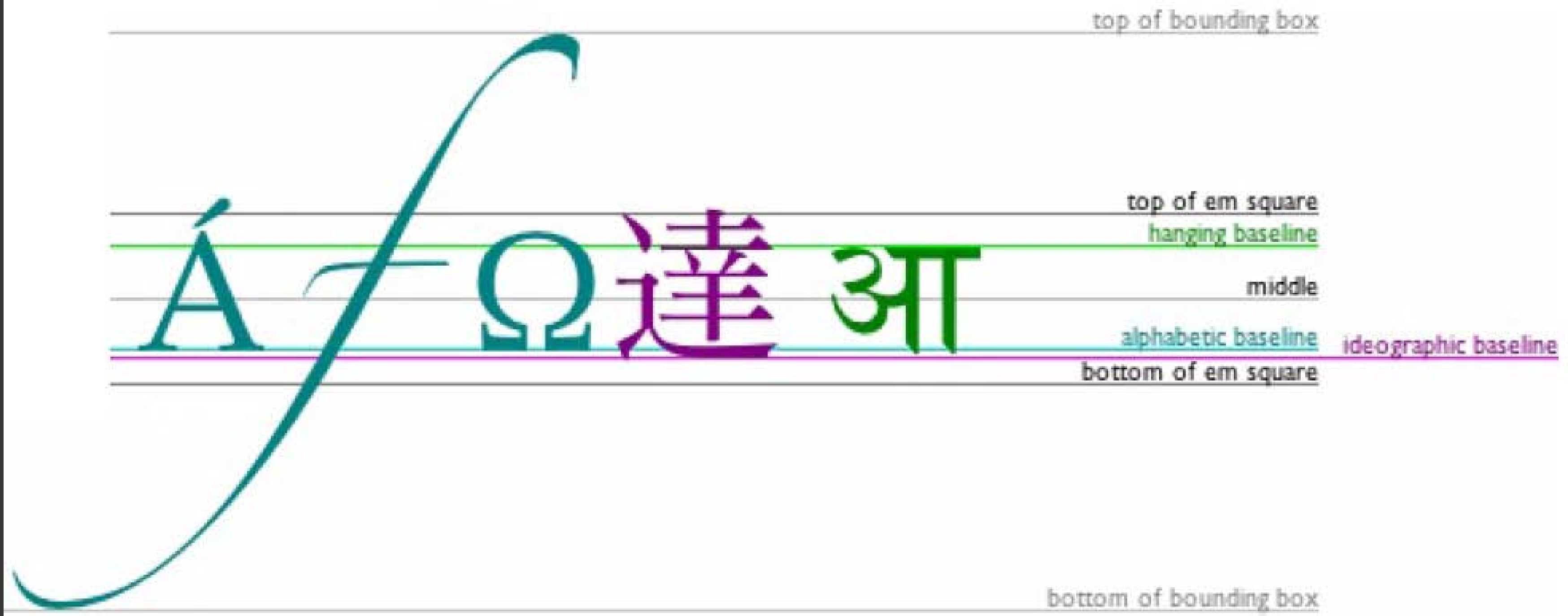
- Pencil Methods

```
context.moveTo(x, y);
context.lineTo(x, y);
```

- Ink Methods

```
context.strokeStyle = "#eee";
context.stroke();
```

Drawing Text



Drawing Text

```
context.font = "bold 12px sans-serif";
context.textBaseline = “top";
context.fillText(“Hello World", 248,
   43);
```

Creating Gradients

```
var my_gradient =
   context.createLinearGradient(0, 0, 0, 225);
my_gradient.addColorStop(0, "black");
my_gradient.addColorStop(1, "white");
context.fillStyle = my_gradient;
context.fillRect(0, 0, 300, 225);
```

- createRadialGradient(x0, y0, r0, x1, y1, r1)

Images

- `drawImage(image, dx, dy)`
- `drawImage(image, dx, dy, dw, dh)`
- `drawImage(image, sx, sy, sw, sh, dx, dy, dw, dh)`

Code

```
<img id="cat" src="images/cat.png"
   alt="sleeping cat"
width="177" height="113">
<canvas id="e" width="177"
   height="113"></canvas>
<script>
window.onload = function() {
var canvas = document.getElementById("e");
var context = canvas.getContext("2d");
var cat = document.getElementById("cat");
context.drawImage(cat, 0, 0);
};
</script>
```

[If IE]

- <!--[if IE]>
- <script src="excanvas.js"></script>
- <![endif]-->

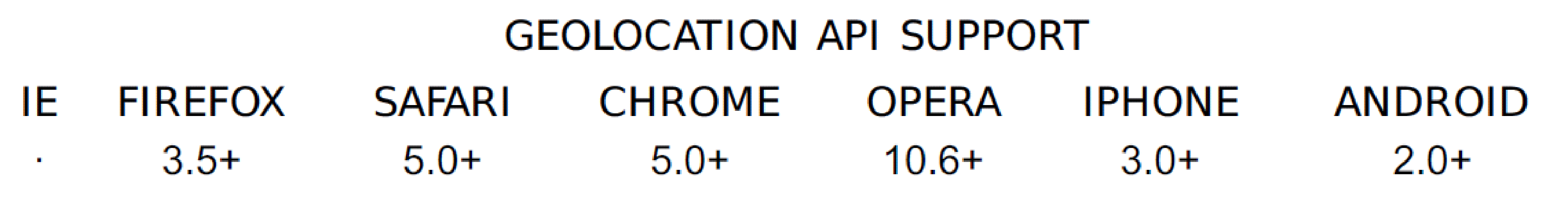
Halma- The Game

Light Stuff

- <input name="q" placeholder="Search Bookmarks and History">
- <input name="q" autofocus>
- <input type="checkbox">
- <input type="radio">
- <input type="password">
- <input type="file">
- <input type="submit">
- <input type="number"
  min="0"
  max="10"
  step="2"
  value="6">
- <input type="range"
  min="0"
  max="10"
  step="2"
  value="6">

