

HTML Version 5



- ❑ Not yet standardized
(started in 2004, target is 5.0 in 2014)
- ❑ Joint effort between W3C HTML WG and WHATWG (Web Hypertext Application Technology Working Group)
- ❑ Timeline:
 - ❖ 1995: HTML 2
 - ❖ 1997: HTML 4
 - ❖ 1999: HTML 4.01

HTML5 Support

- ❑ Already partially supported by current browsers (have a look at <http://html5test.com>)
 - ❖ At least for most wanted features (such as audio/video)
- ❑ Constant evolution
 - ❖ Check, for example, at [en.wikipedia.org/wiki/Comparison_of_layout_engines_\(HTML5\)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(HTML5)) for the supported features of each **layout engine**
 - ❖ Check, for example, at <http://fmbip.com/litmus> for each feature of a specific browser

HTML5 Support Example

	MAC					WIN								
														
	CHROME	FIREFOX	OPERA	SAFARI		CHROME	FIREFOX	OPERA	IE					
	25	20	12.14	5.1	6	25	15	12	6	7	8	9	10	
Local Storage	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	95%
Session Storage	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	96%
Post Message	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	97%
Offline Applications	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	85%
Workers	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	82%
Query Selector	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	95%
WebSQL Database	✓	✗	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	48%
Indexed Database	✓	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✓	20%
Drag and Drop	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	96%
Hash Change (Event)	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	94%
History Management	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	68%
WebSockets	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	53%
GeoLocation	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	89%
Touch	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	6%
File API	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	23%
Meter element	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	17%
Progress element	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	22%

[Snapshot at Apr 3, 2013 , now check: caniuse.com]

Rationale

□ Help web application authors

[from the HTML5 specification abstract]

- ❖ **New elements** are introduced based on prevailing authoring practices
- ❖ Special attention to defining conformance criteria for user agents in an effort to **improve interoperability**

□ In general: simplify and reduce the need for code and external elements whenever possible

Design Principles

- ❑ Compatibility: support existing content, degrade gracefully (not lose essential functionality when not understood)
- ❑ Utility: solve real problems, separation of concerns (content and presentation)
- ❑ Interoperability: well-defined behavior, avoid needless complexity, handle errors
- ❑ Universal access: media independence, support world languages, accessibility

[<http://www.w3.org/TR/html-design-principles/>]

The «HTML5» Term

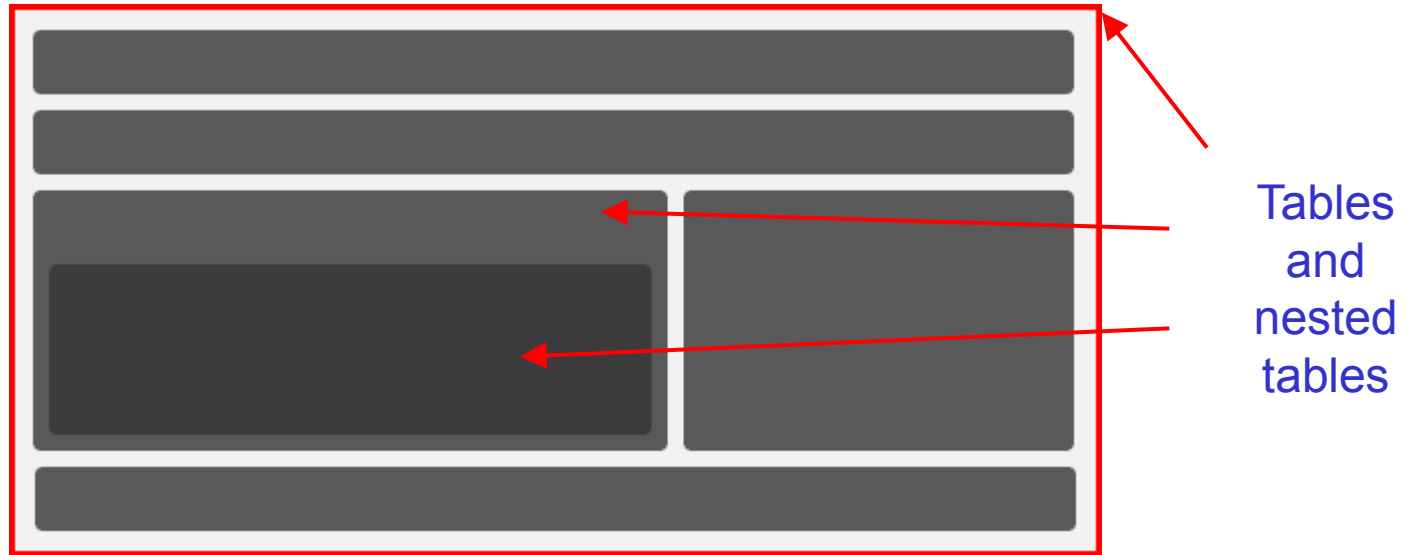
- ❑ Used not only to indicate the HTML5 specification document itself but also for other new technologies and **APIs**, such as:
 - ❖ Local storage
 - ❖ Geolocation
 - ❖ Websockets: full-duplex communications over single TCP channel
 - ❖ WebRTC : real-time communication in the browser

