WESTCLIFF
UNIVERSITY
Educate. Inspire. Empower.

Week 2: CSS Review

Agenda

- Inline vs. Embed vs. External Styles
- CSS Selectors: elements, id, class
- CSS Properties
- Style html tables
- Style html forms
- CSS Pseudo Class
- CSS Pseudo Elements
- In-Class Demo
- Homework

Inline vs. Embed vs. External Styles

Inline vs. Embed vs. External Styles

- There are 3 ways where you can write CSS codes:
  - Inline or Local
  - Embed or Internal
  - External

Inline vs. Embed vs. External Styles

- Inline or Local
  - Codes are embed within HTML open tags using the style attribute. Example:

    ```
    <p style="color: red; font-weight: bold">
    ```

  - Advantages:
    - Quick to write
    - Ability to overwrite global styles
  - Disadvantages:
    - Time consuming to maintain codes as they are spread throughout HTML file and files
    - Duplication of codes on same HTML file and also across multiple HTML files
    - Inflexibility for future changes
  - The least preferred way of writing CSS.

Inline vs. Embed vs. External Styles

- Embed or Internal
  - Codes are group in one specific area of the HTML file – within the <head> area using the style element. Example:

```
<head>
    <style>
        p {color: red; font-weight: bold;}
    </style>
</head>
```

  - Advantages:
    - Organized and easy to maintain codes
  - Disadvantages:
    - Duplication of codes – similar codes repeated on multiple HTML files
    - Time consuming to make code changes – need to chase down codes on multiple files

Inline vs. Embed vs. External Styles

- External
  - Codes are grouped in one document saved as a separate file. There are 2 things you need to do:
    1. A separate file containing css codes saved in css extension (example: styles.css)

    Example of what's in the styles.css file:

```
/* External Stylesheet Codes */

p {
    color: red;
    font-weight: bold;
}
```

Inline vs. Embed vs. External Styles

- External
  2. Link the css file to the HTML file(s). Example:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<p>This is a paragraph.</p>
</body>
</html>
```

  - Advantages:
    - Faster code changes – results in immediate global site updates
    - One stop code maintenance
  - Disadvantages:
    - Risk loss of stylesheet file
    - Could result in a mix use of 2 or more methods that could complicate the order of styles

CSS Selectors: elements, id, class

CSS Selectors: elements, id, class

- What is CSS Selector?
  - It's the part of a CSS rule set that selects the content on the HTML document that you want to style.
  - These are the main types of selectors commonly used to select content on a HTML document:
    - elements
    - id
    - class

CSS Selectors: elements, id, class

- Elements:
  - HTML elements that currently exist on the document
  - Example: body, h1..h6, p, a, img, ol, ul, li, div, span, header, nav, main, footer
  - How to use: h1 { font-family: arial} or p {color:red}

- id:
  - id attribute of HTML elements that currently exist on the document
  - Example: <body id="home"> or <p id="quote">
  - How to use: #home {font-family: arial} or #quote {color:red}

CSS Selectors: elements, id, class

- class:
  - class attribute of HTML elements that currently exist on the document
  - Example: <header class="logo"> or <a class="active">
  - How to use: .logo { width: 0.95em} or .active {color: #ff00ff}

- Other selectors: attribute
  - attribute of HTML elements that currently exist on the document
  - Example: <img alt="London Bridge Photo">
  - How to use: img[alt] { width: 1.5em} or [alt] {width: 1.5em} or
    [alt="London Bridge Photo"] {width: 1.5em}

CSS Properties

CSS Properties

- As seen in a CSS anatomy, property is part of a CSS rule. Property is a particular CSS style that dictates how a HTML element and/or its content can visually look. You cannot create your own properties. A list of properties are already pre-written and available for developers to use.
- Below are some of the most common properties used:
  - color
  - background-color
  - background-image
  - font-family
  - font-size
  - font-weight
  - text-align
  - border
  - margin
  - padding
  - float
  - list-style
  - width
  - height
  - text-decoration

For a comprehensive list of properties, visit https://www.w3schools.com/cssref/

Style html tables

Style html tables

- Example: say you want an outline around the main table structure as well as on all grids within it. Remember, by default a table does not display any outlines. Having outlines, make it easier to visually identify the table separating it from the rest of the page contents and easier to read the data on each row and column.

HTML markup:

```
<table>
    <!-- Row 1 -->
    <tr>
        <!-- Column 1 -->
        <td>Data 1</td>
        <!-- Column 2 -->
        <td>Data 2</td>
    </tr>
    <!-- Row 2 -->
    <tr>
        <!-- Column 1 -->
        <td>Data 3</td>
        <!-- Column 2 -->
        <td>Data 4</td>
    </tr>
</table>
```

CSS styling:

