

Front End Web Development Fundamentals

2_ Learn CSS- SYNC (01 hours 30 minutes)

Learning Objectives:

- Explain what CSS is?
- How to use it to style pages?
- Purpose of using CSS
- Writing Style Sheets:
- Style Sheet Strategies:
- Using Id's
- Using Classes
- font properties
- text properties
- styling links
- styling background
- styling tables
- What is the box model?

Opening: (5 mins)

- **Read the following to the group:**

If you want to make changes in your website, you have to go each and every line of the HTML document from where you want to change the code. Of course, this is very inconvenient and time-consuming way.

All web pages can be broken down into bucketed content areas

These areas can be updated by changing the code on every page

or -

By using cascading style sheets!

What is CSS?

CSS stands for cascading style sheets.

1. Created by Hakon Lie of MIT in 1994
2. Has become the W3C standard for controlling visual presentation of web pages
3. Cascading style-sheets are powerful mechanism to add style to web document
4. Enforce standards and uniformity
5. Create dynamic effects
6. Works by allowing you to specify rules

Purpose of using CSS

- Saves time
- Easy to change
- Pages load faster
- Keep consistency

- Give you more control over layout
- Use styles with JavaScript
- Multiple Device Compatibility

Content: (1hr 15 mins)

Writing Style Sheets:

- Let us understand the differences among inline, internal, and external style sheets

1. In-line styles

Add styles to each tag within the HTML file

Use it when you need to format just a single section in a web page

Style attribute is used to add style

Example

```
<h1 style="color:red; font-family: sans-serif" > IU </h1>
```

2. Embedded/internal styles

A style is applied to the entire HTML file

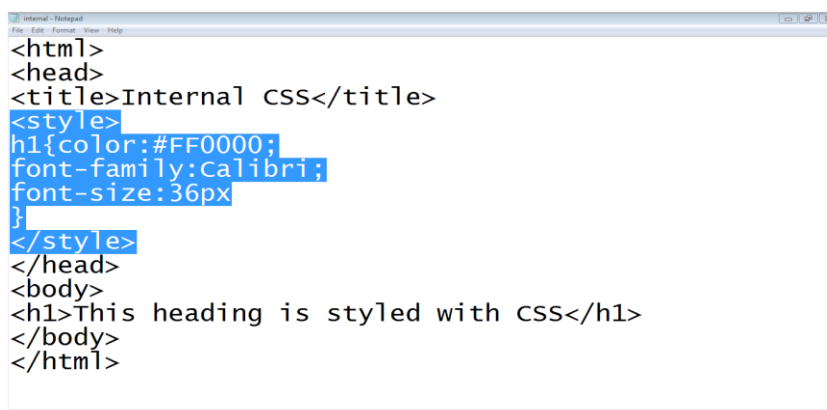
Use it when you need to modify all instances of particular element (e.g., h1) in a web page

Example

```
<style>
```

```
h1 {color:red; font-family:sans-serif}
```

```
</style>
```

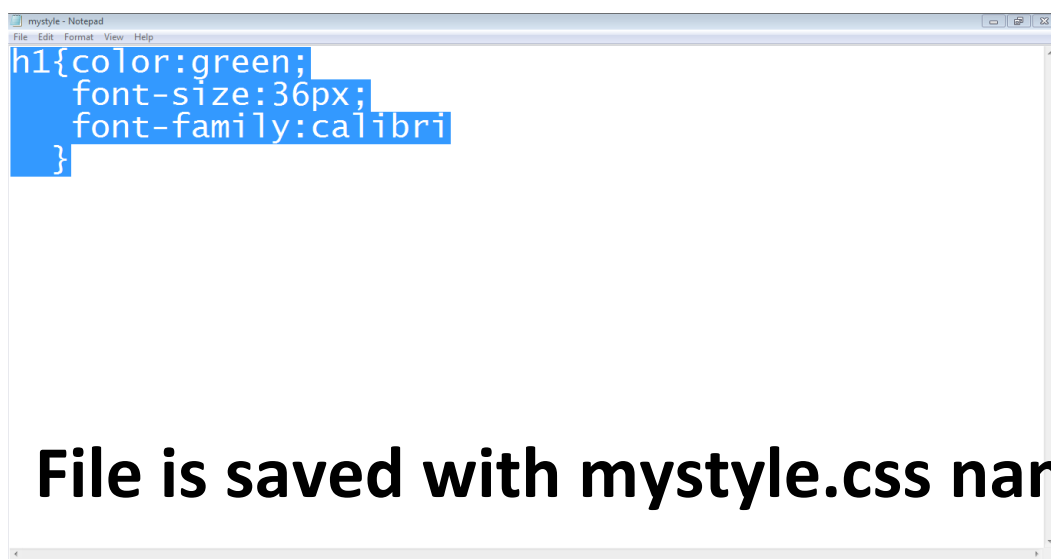


3. External style sheet

An external style sheet is a text file containing the style definition (declaration)