HTML5 in Practice

- ❑ New tags to express the semantic of the document
- ❑ Specific elements for multimedia, e.g., audio and video embedding
- ❑ New elements in forms (**email**, **url**, **search**, **calendar**...)
- ❑ Support for web applications: sessions and local offline storage
- ❑ **canvas** element for graphic drawing

HTML5 Specific Examples

Document (first approach)



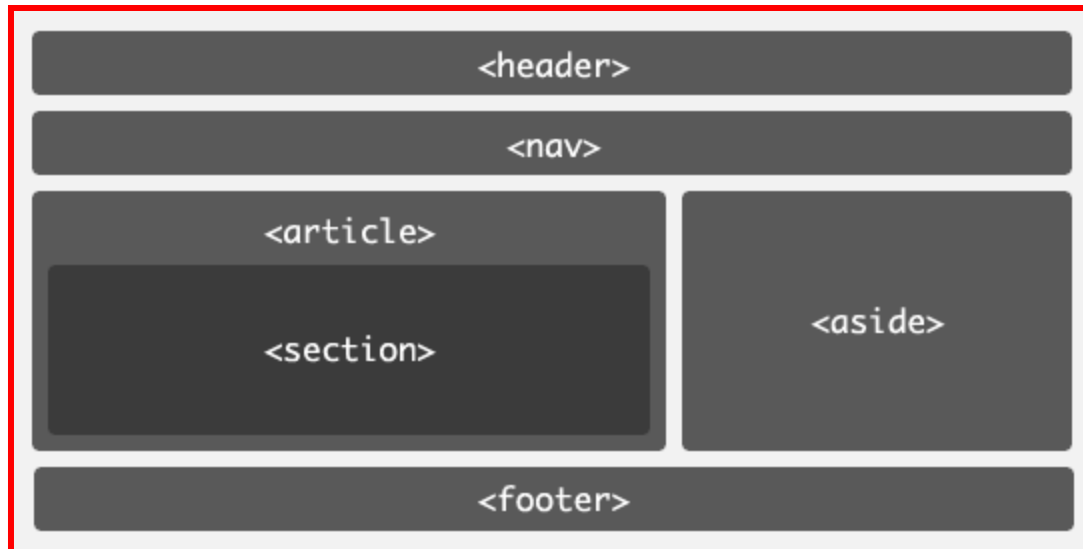
- ❑ Organization and page layout/formatting achieved by means of (nested) TABLE

Document Organization (HTML5)

