

BTI220 - Internet Architecture and Development

**Week 5: More on HTML &
Introduction to CSS**

Agenda

➤ More on HTML

- table, image
- Multimedia - audio and video
- figure elements

➤ Introduction to CSS

- Syntax / Structure
- Selectors
- Web Colors, Units
- CSS Properties - background

HTML Table

- An HTML table is used for presenting tabular data in a grid-like fashion.
- A table is not (/no longer) for the purposes of laying out a web page, or the sections within a web page.
 - unless you have to do so.
- Basic table tags

Tag	Description
<table>	Specifies a table
<caption>	Specifies a table caption
<tr>	Specifies a table row
<th>	Specifies a table heading
<td>	Specifies a table cell / detail

Table Structure

➤ Table structure

- in a table, each piece of information is displayed in a **cell** (`<td>`).
- The cells in a line across the page make up a **row** (`<th>` or `<tr>`).
- The cells in a line down the page make up a **column**.

➤ Example

This is the table caption

Column Heading 1	Column Heading 2
This is row 1 cell 1	This is row 1 cell 2
This is row 2 cell 1	This is row 2 cell 2

[tags-tables.html](#)

Table Attributes - border

- `<table border='value'>`: value (integer) is the thickness of the table border in pixels.
- This attribute has been deprecated in HTML5, so use it only when necessary.
 - Use CSS instead.
- By default, a table has no borders (`border="0"`)

Table Attributes – rowspan

➤ rowspan

```
<th colspan='value'>  
<td colspan='value'>
```

- Value: non-negative integer value that indicates on how many rows does the cell extend.
- By default value ='1'.
- If value ='0', it extends until the end of the table section (<thead>, <tbody> or <tfoot>).

Table Attributes – colspan

➤ colspan

```
<th colspan='value'>  
<td colspan='value'>
```

- Value: non-negative integer value that indicates on how **many rows** does the cell extend.
- By default value ='1'.
- If value ='0', it extends until the end of the table section (<colgroup>).

[tags-tables-col-rowspan.html](#)

Table with `thead`, `tbody` and `tfoot` tags

- The `<thead>`, `<tbody>` and `<tfoot>` elements are used to specify each part/section of a table: table header, body and table footer.
 - **`<thead>`** - table head tags - group the first one or more rows of a table for formatting
 - **`<tbody>`** - table body tags - group the middle rows of a table for formatting
 - **`<tfoot>`** - table foot tags - group the last one or more rows of a table for formatting

[tags-tables-sections.html](#)

HTML Image

- The HTML ** tags** defines an image in a HTML page.
 - e.g.
``
 - The `` tag has 2 **required attributes**:
 - **src**: url of the image
 - **alt**: alternate text for the image
 - The **width** and **height** are supported by HTML5, but suggest to use CSS to define the size:
 - e.g.
``
- image.html

Image Link and Image Map

➤ Image link

```
<a href="http://www.senecacollege.ca/">  
    
</a>
```

➤ Image map

- Define clickable areas on an image.
- Click on the clickable areas will activate different hyperlinks.
- `<map>` and `<area>` tags are used to define an image map.

Image Map Example

```


<map name="tutorials">
  <area shape="poly"
        coords="74,0,113,29,98,72,52,72,38,27"
        href="usemap.png" alt="Image Demo"/>
  <area shape="rect" coords="22,83,126,125"
        href="http://www.w3schools.com/html/" alt="W3C" />
  <area shape="circle" coords="73,168,32"
        href="https://scs.senecac.on.ca/" alt="ICT" />
</map>
```

imageMaps.html

HTML5 - <figure> and <figcaption> tags

- The HTML <figure> tag specifies self-contained content, frequently with a caption (<figcaption>), and is typically referenced as a single unit.
- e.g.

```
<div class="picture">
  <figure>
    <figcaption>
      This is a figure caption
    </figcaption>
    
  </figure>
</div>
```

□ [html5figure-1.html](#)

[html5figure-2.html](#)

<audio> and <video> tags

- About multimedia
 - On the web, multimedia comes in many different formats.
 - It can be almost anything you can hear or see. e.g.
 - ▶ Pictures, music, sound, videos, records, films, animations
- HTML5 introduced a built-in multimedia support via the <audio> and <video> elements, offering the **standard** and easy way to embed media into HTML documents.
 - In the earlier time, most audio/video files are played through a plug-in (like flash).
 - Supported by IE 9, Firefox, Opera, Chrome, and Safari