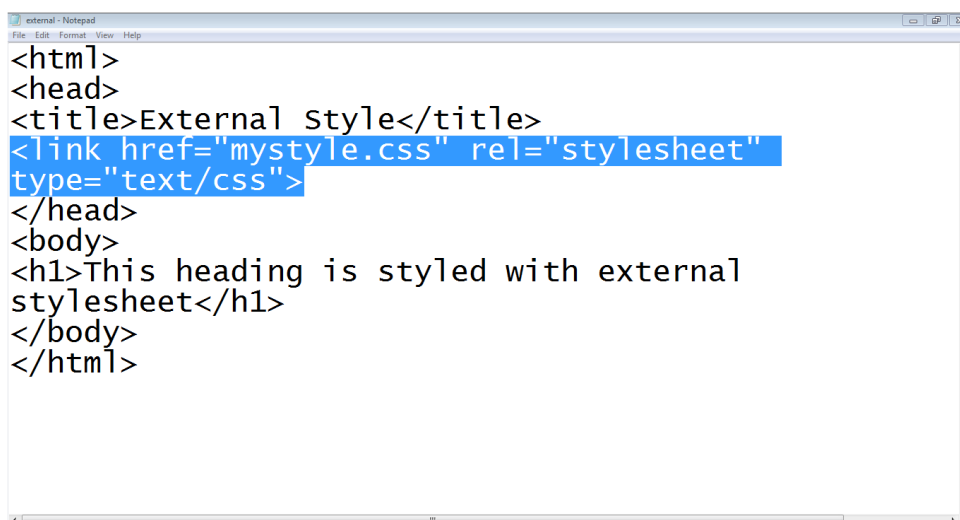
Use it when you need to control the style for an entire website

- Open a new blank document in Notepad
- Type style declarations
 - `h1 {color:red; font-family:calibri;}`
- Do not include `<style>` tags
- Save the document as filename.css
- Open a HTML file
- Between `<head>` and `</head>` add
- `<link href=URL rel="relation_type" type="link_type">`
- URL is the file.css
- `Relation_type="stylesheet"`
- `Link_type="text/css"`
- Save this file and the .css file in the same web server directory



```
mystyle - Notepad
File Edit Format View Help
h1{color:green;
font-size:36px;
font-family:calibri
}
```

File is saved with mystyle.css name



```
external - Notepad
File Edit Format View Help
<html>
<head>
<title>External style</title>
<link href="mystyle.css" rel="stylesheet"
type="text/css">
</head>
<body>
<h1>This heading is styled with external
stylesheet</h1>
</body>
</html>
```

Style Sheet Strategies:

- **Wherever possible, place your styles in external style sheets**
- **Try to avoid using styles in tags**

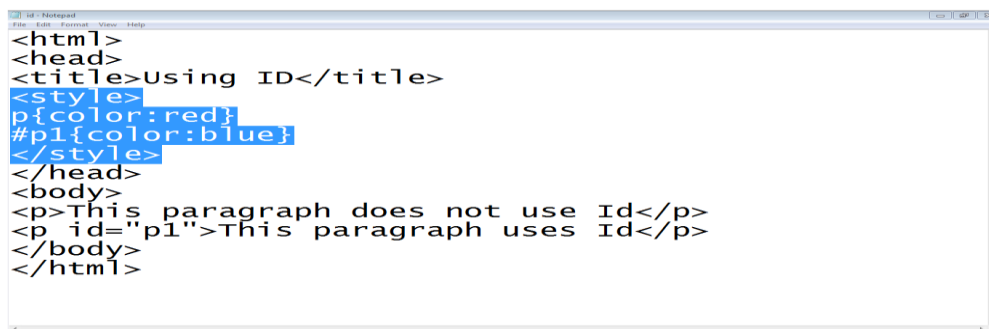
Understand how to declare a style:

- A rule consists of
 - A selector: element or elements the declaration applies to
 - Declaration: how the elements referred to in the selector should be styled
 - property: which is the property of the selected element
 - value: which is a specification for this property

selector declaration
| |
H1 {color: blue;}
| |
property value

Using Id,s:

- Use an id to distinguish something, like a paragraph, from the others in a document
- The id selector is used to specify a style for a single, unique element
- Create a style Id:
 - #id_name {style attributes and values}
- Use a style Id:
 - <tag ID=id_name>



```
<html>
<head>
<title>Using ID</title>
<style>
p{color:red;}
#p1{color:blue;}
</style>
</head>
<body>
<p>This paragraph does not use Id</p>
<p id="p1">This paragraph uses Id</p>
</body>
</html>
```

