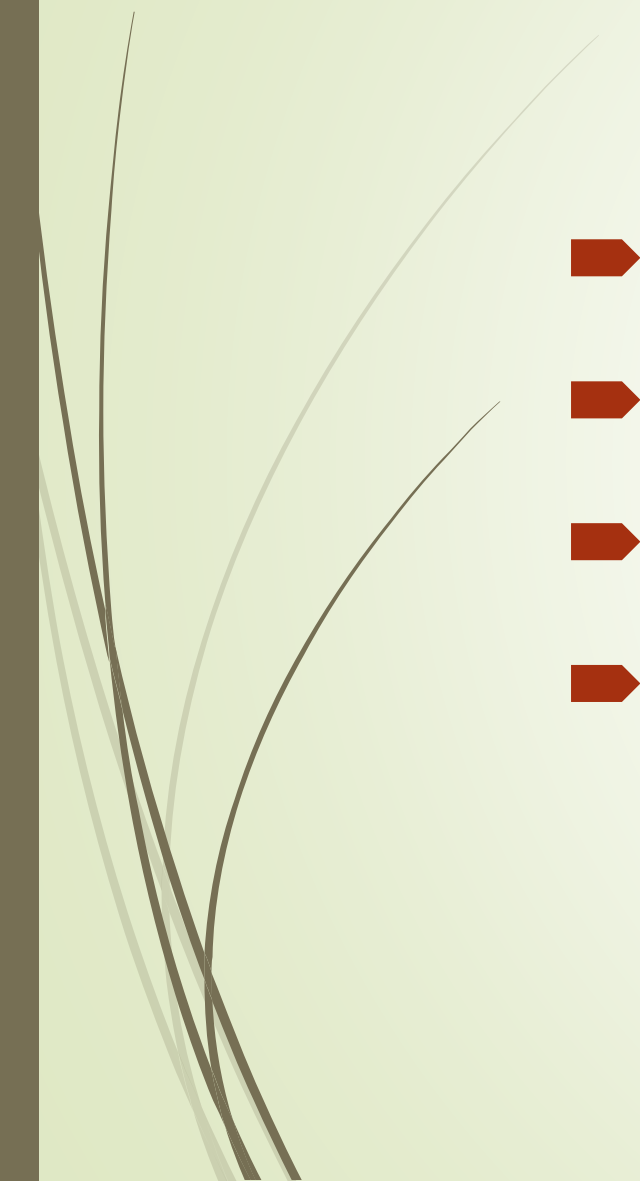# WHAT'S UP, HTML5?

New in HTML5

- A new DOCTYPE
- New elements and attributes
- Obsolete 4.01 elements
- APIs

A minimal DOCTYPE

- The HTML5 declaration is short and sweet:
  - <! DOCTYPE html>
    - Compare that to a declaration for a Strict HTML 4.01 document:
    - <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/HTML4.01/strict.dtd">
    - All the above syntax is case-insensitive
- HTML5 does not have a DTD, which is why we have the simple DOCTYPE declaration.

Character Encoding

- Character Encoding
  - <meta charset="UTF-8">
  - All the above syntax is case-insensitive

The <script> tag

- Syntax
  - <script type="text/javascript" src="scriptfile.js"></script>
- HTML 5 removes extra information required and you can use simply following syntax
  - <script src="scriptfile.js"></script>

The <link> tag

- Syntax
  - <link rel="stylesheet" type="text/css" href="stylefile.css">
- HTML 5 removes extra information required and you can use simply following syntax
  - <link rel="stylesheet" href="stylefile.css">

Elements and attributes

- Besides the new input element which were introduced in the previous slide such as: color, date, datetime… HTML5 also introduce new elements and atributes

article	datalist	header	output	source
aside	details	hgroup	progress	summary
audio	embed	keygen	rp	time
bdi	figcaption	mark	rt	track
canvas	figure	meter	ruby	video
command	footer	nav	section	wbr

Organizing Page Content with HTML5 elements

- Article
- Section
- Aside
- Header
- Footer
- address

Article and Section

<section>…</section>
Thematic group of content
NEW IN HTML5

<article>…</article>
Self-contained, reusable composition

- To divide long web documents into thematic sections
- One section can contain many articles or one article can contain many section

```
<article>
   <h1>Get to Know Helvetica</h1>
   <section>
      <h2>History of Helvetica</h2>
      <p>…</p>
   </section>

   <section>
      <h2>Helvetica Today</h2>
      <p>…</p>
   </section>
</article>
```

```
<section id="essays">
   <article>
      <h1>A Fresh Look at Futura</h1>
      <p>…</p>
   </article>

   <article>
      <h1>Getting Personal with Humanist</h1>
      <p>…</p>
   </article>
</section>
```