HTML5 <audio> Tags

➤ Example

```
<figure>
  <audio controls>
    <source src="Track03.mp3" type="audio/mpeg" />
    <source src="Track03.ogg" type="audio/ogg" />
    Your browser does not support the audio tag used.
  </audio>
  <figcaption>Audio Caption</figcaption>
</figure>
```

- Multiple <source> elements can link to different audio files.
 - The browser will use the first **recognized** format.
- [html5 audio.html](#)

Attributes of <audio> Element

- **controls**
 - Displays the standard HTML5 controls for the audio on the web page.
- **src**
 - It's optional. You may instead use the <source> element with src attribute.
- **autoplay**
- **loop**
- **preload**
- **buffered, muted, played**

The <source> element

- The source element is used to specify multiple media resources for audio and video elements in HTML5. It is an empty element.
- It is commonly used to serve the same media in multiple formats supported by different browsers.
- Attributes
 - src, type, media

HTML5 video Tags

➤ e.g.

```
<figure>
  <video width="320" height="240" controls>
    <source src="movie.mp4"      type="video/mp4"/>
    <source src="movie.ogg"      type="video/ogg" />
    <source src="movie.webm"    type="video/webm" />
    Your browser does not support the video tag / type
  </video>
  <figcaption>Video Caption</figcaption>
</figure>
```

- The **width** and **height** specify the size of the video's display area.
- The **autoplay** and **loop** are additional attributes that can be used with the video tag.
- [html5_video.html](#)

Attributes of <video> Element

- The <video> Element shares many attributes with the <audio> element but has its own attributes:
 - Height, width
 - poster

About Audio/Video Formats

► Audio file formats

- mp3 audio format
- ogg audio/video format

► Video file formats

- mp4 video format
- webm audio/video format

Introduction to CSS

- **CSS: Cascading Style Sheet**
- **HTML:** specify structure and content of a web page
- **CSS:** for presentation, how to display HTML Elements
- CSS are really text files, or text in an HTML file , that allow the use of specific styles, attributes, and positioning of HTML objects.

Introduction to CSS

- CSS defines how HTML elements are to be displayed.
- Separates the layout from the content. Formatting could be removed from HTML doc, and stored in a separate CSS file.
- Control the style and layout of multiple web pages all at once.
- External Style Sheets in CSS files can save a lot of work.
- All browsers support CSS today.

Advantages

- Define the look of your pages in one place, and apply it throughout the whole site.
- Easily change the look of your pages even after they're created. Change style once.
- Pages will be loaded faster, since they aren't filled with tags that define the look. The style definitions are kept in a single CSS document that is only loaded once when a visitor enters your site.

Where to place it

- CSS can be implemented/added in an html document in three different ways:
 - **Inline**
 - **Internal** Embedded
 - **External**
- In addition, each browser has its Browser default CSS settings

Implementing CSS in HTML

1. Browser default

Rules are set by the browser for the various tags

2. Inline

CSS rule is coded / applied on a single element

```
<p style="font-size:16px">This is an inline usage example</p>
```

3. Internal Embedded

CSS rules are included in the head part of an html document and can be used anywhere in the html document.

```
<style type='text/css' media='screen'>
  p { font-size:16px; }
</style>
```

Implementing CSS in HTML

4. External

CSS rules are in a separate CSS file referenced from any html document **using the html <link...> tag**

```
<link rel='stylesheet' href='company.css' type='text/css' media='screen' />
<link rel='stylesheet' href='mystyle.css' type='text/css' media='screen' />
```

or the @import CSS feature

```
<style type='text/css' media='screen'>
  @import url(http://www...../company.css);
  @import url(/project-x/project.css);
</style>
```

CSS – Multiple Style Sheets

➤ The priority order for CSS sources

- If some properties have been set for the same selector in different style sheets, the priority order will be:
(from highest to lowest)
Inline, Internal Embedded, External, Browser default

e.g., external CSS:

```
h3
{
color:red;
text-align:left;
font-size:8pt;
}
```

e.g., Internal CSS:

```
h3
{
text-align:right;
font-size:20pt;
}
```

Final results for
h3:

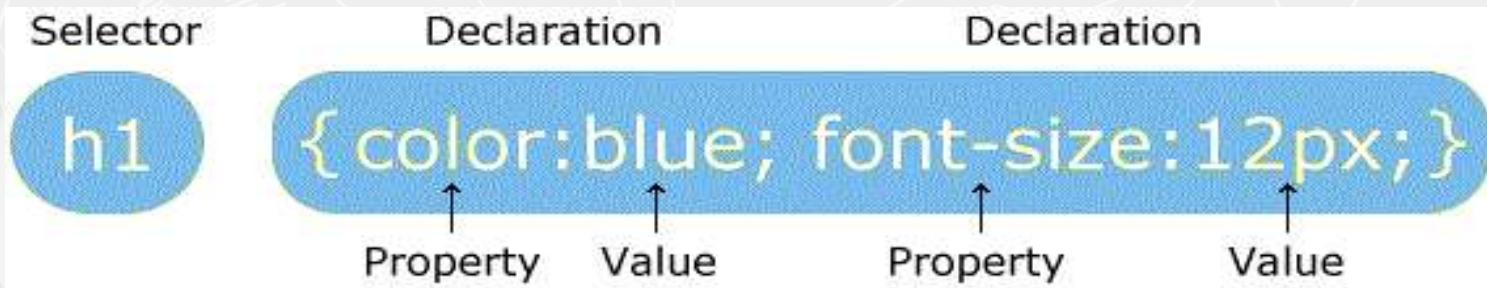
```
color:red;
text-align:right;
font-size:20pt;
```

CSS Cross-browser Consistency

- Browser Defaults
- CSS reset: reset.css
- CSS Normalization: normalize.css
- consistency-default.html
- consistency-reset.html
- consistency-normalize.html

CSS Syntax / Structure

- The CSS syntax is made up of three parts
 - a **selector**
 - a **property**
 - a **value**
- The syntax of a **CSS Rule / Entry:**



CSS Syntax / Structure

- The selector properties and values are always enclosed between { and }
- The property is separated from its value or values by a colon :
- The values, if more than one are separated by a comma ,
- A CSS declaration always ends with a semicolon ;

CSS Syntax / Structure

- The selector is normally the HTML element you want to style.
- Each declaration consists of a property and a value.
- The property is the **style attribute** you want to change. Each property has a value.

CSS Example

```
p { color: red; text-align: center; text-decoration: underline; }
```

- To make the CSS more readable, you can put one declaration on each line, like this:

```
/* This is a CSS comment */  
p  
{  
    background-color: yellow;  
    font-size: small;  
    font-weight: bold;  
}
```

- These will affect all paragraphs in web page that the css rules are applied.
- e.g. css internal inline.html

Basic CSS Selectors

- Selectors:
 - HTML tag Selectors
 - Class Selectors
 - Id Selectors
 - Contextual Selectors
 - Grouping Selectors
 - ...

Tag Selectors

- HTML tag/type Selectors
 - Any html tag is a possible CSS selector. The selector is simply the tag that is linked to a particular style. For example,
`p { text-indent: 20px; }`
 - the selector in above example is the p tag

Class Selectors

- A simple selector can have different classes allowing the same tag to have different styles.

- For example, you may wish to display different paragraph's in different font sizes

```
p.type1 { text-indent: 16px; }
```

```
p.type2 { text-indent: 20px; }
```

- In the above example the class selector can only be used with the tag it is associated with - in this case the **p** tag.

```
<p class="type1" >.....</p>
```

```
<p class="type2" >.....</p>
```

Class Selectors

- Classes may also be declared without an associated tag.

```
.note { font-weight: bold; }
```

- In this case, the note class may be used with any tag.

```
<p class="note" >.....</p>
```

```
<h3 class="note" >.....</h3>
```

```
<p>.....<span class="note" >....</span>.....</p>
```

- A class selector can be used as many times as you wish within an HTML document.

Id Selectors

- Id selectors can only be used once in an html document and are individually assigned for a specific purpose. The Id selector type should only be used sparingly because of its limitations.
- An Id selector is assigned by using the indicator "#" to precede a name. For example,

```
#xyz656 { text-indent: 1em; }
```

```
<p id="xyz656" >.....</p>
```

Or:

```
<p>.....<span id="zyx565" >.....</span>.....</p>
```

- The value of Id attribute should be unique in a HTML file.