HTML 4



HTML 5



□ Special tags indicate the content

HTML5: more tags, less scripts and complex syntax

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/1999/xhtml1-transitional.dtd">
<html lang="en" xml:lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset="utf-
8">
  <title>Sample page</title>
  <link rel="stylesheet" href="style-original.css" type="text/css" />
</head>
```

HTML4

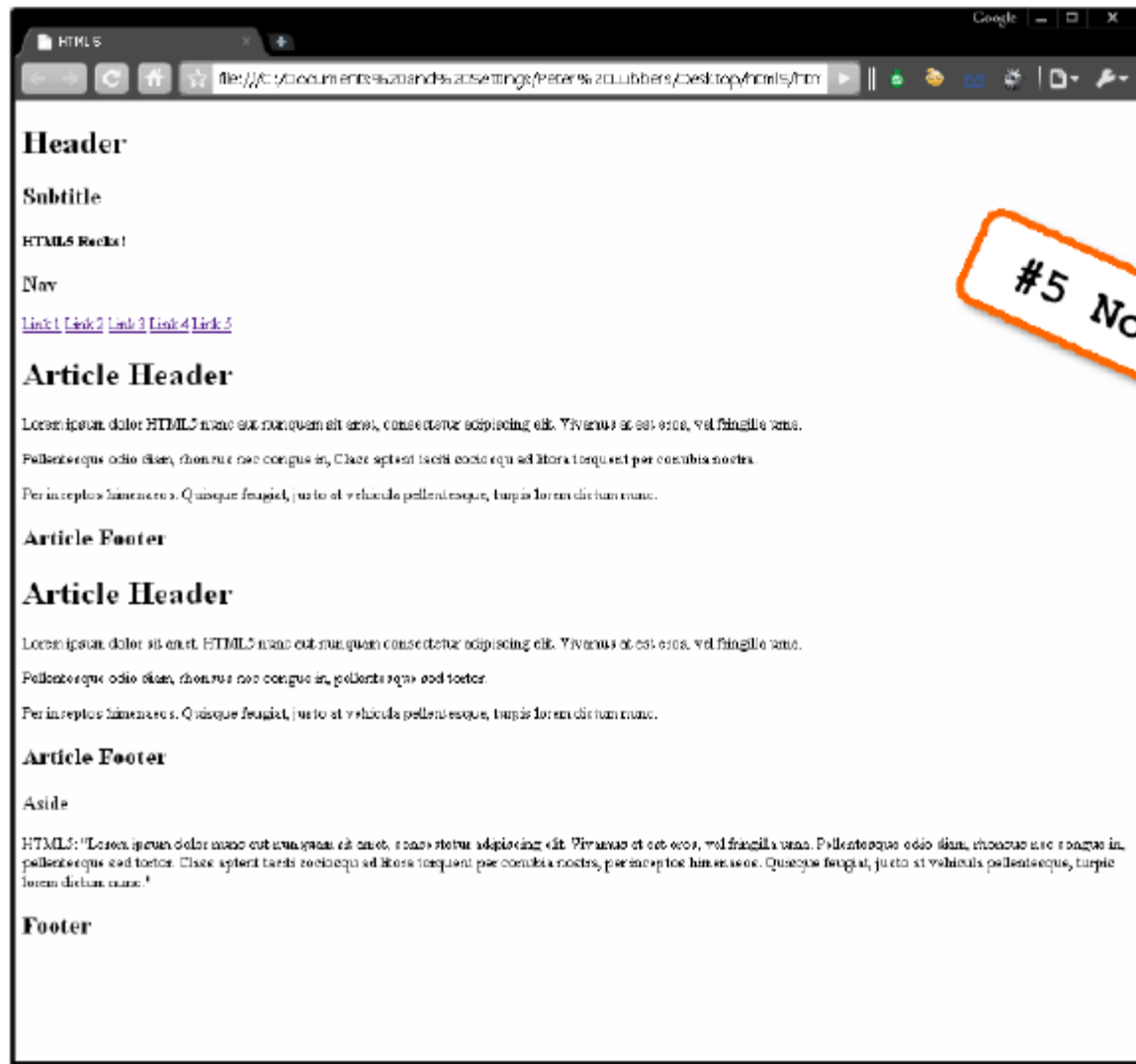
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Sample page</title>
  <link rel="stylesheet" href="style-original.css" />
</head>
```