Aside (sidebars)

- The **aside** element identifies content that is related but tangential to the surrounding content

```
<h1>Web Typography</h1>
<p>Back in 1997, there were competing font formats and tools for making them…</p>
<p>We now have a number of methods for using beautiful fonts on web pages…</p>
<aside>
  <h2>Web Font Resources</h2>
  <ul>
  <li><a href="http://typekit.com/">Typekit</a></li>
  <li><a href="http://www.google.com/webfonts">Google Fonts</a></li>
  </ul>
</aside>
```

Navigation

<nav>...</nav>
Primary navigation links

- The new **nav** element gives developers a semantic way to identify navigation for a site

```
<nav>
<ul>
  <li><a href="">Serif</a>/li>
  <li><a href="">Sans-serif</a></li>
  <li><a href="">Script</a></li>
  <li><a href="">Display</a></li>
  <li><a href="">Dingbats</a>/li>
</ul>
</nav>
```

Headers

- Header can appear at the beginning of the web page or in an individual article or section

```
<header>
  <img src="/images/logo.png">
  <hgroup>
  <h1>Nuts about Web Fonts</h1>
  <h2>News from the Web Typography Front</h2>
  </hgroup>
  <nav>
    <ul>
      <li><a href="">Home</a></li>
      <li><a href="">Blog</a></li>
      <li><a href="">Shop</a></li>
    </ul>
  </nav>
</header>
```

```
<article>
  <header>
    <h1>More about WOFF</h1>
    <p>by Jennifer Robbins, <time datetime="11-11-2011"
    pubdate>November 11, 2011</time></p>
  </header>
  <p>...article content starts here...</p>
</article>
```

Footer

\<footer>...\</footer>

Footer for page, section, or article

```
<article>
  <header>
    <h1>More about WOFF</h1>
    <p>by Jennifer Robbins, <time datetime="11-11-2011"
    pubdate>November 11, 2011</time></p>
  </header>
  <p>...article content starts here...</p>
  <footer>
    <p><small>Copyright &copy;2012 Jennifer Robbins.</small></p>
    <nav>
    <ul>
      <li><a href="">Previous</a></li>
      <li><a href="">Next</a></li>
    </ul>
    </nav>
  </footer>
</article>
```

HTML5 Document - 1

- The following tags have been introduced for better structure

- **section:** This tag represents a generic document or application section. It can be used together with h1-h6 to indicate the document structure.

- **article**: This tag represents an independent piece of content of a document, such as a blog entry or newspaper article.

- **aside**: This tag represents a piece of content that is only slightly related to the rest of the page.

- **header**: This tag represents the header of a section.

HTML5 Document - 2

- **footer**: This tag represents a footer for a section and can contain information about the author, copyright information, etc.

- **nav**: This tag represents a section of the document intended for navigation.

- **dialog**: This tag can be used to mark up a conversation.

- **figure**: This tag can be used to associate a caption together with some embedded content, such as a graphic or video.

```
<!DOCTYPE html>

<html>

    <head>
        <meta charset="utf-8">
        <title>...</title>
    </head>

    <body>

        <header role="banner">
            <h1>HTML5 Document Structure Example</h1>
            <p>This page should be tried in safari, chrome or Mozila.</p>
        </header>

        <nav>

            <ul>
                <li><a href="#">HTML Tutorial</a></li>
                <li><a href="#">CSS Tutorial</a></li>
                <li><a href="#">JavaScript Tutorial</a></li>
            </ul>

        </nav>
```

```html
<article>
    <section>
        <p>Once article can have multiple sections</p>
    </section>
</article>

<aside>
    <p>This is  aside part of the web page</p>
</aside>

<figure align="right">
    <img src="logo.png" alt="QuangHD" width="200" height="100">
</figure>

<footer>
    <p>Created by <a href="#">QuangHD</a></p>
</footer>

</body>
</html>
```

Result

HTML5 Document Structure Example

This page should be tried in safari, chrome or Mozila.

- HTML Tutorial
- CSS Tutorial
- JavaScript Tutorial

Once article can have multiple sections

This is aside part of the web page

Created by QuangHD

Standard Attributes - 1

Attribute	Options	Function
accesskey	User Defined	Specifies a keyboard shortcut to access an element.
align	right, left, center	Horizontally aligns tags
background	URL	Places an background image behind an element
bgcolor	numeric, hexidecimal, RGB values	Places a background color behind an element
class	User Defined	Classifies an element for use with Cascading Style Sheets.
contenteditable	true, false	Specifies if the user can edit the element's content or not.
contextmenu	Menu id	Specifies the context menu for an element.

