HTML5

New Features of HTML5

Objectives

- At the end of this session, you would be able to:
  - Build interactive, cross-browser compatible and device independent HTML5 web pages by following core principles and best practices.
  - Utilize the various new HTML5 features such as local storage, geo location, web sockets, multimedia & graphics, semantics etc. considering their current state of development.

Contents

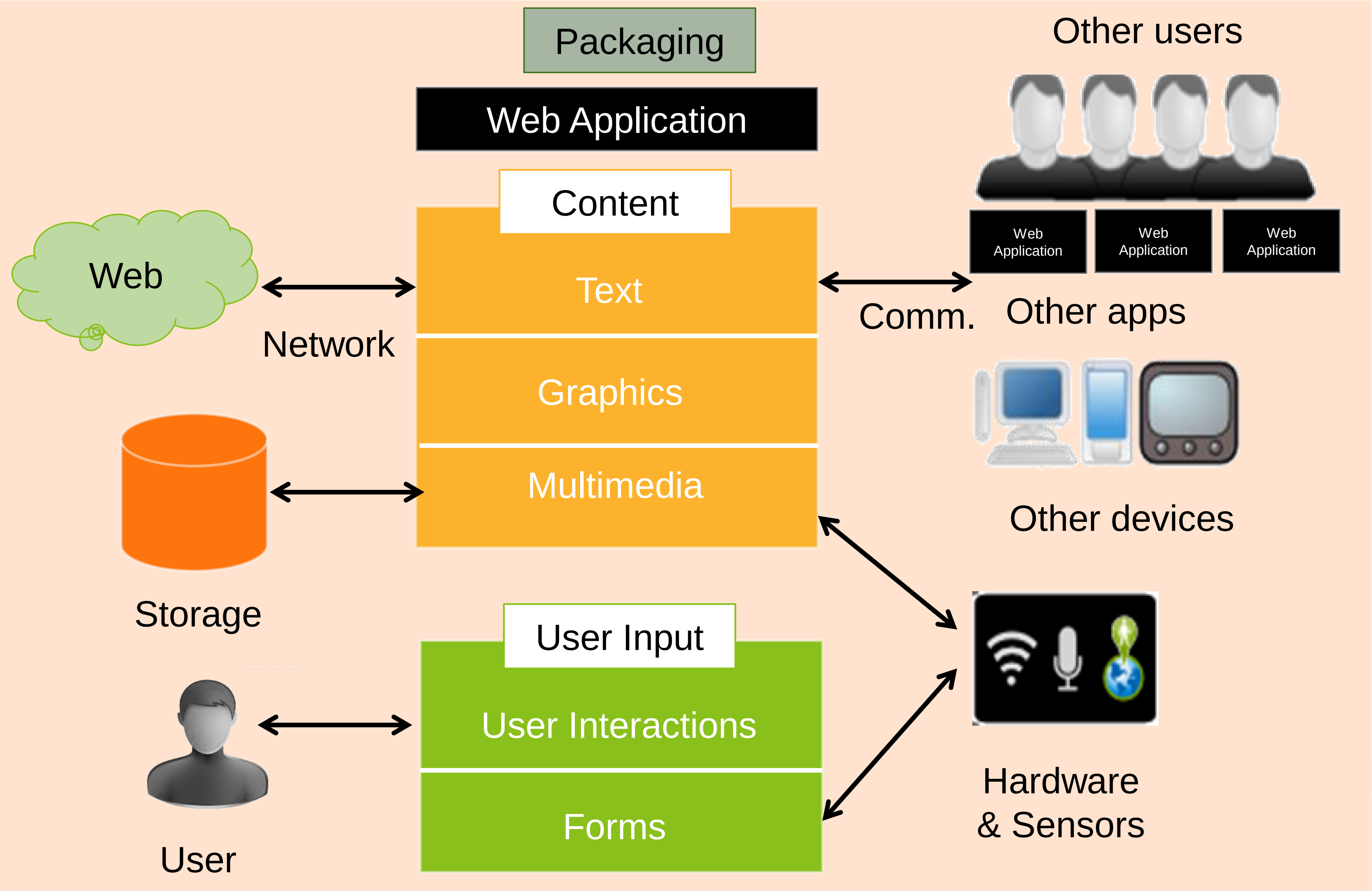
Table of Contents

- The Web as an Application Development Platform
- Getting started with HTML5
- The Taxonomy of HTML5
- The Web Storage API
- The Communication API
- Playing Multimedia
- The Device Access API
- Semantics and Markup
- Performing Graphics
- Tools & IDE’s
- The Road Ahead

What is HTML5

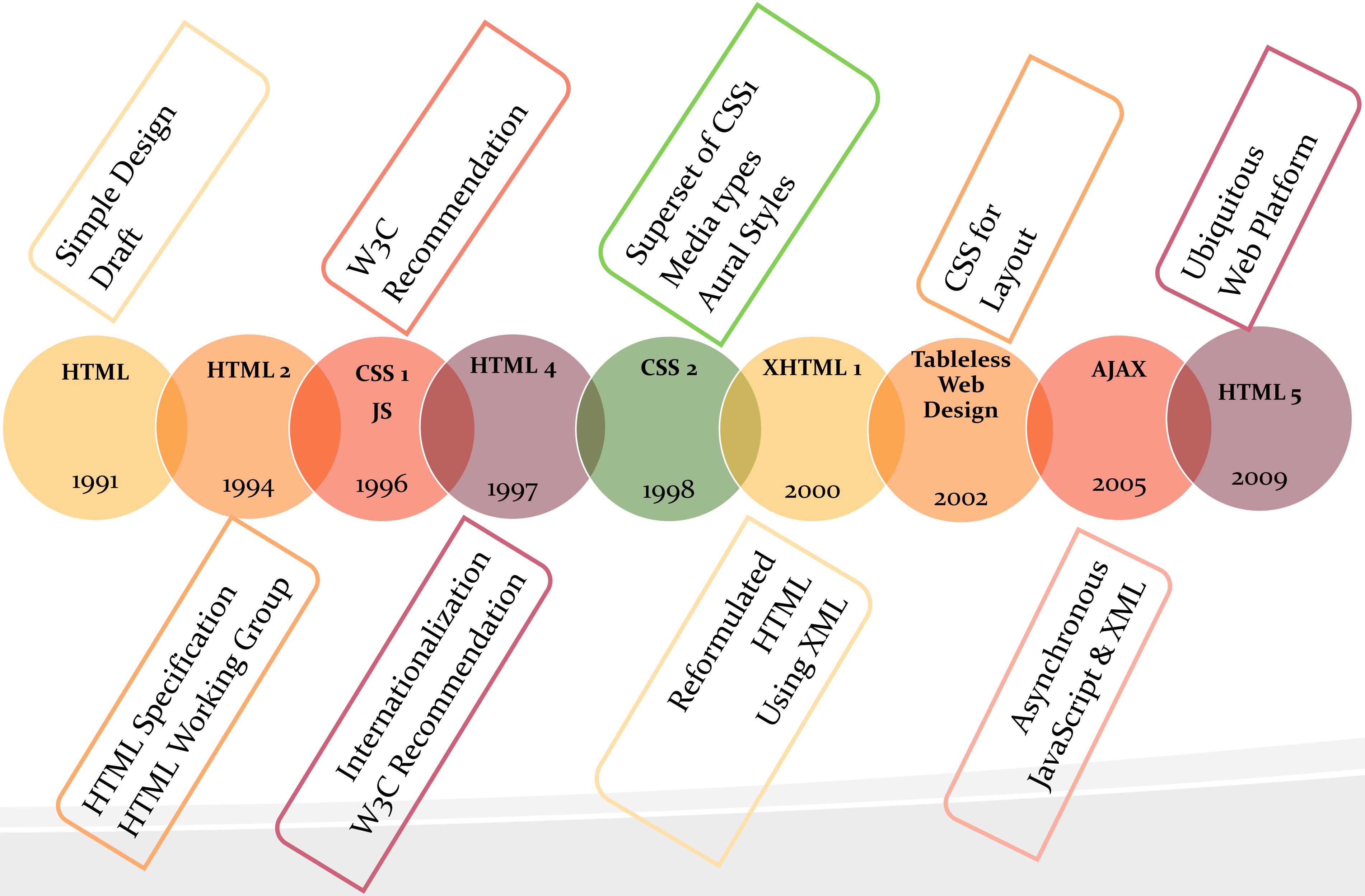
- The latest evolution of the standard that defines HTML
  - new version of the language HTML
  - a larger set of technologies
- Allows the building of
  - more diverse and powerful Web sites and applications.

Today's Web: An Application Development Platform



[source : W3C]

Evolution of HTML



HTML
1991
Simple Design
Draft

HTML 2
1994
HTML Specification
HTML Working Group

CSS 1
JS
1996
W3C
Recommendation

HTML 4
1997
Internationalization
W3C Recommendation

CSS 2
1998
Superset of CSS1
Media types
Aural Styles

XHTML 1
2000
Reformulated
HTML
Using XML

Tableless
Web
Design
2002
CSS for
Layout

AJAX
2005
Asynchronous
JavaScript & XML

HTML 5
2009
Ubiquitous
Web Platform

HTML5 Standard Bodies

- The HTML standard bodies that control the web standards are:

W3C® WORLD WIDE WEB consortium	?	IETF®
The World Wide Web consortium (W3C)	The Web Hypertext Application Technology Working Group (WHATWG)	The Internet Engineering Task Force (IETF)

HTML 5 vs. HTML 4 (1 of 2)

HTML 4 Markup

HTML 5 Markup

HTML 4 Markup

Common Structures

div.

```
<html>
<title>HTML</title>
<body>
This is HTML!
</body>
</html>
```

HTML 5 Markup

Richer Semantics

header, footer, section, article, aside

HTML 4 API

HTML 5 API

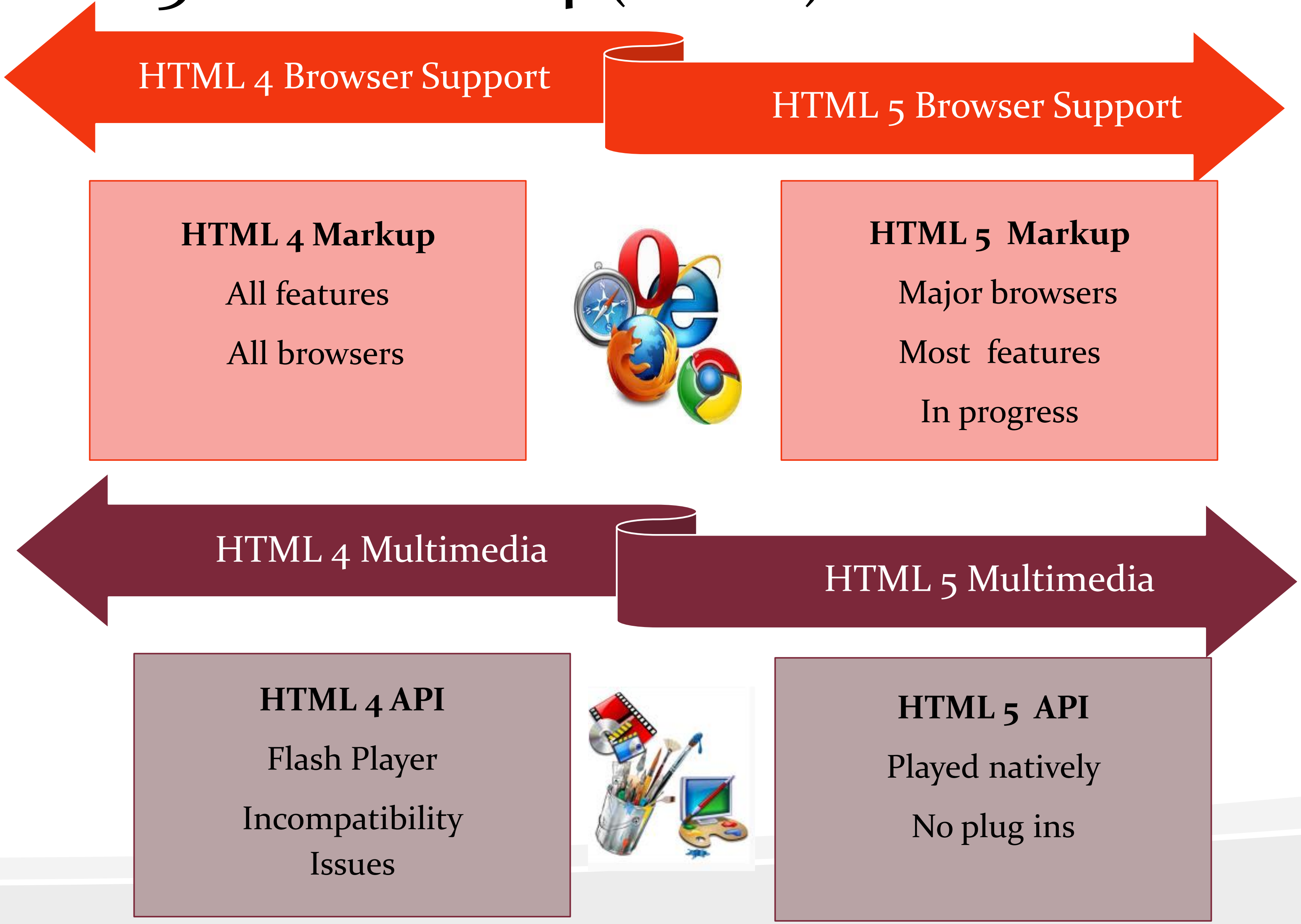
HTML 4 API

No such API

HTML 5 API

API for geolocation, offline storage, canvas, audio & video

HTML 5 vs. HTML 4 (2 of 2)