GeoLocation API

- The geolocation API lets you share your location with trusted web sites.
- The latitude and longitude are available to JavaScript on the page,
- This is sent back to the remote web server to fetch fancy results



GEOLOCATION API SUPPORT

IE	FIREFOX	SAFARI	CHROME	OPERA	IPHONE	ANDROID
·	3.5+	5.0+	5.0+	10.6+	3.0+	2.0+

getCurrentPosition

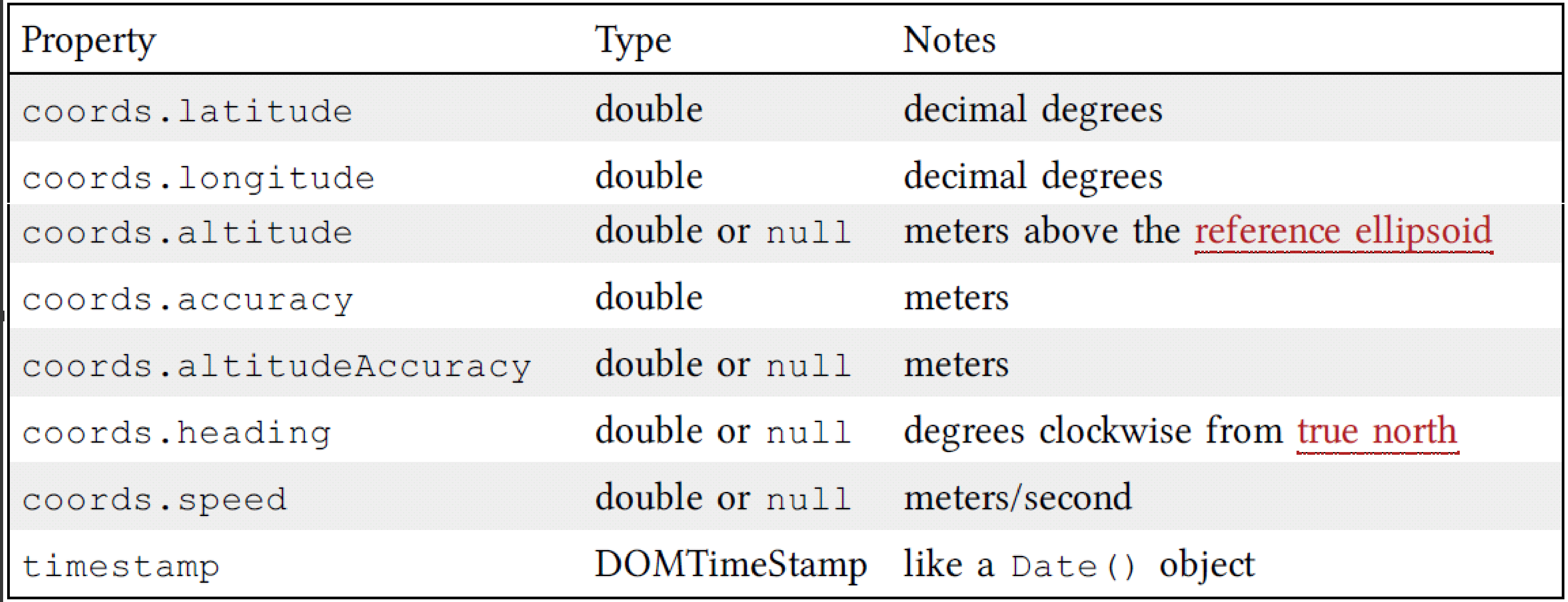
```
function get_location() {
navigator.geolocation.getCurrentPosition(show_map);
}
```

show_map is a callback function,

```
function show_map(position) {
var latitude = position.coords.latitude;
var longitude = position.coords.longitude;
// let's show a map or do something interesting!
}
```

`getCurrentPosition` actually takes 3 parameters – success(position), error(positionerror), positionoptions Object

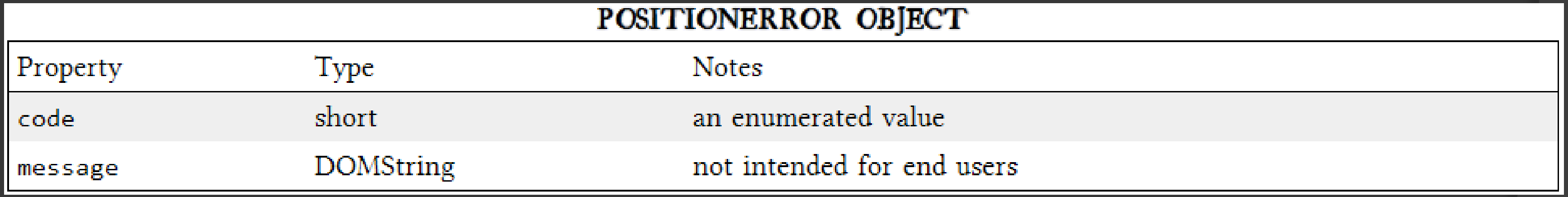
Positions Object



Property	Type	Notes
`coords.latitude`	double	decimal degrees
`coords.longitude`	double	decimal degrees
`coords.altitude`	double or `null`	meters above the reference ellipsoid
`coords.accuracy`	double	meters
`coords.altitudeAccuracy`	double or `null`	meters
`coords.heading`	double or `null`	degrees clockwise from true north
`coords.speed`	double or `null`	meters/second
`timestamp`	DOMTimeStamp	like a `Date()` object

