WESTCLIFF
UNIVERSITY
Educate. Inspire. Empower.

Week 1: HTML Review

Agenda

- Review of This Week's HTML Concepts
  - Basics of HTML, CSS, and JavaScript
  - HTML building blocks
  - Inline vs. Block elements
  - ID vs Class
  - HTML comments
  - Attributes
  - Hypertext and Anchor Links
  - Media Elements
  - HTML Tables
  - HTML Forms & Form Elements
  - Lab Assignment
  - Homework

Basics of HTML, CSS, and JavaScript and how they are used

Basics of HTML, CSS, and JavaScript and how they are used

An overview:

- **HTML** provides the *basic structure* of sites (known as structural or markup language), which is enhanced and modified by other technologies like CSS and JavaScript.
- **CSS** is used to control *presentation, formatting, and layout* – hence known as presentation language.
- **JavaScript** is used to control the *behavior* of different elements and is known as behavioral language.

HTML building blocks

HTML building blocks

Main HTML structural code blocks

```
<!DOCTYPE html>
<html>
<head>
<title>
<body>
```

HTML building blocks

HTML5 Elements

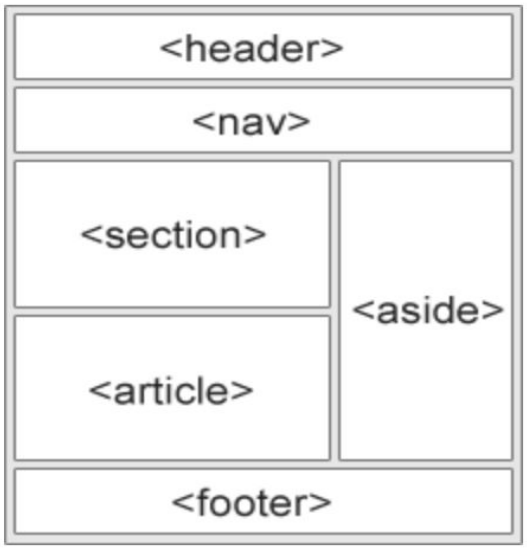
- These are some of the common HTML5 building block elements

```
<header> - Defines a header for a document or section
<nav> - Defines a set of navigation links
<section> - Defines a section in a document
<article> - Defines an independent, self-contained content
<aside> - Defines content aside from content (like a sidebar)
<footer> - Defines a footer for a document or a section
<details> - Defines additional details user open/close on demand
<summary> - Defines a heading for the <details> element
```

HTML building blocks

HTML5 Elements

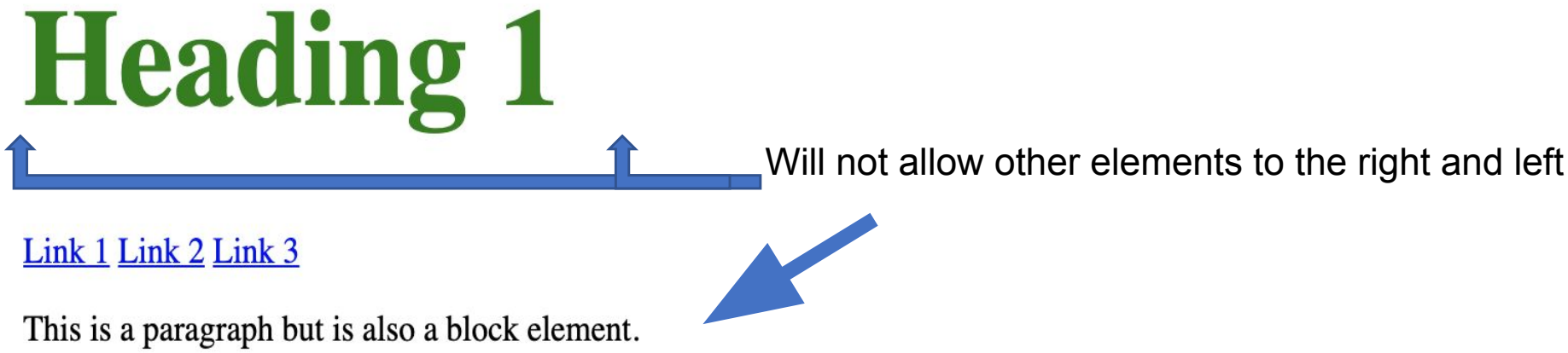
- Example of Page Layout using HTML5 elements