HTML 4 Browser Support

HTML 5 Browser Support

HTML 4 Markup

All features

All browsers

HTML 5 Markup

Major browsers

Most features

In progress

HTML 4 Multimedia

HTML 5 Multimedia

HTML 4 API

Flash Player

Incompatibility Issues

HTML 5 API

Played natively

No plug ins

App vs. Native App

Feature	HTML5 App	Native App
Accessibility	URL	Download, Install, Update
End User Experience	Progressive Enhancement	Standards
Hardware	Offline Storage Geolocation Canvas Audio Video Camera	Accelerometer Compass Camera Gyroscope GPS
Development	HTML5 CSS3 Javascript	Objective C C++ Java Microsoft.NET

HTML 5: A Suite

Meaningful Document Structure

Rich Media Capabilities

Locally Stored Information

Several Graphic Options

Semantics
Multimedia
Offline & Storage
3D & Effects
Device Access
Performance & Integration
Connectivity
CSS3 & Styling

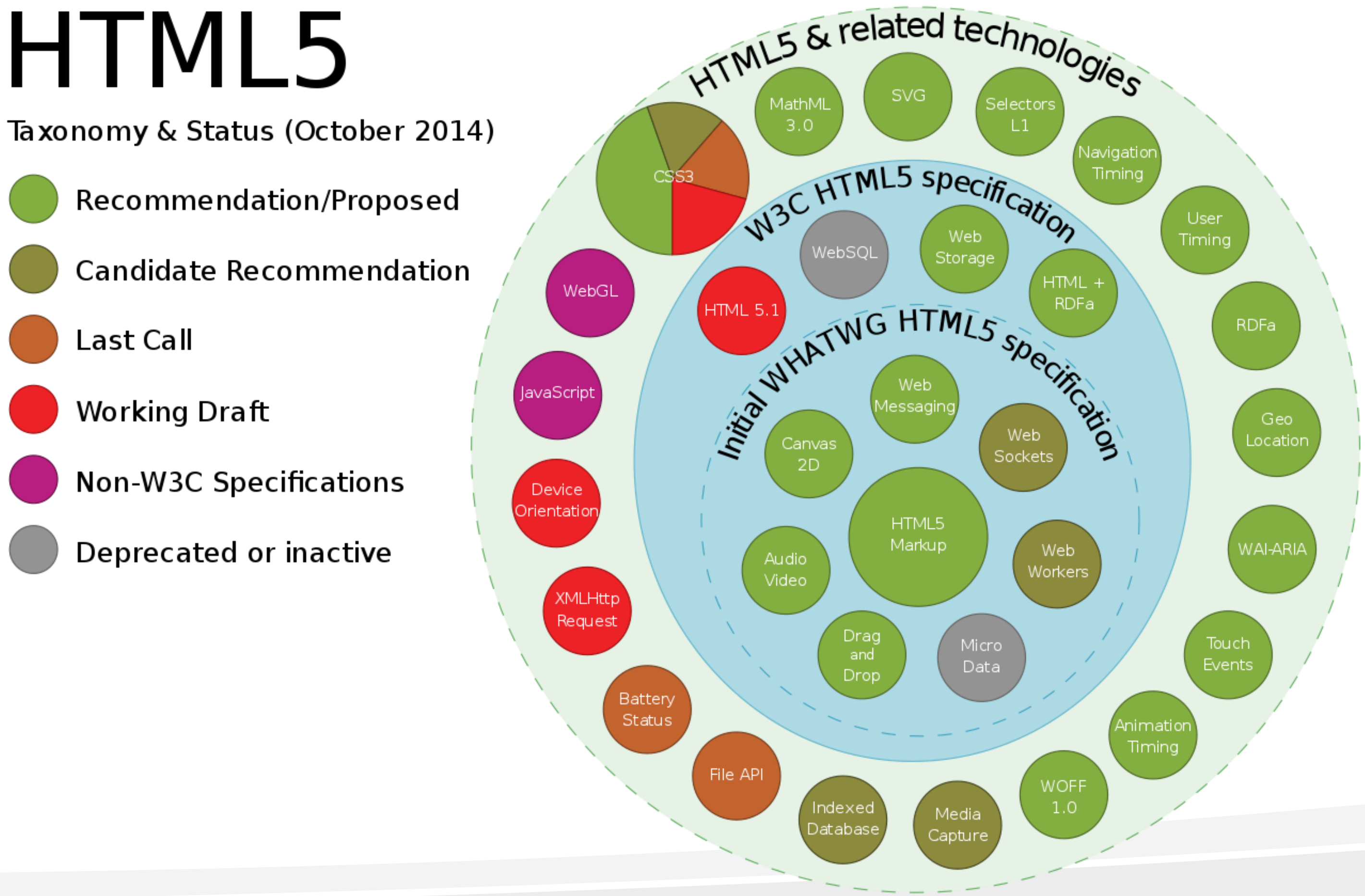
Mobile Device Component Access

High Performance Pages

Efficient Data Exchange

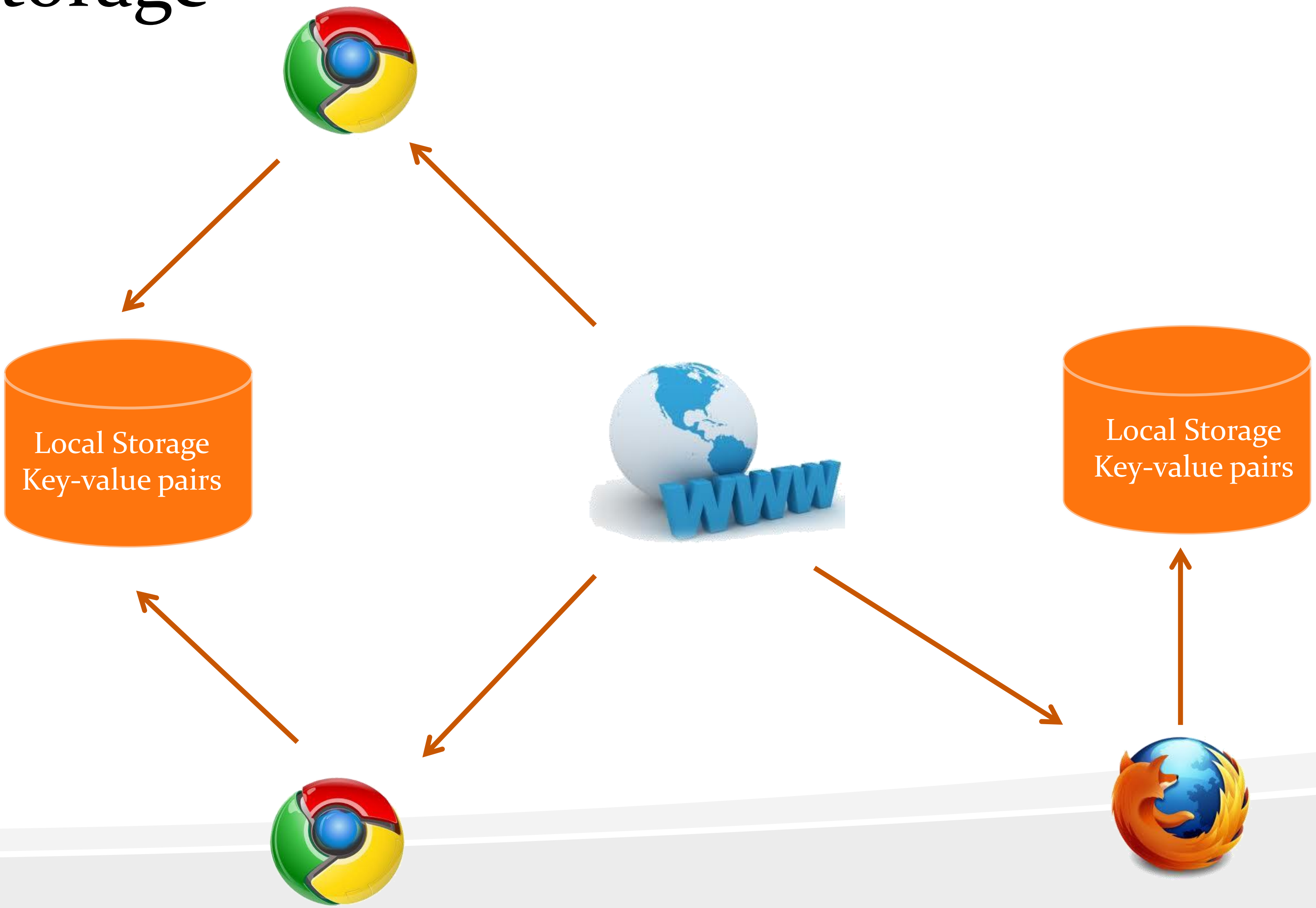
Rich Styling Effects

HTML5 Taxonomy



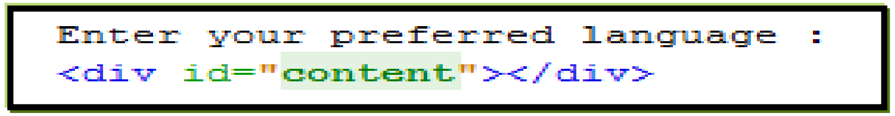
Web Storage

Local Storage



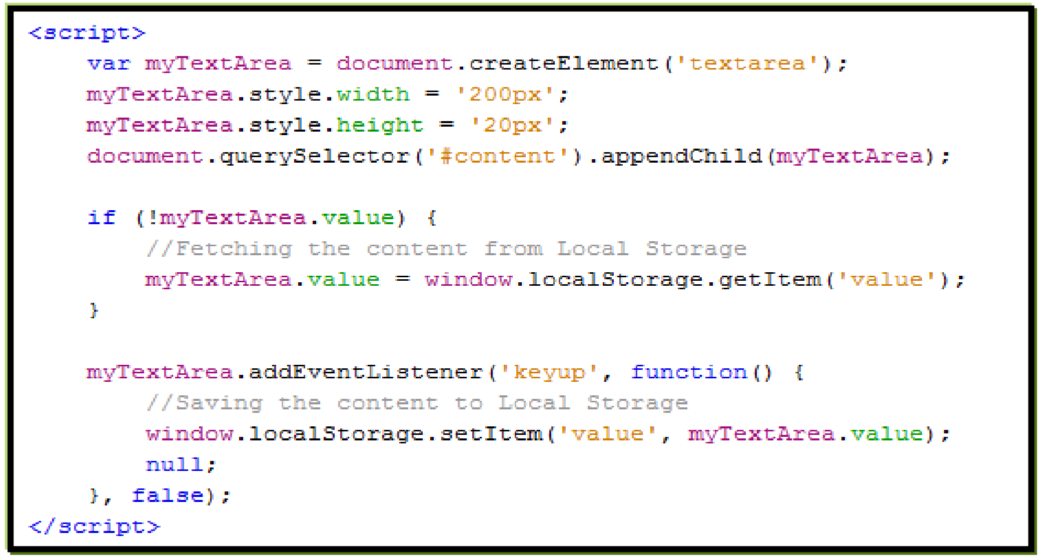
Local Storage: Code Snippet

The User Interface



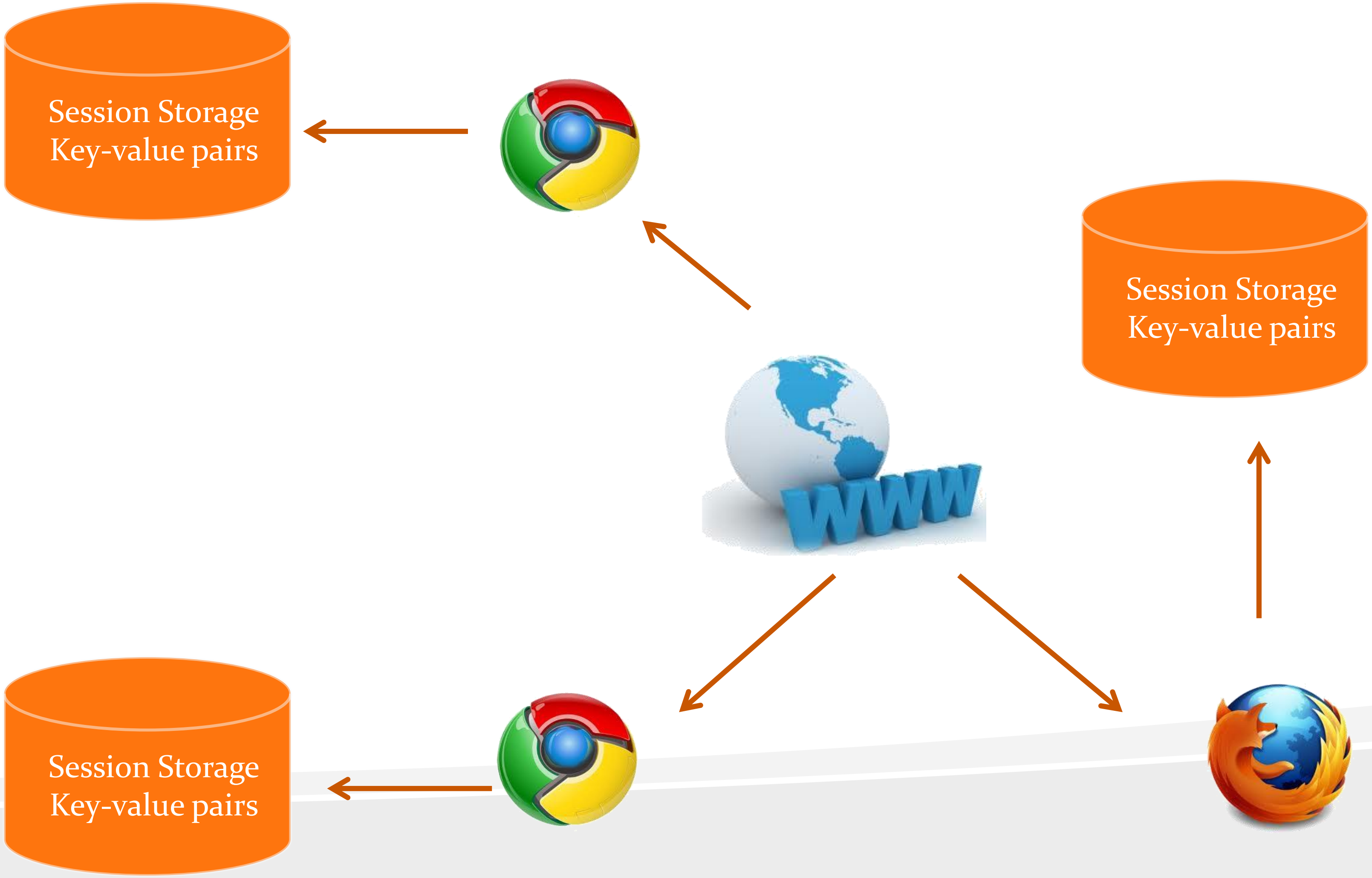
```
Enter your preferred language :
<div id="content"></div>