PositionsError Object

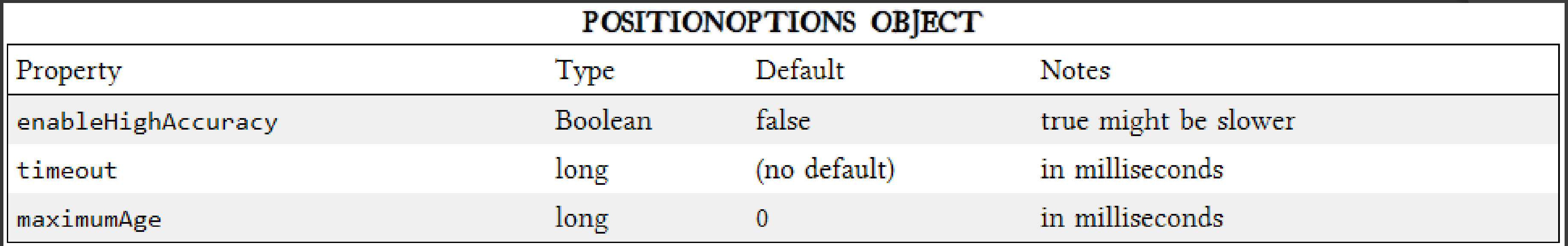
```
navigator.geolocation.getCurrentPosition(show_map, handle_error);
function handle_error(positionerror) {
   if (positionerror.code == 1)
   {
     // user said no!
   }
}
```



POSITIONERROR OBJECT

Property	Type	Notes
code	short	an enumerated value
message	DOMString	not intended for end users

PositionOptions Object



POSITIONOPTIONS OBJECT

Property	Type	Default	Notes
enableHighAccuracy	Boolean	false	true might be slower
timeout	long	(no default)	in milliseconds
maximumAge	long	0	in milliseconds

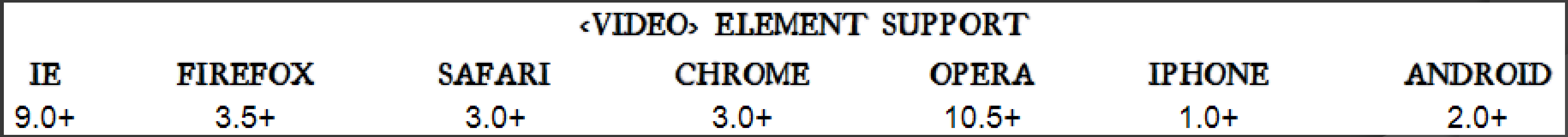
```
navigator.geolocation.getCurrentPosition(
   success_callback, error_callback, {maximumAge: 75000});
```

Continous Updates

- watchPosition()
- clearPosition()

Media

- YouTube Videos
- Formats – mp4, avi etc
- Containers
- Codecs



<VIDEO> ELEMENT SUPPORT

IE	FIREFOX	SAFARI	CHROME	OPERA	IPHONE	ANDROID
9.0+	3.5+	3.0+	3.0+	10.5+	1.0+	2.0+

Video Containers

- Container contains – Audio tracks, video tracks, track metadata, container metadata
- Popular containers
  - MPEG -4
  - OGG
  - WebM
  - Flash
  - Audio Video Interleave

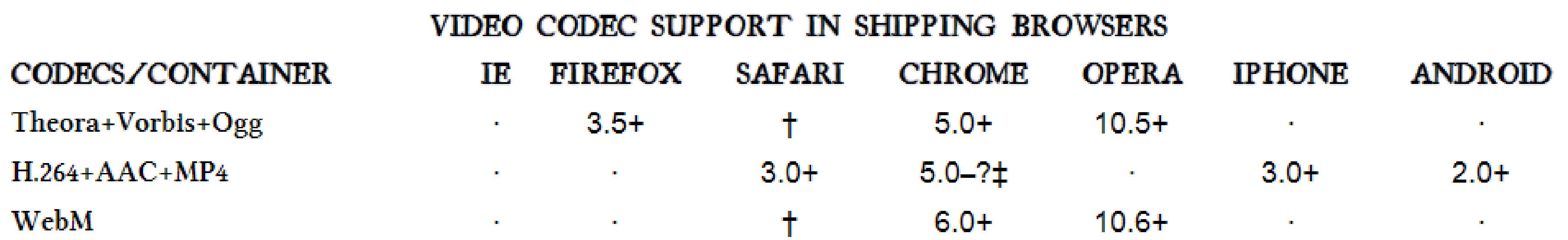
What does a Video Player do

- Interpreting the container format
- Decoding the video stream and displaying a series of images on the screen
- Decoding the audio stream and sending the sound to your speakers

Video Codec

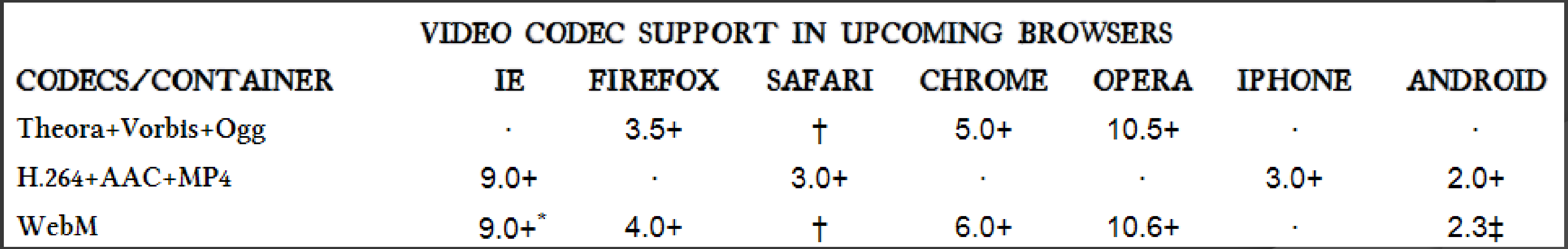
- A *video codec* is an algorithm by which a video stream is encoded
- Popular Video codecs
  - H.264
  - Theora
  - VP8
- Popular Audio codecs
  - MPEG -1
  - Advanced Audio Coding
  - Vorbis

Video Support



VIDEO CODEC SUPPORT IN SHIPPING BROWSERS

CODECS/CONTAINER	IE	FIREFOX	SAFARI	CHROME	OPERA	IPHONE	ANDROID
Theora+Vorbis+Ogg	·	3.5+	†	5.0+	10.5+	·	·
H.264+AAC+MP4	·	·	3.0+	5.0–?‡	·	3.0+	2.0+
WebM	·	·	†	6.0+	10.6+	·	·