HTML5

New TAGS

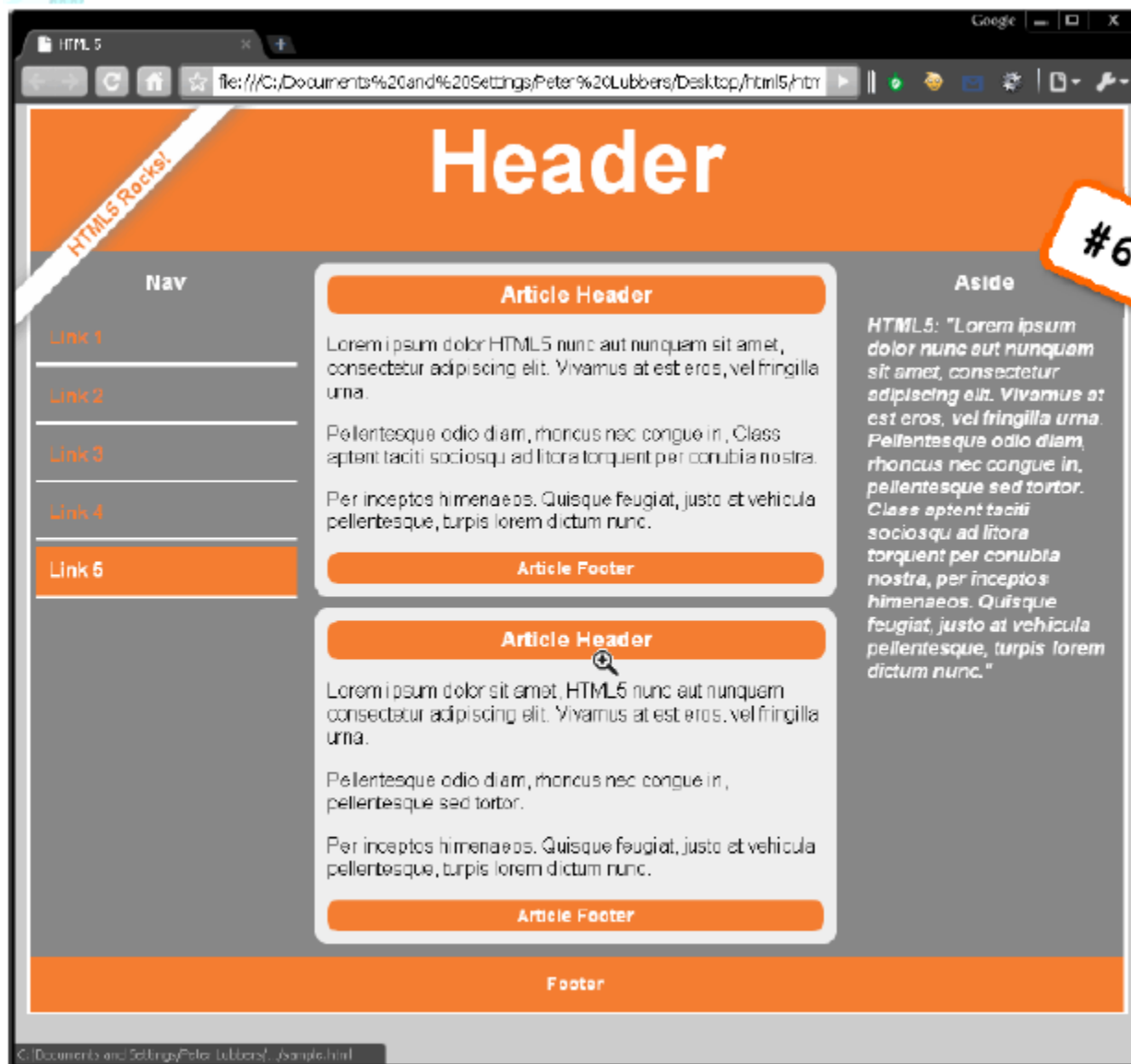
- ❑ New tags have been defined to specify the semantic structure of the document:
`<article>`, `<footer>`, `<header>`,
`<nav>`, `<section>`, `<source>`,
`<figure>`, `<datalist>`, ...)
- ❑ Other tags are for special content such as audio, video, or special purposes such as controls in forms.