```

The Local Storage Script



```
<script>
    var myTextArea = document.createElement('textarea');
    myTextArea.style.width = '200px';
    myTextArea.style.height = '20px';
    document.querySelector('#content').appendChild(myTextArea);

    if (!myTextArea.value) {
        //Fetching the content from Local Storage
        myTextArea.value = window.localStorage.getItem('value');
    }

    myTextArea.addEventListener('keyup', function() {
        //Saving the content to Local Storage
        window.localStorage.setItem('value', myTextArea.value);
        null;
    }, false);
</script>
```

Session Storage



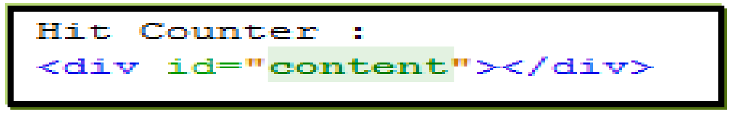
Session Storage
Key-value pairs

Session Storage
Key-value pairs

Session Storage
Key-value pairs

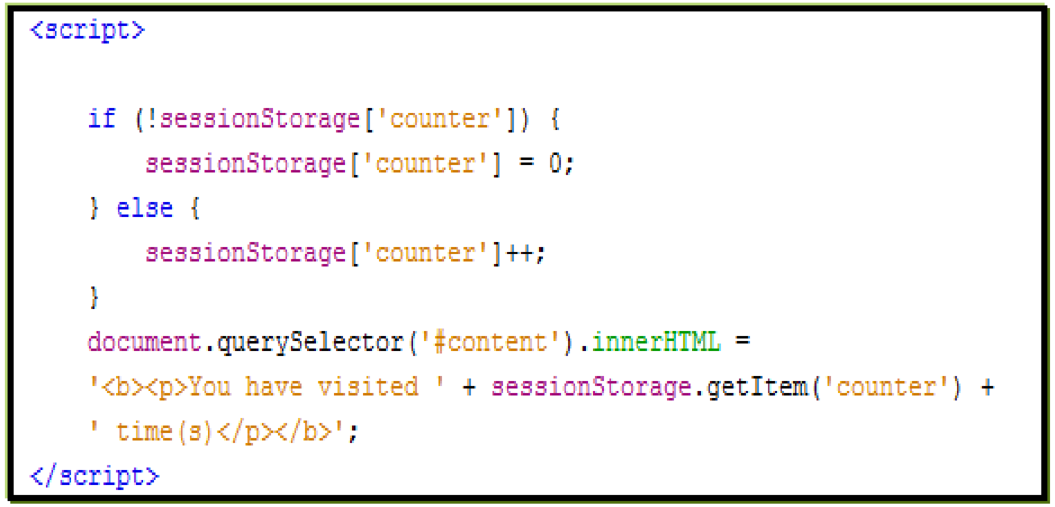
Session Storage- Code Snippet

The User Interface



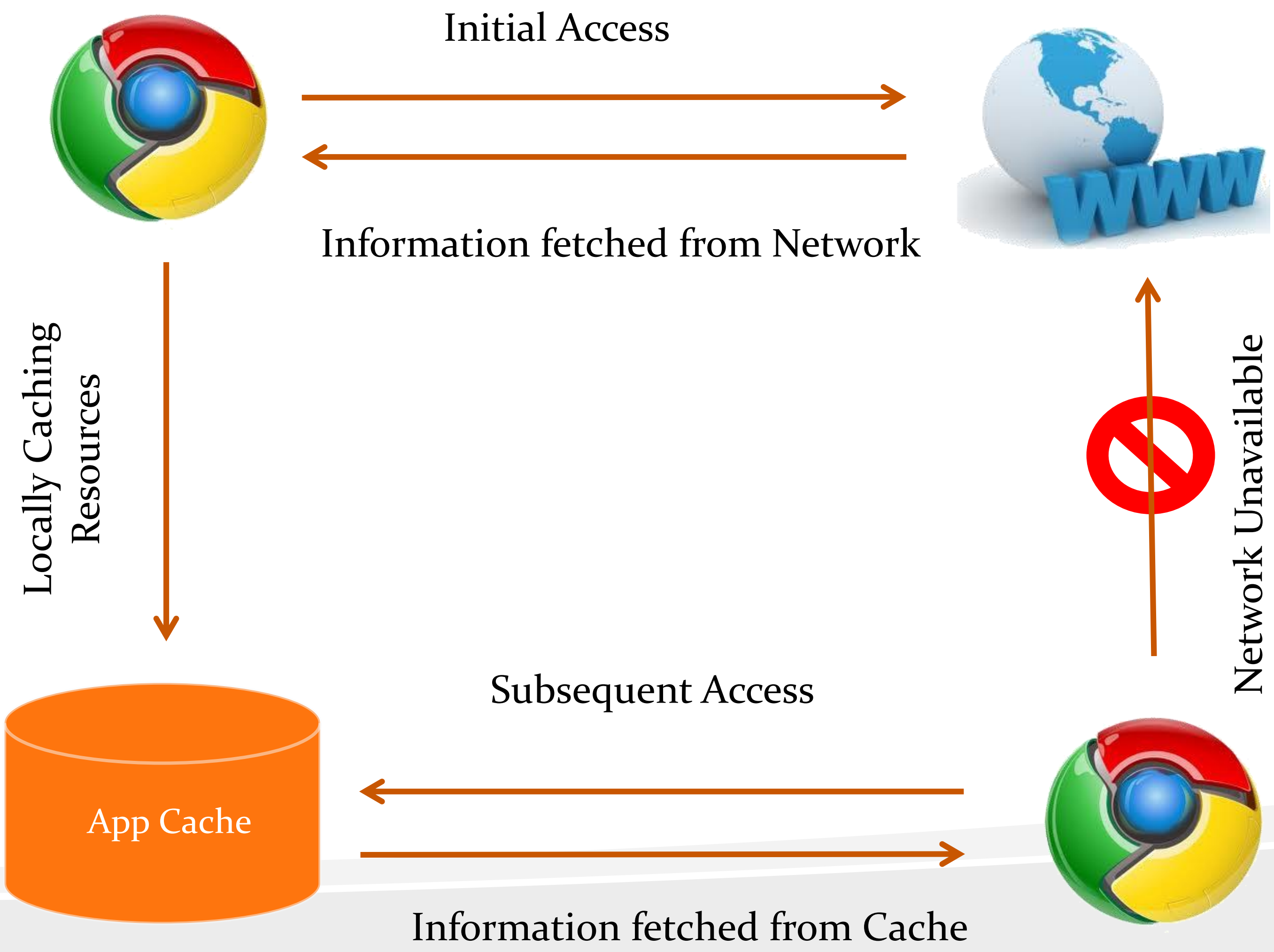
```
Hit Counter :
<div id="content"></div>
```

The Session Storage Script



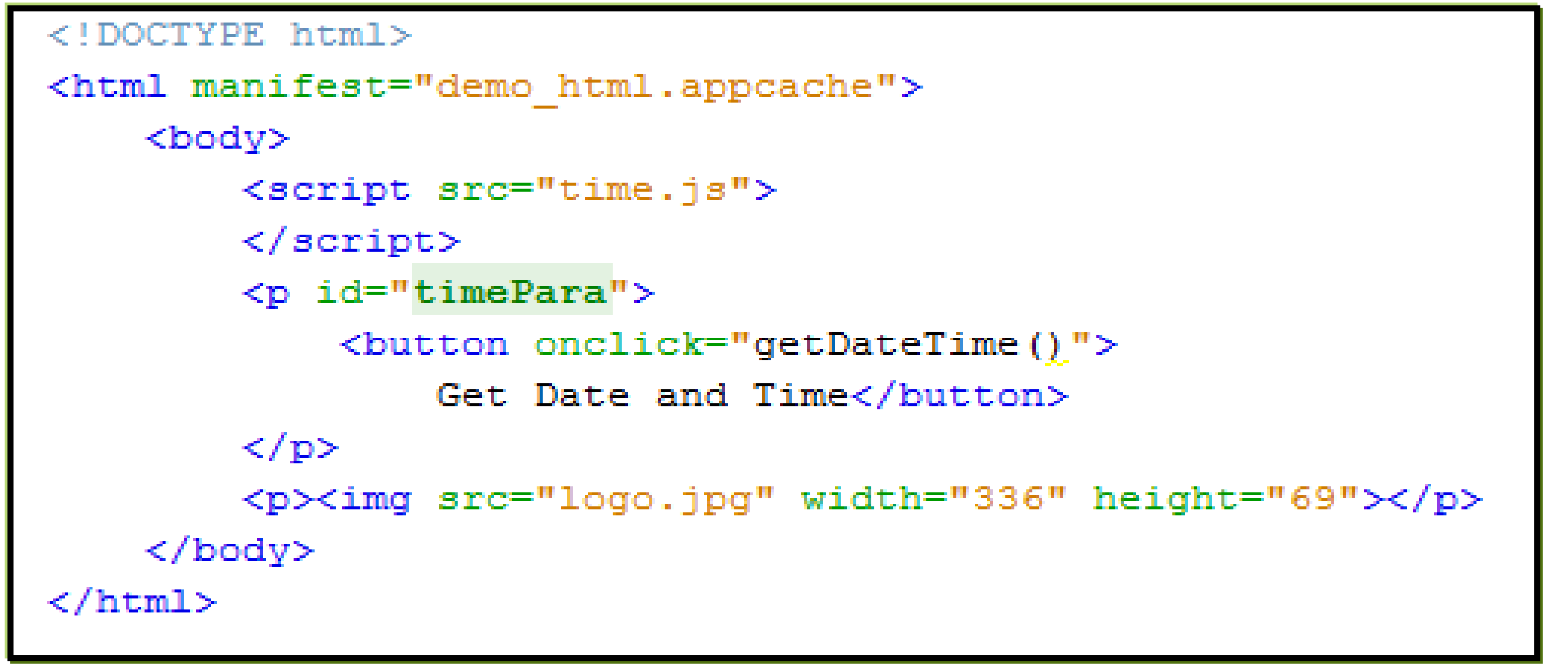
```
<script>

    if (!sessionStorage['counter']) {
        sessionStorage['counter'] = 0;
    } else {
        sessionStorage['counter']++;
    }
    document.querySelector('#content').innerHTML =
    '<b><p>You have visited ' + sessionStorage.getItem('counter') +
    ' time(s)</p></b>';
</script>
```