Inline vs. Block elements

Inline vs. Block elements

- **Block elements** are those that take up the full width available on a web page, effectively *blocking out* any other elements from sitting next to it on the left or right. Examples of <h1> and <p> elements:



Inline vs. Block elements

- **Block Elements**

<address>	<article>	<aside>	<blockquote>	<canvas>	<dd>
	<dl>	<dt>	<fieldset>	<figcaption>	<figure>
<footer>	<form>	<h1>-<h6>	<header>	<hr>	
<main>	<nav>	<noscript>		<p>	<pre>
<section>	<table>	<tfoot>		<video>	

Inline vs. Block elements

- **Inline Elements**

<a>	<abbr>	<acronym>	<b>	<bdo>	<big>
 	<button>	<cite>	<code>	<dfn>	
<i>	<img>	<input>	<kbd>	<label>	<map>
<object>	<output>	<q>	<samp>	<script>	<select>
<small>			<sub>	<sup>	<textarea>
<time>	<tt>	<var>			

ID vs Class

ID vs Class

- What is ID and Class?
  - Block and inline elements are often given additional identity so that other languages like CSS and JavaScript can communicate with that element via ID and class.
  - They are essentially just another piece of code added to the element, more specifically the open tag and as an attribute.
- ID and Class Rules
  - ID names are unique. An ID name given on a page cannot be repeated on the same page. They can however be used again on other pages.
  - Unlike ID names, class names can be repeated – on the same page and other pages.
  - When creating ID or class names, do not use symbols other than underscore (_)and dash (-), do not begin names with a number and no spaces between 2 or more words.

ID vs Class

- When to use ID vs Class?
  - Use ID on main block elements ex: body, header, nav, main, footer, div, where their names are unique (not repeated elsewhere on the page).
  - Use class on smaller blocks and inline elements ex: div, ul, li, img, span, where these elements are typically not unique (likely repeat elsewhere on the page).
- How they are used:
  - <body id="aboutUs">
  -
  - <img class="gallery1" ……>
  -

HTML comments

HTML comments

- Here's an example of comments used on an HTML document:

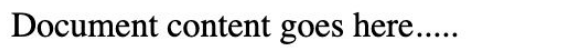
```
<!DOCTYPE html>
<html>

   <head>  <!-- Document Header Starts -->
      <title>This is document title</title>
   </head> <!-- Document Header Ends -->

   <body>
      <h1>HTML Comments</h1>
      <p>Document content goes here.....</p>
   </body>

</html>
```

What it looks like on the browser:



Document content goes here.....

HTML comments

- Another common use of comments is to deactivate codes – instead of erasing unused codes, deactivate it so in the event if you decide to use it, saves you time to rewrite it. Here's an example:

```
<!DOCTYPE html>
<html>

   <head>  <!-- Document Header Starts -->
      <title>This is document title</title>
   </head> <!-- Document Header Ends -->

   <body>
      <!-- <h1>HTML Comments</h1> -->
      <h2>How Comments Are Used</h2>
      <p>Document content goes here.....</p>
   </body>

</html>
```

What it looks like on the browser:

How Comments Are Used

Document content goes here.....

Attributes

Attributes

What are attributes?

- And then there are some that are recommended though not a must. For instance, the *width* and *height* attributes are recommended for the <img> element. This informs the browser in advance of the image dimensions so the loading time can be shortened.
- Other recommended attributes: *id* and *name*.
  - **Examples**:

     … *id* attribute plays an important role for JavaScript and CSS.

    <input name="gender"> … *name* attribute is important for certain form elements.

Attributes

What are attributes?