VIDEO CODEC SUPPORT IN UPCOMING BROWSERS

CODECS/CONTAINER	IE	FIREFOX	SAFARI	CHROME	OPERA	IPHONE	ANDROID
Theora+Vorbis+Ogg	·	3.5+	†	5.0+	10.5+	·	·
H.264+AAC+MP4	9.0+	·	3.0+	·	·	3.0+	2.0+
WebM	9.0+*	4.0+	†	6.0+	10.6+	·	2.3‡

Video Encoding

- FFMPEG + Theora
- H.264 + AAC

The Code

```
<video src="pr6.webm"
  width="320" height="240"
  controls preload autoload>
</video>
```

GreaseMonkey for FF

```
// ==UserScript==
// @name          Disable video autoplay
// @namespace     http://diveintomark.org/projects/greasemonkey/
// @description   Ensures that HTML5 video elements do not autoplay
// @include       *
// ==/UserScript==

var arVideos = document.getElementsByTagName('video');
for (var i = arVideos.length - 1; i >= 0; i--)
{
    var elmVideo = arVideos[i];
    elmVideo.autoplay = false;
}
```

More types…More Files

```
<video width="320" height="240" controls>
    <source src="pr6.mp4" type='video/mp4; codecs="avc1.42E01E,
      mp4a.40.2"'>
    <source src="pr6.webm" type='video/webm; codecs="vp8, vorbis"'>
    <source src="pr6.ogv" type='video/ogg; codecs="theora, vorbis"'>
</video>
```

Points to remember

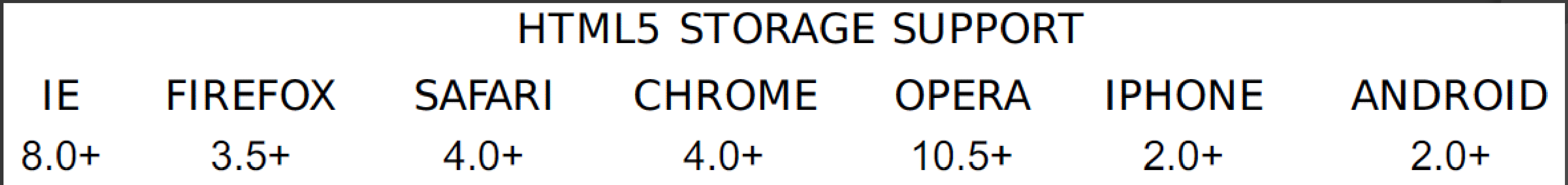
- Setting the right MIME types in your HTML code is very (very) important.
- Also ensure that your web server includes the proper MIME type in the Content-Type HTTP header.

E.g.

```
AddType video/ogg .ogv
AddType video/mp4 .mp4
AddType video/webm .webm
```

Local Storage

- Also referred to as Web Storage or DOM storage
- It’s a way for web pages to store named key/value pairs locally, within the client web browser.
- You mean Cookies???



HTML5 STORAGE SUPPORT

IE	FIREFOX	SAFARI	CHROME	OPERA	IPHONE	ANDROID
8.0+	3.5+	4.0+	4.0+	10.5+	2.0+	2.0+

Details

- Data is stored as a named key/value pairs.
- Any kind of data supported by JS is stored
- Data is always stored as String
- Use parseInt() or parseFloat to coerce data

```
interface Storage {
getter any getItem(in DOMString key);
setter creator void setItem(in DOMString key,
  in any data);
};
```

Something like...

```
var foo = localStorage.getItem("bar");
// ...
localStorage.setItem("bar", foo);
```

OR

```
var foo = localStorage["bar"];
// ...
localStorage["bar"] = foo;
```

Some more methods

```
interface Storage {
deleter void removeItem(in DOMString
  key);
void clear();
};

interface Storage {
readonly attribute unsigned long
  length;
getter DOMString key(in unsigned long
  index);
};
```

`storage` Event

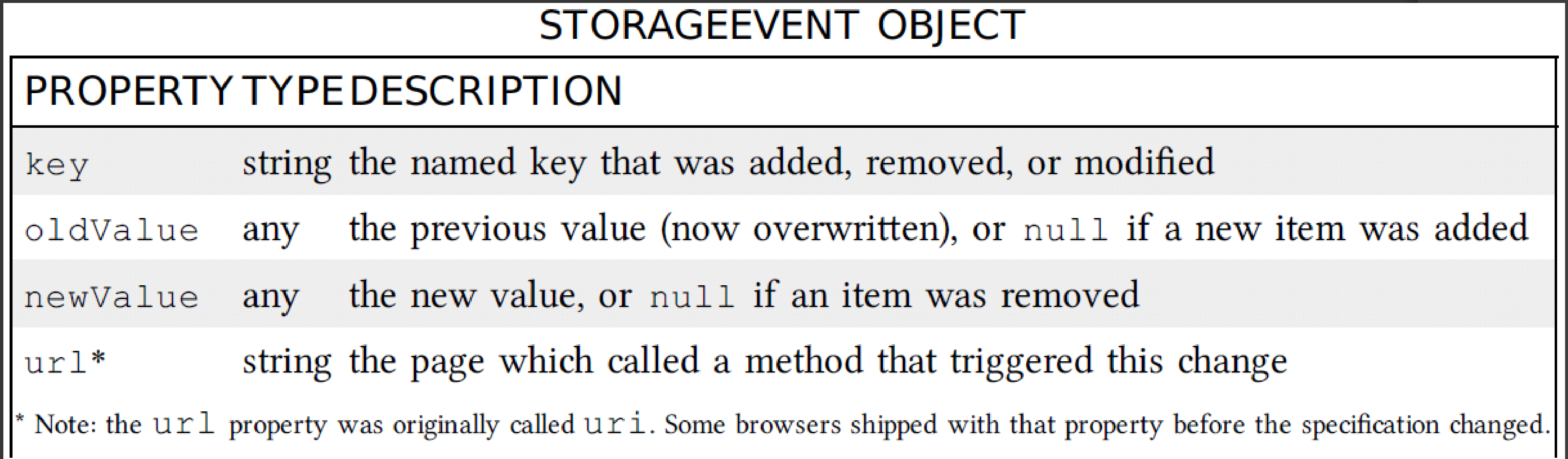
- Fired when changes occur or when `setItem()`, `removeItem()` or `clear()` is called.

```
if (window.addEventListener) {
window.addEventListener("storage", handle_storage,
  false);
} else {
window.attachEvent("onstorage", handle_storage);
};
```