The Application Cache


Initial Access
Information fetched from Network
Locally Caching Resources
Network Unavailable
Subsequent Access
App Cache
Information fetched from Cache

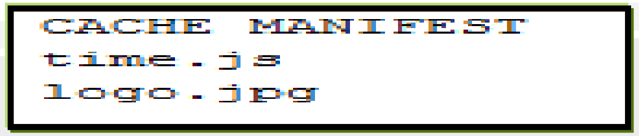
The Application Cache: Code Snippet (1 of 2)

The User Interface



```
<!DOCTYPE html>
<html manifest="demo_html.appcache">
    <body>
        <script src="time.js">
        </script>
        <p id="timePara">
            <button onclick="getDateTime()">
                Get Date and Time</button>
        </p>
        <p><img src="logo.jpg" width="336" height="69"></p>
    </body>
</html>
```

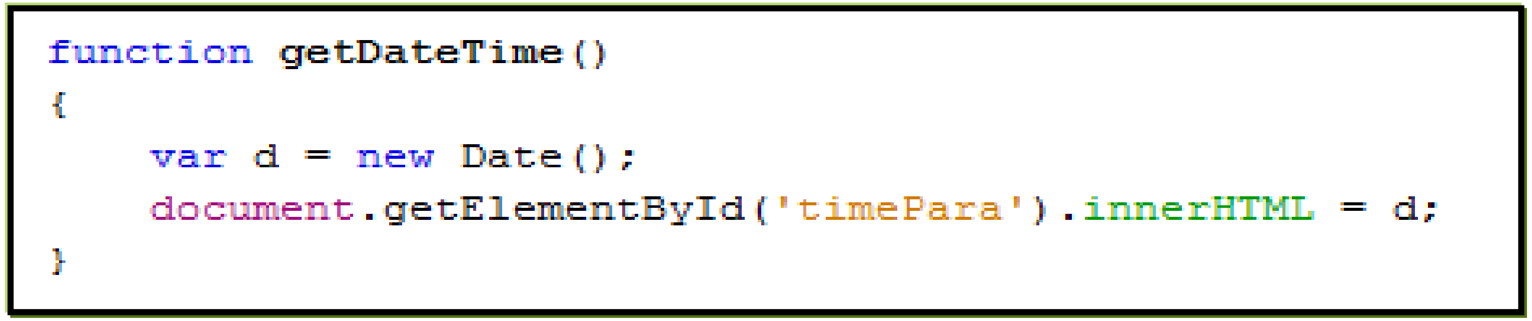
The Manifest File



```
CACHE MANIFEST
time.js
logo.jpg
```

The Application Cache: Code Snippet (2 of 2)

The JavaScript File



```
function getDateTime()
{
    var d = new Date();
    document.getElementById('timePara').innerHTML = d;
}
```

Storing Information on the Client

Connectivity

The Web Sockets

- A communication channel

Features

- Full-duplex
- Bi-directional
- Web Sockets API

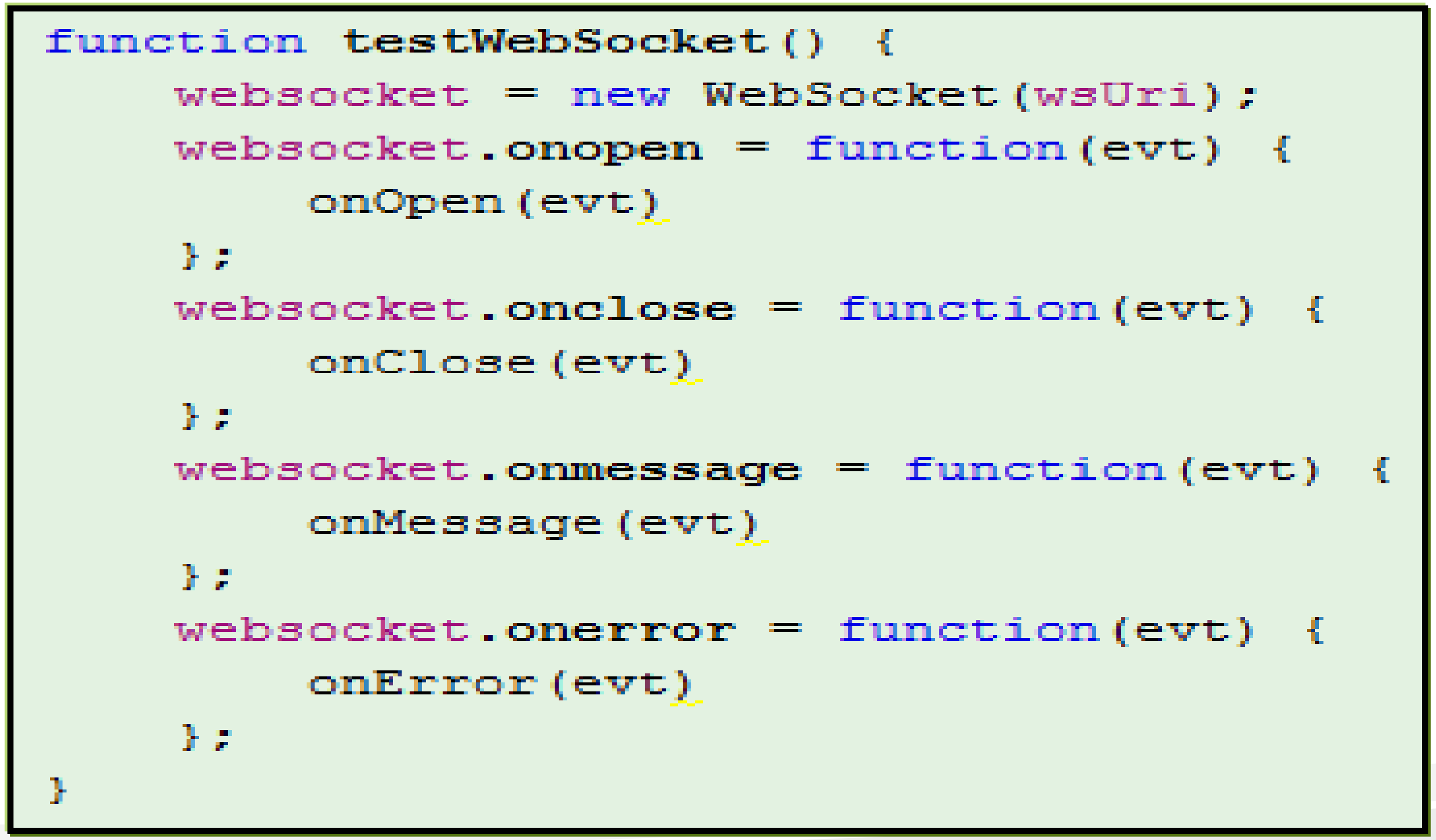
Advantages

- Less network traffic
- Less latency time
- Works on secure connections

“A WebSocket connection is established by upgrading from the HTTP protocol to the Web Socket protocol during the initial handshake between the client and the server, over the same underlying TCP/IP connection”

Web Sockets: Code Snippet (1 of 2)

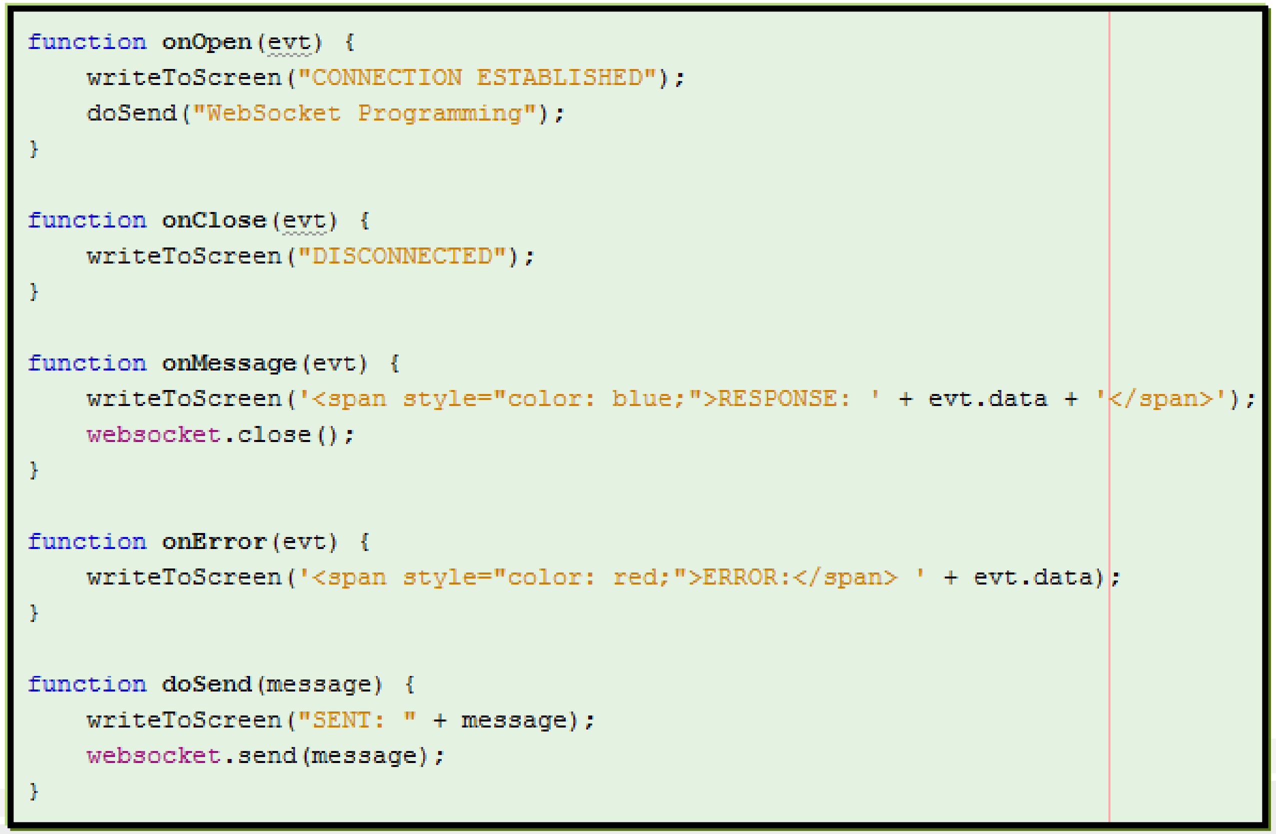
Creating Web Socket



```
function testWebSocket() {
    websocket = new WebSocket(wsUri);
    websocket.onopen = function(evt) {
        onOpen(evt)
    };
    websocket.onclose = function(evt) {
        onClose(evt)
    };
    websocket.onmessage = function(evt) {
        onMessage(evt)
    };
    websocket.onerror = function(evt) {
        onError(evt)
    };
}
```

Web Sockets: Code Snippet (2 of 2)

Handling Events



```
function onOpen(evt) {
    writeToScreen("CONNECTION ESTABLISHED");
    doSend("WebSocket Programming");
}

function onClose(evt) {
    writeToScreen("DISCONNECTED");
}

function onMessage(evt) {
    writeToScreen('<span style="color: blue;">RESPONSE: ' + evt.data + '</span>');
    websocket.close();
}

function onError(evt) {
    writeToScreen('<span style="color: red;">ERROR:</span> ' + evt.data);
}

function doSend(message) {
    writeToScreen("SENT: " + message);
    websocket.send(message);
}
```

Real-time Communication using Web Sockets

Multimedia

HTML5 Multimedia Elements

HTML 5 Multimedia elements provides:

