

TOPIC 2.2

WEB PAGE STYLING USING CSS

OBJECTIVES

- Using CSS to add style to web documents
 - Syntax of CSS
 - Selectors
 - Ways to use CSS
 - Formatting Text & Font
 - Formatting Background

HTML AND STYLE SHEETS

- **HTML** marks the **different parts of a document**, but it does not indicate how document content should be displayed by browsers
- For this reason, the exact **appearance of each page element** is described in a separate document known as a **style sheet**

WHAT IS CSS?

- CSS stands for **Cascading Style Sheets**
- Controls format/style/presentation of web pages
- These include font, text, colour, alignment, layout, borders and even some simple animation to enhance the presentation of the document
- It enables the different presentation of web pages in different media
- CSS standards: CSS1, CSS2, CSS3(evolution)

EXAMPLE ON USING CSS

- Page with no document structure
 - [Example0](#)
- Without Styling
 - [Example1](#)
- Different styling
 - [Example2](#)
 - [Example3](#)

SYNTAX OF CSS

- It consists only of 3 parts:
`selector { property: value; }`
- **selector** ==> the **element** you want to style
or
user-defined selector
(ID selector or Class selector)
- Styles are typically specified in a **property: value** pair
`color : red;`

SYNTAX OF CSS

- **property** is the attribute you wish to change
- Each property can have a **value**
- The **property** and **value** are *separated by a colon* and surrounded by *curly braces*
- Example :
body {**color**: **black**;}
- For multiple words, put quotes around the value:

p {**font-family**: "**sans serif**"}

SYNTAX OF CSS

selector { property: value }

h1

```
{  
font-family:Comic Sans Ms;  
color:white;  
background-color:orange;  
}
```

semi colon to separate the rules

SYNTAX OF CSS

- You can specify more property-value pair as follows:

```
body {color : black;  
      font-size : 10px;  
      text-align:center;}
```

SYNTAX OF CSS

- You can **group** selectors together and separate each selector with a comma
- Example: group all header elements and each header element is green

h1, h2, h3, h4, h5, h6

{

color : green;

}

THE ID SELECTOR

- The ID selector is used to specify a style for a **single, unique element**
- Uses the id attribute of the HTML element, and is defined with a **"#"**

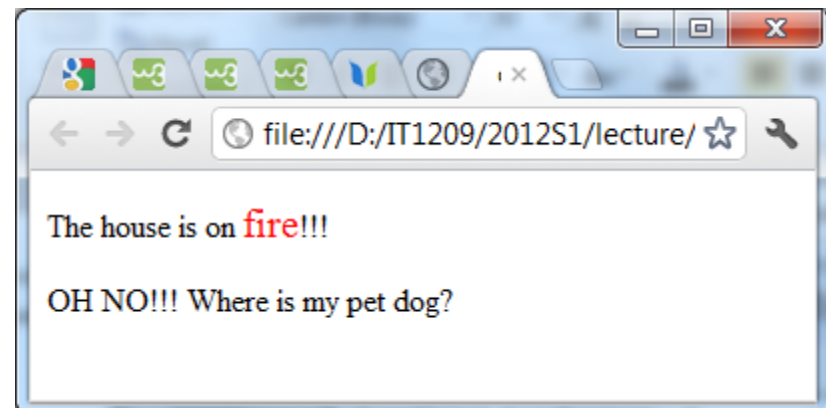
THE ID SELECTOR

- First, set up an ID inside the element with an unique identifier

```
<p>The house is on <span id="fire">fire</span>!!!</p>  
<p>OH NO!!! Where is my pet dog?</p>
```

- Then, apply CSS styles using the ID with a “#”

```
#fire  
{color : red;  
font-size : 20px;}
```



ID SELECTOR EXAMPLE

CSS

```
<style>  
#para1  
{  
text-align:center;  
color:red;  
}  
</style>
```

HTML

```
<p id="para1">Hello World!</p>  
<p>This paragraph is not affected by the style.</p>
```

Result

Hello World!

This paragraph is not affected by the style.

