

---

---

# HTML/HTML5 for Front End Developers

— [phuong.vu@htklabs.com](mailto:phuong.vu@htklabs.com) —

---

---

# Agenda

- HTML Structure
- HTML5 Video, Audio, Local Storage
- HTML5 Canvas
- HTML5 Offline
- HTML5 History
- HTML Geolocation
- Q&A

# HTML Structure

- HTML is the markup that contains all the actual stuff that a web page has.
- Gives content structure and meaning by defining that content as, for example, headings, paragraphs, or images
- HTML will always represent content

# HTML Structure - Common HTML Terms

- **Tags:** The use of less-than and greater-than angle brackets surrounding an element creates what is known as a tag.
- **Elements:** are written with a **start tag**, with an **end tag**, with the content in between
- **Attributes:** Attributes provide additional information about HTML elements.

# HTML Structure - Coding Conventions

- **Use Correct Document Type:** Always declare the document type as the first line in the document.

- Ex: `<!DOCTYPE html>`

- **Use Lower Case Element Names, Attribute Name:**

```
<section>
  <p>This is very bad.</p>
</SECTION>
<SECTION>
  <p>This is bad.</p>
</SECTION>
<section>
  <p>This is good.</p>
</section>
<div CLASS="menu"> #This is looking bad
<div class="menu"> #This is looking good
```

# HTML Structure - Coding Conventions (con't)

- **Close All HTML Elements:**

```
<p>This is looking bad.  
<p>This is looking good.</p>
```

- **Close Empty HTML Elements:**

```
<meta charset="utf-8"> #This is looking bad  
<meta charset="utf-8" /> # This is looking good
```

- **Quote Attribute Values:**

```
<table class=table striped> #This is bad  
<table class="table striped"> #This is good
```

- **Always use the *alt* attribute with images:**

- **Spaces and Equal Signs:**






```
<link rel = "stylesheet" href = "styles.css"> # This is looking bad  
<link rel="stylesheet" href="styles.css"> # This is looking good
```

# HTML Structure - Coding Conventions (con't)

- Do not add blank lines without a reason. Add blank lines to separate large or logical code blocks.
- Add 2 spaces of indentation. Do not use TAB.
- Do not omit the **<html>** and **<body>** tags.
- Short comments should be written on **one line**, with **a space after** `<!--` and **a space before** `-->`
- Long comments, spanning many lines, should be written with `<!--` and `-->` on separate lines.

# HTML5 Audio

- Before HTML5, there was no standard for playing audio files on a web page.
- Before HTML5, audio files could only be played with a plug-in (like flash).
- The HTML5 **<audio>** element specifies a standard way to embed audio in a web page.
- Browser Support:

Element					
<audio>	4.0	9.0	3.5	4.0	10.5








# HTML5 Audio (con't)

- How it works:
  - The **controls** attribute adds audio controls, like play, pause, and volume.
  - Text between the **<audio>** and **</audio>** tags will display in browsers that do not support the **<audio>** element.
  - Multiple **<source>** elements can link to different audio files. The browser will use the first recognized format.
- File format, Media type and Browser support:

File Format	Media Type	Browser	MP3	Wav	Ogg
MP3	audio/mpeg	Internet Explorer	YES	NO	NO
Ogg	audio/ogg	Chrome	YES	YES	YES
Wav	audio/wav	Firefox	YES	YES	YES
		Safari	YES	YES	NO
		Opera	YES	YES	YES

# HTML5 Video

- Before HTML5, there was no standard for showing videos on a web page.
- Before HTML5, videos could only be played with a plug-in (like flash).
- The HTML5 **<video>** element specifies a standard way to embed a video in a web page.
- Browser Support:

Element					
<video>	4.0	9.0	3.5	4.0	10.5

# HTML5 Video (con't)

- How it works:
  - The **controls** attribute adds video controls, like play, pause, and volume.
  - It is a good idea to always include **width** and **height** attributes. If height and width are not set, the browser does not know the size of the video. The effect will be that the page will change (or flicker) while the video loads.
  - Text between the **<video>** and **</video>** tags will only display in browsers that do not support the **<video>** element.
  - Multiple **<source>** elements can link to different video files. The browser will use the first recognized format.
- File file, Media type, Browser support

		Browser	MP4	WebM	Ogg
File Format	Media Type	Internet Explorer	YES	NO	NO
MP4	video/mp4	Chrome	YES	YES	YES
WebM	video/webm	Firefox	YES	YES	YES
		Safari	YES	NO	NO
Ogg	video/ogg	Opera	YES (from Opera 25)	YES	YES

# HTML5 LocalStorage

- Web applications can store data locally within the user's browser.
- Is more secure, and large amounts of data can be stored locally, without affecting website performance.
- The storage limit is far larger (at least 5MB) and information is never transferred to the server.
- Is per origin (per domain and protocol). All pages, from one origin, can store and access the same data.
- Has no expiration time. Data stored in sessionStorage gets cleared when the browsing session ends.