Note: HTML requires each id be unique– therefore an id value can only be used once in a document

Using Classes:

You can mark a group of elements with a common identifier using the class attribute

- To create a class
 - tag.class_name {style attributes} or
 - .class_name {style attributes}
- To apply a style
 - <tag CLASS=class_name>
 - <h1 CLASS=FirstHeader>IU</h1>

```
<html>
<head>
<title>Using ID</title>
<style>
p{color:red;}
p.p1{color:blue;}
.p2{color:yellow;}
</style>
</head>
<body>
<p>This paragraph does not use class</p>
<p class="p1">This paragraph uses p1 class</p>
<p class="p2">This paragraph uses p2 class</p>
</body>
</html>
```

Difference between classes and Id,s:

- you can't have more than one tag with the same ID value
 - You can apply the same Class value to multiple document tags
 - Classes or Id?
 - use ID's for any elements that are simply used once on a page
- OR
- only use classes to style websites, but, when you have to use an element in JavaScript, use an identifier
 - You can apply a style to many selectors if you like

```
h1, h2, h3 {
color: #36C;
font-weight: normal;
letter-spacing: .4em;
margin-bottom: 1em;
text-transform: lowercase;
}
```

CSS properties:

- *font properties:*

Common font properties:

1. font-family:
 - Specifies the typeface or family of font that should be used

Common values:

- Arial, Courier/Courier New, Georgia, Times/Times New Roman and Verdana
- Example:

H1{font-family:arial}

2. *font-size*

- Specifies the size of a font

Common values:

- In pixels (12px,20px etc.)
- Absolut size (small, medium, lage, x-large etc.)

Example:

- H1{font-size:20px}

3. *font-weight*:

Specifies whether the font should be bold or normal

Common values:

Normal, bold, bolder, lighter

Example:

H1{font-weight:bold}

- *font-style*:
 - Specifies whether the font should be normal, italic or oblique
 - Common values:
 - Normal, italic, oblique

Example:

H1{font-style:italic}

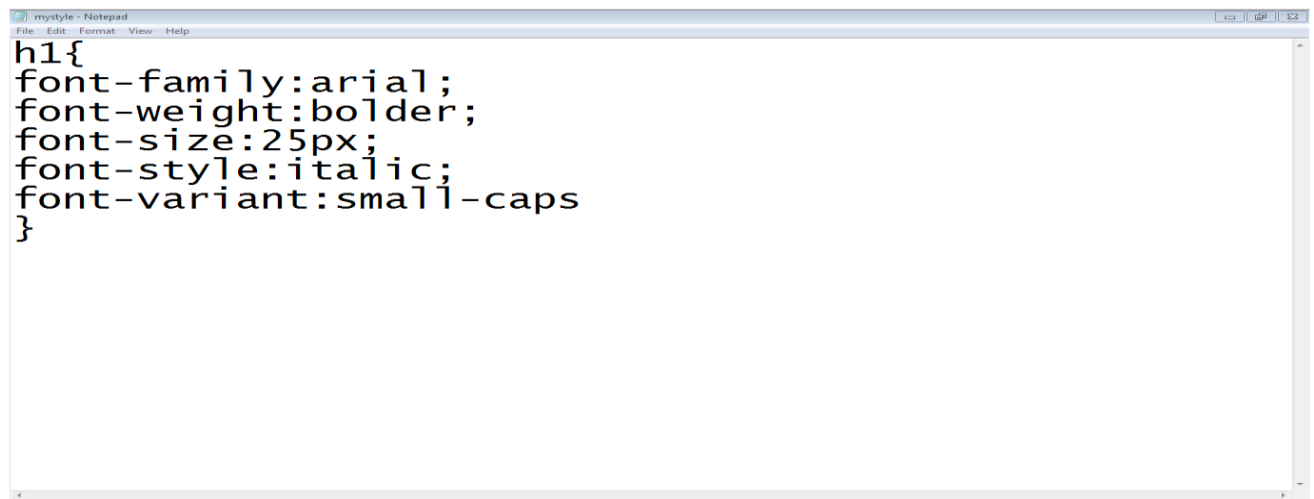
- *font-variant*:
 - Specifies whether the font should be normal or small-caps (smaller version of upper case)

Common values:

- Normal, small-caps

Example:

H1{font-variant:small-caps}



```
h1{
font-family:arial;
font-weight:bolder;
font-size:25px;
font-style:italic;
font-variant:small-caps
}
```

- *TEXT properties:*
 - *color:*
 - specifies the color of the text
 - P{color : green}
 - *text-align:*
 - horizontal alignment of the text
 - Left, right, center or justify
 - *vertical-align:*
 - Vertical alignment of the text
 - Sub, super, top, middle, bottom
 - *text-decoration:*
 - specifies the whether the text should be underline, overline, line-through or blinking
 - *text-transform:*
 - text should be lowercase, uppercase or capitalized
 - *letter-spacing:*
 - Specifies the space between letters