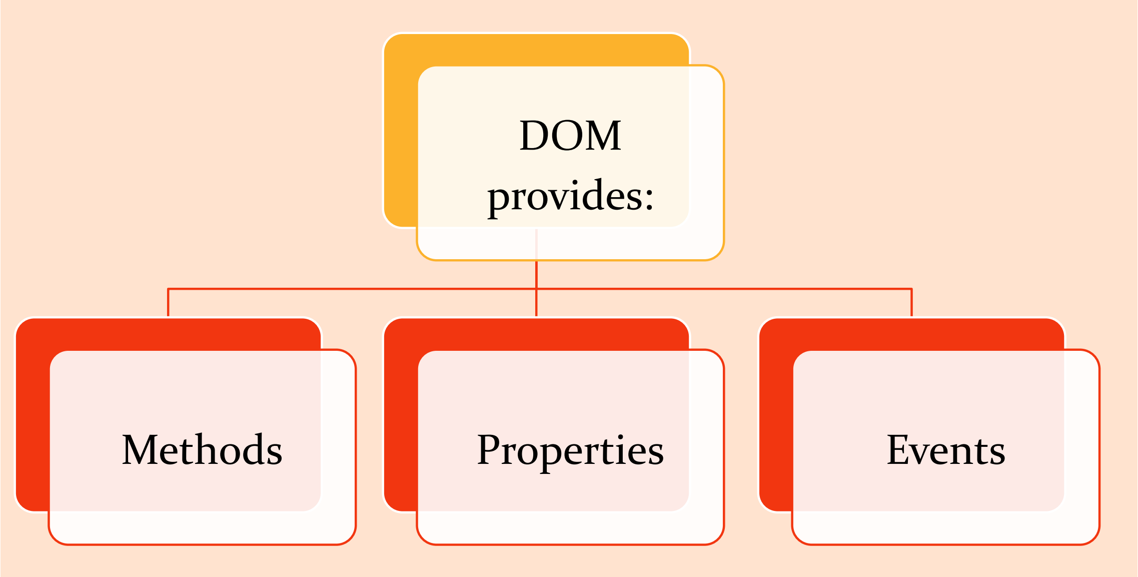
- Native Audio & Video Support
- Scriptable containers
- DOM support

No plug-in are required

“The real power of HTML5 multimedia elements stand out when they are combined with the other technologies of the web stack such as Canvas, SVG, CSS or even WebGL”

The Audio Element

- Performs native audio playback

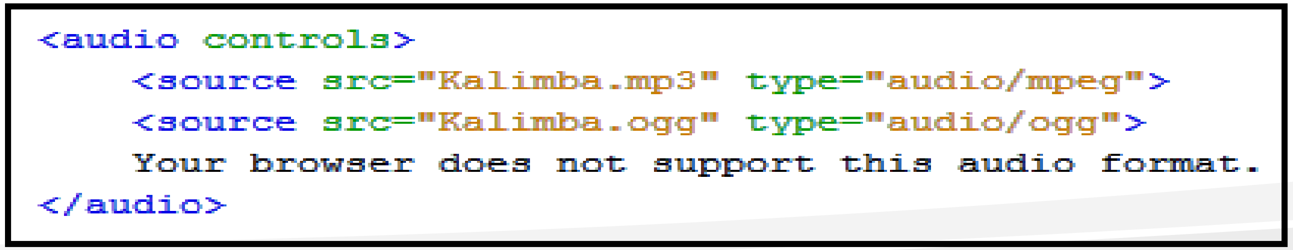


DOM provides:

Methods

Properties

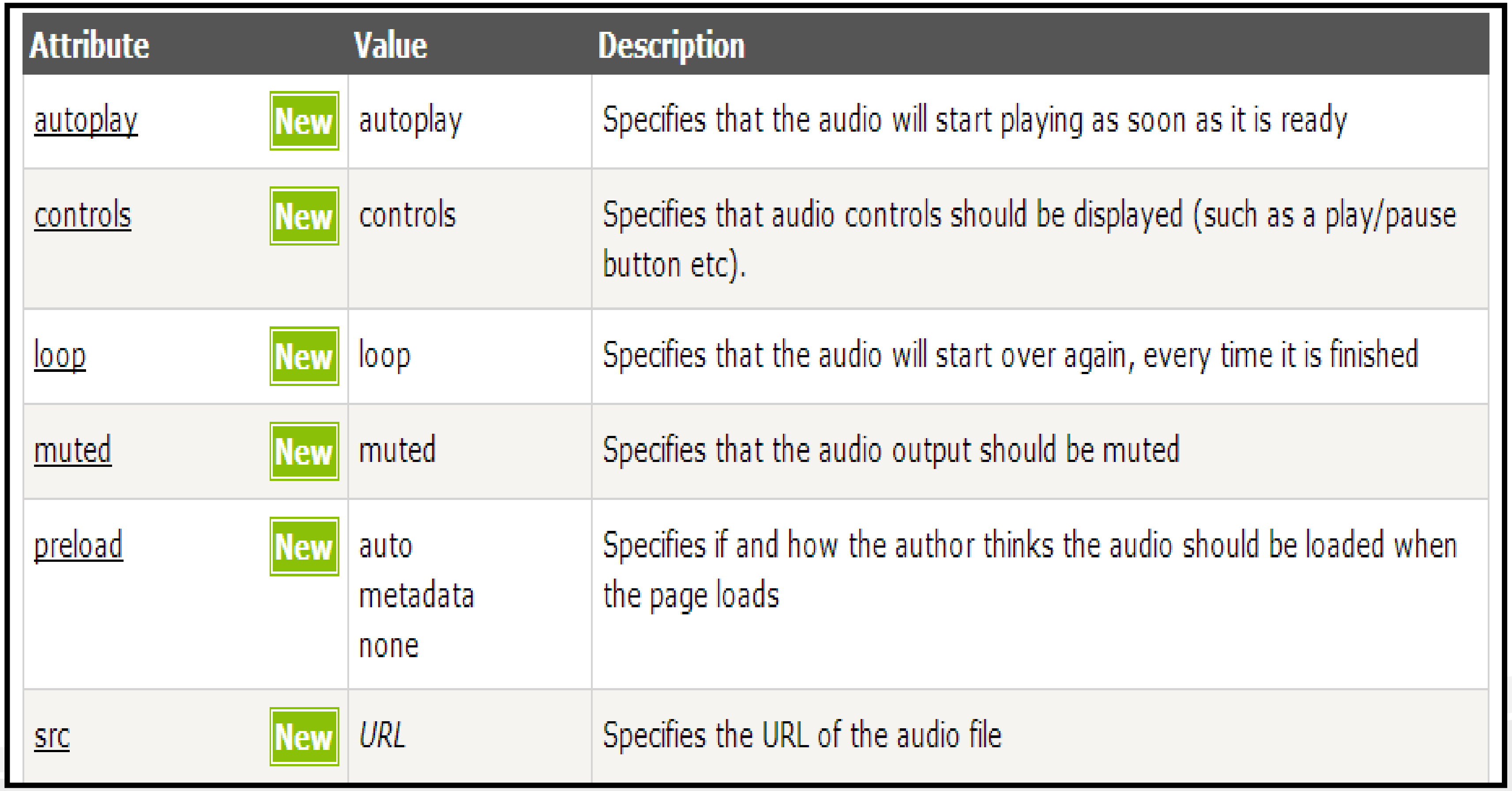
Events



```
<audio controls>
    <source src="Kalimba.mp3" type="audio/mpeg">
    <source src="Kalimba.ogg" type="audio/ogg">
    Your browser does not support this audio format.
</audio>
```

Audio Attributes

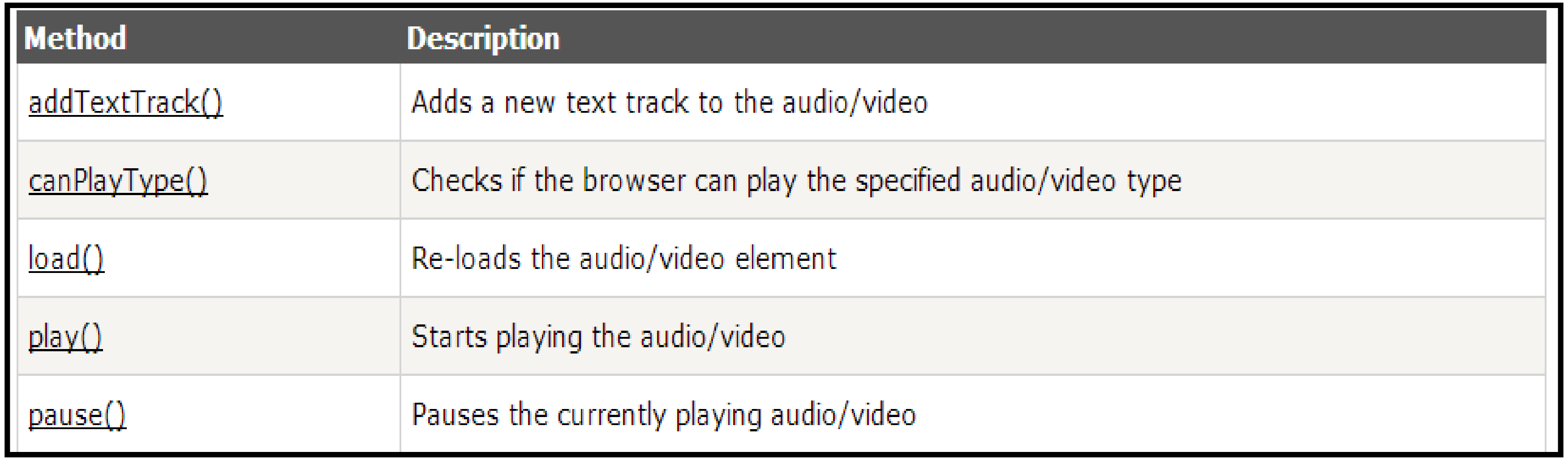
Attributes



Attribute		Value	Description
autoplay	New	autoplay	Specifies that the audio will start playing as soon as it is ready
controls	New	controls	Specifies that audio controls should be displayed (such as a play/pause button etc).
loop	New	loop	Specifies that the audio will start over again, every time it is finished
muted	New	muted	Specifies that the audio output should be muted
preload	New	auto metadata none	Specifies if and how the author thinks the audio should be loaded when the page loads
src	New	*URL*	Specifies the URL of the audio file

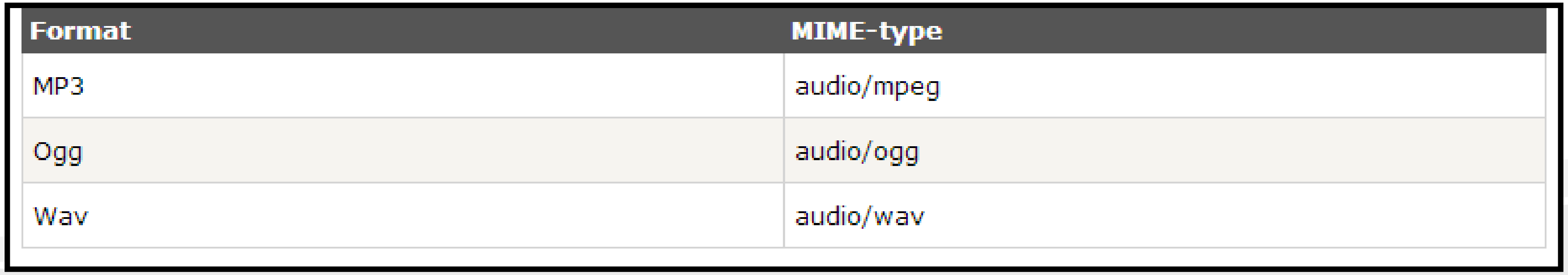
Audio Methods and MIME Types

HTML Audio/ Video Methods



Method	Description
addTextTrack()	Adds a new text track to the audio/video
canPlayType()	Checks if the browser can play the specified audio/video type
load()	Re-loads the audio/video element
play()	Starts playing the audio/video
pause()	Pauses the currently playing audio/video