Contextual Selectors

- Contextual selectors are used to indicate the context of a selector.
- The context of a selector is determined by what its parent element is. In other words, what the **element is nested** within or what precedes it in the document.
 - For example, if you want unordered lists that are nested under ordered lists to have a font size of 16px, then you would use
- **ol ul { font-size: 16px; }**
- This can be read as "for any unordered list that is nested within an ordered list" - change the font size to 16px.

Grouping Selectors

- To reduce the size of the style sheets, one can group selectors in comma-separated list

```
h1, h2, h3 { font-family: Serif; color: blue;}
```

- Example: css selectors.html

CSS how to - div

- The <div> tag defines logical divisions, acting like a paragraph tag.
- The <div> tag is a block-level element
- The <div> tag can contain nearly any other tag
- The <div> tag cannot be inside <p> tags
- You can use the <div> tag when you want to center or position a content block on the page.

Tips of Using tag *div*

- The `<div>` tag is **not** a replacement `<p>` tag.
- The `<p>` tag is for paragraphs only,
- The `<div>` tag defines more general divisions within a document.
- Don't replace `<p>` tags with `<div>` tags.
- It's always a good idea to close your `<div>` tags as soon as you open them. Then place the contents within the element.
- If you nest your `<div>` tags, be sure that you know where your content is going (in other words, which DIV it should be part of).

CSS how to - span

- Similar to the `div` element in function with one difference,
- *span* is an inline element - do not begin a new line.
- Can only contain other inline elements.
- The *span* element can be a selector and accept *style*, *class* and *id* attributes.
- The primary difference between the `` and `<div>` tags is:
 - `` doesn't do any formatting of its own.
 - `<div>` includes a paragraph break, because it is defining a logical division in the document.
 - The `` tag simply tells the browser to apply the style rules to whatever is within the ``.

Units used in CSS

1em = 12pt = 16px = 100%

- *Ems* - **em**
- The "**em**" is a scalable unit that is used in web document media. An **em** is equal to the current font-size, for instance,
 - if the font-size of the document is 12pt, 1em is equal to 12pt.
- *Ems* are scalable in nature, so
 - 2em would equal 24pt,
 - .5em would equal 6pt, etc.
- *Ems* are becoming increasingly popular in web documents due to scalability and their mobile-device-friendly nature.
- **em** stands for "M", the letter M being the widest character in a font.

Units used in CSS

$1\text{em} = 12\text{pt} = 16\text{px} = 100\%$

- *Points* - **pt**
- *Points* are traditionally used in **print** media (anything that is to be printed on paper, etc.).
- One *point* is equal to 1/72 of an inch.
- *Points* are much like pixels, in that they are fixed-size units and cannot scale in size.

Units used in CSS

$1\text{em} = 12\text{pt} = 16\text{px} = 100\%$

- **Pixels - px**
- **Pixels** are fixed-size units that are used in screen media (i.e. to be read on the computer screen).
- One **pixel** is equal to one dot on the computer screen (the smallest division of your screen's resolution).
- Many web designers use **pixel units** in web documents in order to produce a pixel-perfect representation of their site as it is rendered in the browser.
- One problem with the **pixel unit** is that
 - it does not scale upward for visually-impaired readers
 - or downward to fit mobile devices.

Units used in CSS

$1\text{em} = 12\text{pt} = 16\text{px} = 100\%$

- *Percent - %*
- The *percent* unit is much like the "em" unit, save for a few fundamental differences.
 - First and foremost, the current font-size is equal to 100% (i.e. $12\text{pt} = 100\%$).
 - While using the percent unit, your text remains fully scalable for mobile devices and for accessibility.

□ [font-units.html](#)

Units used in CSS

➤ The other ways of measurement

- xx-small, x-small, small, medium, large, x-large, xx-large,
- smaller, larger
- thin, medium, thick

Web colors

- Primary colors are sets of colors that can be combined to make a range of colors.
 - Normally, **red, green and blue** are used as primary colors - the RGB (Red-Green-Blue) color model.
- CSS colors are specified in 3 formats:
 - Hexadecimal Value Notation
 - Hex triplet: written as 3 double digit numbers, starting with a # sign.
 - e.g. h1 { background-color: #800080; }
 - RGB Value Notation
 - the combination of Red, Green, and Blue color values (RGB).
 - e.g. P { color: rgb(128,0,128); }
 - Named colors

Color Examples

Color (Named)	Color HEX	Color RGB
Black	#000000	rgb(0,0,0)
Red	#FF0000	rgb(255,0,0)
Green	#00FF00	rgb(0,255,0)
Blue	#0000FF	rgb(0,0,255)
Yellow	#FFFF00	rgb(255,255,0)
Aqua	#00FFFF	rgb(0,255,255)
Fuchsia	#FF00FF	rgb(255,0,255)
Gray	#808080	rgb(128, 128, 128)
Silver	#C0C0C0	rgb(192,192,192)
White	#FFFFFF	rgb(255,255,255)

Web safe colors

➤ Web safe colors

- Many years ago, computers supported max 256 different colors.
- The 216 cross-browser color palette was created to ensure that all computers would display the colors correctly.
- This is not important now.