Example: Structure



Copyright © 2010 - Kaazing Corporation. All rights reserved.

Example with Formatting



Removed TAGS

- ❑ Many tags have been removed:
`<applet>`, `<big>`, ``, `<big>`,
`<center>`, `<strike>`, `<u>`, `<tt>`,
`<frame>`, `<frameset>`, `<noframes>`
- ❑ The same effect can still be achieved, but using other «cleaner» ways, for instance:
 - ❖ Formatting can be done using CSS
 - ❖ Frames can be created with DIV elements + CSS + scripts

Embedding multimedia

- ❑ The only multimedia content that can be easily embedded before HTML5 were images through the `` tag
- ❑ To include embedded video, audio or animations in the page plug-ins were needed
 - ❖ A plug-in is a separate code module that behaves as though it is part of the browser.

Plug-ins









- ❑ Plug-ins can be used to extend the browser with a wide range of interactive and multimedia capabilities, and they handle one or more data (MIME) types.
- ❑ Design goals
 - ❖ To extend the capabilities of Browser by providing inline viewers for types of data not natively supported
 - ❖ To provide an API (originally defined by Netscape) that is as simple and concise as possible, making it relatively easy to leverage existing native code libraries or convert existing applications to take advantage of the web.

Embedding multimedia (pre-HTML5)

- ❑ Use an applet with video playback capabilities (not easily portable since audio/video decoding was usually done using native code libraries)
- ❑ Rely on Flash plugin player capabilities directly (FLV format, from v.9 supports H.264 directly)
- ❑ Rely on Flash plugin to build a video player (e.g. JWPlayer)
- ❑ Other plugins, e.g., Video Lan Client (VLC), also developed for several platforms

Embedding Video in HTML5

- HTML5 introduces the new `<video>` tag that automatically can handle the following three standards:
 - ❖ **MPEG4**, with AAC audio and H.264 video
 - ❖ **WebM**, with Vorbis audio and VP8 video
 - ❖ **Ogg**, with Vorbis audio and Theora video

	MAC					WIN								
														
	CHROME	FIREFOX	OPERA	SAFARI		CHROME	FIREFOX	OPERA	IE					
	25	20	12.14	5.1	6	25	15	12	6	7	8	9	10	
Video: ogg/theora	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	71%
Video: H.264	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	48%
Video: WebM	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	56%

Embedding Video in HTML5

- ❑ H.264 situation is slowly improving
 - ❖ MPEG-LA had said it wouldn't collect royalties from those freely distributing AVC/H.264 video forever
 - ❖ Cisco released a license-free binary module

#	MPEG-4/H.264 video format - Other										
Commonly used video compression format (not royalty-free)	*Usage stats:							Global			
	Support:							61.42%			
	Partial support:							19.74%			
	Total:							81.16%			
Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Android Browser	Opera Mobile	Blackberry Browser	IE Mobile	
	8.0		31.0			4.2-4.3	4.0	11.5			
	9.0		32.0			5.0-5.1	4.1	12.0			
	10.0	27.0	33.0			6.0-6.1	4.2-4.3	12.1	7.0		
Current	11.0	28.0	34.0	7.0	20.0	7.0	4.4	16.0	10.0	10.0	
Near future		29.0	35.0		21.0						
Farther future		30.0	36.0		22.0						
3 versions ahead		31.0	37.0								

Especially
important
for webRTC

Example

```
<video width="320" height="240"
```

```
  controls="controls">
```

Shows the control
bar

```
    <source src="movie.ogg"
```

```
      type="video/ogg" >
```

The first format that can be
reproduced is played back

```
    <source src="movie.mp4"
```

```
      type="video/mp4" >
```

```
    <source src="movie.webm"
```

```
      type="video/webm" >
```

Your browser does not support the
video tag.

```
</video>
```



Note the `controls` attribute: it allows to
avoid a lot of scripting to control playback!