- Attributes define additional characteristics or properties of the element such as width and height of an image. Attributes are always specified in the start tag (or opening tag) and usually consist of name/value pairs like name="value". Attribute values should always be enclosed in quotation marks.
- Most attributes are optional, but some are required for certain elements. For instance, the <img> element must contain a *src* and *alt* attribute. Example:

```
<img src="images/flower.jpg" alt="Nice Flower Image">
```

Attributes

Common HTML attributes

- **href** – this is used inside the <a> element and specifies the url of a file to open.
- **src** – the path of the file name of the image to be displayed
- **alt** – to provide a short description of the image, which is useful for vision impaired users who use text-to-speech synthesizers.
- **title** - adds information related to the element. Hovering on it will cause a tooltip with the title you created to appear.
- **width & height** – specify or change the element's dimensions.
- **id** – used to identify HTML elements for styling purposes and also for JavaScript calling.
- **class** – used to identify HTML elements for styling purposes and could also be used for JavaScript calling.
- **style** - specifies an inline style for an element.

Hypertext and Anchor Links

Hypertext and Anchor Links

- Hypertext links allows us to create link between files with user clicks. The object that is click on are usually texts which is markup with this special element <a>. Example: <a>Click Here
- That still won't work yet, until you specify this attribute: href. This attribute tells HTML which file to open next. Example: Click Here
- Use the target attribute and _blank value to open the file in a new browser tab window – example: Click Here
- Note that all hypertext links by default are underlined with blue font color.

Hypertext and Anchor Links

- Anchor links allows us to create links between different areas within the file. This is refer to as shortcut links. Some examples: return to top of page, jump to an alphabetically categorized page content.
- Instead of specifying a file to open, you specify which *id* name to jump to. So therefore, make sure you have ids created in the areas or it won't find it.
- Example:
  - Step 1: create the link: Return to Top
  - Step 2: add an id attribute to the element that you want it to jump to.
- Note that you use the **#** symbol when linking to an id.

Media Elements

Media Elements

- Besides texts, site visitors are expecting to see images and graphics on the website. Images and graphics help enhance the story telling of the subject matter. Graphics can also enhance the visual appeal of the layout.
- To display an image or graphic, use <img> element. Acceptable popular formats: png, jpg, gif and svg.
- How it is used:

```
<img src="images/sunset.jpg" alt="Sunset Photo" width="400" height="275">
```

Required attributes: src, alt
Recommended attributes: width, height

Media Elements

- Apart from images, HTML also has elements specifically for embedding medias such as video and audio for playback on a web page.
- For videos, use <video></video> to display video content. Current recommended formats are: MP4, WebM and Ogg.
- How it is used:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
</video>
```

Media Elements

- For audios, use <audio></audio> to playback sound or music. Current recommended formats are: MP3, Ogg and Wav.
- How it is used:

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
</audio>
```

Note: on both video and audio elements, the controls attribute is to turn on the controller (play, stop, pause, forward, backward buttons and volume slider). Other attributes: autoplay, loop and muted (initial setting when it first loads)

HTML Tables

HTML Tables

- What is an HTML table?
  - A table is a grid structure made up of rows and columns to display tabular data. A table allows you to quickly and easily look up values that indicate the the connection between different types of data. For example the calories of different types of food, or a person's age like below:

Person	Age
Chris	38
Dennis	45
Sarah	29
Karen	47

HTML Tables

- Or the timetable for a local swimming pool:

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Public Swim 06:30 - 10:30	Public Swim 06:30 - 09:00	Public Swim 06:30 - 09:00	Public Swim 06:30 - 11:15	Public Swim 06:30 - 09:00	Lane Swim 08:00 - 09:00	Lane Swim 08:00 - 09:00
Aquacise 10:30 - 11:15	Aqua Jog 09:15 - 10:00	Education Swimming Lessons 09:00 - 12:00	Aquacise 11:15 - 12:00	Education Swimming Lessons 09:00 - 12:00	Oldham Active Kids Swimming Lessons 09:00 - 13:00	Public Swim 09:00 - 11:00
Lane Swim 11:30 - 13:00	Parent & Baby Class 09:30 - 10:15	Lane Swim 12:00 - 13:00	Lane Swim 12:00 - 13:00	Lane Swim 12:00 - 13:00	Parent and Baby 12:00 - 12:45	Aquacise 11:00 - 11:45
Education Swimming Lessons	Public Swim 10:00 - 11:45	Public Swim 13:00 - 16:00	Education Swimming Lessons	Oldham Active Kids Swimming	Public Swim 13:00 - 17:00	Public Swim 11:45 - 13:00