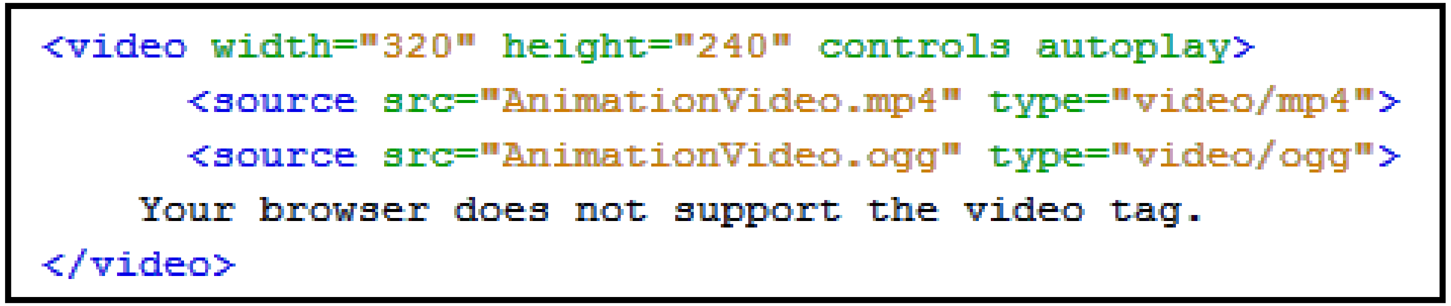
MIME Types for Audio Formats



Format	MIME-type
MP3	audio/mpeg
Ogg	audio/ogg
Wav	audio/wav

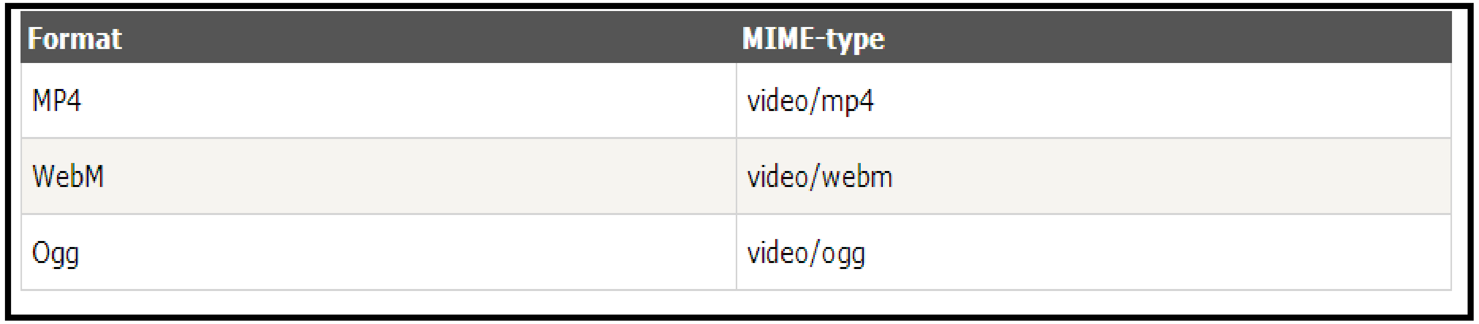
The Video Element

Performs native video playback

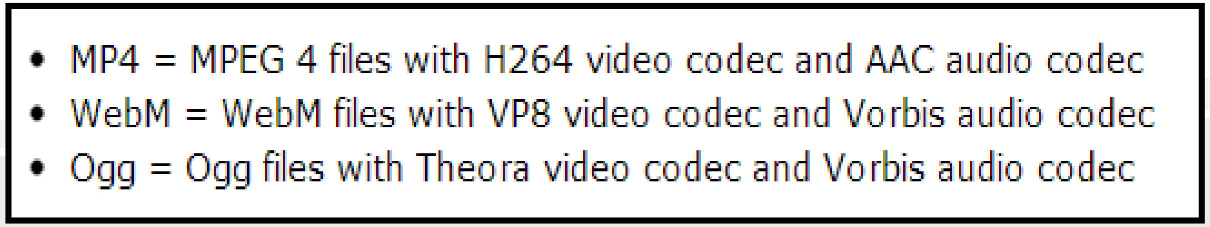


```
<video width="320" height="240" controls autoplay>
      <source src="AnimationVideo.mp4" type="video/mp4">
      <source src="AnimationVideo.ogg" type="video/ogg">
    Your browser does not support the video tag.
</video>
```

MIME Types for Video Formats

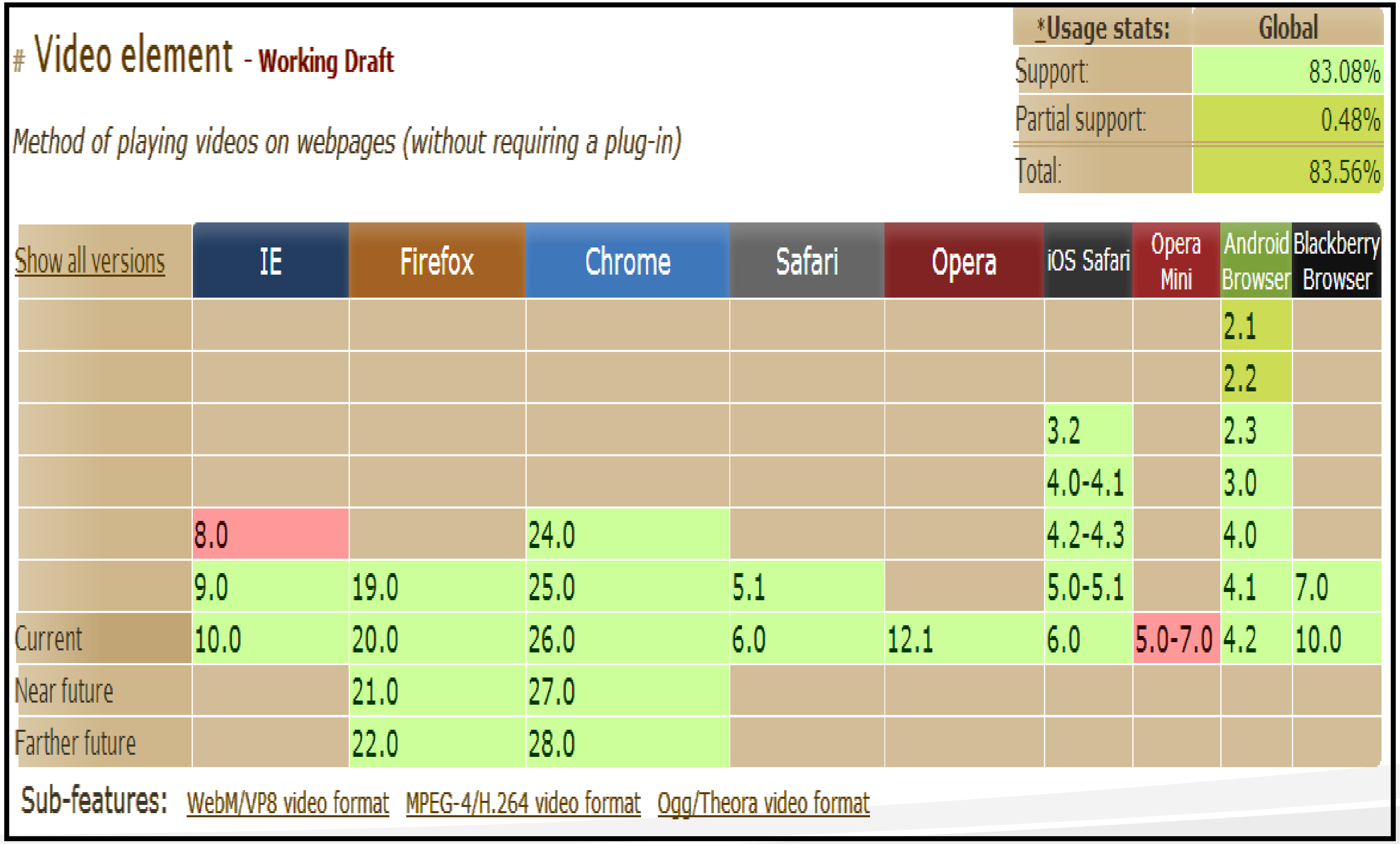


Format	MIME-type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg



- MP4 = MPEG 4 files with H264 video codec and AAC audio codec
- WebM = WebM files with VP8 video codec and Vorbis audio codec
- Ogg = Ogg files with Theora video codec and Vorbis audio codec

The Video: Browser Support



# Video element - **Working Draft**

Method of playing videos on webpages (without requiring a plug-in)

*Usage stats:	Global
Support:	83.08%
Partial support:	0.48%
Total:	83.56%

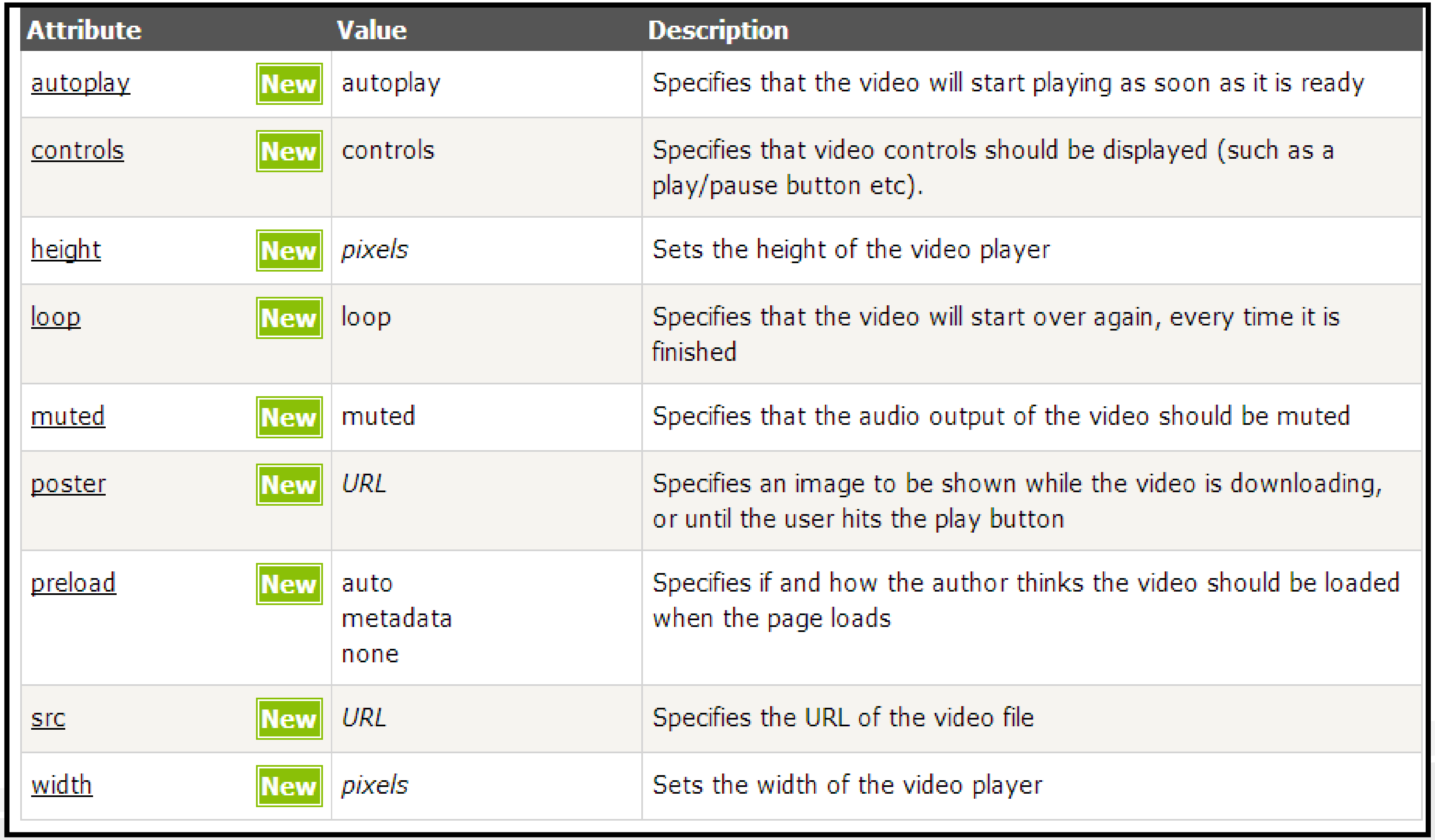
Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser
								2.1	
								2.2	
						3.2		2.3	
						4.0-4.1		3.0	
	8.0		24.0			4.2-4.3		4.0	
	9.0	19.0	25.0	5.1		5.0-5.1		4.1	7.0
Current	10.0	20.0	26.0	6.0	12.1	6.0	5.0-7.0	4.2	10.0
Near future		21.0	27.0						
Farther future		22.0	28.0						

Sub-features: WebM/VP8 video format MPEG-4/H.264 video format Ogg/Theora video format

[source : caniuse.com]

Video Attributes

Optional Attributes



Attribute		Value	Description
autoplay	New	autoplay	Specifies that the video will start playing as soon as it is ready
controls	New	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
height	New	*pixels*	Sets the height of the video player
loop	New	loop	Specifies that the video will start over again, every time it is finished
muted	New	muted	Specifies that the audio output of the video should be muted
poster	New	*URL*	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload	New	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
src	New	*URL*	Specifies the URL of the video file
width	New	*pixels*	Sets the width of the video player