- For IE

```
function handle_storage(e) {
if (!e) { e = window.event; }
}
```

StorageEvent Object



STORAGEEVENT OBJECT

PROPERTY	TYPE	DESCRIPTION
`key`	string	the named key that was added, removed, or modified
`oldValue`	any	the previous value (now overwritten), or `null` if a new item was added
`newValue`	any	the new value, or `null` if an item was removed
`url*`	string	the page which called a method that triggered this change

* Note: the `url` property was originally called `uri`. Some browsers shipped with that property before the specification changed.

ActionTime

Forms continued

- <input type="email">
- <input type="url">.
- <input
  - type="date"
  - type="month"
  - type="week"
  - type="time"
  - type="datetime"
  - type="datetime-local“
- <input name="q" type="search">
- <input type="color">
- <form novalidate>

2D Transforms : Rotate

```
.rotate-45 {
-webkit-transform: rotate(-45deg);
-moz-transform: rotate(-45deg);
-o-transform: rotate(-45deg);  }
```

2D Transforms - scale

```
.scale02  {
-webkit-transform: scale(0.2)
-moz-transform: scale(0.2);
-o-transform: scale(0.2);
}
```

2D Transforms - Skew

```
.skewX10 {
-webkit-transform:skewX(10deg);
-moz-transform: skewX(10deg);
-o-transform: skewX(10deg);
}
```

Transitions

```
#content {

        -webkit-transition:
  margin-left 1s ease-in-out;
        -moz-transition: margin-
  left 1s ease-in-out;
        -o-transition: margin-
  left 1s ease-in-out;
}
```

3D Transforms

```
#anim1 iframe {
-webkit-transition: -webkit-transform 1s
  ease-in-out;
-webkit-transform: rotate3d(0, 1, 1, 30deg);
 transition: transform 1s ease-in-out;
 transform: rotate3d(0, 1, 1, 30deg);
      }

#anim1 iframe:hover {
 -webkit-transform: rotate3d(0, 0, 1, 30deg);
 transform: rotate3d(0, 0, 1, 30deg);
      }
```

Animations

```
@-webkit-keyframes pulse {
      from {
        opacity: 0.0;
        font-size: 100%;
      }
      to {
        opacity: 1.0;
        font-size: 200%;
      }
    }

    div {
      -webkit-animation-name: pulse;
      -webkit-animation-duration: 2s;
      -webkit-animation-iteration-count: infinite;
      -webkit-animation-timing-function: ease-in-out;
}
```

Notification

- I love popups?!?!
- How about FYI’s then?
- Notifications API

User is the king

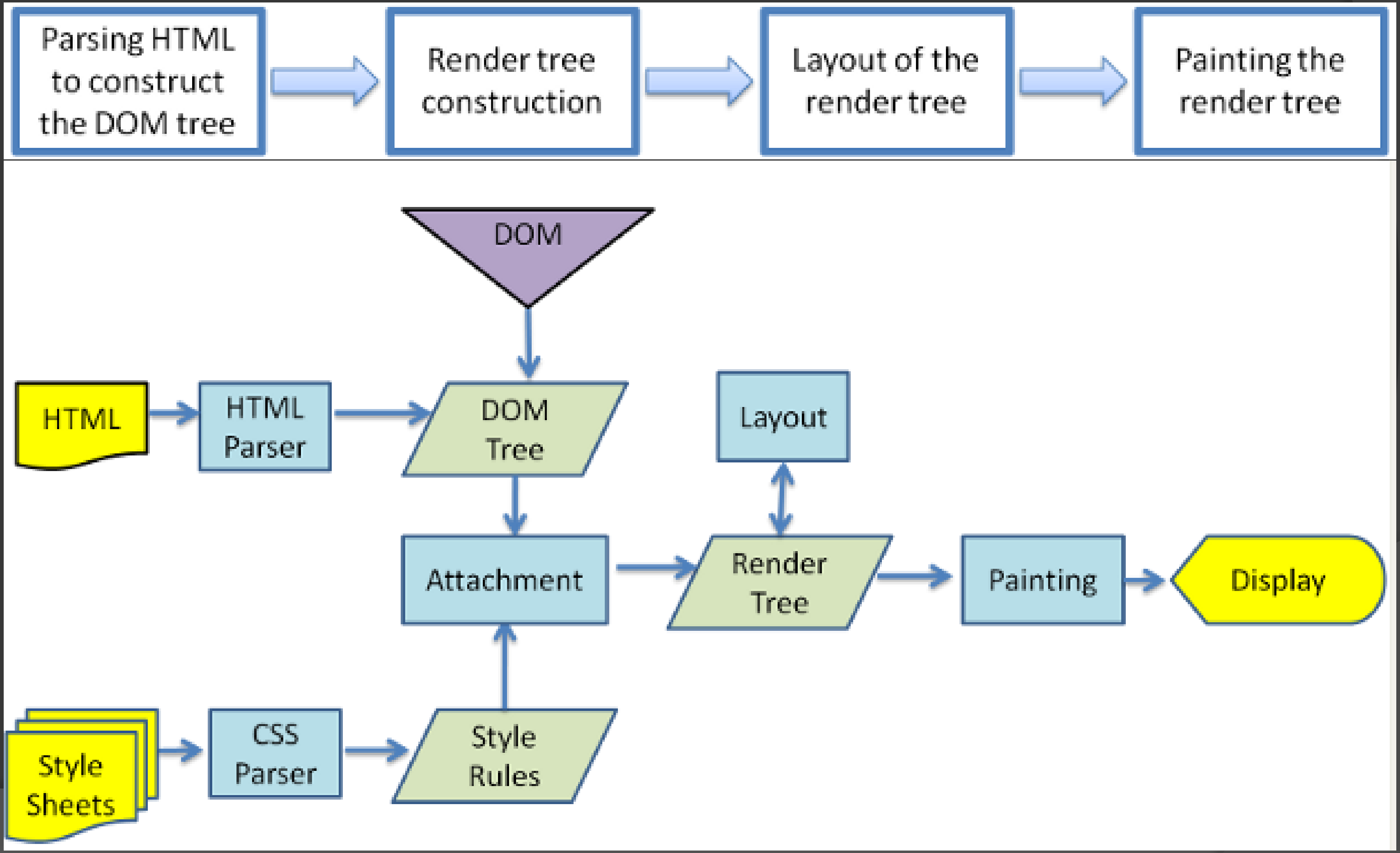
- Request permission
- Check Users response
- Create Notification
- Show notification
- Remove it