Standard Attributes - 2

Attribute	Options	Function
data-XXXX	User Defined	Custom attributes. Authors of a HTML document can define their own attributes. Must start with "data-".
draggable	true,false, auto	Specifies whether or not a user is allowed to drag an element.
height	Numeric Value	Specifies the height of tables, images, or table cells.
hidden	hidden	Specifies whether element should be visible or not.
id	User Defined	Names an element for use with Cascading Style Sheets.
item	List of elements	Used to group elements.
itemprop	List of items	Used to group items.

Standard Attributes - 3

Attribute	Options	Function
spellcheck	true, false	Specifies if the element must have it's spelling or grammar checked.
style	CSS Style sheet	Specifies an inline style for an element.
subject	User define id	Specifies the element's corresponding item.
tabindex	Tab number	Specifies the tab order of an element.
title	User Defined	"Pop-up" title for your elements.
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
width	Numeric Value	Specifies the width of tables, images, or table cells.

Custom Attributes

- A new feature being introduced in HTML 5 is the addition of custom data attributes.

- A custom data attribute starts with **data-** and would be named based on your requirement. Following is the simple exampleof custom data attributes.

  - <div class="example" data-subject="physics" data-level="complex">

  - ...

  - </div>

- The above will be perfectly valid HTML5 with two custom attributes called*data-subject* and *data-level*.

- You would be able to get the values of these attributes using JavaScript APIs or CSS in similar way as you get for standard attributes.

HTML5 – Web Forms 2.0 - 1

- The <input> element in HTML4

text	password
checkbox	radio
submit	file
image	hidden
select	textarea
button	

Example

```html
<form action="Controller" method="post">
<p>

    <label for="firstname">first name: </label>
    <input type="text" id="firstname"><br />

    <label for="lastname">last name: </label>
    <input type="text" id="lastname"><br />

    <label for="email">email: </label>
    <input type="text" id="email"><br>

    <input type="radio" name="sex" value="male"> Male<br>
    <input type="radio" name="sex" value="female"> Female<br>
    <input type="submit" value="send"> <input type="reset">

</p>
</form>
```

HTML5 – Web Forms 2.0 - 2

- New value for type attirbute of <input> element in HTML5

datetime	datetime-local
date	month
week	time
number	range
email	url

The placeholder attribute

- HTML5 introduced a new attribute called **placeholder**.
- Use on <input> and <textarea> elements provides a hint.
- Examples:

```
<form action="Controller" method="get">
    Enter email : <input type="email" name="email"
        placeholder="email@example.com"/>
    <input type="submit" value="submit" />
</form>
```

The required attribute

- HTML5 introduced a new attribute called **required** replace javascript for client side validations empty text box

- Example:

```html
<form action="Controller" method="get">
    Enter email : <input type="email" name="email"
        placeholder="email@example.com" required/>
    <input type="submit" value="submit" />
</form>
```

Obsolete HTML 4.01 Markup

acronym	dir	noframes
applet	font	strike
basefont	frame	tt
big	frameset	
center	isindex	

APIs

- Editing API
- Drag and Drop API
- Canvas AP
- Web Storage API
- Geolocation API
- Web Workers API
- Web Sockets API

Drag and Drop

- Drag and Drop (DnD) is powerful User Interface concept
- allows the user to click and hold the mouse button down over an element
  - drag it to another location
  - release the mouse button to drop the element there
- DnD is supported by all the major browsers like Chrome, Firefox 3.5 and Safari 4 etc

DnD Events

- dragstart: Fires when the user starts dragging of the object.
- dragenter: Fired when the mouse is first moved over the target element while a drag is occuring.
- dragover: This event is fired as the mouse is moved over an element when a drag is occuring.
- dragleave: This event is fired when the mouse leaves an element while a drag is occuring.
- drag: Fires every time the mouse is moved while the object is being dragged.
- drop: The drop event is fired on the element where the drop was occured at the end of the drag operation.
- dragend: Fires when the user releases the mouse button while dragging an object.