Other Attributes

❑ **muted="muted"**

- ❖ Defines the initial state of the audio (the only possible value is disabled)

❑ **autoplay="autoplay"**


- ❖ Playback starts immediately

❑ **poster**

- ❖ The URL of an image that represents the video, to be shown while video is not playing

Embedding Audio in HTML5

- ❑ A new tag is introduced: `<audio>`
- ❑ Recognized formats:
 - ❖ WAV
 - ❖ MP3 / AAC
 - ❖ Vorbis

	MAC					WIN									
															
	CHROME	FIREFOX	OPERA	SAFARI		CHROME	FIREFOX	OPERA	IE						
	25	20	12, 14	5.1	6	25	15	12	6	7	8	9	10		
Audio: ogg/vorbis	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	72%	
Audio: mp3	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	48%	
Audio: wav	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	77%	
Audio: AAC	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	48%	

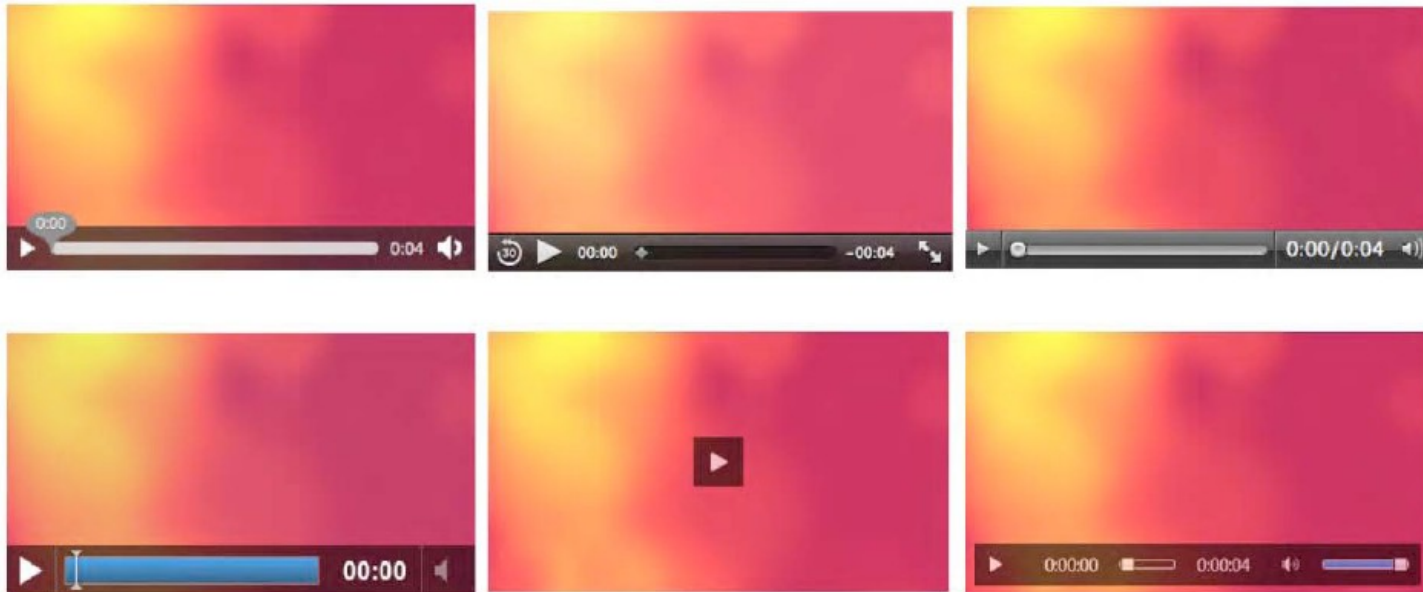
Audio Attributes

□ Attributes

- ❖ **autoplay, src, controls** Same meaning as in video
- ❖ **loop="loop"** Audio is reproduced continuously, restarting immediately the playback when the end of file is reached

Uniform Controls for Multimedia

- ❑ Every browser shows controls with its own style.