HTML Tables

- Always begin your table code structure with: <table></table>
- Tables contain rows and columns:
  - Always create rows first. Use <tr></tr> to create a row. If you need 3 rows, then you need to create 3 sets of <tr></tr> .
  - Each row is divided into columns. Use <td ></td> to create column or data cells. If you need 2 columns, then you need to create 2 sets of <td ></td> within that row.
- How many tables can I create on a single html document? Answer: there's no limit. However, you want to keep it to a minimum unless it is necessary. *Reason: the more tables or more complex a table can affect the download speed of a web page.*

HTML Tables

- To ensure your table data makes more sense to the viewer, it is recommended that you create table headings for each column.
- Additionally, you may also want to use captions to create a title for the table. This will be helpful to viewers to know what the table is about.
- Use <th> for headings and <caption> for title of the table:

```
<table>
    <caption>Title of Table</caption>
    <thead>
        <tr>
            <th>Heading Col 1</th>
            <th>Heading Col 2</th>
        </tr>
    </thead>
```

HTML Tables

- Example of a simple table with all the table elements we'd discussed:

```
<table>
    <caption>Calories Comparison of Popular Burgers</caption>
    <thead>
        <tr>
            <th>Big Mac</th>
            <th>Whopper</th>
            <th>Dave's Double</th>
            <th>Whataburger</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>563</td>
            <td>677</td>
            <td>810</td>
            <td>590</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td>MacD</td>
            <td>Burger King</td>
            <td>Wendys</td>
            <td>Whataburger</td>
        </tr>
    </tfoot>
</table>
```

What it looks like on the browser with the border turned on:

Calories Comparison of Popular Burgers			
Big Mac	**Whopper**	**Dave's Double**	**Whataburger**
563	677	810	590
MacD	Burger King	Wendys	Whataburger

Note:
<caption> will auto span across columns and centered align
<th> will auto bold texts

HTML Tables

Additional notes about tables:

- By default - outlines (borders) are not visible.
- By default - spacing between all table data cells.
- By default - all content are left and vertically centered aligned in each data cell.
- By default - no formatting or styling on the table and all of its cells' content.
- By default - the width and height of the table and its data cells are determined by the content in each cell.
- To turn on outlines/borders, remove unwanted spaces, style the table and data cells, control the width and height - use CSS, which is the recommended method.