Displaying Multimedia Content on the Web Page

Semantics and Markup

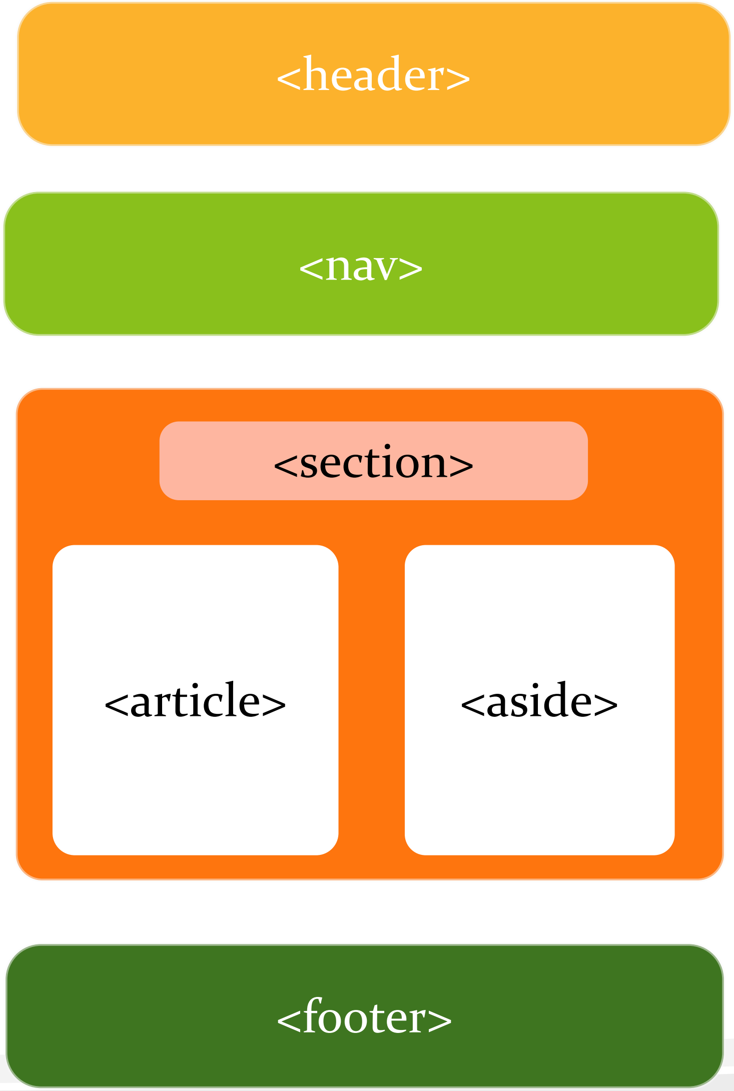
HTML5 Semantics

HTML5 semantics

- Allow describing the elements content
- Simplifies the design of HTML web pages
- Search engines can easily take advantage of the semantic elements to identify content quickly

“The browsers do not actually understand these elements, but the real job is done by CSS”

Using Semantic Elements- HTML 5 Page Layout



<header>

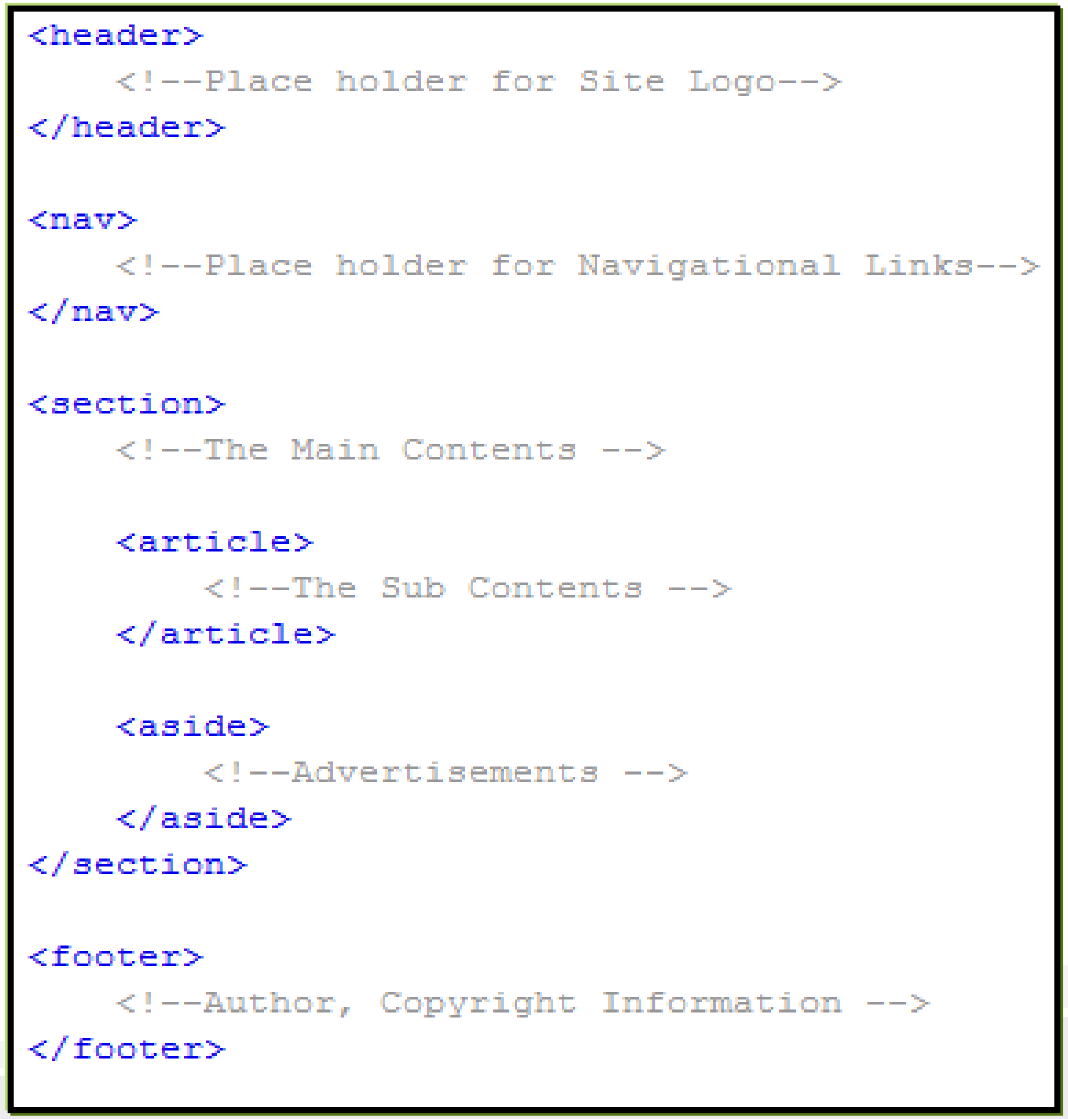
<nav>

<section>

<article>

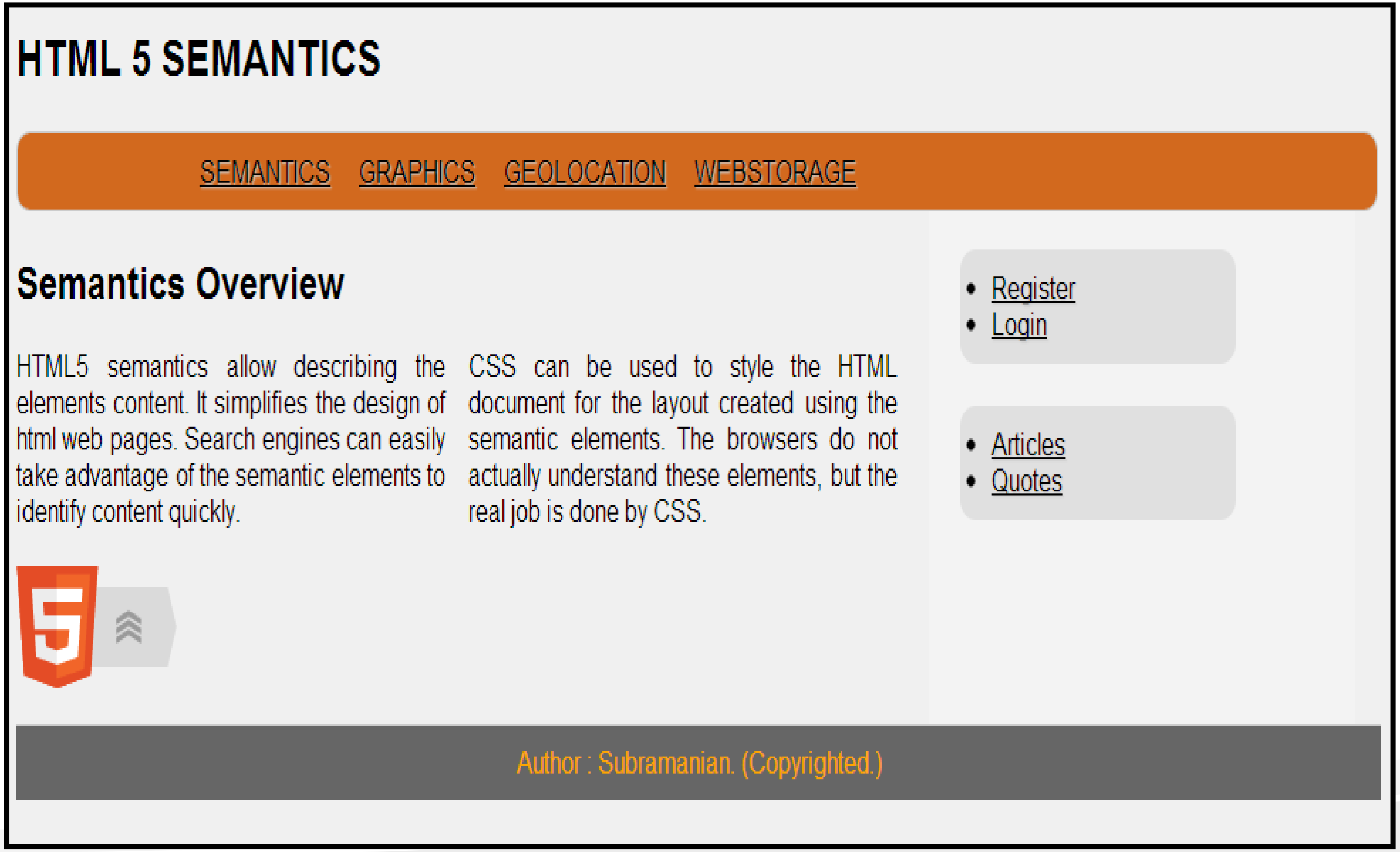
<aside>

<footer>



```
<header>
    <!--Place holder for Site Logo-->
</header>

<nav>
    <!--Place holder for Navigational Links-->
</nav>

<section>
    <!--The Main Contents -->

    <article>
        <!--The Sub Contents -->
    </article>

    <aside>
        <!--Advertisements -->
    </aside>
</section>

<footer>
    <!--Author, Copyright Information -->
</footer>
```

Identify the Semantic Layout Elements



HTML 5 SEMANTICS

SEMANTICS GRAPHICS GEOLOCATION WEBSTORAGE

Semantics Overview

HTML5 semantics allow describing the elements content. It simplifies the design of html web pages. Search engines can easily take advantage of the semantic elements to identify content quickly.

CSS can be used to style the HTML document for the layout created using the semantic elements. The browsers do not actually understand these elements, but the real job is done by CSS.

- Register
- Login

- Articles
- Quotes

Author : Subramanian. (Copyrighted.)