From left to right, top to bottom: `<video>` with controls in Firefox, Safari and Opera, Chrome, IE with width 300px, and IE with width 400px

Uniform Controls for Multimedia

- ❖ It is possible to make a **uniform interface across browsers**: do NOT use `controls` attribute but create an HTML layout with CSS for buttons + Javascript to handle press button events and interact with the HTML5 API of the video object



[From: <http://icant.co.uk/easy-youtube>]

Elements («controls») in forms

- ❑ In previous HTML versions, most of the input was done using `<input type="text" ...>` elements
- ❑ Often, there is need to check input validity
For example:
number : only digits
email address: correct syntax with @
etc...
 - ❖ This can traditionally be done only using scripts

New elements in HTML5 forms

❑ `email, url, number, range, tel, color, search`

- ❖ They automatically validate the input
- ❖ In the case of number and range, a range check is performed
- ❖ In case of range, values are shown as a combo box (in certain browsers)

❑ `date, month, week, time, datetime, datetime local`

- ❖ Allows to easily select date and time showing a calendar etc.

HTML5 Form Support

	MAC					WIN								
														
	CHROME	FIREFOX	OPERA	SAFARI		CHROME	FIREFOX	OPERA	IE					
	25	20	12.14	5.1	6	25	15	12	6	7	8	9	10	
Form: Search	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	69%
Form: Phone	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	69%
Form: URL	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	68%
Form: Email	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	68%
Form: DateTime	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	5%
Form: Date	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	14%
Form: Month	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	6%
Form: Week	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	6%
Form: Time	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	9%
Form: LocalTime	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	6%
Form: Number	✓	✗	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓	37%
Form: Range	✓	✗	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓	46%
Form: Colour	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	11%

Session

- ❑ The HTTP protocol is stateless
 - ❖ Each request is independent of the others
- ❑ How to maintain information between request?
 - ❖ A session is needed: way to store information (on the server or the client) needed while navigating a website (e.g., shopping cart)
- ❑ Old approach
 - ❖ Use **cookies** (small pieces of information stored in the browser, in the form of id=value)
 - ❖ Pass them as **parameters** of the HTTP request (difficult and inefficient depending on data amount)

Session in HTML5

- ❑ It is possible to replace cookies to create sessions

- ❑ A new Javascript object is available

sessionStorage

in which some data about the current session can be stored

- ❖ Global to the browser (shared across tabs)

- ❑ The memory associated with the session is released when the browser is closed

Local storage

- ❑ It is a JavaScript object
- ❑ It allows to permanently store data (on disk) associated with the page
 - ❖ Persistent also when browser is closed
- ❑ Everything is stored in textual form
- ❑ JavaScript provides functions to convert string to numbers, arrays, etc., and vice versa
- ❑ NB: Different from cookies.
 - ❖ Data on disk are only read on request from code, cookies always sent in HTTP requests

Web Application Features

- Local and Session storage, support for offline applications, Workers, Drag-and-drop, simple databases and similar features make HTML5 a framework to develop feature-rich applications which **only need a browser** to run!

	MAC					WIN								
														
	CHROME	FIREFOX	OPERA	SAFARI		CHROME	FIREFOX	OPERA	IE					
	25	20	12.14	5.1	6	25	15	12	6	7	8	9	10	
Local Storage	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	95%
Session Storage	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	96%
Offline Applications	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	85%
Workers	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	82%
Drag and Drop	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	96%

Graphics

- ❑ The `<canvas>` tag can be used as a «portrait» on which it is possible to draw using **JavaScript**
- ❑ Images from a file can be loaded into the canvas
- ❑ Several shapes are available for drawing through **scripts**

Example

```
<!DOCTYPE HTML>
<html>
<body>
<canvas id="tela" width="1000" height="500" style="border:1px solid
    #c3c3c3;">
```

Your browser does not support the canvas element.

```
</canvas>
<script type="text/javascript">
var c=document.getElementById("tela");
var cxt=c.getContext("2d");
var img=new Image();
//load an image from file
img.src="./questions.jpg";
cxt.drawImage(img,0,0);
//draw a red circle
cxt.fillStyle="#FF0000";
cxt.beginPath();
cxt.arc(70,18,15,0,Math.PI*2,true);
cxt.closePath();
cxt.fill();
</script>
</body>
</html>
```