- `window.webkitNotifications`

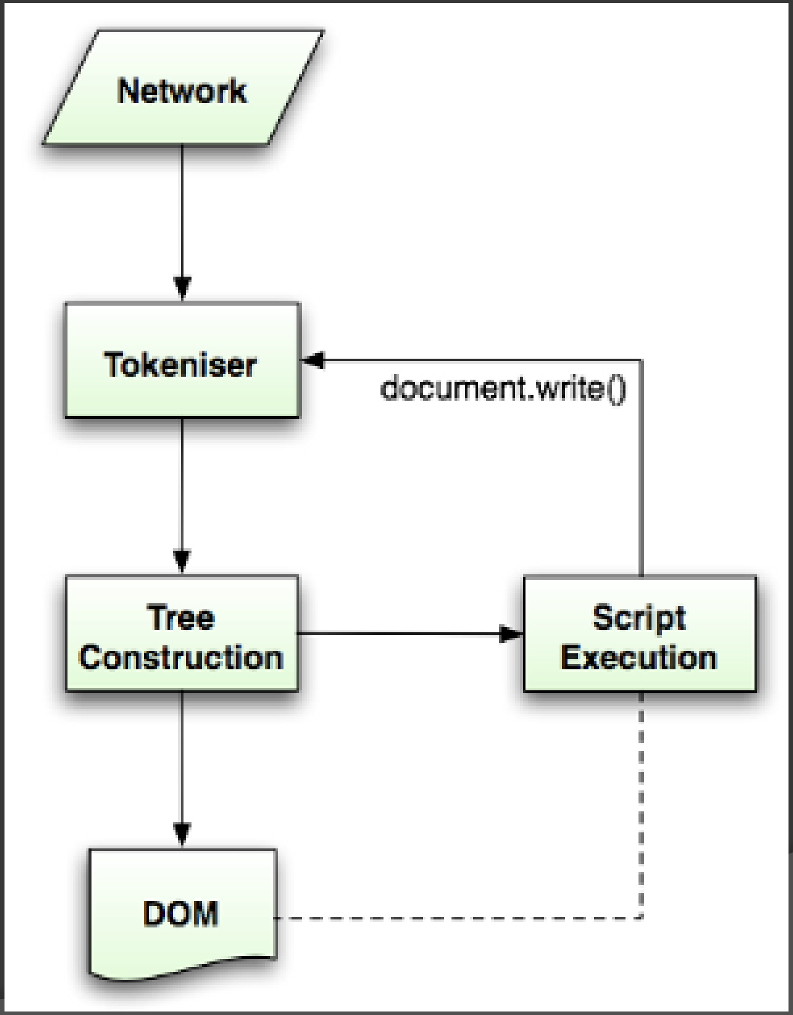
2 types

- Simple Notifications
- Web Notifications

How browsers operate



HTML Parser


Network
Tokeniser
document.write()
Tree Construction
Script Execution
DOM

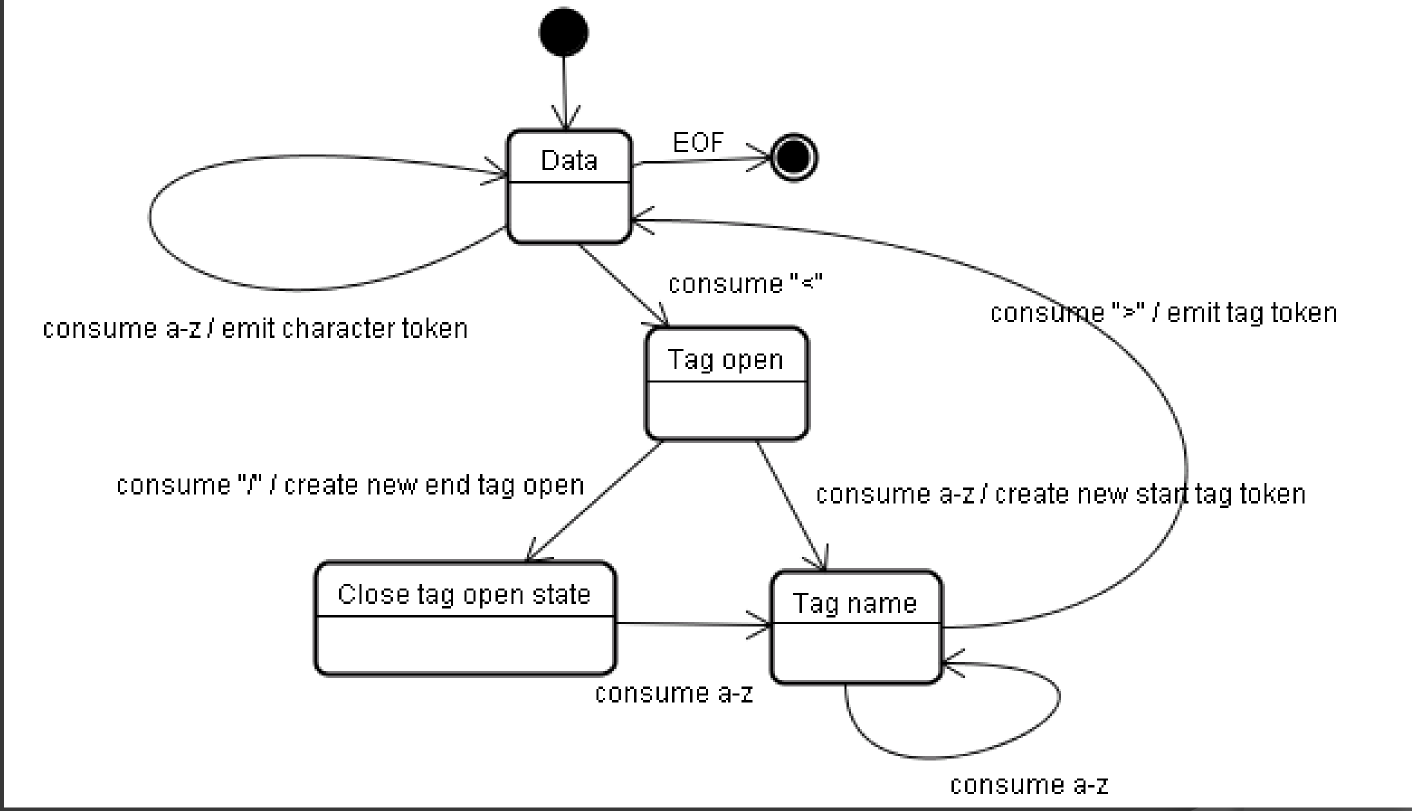
2 step process

- Tokenization algorithm
- Tree Construction algorithm

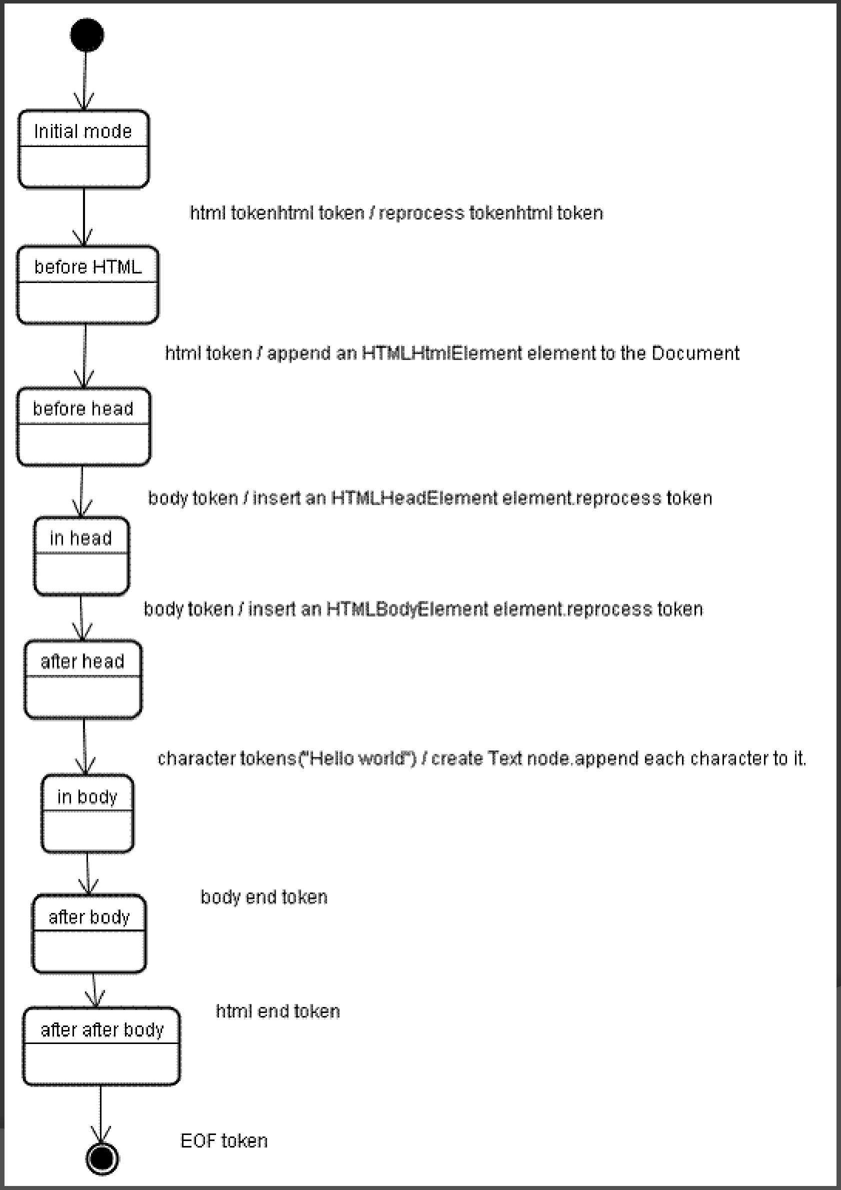
Tokenization

```
<html>
   <body>
      Hello world
   </body>
</html>
```

```
Output
<html>, <body>,
Hello World,
</body>,
</html>
```


Data
EOF
consume a-z / emit character token
consume "<"
consume ">" / emit tag token
Tag open
consume "/" / create new end tag open
consume a-z / create new start tag token
Close tag open state
Tag name
consume a-z
consume a-z

Tree Construction



Initial mode

html tokenhtml token / reprocess tokenhtml token

before HTML

html token / append an HTMLHtmlElement element to the Document

before head

body token / insert an HTMLHeadElement element.reprocess token

in head

body token / insert an HTMLBodyElement element.reprocess token

after head

character tokens("Hello world") / create Text node.append each character to it.