Red	Green	Blue	R	G	B	No. of colors
00 33 66	00 33 66	00 33 66	# Any red	Any green	Any blue	216
99 CC FF	99 CC FF	99 CC FF				

Web smart colors

- Web smart colors
 - With advances in technology and the prevalence of monitors displaying High colors (65,000 colors) and True colors (16million colors), the limitations of the Web-safe colors became evident.
 - The Web-smart colors was therefore used.
- 4096 Color Wheel

Red	Green	Blue	R	G	B	No. of colors
00 11 22 33	00 11 22 33	00 11 22 33	# Any red	Any green	Any blue	4096
44 55 66 77	44 55 66 77	44 55 66 77				
88 99 AA BB	88 99 AA BB	88 99 AA BB				
CC DD EE FF	CC DD EE FF	CC DD EE FF				

CSS Properties and Values

- A **property** is assigned to a selector in order to manipulate its style.
 - Examples of properties include color, margin, font and many more.

```
p { text-indent: 1em; }
```
- A property can have one or more value. **Values** must be spelled exactly as described in the CSS rules.

```
p { font-family: "Times New Roman", serif; }
```

 - **Times New Roman** and **serif** in the above are two value examples for the font-family.
 - Multiple words for any value must be in quotations
- There are large number of properties and their values. We cannot cover all of them in the courses.

background - Properties

- **Background-image:**
 - `background-image: url(image.jpg);`
where image.jpg may be a relative or absolute path
- **background-color:** can still be used, and will provide colour where the image is not displayed
- **background-position:**
 - Values: left top (default), right bottom, center center
- **background-repeat:**
 - Values: repeat (default), repeat-x, repeat-y, no-repeat
- **Shorthand property:**

`body{ background: url("../img/seneca_logo.gif") no-repeat grey right top; }`
- [Bg.html](#), [bg.css](#)

CSS3 Backgrounds

- Property “**background-size**”: specifies the size of the background image.
- Before CSS3, the background image size was determined by the actual size of the image.
- In CSS3 it is possible to **specify the size** of the background image, which allows us to **re-use background images** in different contexts.
- You can specify the size in **pixels** or in **percentages**.
- If you specify the size as a percentage, the size is **relative to the width and height of the parent element**.

CSS3 Backgrounds

- Resize a background image:

```
body {  
    background:url(seneca_logo.gif);  
    background-size: 80px 60px;  
    -moz-background-size:80px 60px; /* Firefox 3.6 */  
    background-repeat: no-repeat;  
    padding-top: 40px;  
}
```

Bg_new.html

CSS3 Background

- Stretch the background image to completely fill the **content area**:

```
p {  
  background:url(seneca_logo.gif);  
  background-size: 100% 100%;  
  -moz-background-size:100% 100%; /* Firefox 3.6 */  
  background-repeat: no-repeat;  
  padding-top: 40px;  
}
```

[Bg new 100.html](#)

CSS3 Background

- Property “**background-origin**”:
- Specifies the positioning area of the background images.
- The background image can be placed within the content-box, padding-box, or border-box area.



[Bg_new_origin.html](#)

Support Browsers

- Firefox 3.6 and earlier does not support the background-origin property, and requires the prefix `-moz-` to support the background-size property.
- Safari 4 requires the prefix `-webkit-` to support the new background properties.
- Internet Explorer 9, Firefox 4, Chrome, Safari 5 and Opera support the new background properties.

Resourceful Links

- MDN - [HTML element reference](#)
- MDN - [Articles tagged: Multimedia](#)
- CSS Reference
<http://reference.sitepoint.com/css>
- CSS Selectors
<http://reference.sitepoint.com/css/selectorref>
- CSS properties
<http://reference.sitepoint.com/css/propertyref>

Thank You!