Example

```html
<!DOCTYPE HTML>
<html>
    <head>
        <style type="text/css">
            #boxA, #boxB {
                float:left;padding:10px;margin:10px; -moz-user-select:none;
            }
            #boxA { background-color: #6633FF; width:75px; height:75px;   }
            #boxB { background-color: #FF6699; width:150px; height:150px; }
        </style>
        <script type="text/javascript">
            function dragStart(ev) {
                ev.dataTransfer.effectAllowed='move';
                ev.dataTransfer.setData("Text", ev.target.getAttribute('id'));
                ev.dataTransfer.setDragImage(ev.target,0,0);
                return true;
            }
        </script>
    </head>
    <body>
        <center>
            <h2>Drag and drop HTML5 demo</h2>
            <div>Try to drag the purple box around.</div>
            <div id="boxA" draggable="true"
                ondragstart="return dragStart(ev)">
                <p>Drag Me</p>
            </div>
            <div id="boxB">Dustbin</div>
        </center>
    </body>
</html>
```

Videos supported

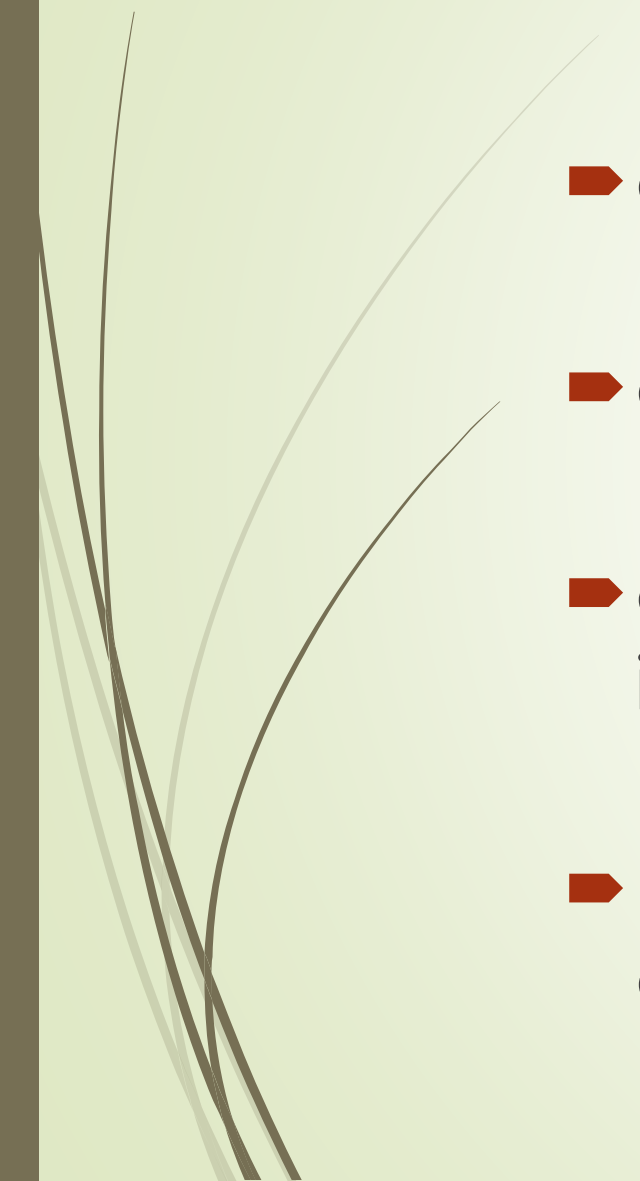| Format	Type	IE	Chrome	Firefox	Safari	Opera Mobile	Mobile Safari	Android
Ogg Theora	video/ogg	–	5.0+	3.5+	–	10.5+	–	–
MP4/H.264	video/mp4	9.0+	–	–	3.1+	–	3.0+	2.0+
WebM	video/webm	9.0+	6.0+	4.0+	–	11+	–	2.3.3+

Adding a video to a page

`<video>...</video>`

Adds a video player to the page

```
<video src="highlight_reel.mp4" width="640" height="480"
poster="highlight_still.jpg" controls autoplay>
</video>
```

Video Attributes - 1

- autoplay: video will automatically begin to play back

- autobuffer: video will automatically begin buffering

- controls: allow to control video playback, including volume, seeking, and pause/resume playback

- height: specifies the height of the video's display area, in CSS pixels.

Video Attributes - 2

- loop: allow video automatically seek back to the start after reaching at the end

- preload: video will be loaded at page load, and ready to run. Ignored if autoplay is present

- poster: URL of an image to show until the user plays or seeks

- src: The URL of the video to embed.

- width: width of the video's display area, in CSS pixels.