```
table, td {
    border: 1px solid #cccccc;
}
td {
    width: 100px;
    height: 50px;
    text-align:center;
}
```

View on the Browser:

Data 1	Data 2
Data 3	Data 4

Style html forms

Style html forms

- Example: say you have a simple form that needs to be styled and formatted so that it can be better presented to the user.

HTML markup:

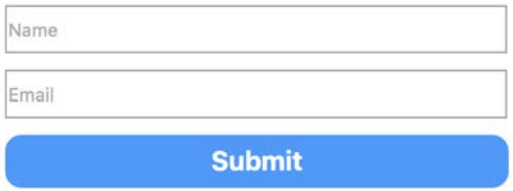
```
<form>
    <input type="text" placeholder="Name" />
    <input type="text" placeholder="Email" />
    <input type="submit" />
</form>
```

Browser view before styling:

CSS Styling:

```
input {
    width:300px;
    height:25px;
    border: 1px solid #999;
    display:block;
    margin-top:10px;
}
input[type="submit"] {
    background-color:#39f;
    color:#fff;
    font-weight:bold;
    font-size:1em;
    border:none;
    width:305px;
    height:32px;
    border-radius:10px;
}
```

Browser view after styling:



CSS Pseudo Class

CSS Pseudo Class

- A pseudo-class is used to define a special state or characteristic of an element. For example, it can be used to:
  - Style an element when a user mouse over it
  - Style visited and unvisited links differently
  - Style an element when it gets focus
- The syntax:

```
selector:pseudo-class {
  property: value;
}
```

- Next, we will look at some pseudo class types.

CSS Pseudo Class

- Anchor pseudo-class: this is to display the hypertext links in the specified state. Those states are:
  - a:link (yet to click)
  - a:visited (after click)
  - a:hover (mouse over)
  - a:active (mouse down)
- An example how it's used:

Common: Apply to an anchor <a> element.

```
a:hover {
  color: #ff0000;
}
```

Not common: Apply to a block element.

```
div:hover {
  background-color: blue;
}
```

CSS Pseudo Class

- Other common and useful pseudo-classes:
  - :focus (selects the target element that currently has focus)
  - :first-child (selects every target element that is the first child of its parent)
  - :last-child (selects every target element that is the last child of its parent)
  - :nth-child(n) (selects every target element that is the n child of its parent)
  - :nth-last-child(n) (selects every target element that is the n child of its parent, counting from the last child)
- Let's look at some examples on the next slide.

CSS Pseudo Class

HTML markup:

```
<p>This is some text.</p>
<p>This is some text.</p>
```

CSS styling:

```
p:first-child {
  color: blue;
}
```

View on the browser:

This is some text.

This is some text.

HTML

```
<div id="group1">
<p>I am a <i>strong</i> person.</p>
<p>I am a <i>stronger</i> person.</p>
</div>
<div id="group2">
<p>I am a <i>kind</i> person.</p>
<p>I am a <i>kinder</i> person.</p>
</div>
```

CSS styling:

```
p:nth-child(2) {
  color: blue;
}
```

View on the browser:

I am a *strong* person.

I am a *stronger* person.

I am a *kind* person.

I am a *kinder* person.

CSS Pseudo Elements

CSS Pseudo Elements

- CSS pseudo-element is used to style specified parts of an element.
- For example, it can be used to:
  - Style the first letter, or line, of an element
  - Insert content before, or after, the content of an element
- The syntax:

```
selector::pseudo-element {
  property: value;
}
```

- Next, we will look at some pseudo element types.

CSS Pseudo Elements

- The following are some common ones:
  - ::first-line (add a special style to the first line of a text)
  - ::first-letter (add a special style to the first letter of a text)
  - ::before (to insert some content before the content of an element)
  - ::after (to insert some content after the content of an element)
  - ::selection (matches the portion of an element that is selected by a user)
- Examples of how it's used:

```
p::first-line {
  color: #ff0000;
  font-variant: small-caps;
}
```

```
h1::after {
  content: url(smiley.gif);
}
```

CSS Pseudo Elements

- More complete examples:

HTML

```
<p>This example shows how to combine the ::first-letter and
::first-line pseudo-elements to add a special effect to the first
letter and the first line of a text!</p>
```

CSS

```
p::first-letter {
  color: #ff0000;
  font-size: xx-large;
}
p::first-line {
  color: #0000ff;
  font-variant: small-caps;
}
```

View on the

THIS EXAMPLE SHOWS HOW TO COMBINE THE ::FIRST-
letter and ::first-line pseudo-elements to add a special effect to the first letter and the first line of a text!

CSS Pseudo Elements

- More complete examples:

HTML

```
<h1> This is a heading</h1>
<p>This example shows the ::before pseudo-
element inserts content before the content of
an element.</p>
```

CSS

```
h1::before {
  content: url(smiley.gif);
}
```

View on the

This is a heading

This example shows the ::before pseudo-element inserts content before the content of an element.

Questions?

Resources

https://www.w3schools.com/css/css_website_layout.asp

https://www.studytonight.com/cascading-style-sheet/basic-webpage-structure

https://www.w3schools.com/css/css_table.asp

https://www.htmlgoodies.com/tutorials/forms/article.php/3888746/HTML-Forms-From-Basics-to-Style-Layouts.htm

https://www.w3schools.com/css/css_link.asp

https://blog.logrocket.com/a-guide-to-css-pseudo-elements/