Designing a Web Page Layout using Semantic Elements

Graphics

Graphics: The Canvas

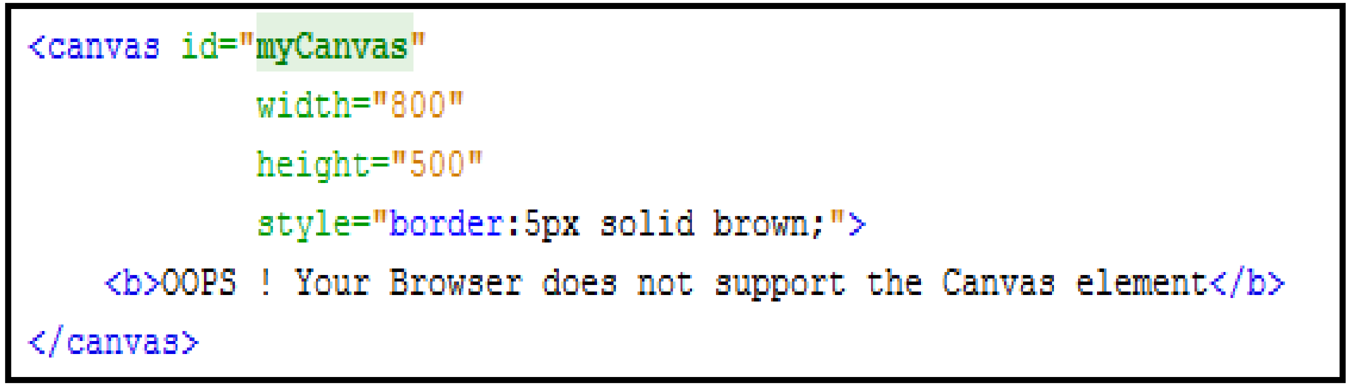
The Canvas Element

- A container
- Perform runtime graphics
- Javascript to draw:
  - Shapes
  - Texts
  - Gradients
  - Images
- 2 D drawing API

> “HTML supports inline SVG and WebGL. WebGL is a context of the canvas HTML element that provides a 3D computer graphics API without the use of plug-ins”

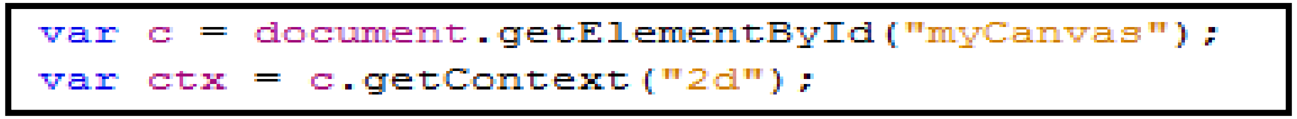
The Canvas- Code Snippets (1 of 2)

Create the Canvas



```
<canvas id="myCanvas"
            width="800"
            height="500"
            style="border:5px solid brown;">
    <b>OOPS ! Your Browser does not support the Canvas element</b>
</canvas>
```

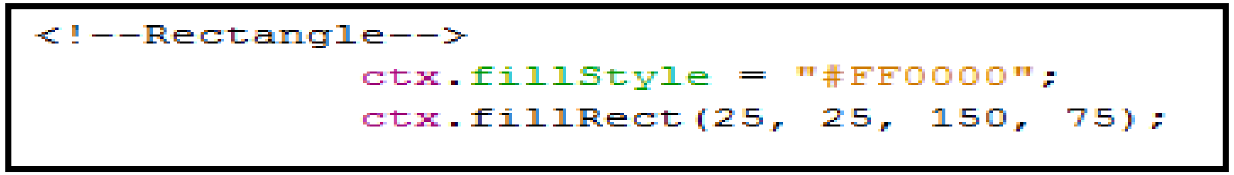
Get the context



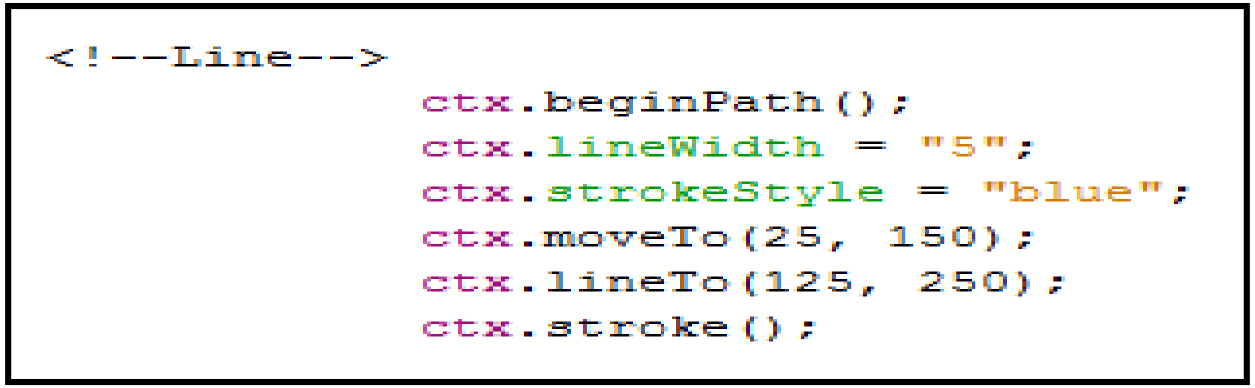
```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
```

The Canvas- Code Snippets (2 of 2)

Draw Shapes

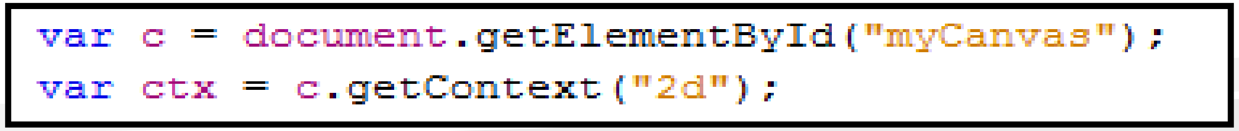


```
<!--Rectangle-->
            ctx.fillStyle = "#FF0000";
            ctx.fillRect(25, 25, 150, 75);
```



```
<!--Line-->
            ctx.beginPath();
            ctx.lineWidth = "5";
            ctx.strokeStyle = "blue";
            ctx.moveTo(25, 150);
            ctx.lineTo(125, 250);
            ctx.stroke();
```

Draw Text



```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
```

Drawing on the Canvas

Geolocation

Geolocation

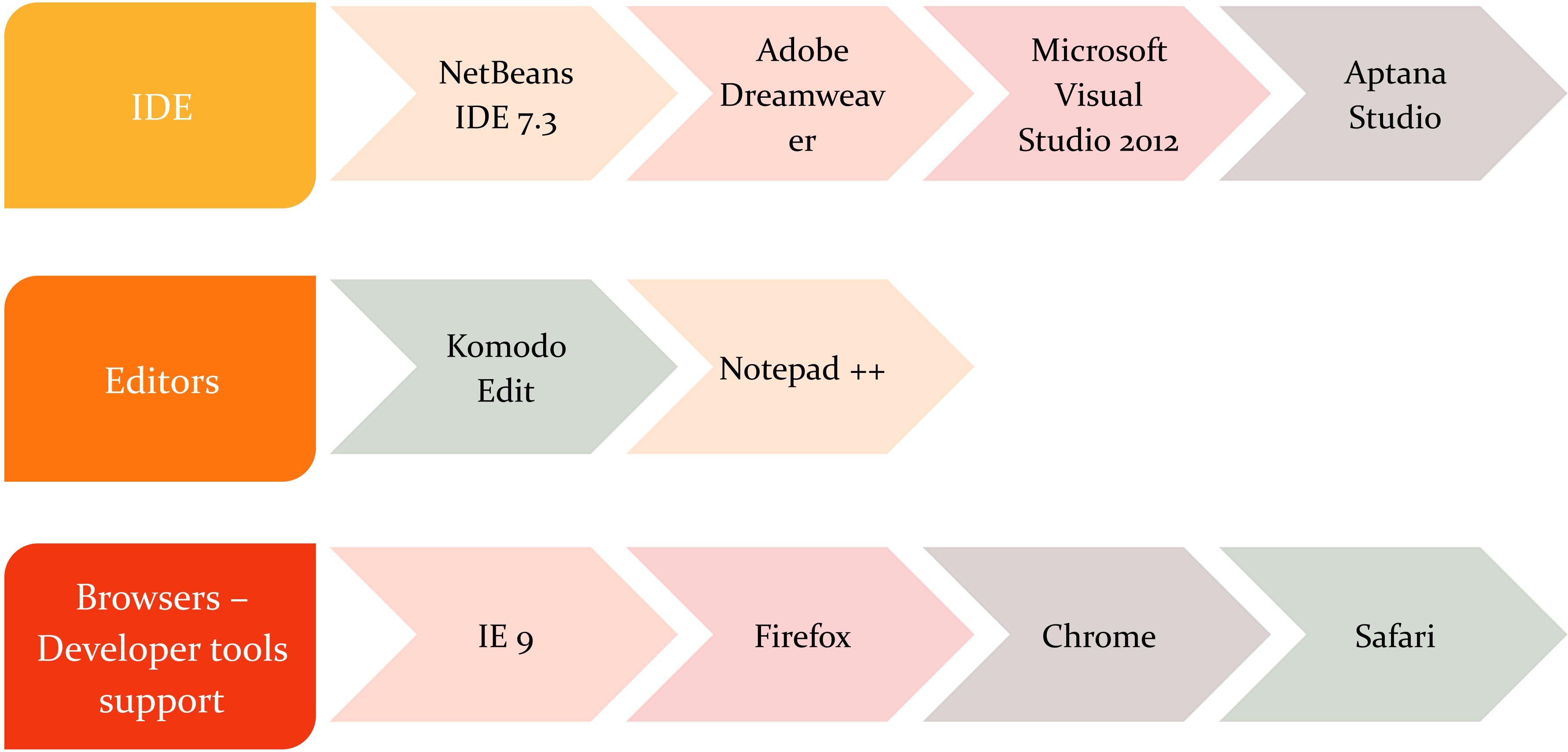
The HTML5 Geolocation API

- Fetches the geographical position of the user.
- Data Protection
- Sources such as WiFi, IP, GPS etc.
- API methods
  - getCurrentPosition
  - watchPosition
  - clearWatch
- API object
  - The Position Object
    - Coordinate properties

Fetching Geographical Location

Tools and IDE’s

Tools & IDE's



IDE

- NetBeans IDE 7.3
- Adobe Dreamweaver
- Microsoft Visual Studio 2012
- Aptana Studio

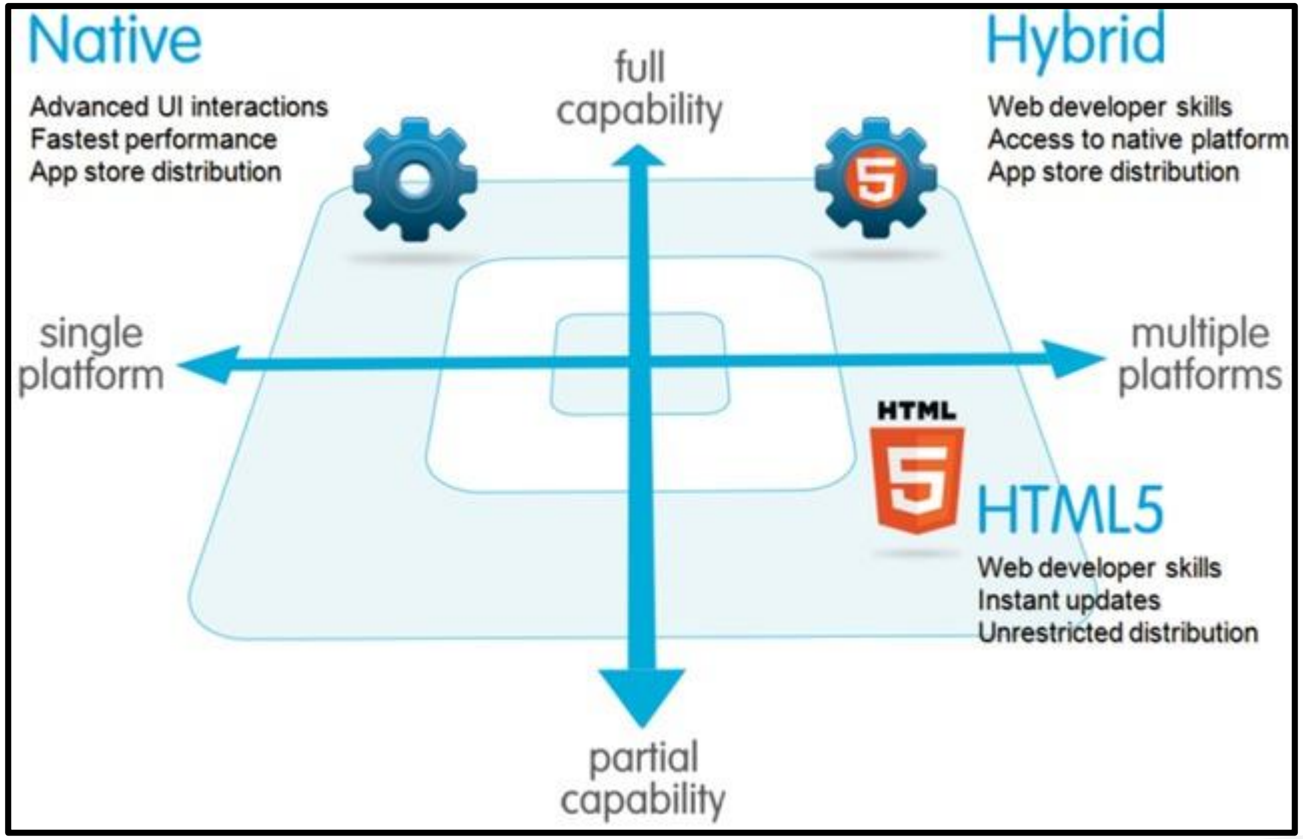
Editors

- Komodo Edit
- Notepad ++

Browsers – Developer tools support

- IE 9
- Firefox
- Chrome
- Safari

The Road Ahead: Web, Native & Hybrid Applications



The Road Ahead

Responsive Web Design (RWD)

- Provides optimal viewing experience
- Addresses ever-changing landscape of devices, browsers, screen sizes and orientations

Thank you!