in body

body end token

after body

html end token

after after body

EOF token

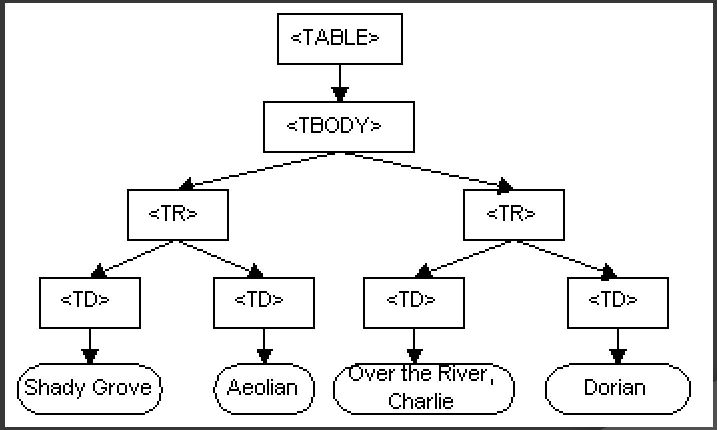
DOM- Document Object Model

- A representation — a model — of a document and its content
- An application programming interface (*API*) for valid *HTML* and well-formed *XML* documents.
- Defines the logical structure of documents
- Defines the way a document is accessed and manipulated
- Platform- and language-independent

…

- Build documents, navigate their structure, add-modify-delete elements and content
- It is an object structure that mirrors the structure of a document it models
- Defines objects and properties of all HTML elements, and the methods (interface) to access them

```
<TABLE>
    <TBODY>
        <TR>
          <TD>Shady Grove</TD>
          <TD>Aeolian</TD>
        </TR>
        <TR>
          <TD>Over the River, Charlie</TD>
          <TD>Dorian</TD>
        </TR>
    </TBODY>
</TABLE>
```



DOM Nodes

- Everything in an HTML document is a node
- The entire document is a document node
- Every HTML element → element node
- The text in the HTML elements → text nodes
- Every HTML attribute → attribute node
- Comments → comment nodes

Text Nodes

- Text of an element node → text node
- The value of the text node can be accessed by the innerHTML property

Talking to the nodes

- Nodes can be accessed with JavaScript
- Some DOM properties:
  - x.innerHTML - the text value of x
  - x.nodeName - the name of x
  - x.nodeValue - the value of x
  - x.parentNode - the parent node of x
  - x.childNodes - the child nodes of x
  - x.attributes - the attributes nodes of x
- Some DOM methods:
  - x.getElementById(*id*) - get the element with a specified id
  - x.getElementsByTagName(*name*) - get all elements with a specified tag name
  - x.appendChild(*node*) - insert a child node to x
  - x.removeChild(*node*) - remove a child node from x

innerHTML

- Easiest way to get or modify the content of an element

```
<html>
<body>
<p id="intro">Hello World!</p>
<script type="text/javascript”>
txt=document.getElementById("intro“).innerHTML;
document.write("<p>The text from the intro
  paragraph: " + txt + "</p>");
</script>
</body>
</html>
```

childNodes and nodeValue

```
txt=document.getElementById
("intro").childNodes[0].nodeVal
  ue;
```

Accessing Nodes

- getElementById()
- getElementsByTagName()
- By navigating the node tree, using the node relationships

getElementById

- *node*.getElementById(*"id"*);

```
<p id="intro">bhakti!</p>
X=document.getElementById("intro");
document.write(x.innerHTML);
```

getElementByTagName

- *node*.getElementsByTagName(*"tagname"*);

```
<p>Hello World!</p>
<p>The DOM is very useful!</p>
<p id=“main”>This example demonstrates the
   <b>getElementsByTagName</b> method.</p>

<script type="text/javascript">
x=document.getElementsByTagName("p");
document.write("Text of first paragraph: " +
   x[0].innerHTML);
</script>
```

- document.getElementById("main").getElementsByTagName("p");

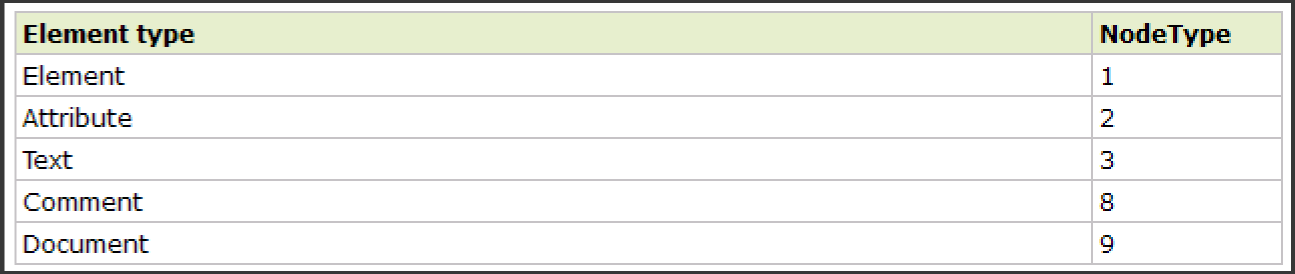
Navigating node relationships

```
<html>
  <body>

  <p id="intro">Hello World!</p>

  <script type="text/javascript">
  x=document.getElementById("intro");
  document.write(x.firstChild.nodeValue);
  </script>

  </body>
</html>
```

- document.documentElement - returns the root node of the document
- document.body - gives direct access to the <body> tag

Node Properties

- Every node in the tree has 3 properties
- nodeName, nodeType, nodeValue



Element type	NodeType
Element	1
Attribute	2
Text	3
Comment	8
Document	9

Change an HTML Element

```
<html>
  <body>

  <script type="text/javascript">
  document.body.bgColor="lavender"
  ;
  </script>

  </body>
  </html>
```

Change text

```
<html>
  <body>

  <p id="p1">Hello World!</p>

  <script type="text/javascript">
  document.getElementById("p1").innerHTML="New text!";
  </script>

  </body>
  </html>
```

Change element using Events

```
<html>
  <body>

  <input type="button"
  onclick="document.body.bgColor='
  lavender';"
  value="Change background color"
  />

  </body>
  </html>
```

Using Style Object

```
<html>
  <head>
  <script type="text/javascript">
  function ChangeBackground()
  {
  document.body.style.backgroundColor="lavender";
  }
  </script>
  </head>

  <body>
  <input type="button"
  onclick="ChangeBackground()"
  value="Change background color" />
  </body>
  </html>
```

Events

Useful link

- http://livedom.validator.nu/