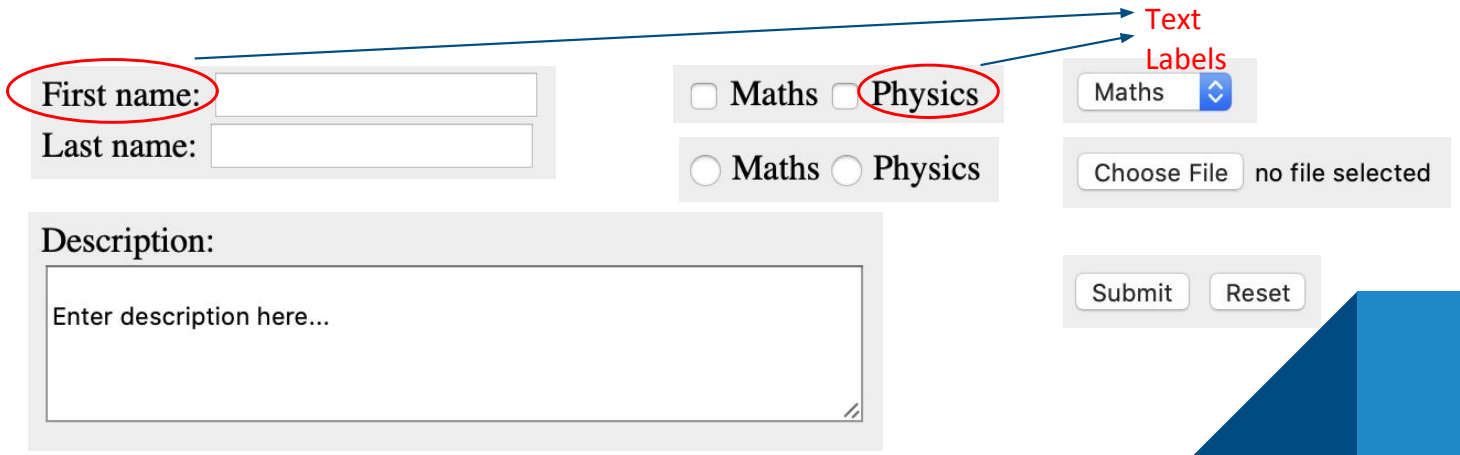
HTML Forms & Form Elements

HTML Forms

- What is HTML forms?
  - You need HTML forms on your website or application when you want to collect data from the site visitor. For example, on a user registration form, you will likely collect information such as name, email address, credit card information and so on.
  - A form will take input from the site visitor and then post it to a back-end application via JavaScript, CGI, ASP Script or PHP script, etc. The back-end application will perform the required processing on the collected data based on defined business logic inside the application.

HTML Forms

- What is in a HTML form?
  - A form typically contains text labels, boxes, menus and buttons– called form elements.
  - Some common elements: text fields, textarea fields, drop-down menus, radio buttons, checkboxes, file uploads, submit and reset buttons. Below are some examples of what they look like:



HTML Forms

- Example of a common html form:

Registration form

User personal information

Enter your full name

Enter your email

Enter your password

confirm your password

Enter your gender

- Male
- Female
- others

Enter your Address:

sign-up

Note:
This form has been formatted.

HTML Forms

- A look at the syntaxes of a html form:
  - All forms should begin with this element: <form></form>.
  - Within the start tag <form>, you need these attributes to direct the form to be processed when it is submitted:
    - action – the URL that will process the form data
    - method – can be get or post

      example: <form action="/signup" method="post">
  - Within the <form></form> tag will contain form elements that you want to collect data from the user.
  - The next slide lists the syntaxes of the main elements.

HTML Forms

- The main syntax for these elements - some as discussed previously:

Tag	Description
<form>	It defines an HTML form to enter inputs by the used side.
<input>	It defines an input control.
<textarea>	It defines a multi-line input control.
<label>	It defines a label for an input element.
<fieldset>	It groups the related element in a form.
<legend>	It defines a caption for a <fieldset> element.
<select>	It defines a drop-down list.
<optgroup>	It defines a group of related options in a drop-down list.
<option>	It defines an option in a drop-down list.
<button>	It defines a clickable button.

HTML Forms

- The <input> elements have different types within its group:
  - <input type="text">
  - <input type="email">
  - <input type="password">
  - <input type="radio">
  - <input type="checkbox">
  - <input type="submit">
  - <input type="reset">

HTML Forms

- An example of a markup HTML form:

```
<form action="/signup" method="post">
     <fieldset>
        <legend>User personal information</legend>
        <label>Enter your full name</label><br>
        <input type="text" name="name"><br>
         <label>Enter your email</label><br>
         <input type="email" name="email"><br>
         <label>Enter your phone</label><br>
         <input type="text" name="phone"><br>
         <br><label>Select your gender</label><br>
         <input type="radio" name="gender" value="male"/>Male
         <input type="radio" name="gender" value="female"/>Female <br/>
         <br><label>How should we contact you?</label><br>
         <select>
            <option>Email</option>
            <option>Phone</option>
         </select><br>
         <br>Tell us the reason of your visit today:<br>
         <textarea></textarea><br>
         <input type="submit" value="sign-up">
     </fieldset>
  </form>
```

User personal information

Enter your full name

Enter your email

Enter your phone

Select your gender

Male Female

How should we contact you?

Email

Tell us the reason of your visit today:

sign-up

HTML Forms

- Additional notes about HTML forms:
  - By default, form elements are inline elements. You can use
 or wrap with block elements such as or use CSS - which is the better option to make them appear as blocks.
  - By default, form elements such as input text and email boxes are 20 characters wide. You can specify <input type="text" size="40"> to change the width or, preferably, use CSS.
  - Unlike checkboxes, user selects only one of the radio buttons, not multiple. For this to work, make sure the name attribute value is the same:

```
<input type="radio" name="gender" value="male">
<input type="radio" name="gender" value="female">
```