H1 {letter-spacing:3px }

- *word-spacing:*
 - Specifies the space between words

H1 {word-spacing:4px }

- *Styling links:*
 1. *color:*
 - Changes the color of the links
 2. *background-color:*
 - Highlights the link, as if it had been highlighted with a highlighter pen
 3. *text-decoration:* Underline, strike through, over-line, blink
- *Styling Background:*
 1. *background-color:*
 - Specifies the background color

2. *background-image*:
 - Specifies the background image
3. *background-repeat*:
 - Specifies whether the image should repeat or not
4. *background-position*:
 - Where an image should be positioned

```
body{
  background-color: pink;
  background-image: url("download.jpg");
  background-repeat: no-repeat;
  background-position: top right;
}
```

- *Styling tables*:

1. *text and font*
2. *vertical – align*
3. *width*
4. *height*
5. *background - color*
6. *background – image*
7. *border*:

- border-style (solid, dashed, dotted, double etc.)
- border-color
- border-bottom (solid, dashed, dotted, double etc.)

8. padding
 - Padding-left
 - Padding-right
 - Padding-top
 - padding-bottom

```
body{
  background-color: pink;
  background-image: url("download.jpg");
  background-repeat: no-repeat;
  background-position: top right;
}

table{
  border-style: dashed;
  border: 30px;
  width: 300px;
}

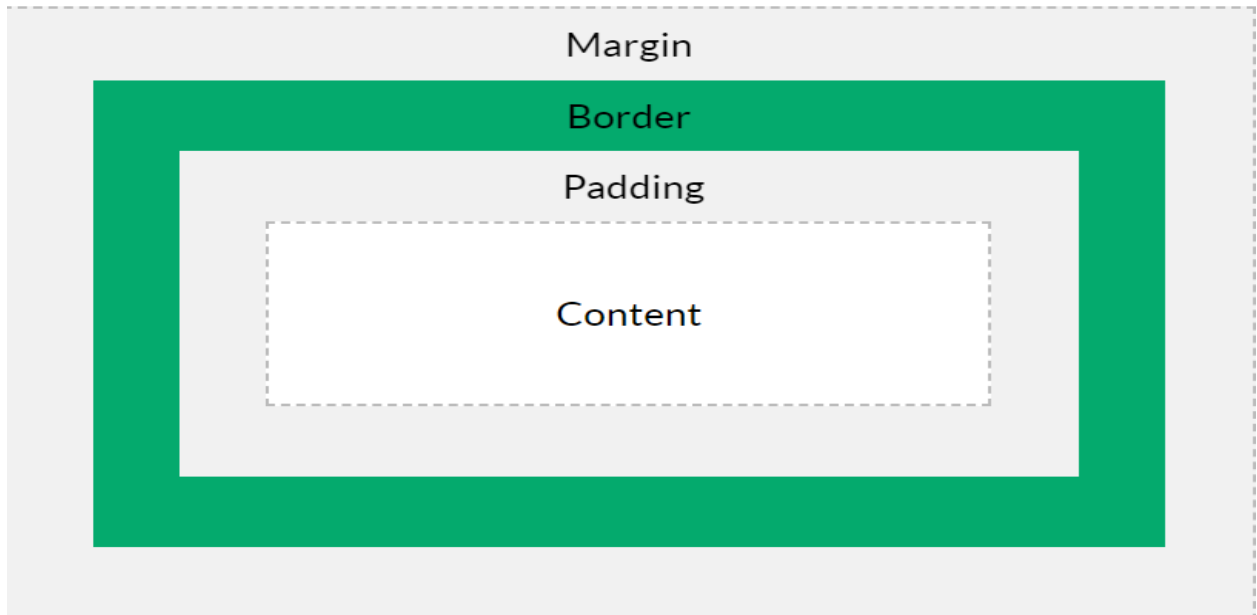
th{
  font-family: calibri;
  background-color: gray;
}

td{
  background-color: lightblue;
  vertical-align: top;
}

tr{
}
```

What is the box model?

1. Every element on a page is a rectangular box and may have width, height, padding, borders, and margins.
2. The box model allows us to add a border around elements, and to define space between elements.



- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        width: 320px;
        padding: 10px;
        border: 5px solid gray;
        margin: 0;
      }
    </style>
  </head>
  <body>

    <h2>Hello everyone</h2>

    
    <div>The picture above is 350px wide. The total width of this element is also 350px.</div>

  </body>
</html>
```

CLOSING (15 mins)

- Ask participants to summarize what they learned in today's session.
- Let participants know what comes next -