Audio supported

Format	Type	IE	Chrome	Firefox	Safari	Opera Mobile	Mobile Safari	Android
MP3	audio/mpeg	9.0+	5.0+	–	4+	–	3.0+	2.0+
WAV	audio/wav or audio/wave	–	5.0+	3.5+	4+	10.5+	3.0+	2.0+
Ogg Vorbis	audio/ogg	–	5.0+	3.5+	–	10.5+	–	2.0+
MPEG-4/AAC	audio/mp4	9.0+	5.0+	–	4+	–	3.0+	2.0+
WebM	audio/webm	9.0+	6.0+	4.0+	–	11+	–	2.3.3+

Adding audio to a page

<audio>...</audio>

Adds an audio file to the page

```
<audio id="soundtrack" controls preload="auto">
  <source src="soundtrack.mp3" type="audio/mp3">
  <source src="soundtrack.ogg" type="audio/ogg">
  <source src="soundtrack.webm" type="audio/webm">
</audio>
<p>Download the Soundtrack song:</p>
<ul>
  <li><a href="soundtrack.mp3">MP3</a></li>
  <li><a href="soundtrack.ogg">Ogg Vorbis</a></li>
</ul>
```
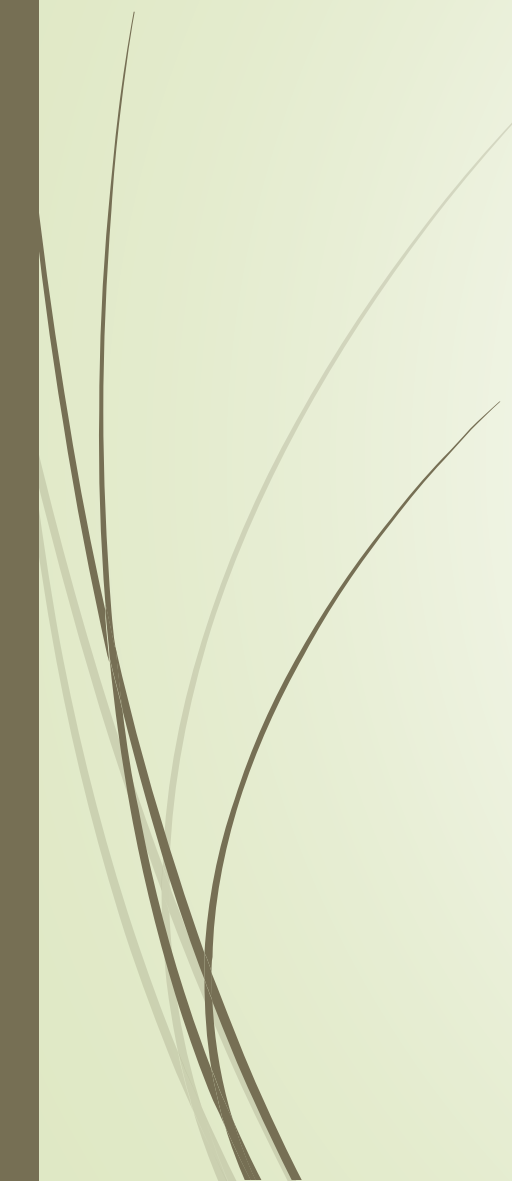
Audio Attributes

- autoplay: video will automatically begin to play back

- autobuffer: video will automatically begin buffering

- controls: allow to control video playback, including volume, seeking, and pause/resume playback

- loop: allow video automatically seek back to the start after reaching at the end

- preload: video will be loaded at page load, and ready to run. Ignored if autoplay is present

- src: The URL of the audio to embed.

Canvas

- Gives an easy and powerful way to draw graphics using JavaScript.

- Can be used to draw graphs

- Make photo compositions

- Do simple (and not so simple) animations

Example

```html
<!DOCTYPE HTML>
<html>
    <head>
        <style>
            #test {
                width: 150px;
                height:150px;
                margin: 0px auto;
                border: 1px solid red;
            }
        </style>
        <script type="text/javascript">
            function drawShape(){
                var canvas = document.getElementById('mycanvas');
                // Make sure we don't execute when canvas isn't supported
                if (canvas.getContext){
                    // use getContext to use the canvas for drawing
                    var ctx = canvas.getContext('2d');
                    ctx.fillRect(25,25,100,100);
                }
                else { alert('You need Safari or Firefox 1.5+ to see this demo.'); }
            }
        </script>
    </head>
    <body id="test" onload="drawShape();">
        <canvas id="mycanvas"></canvas>
    </body>
</html>
```

Summary

- Introduce new elements in HTML5: section, article, header…
- Adding videos, audio to a web page