# HTML5 Canvas

- Is used to draw graphics on a web page, on the fly, via scripting (usually JavaScript). All drawing on the HTML canvas must be done with JavaScript.
- Is only a container for graphics. You must use a script to actually draw the graphics.
- Is a rectangular area on an HTML page. By default, a canvas has no border and no content.
- The markup looks like this: `<canvas id="myCanvas" width="200" height="100"></canvas>`
- Always specify an id attribute (to be referred to in a script), and a width and height attribute to define the size of the canvas.
- Fallback content inside it will display in older browsers not supporting it.

# HTML5 Canvas can

- **Draw Text:** draw colorful text, with or without animation.
- **Draw Graphics:** has great features for graphical data presentation with an imagery of graphs and charts.
- **Be Animated:** Object can move. Everything is possible: from simple bouncing balls to complex animations.
- **Be Interactive:** can respond to JavaScript events, can respond to any user action (key clicks, mouse clicks, button clicks, finger movement).
- **Be Used in Games:** offer a lot of possibilities for HTML gaming applications.

# HTML5 Offline

- **HTML5 Application Cache:** With application cache it is easy to make an offline version of a web application, by creating a cache manifest file.
- **Given three advantages:**
  - **Offline browsing** - users can use the application when they're offline
  - **Speed** - cached resources load faster
  - **Reduced server load** - the browser will only download updated/changed resources from the server.
- To enable application cache, include the manifest attribute in the document's <html> tag: `<html manifest="demo.appcache">`

# HTML5 Offline - Manifest File

- The manifest file is a simple text file, which tells the browser what to cache (and what to never cache).
- The recommended file extension for manifest files is: **".appcache"**.
- The manifest file has three sections:
  - **CACHE MANIFEST** - Files listed under this header will be cached after they are downloaded for the first time
  - **NETWORK** - Files listed under this header require a connection to the server, and will never be cached
  - **FALLBACK** - Files listed under this header specifies fallback pages if a page is inaccessible
- Lines starting with a "#" are comment lines, but can also serve another purpose.



# HTML5 Offline - Update Caching

- Once an application is cached, it remains cached until one of the following happens:
  - The user clears the browser's cache
  - The manifest file is modified (see tip below)
  - The application cache is programmatically updated
- Updating the date and version in a comment line is one way to make the browser re-cache your files.
- Once a file is cached, the browser will continue to show the cached version, even if you change the file on the server. To ensure the browser updates the cache, you need to change the manifest file.
- Browsers may have different size limits for cached data (some browsers have a 5MB limit per site)

# HTML5 History

- History Object:
  - Contains the URLs visited by the user (within a browser window).
  - Is part of the window object and is accessed through the window.history property.
- Object Property:
  - length: Returns the number of URLs in the history list
- Object Method:
  - back(): Loads the previous URL in the history list
  - forward(): Loads the next URL in the history list
  - go(): Loads a specific URL from the history list

# HTML5 Geolocation

- Is used to get the geographical position of a user.
- Since this can compromise privacy, the position is not available unless the user approves it.
- Geolocation is also very useful for location-specific information, like:
  - Up-to-date local information
  - Showing Points-of-interest near the user
  - Turn-by-turn navigation (GPS)

# Assignment - Make static web follow some rule

- Only using HTML, not CSS.
- All pages have to have:
  - The title.
  - Button to next page and previous page
  - Button to go to direct to Home page
- Page 1: Home page has header, footer, and all links in the project.
- Page 2: Display all heading that HTML support. In each heading, display the info of the heading about the font, font-size,.. (Using debug tool bar to get those info).

# Assignment (con't)

- Page 3: Display the <p> elements. In each element, display format of some character as: Bold text, Important text, Italic text, Emphasized text, Marked text, Small text, Deleted text, Inserted text, Subscripts, Superscripts, Short Quotations, Long Quotations,
- Page 4: Display some list with some type:
  - Type ●
  - Type ○
  - Type ■
  - Type 1, 2,..
  - Type A, B,..
  - Type a, b,...
  - Type I, II,...
  - Type i,ii,...

# Assignment (con't)

- Page 5: Create 1 form have ability submit file with some element as below:
  - Input short text
  - Input password
  - Input long text that has default width and height
  - Checkbox that select only one
  - Checkbox that select one more
  - Select list
  - Button submit
- Page 6: Has 2 audio file. Default play second file.
- Page 7: Has 2 video file. Default play first file.

# Assignment (con't)

- Page 8: Anything that applies the canvas for drawing.
- Page 9: Display the 3 image with the second image can not be located.
- Page 10: Make the table as same as:

Name	Week 50				
	Mon	Tue	Wed	Thu	Fri
Person 1	8	6	4	5	7
Person 2	8	6	4	5	7
Total	16	12	8	10	14

- Make the page 3, 4 can display in offline.
- Try to make the page 6, 7 can play in offline mode and answer the question. Does the playing audio or video play in offline?



# Q&A





---

---

**Thanks for your listening!**

---

---