THE CLASS SELECTOR

- It is used to specify a style for **a group of elements**
- Most often used on several elements
- Uses the HTML class attribute, and is defined with a **"."**

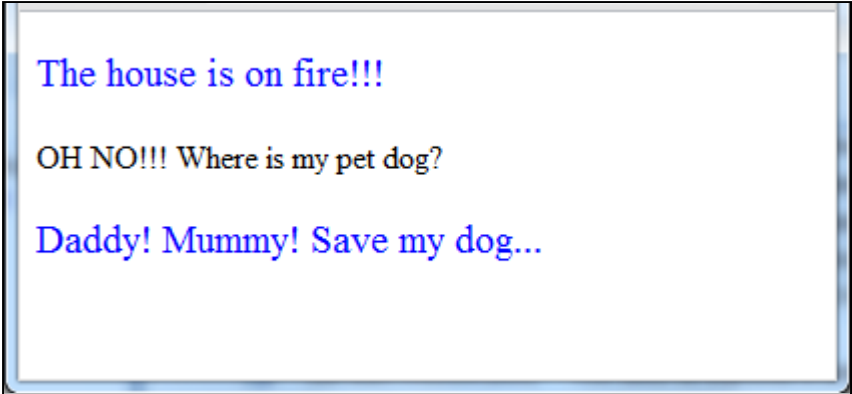
THE CLASS SELECTOR

- First, set up elements with the same class

```
<p class="odd">The house is on <span id="fire">fire</span>!!!</p>  
<p>OH NO!!! Where is my pet dog?</p>  
<p class="odd">Daddy! Mummy! Save my dog...</p>
```

- Then, apply CSS styles using the class with a "."

```
.odd  
{ color:blue;  
  font-size : 20px;}
```



The house is on fire!!!

OH NO!!! Where is my pet dog?

Daddy! Mummy! Save my dog...

CLASS SELECTOR EXAMPLE

CSS

```
<style>  
.center  
{  
text-align:center;  
}  
</style>
```

HTML

```
<h1 class="center">Center-aligned heading</h1>  
<p class="center">Center-aligned paragraph.</p>
```

Result

Center-aligned heading

Center-aligned paragraph.

CLASS SELECTOR EXAMPLE

CSS

```
<style>  
p.center  
{  
text-align:center;  
}  
</style>
```

HTML

```
<h1 class="center">Center-aligned heading</h1>  
<p class="center">Center-aligned paragraph.</p>
```

Result

This heading will not be affected

This paragraph will be center-aligned.

THREE WAYS TO USE CSS

- **Inline:** apply CSS rules directly within the HTML element
- **Embedded/Internal:** apply CSS rules locally for a web page
- **External:** store the CSS rules in a .css file; link the css file to any web page which need to apply the CSS rules.

THREE WAYS TO USE CSS

- **Inline styles:**

- refers to using of the style attribute in the relevant tag

```
<body>
<p>The house is on <span style="color:red">fire</span>!</p>
</body>
```

- loses many of the advantages of style sheets by mixing content with presentation
- use this method sparingly

THREE WAYS TO USE CSS

- **Internal(or embedded) style sheet:**
 - styles are defined within the **head** section of the page

```
<!doctype html>
<html>
<head>
<title>Internal CSS</title>
  <style type="text/css">
    .odd
    {color:blue;
     font-size : 20px;}
    h1 {text-shadow:3px 3px 5px #FF0000;}
  </style>
</head>
```

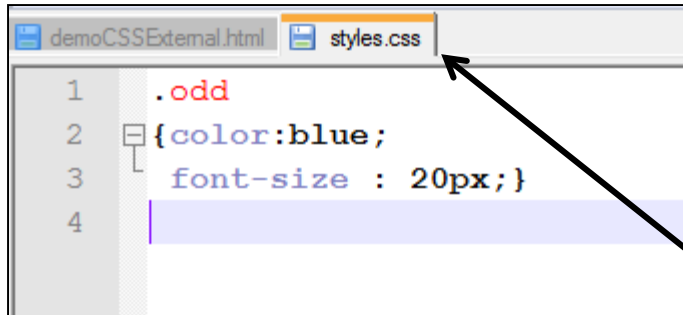
Style element are defined with attribute **type="text/css"**

- Style rules are applied to this particular web page only.

THREE WAYS TO USE CSS

- **External style sheet:**
 - All styles are defined in another separate document with file extension **.css**
 - The styles can be applied to many pages within the same website
 - The look of the entire website can be changed by just changing one file
 - Each page must link to the style sheet using the **<link>** tag

THREE WAYS TO USE CSS



```
1  .odd
2  {color:blue;
3    font-size : 20px;}
4
```

Link element in
HTML page
specifies linkage to
"styles.css"



```
1  <!doctype html>
2  <html>
3  <head>
4    <title>Internal CSS</title>
5    <link rel="stylesheet" type="text/css" href="styles.css" />
6  </head>
7  <body>
8    <p class="odd">The house is on <span id="fire">fire</span>!!
9    <p>OH NO!!! Where is my pet dog?</p>
```

STYLE CASCADE

Cascading Order (in order of priority)

1. Browser default ← lowest priority
2. External Style Sheet (in a standalone .css file)
3. Embedded Style Sheet (inside the <head> tag)
4. Inline Style (inside HTML element) ← highest priority

STYLE CASCADE

- As a general rule of thumb, all other things being equal, the **more specific style is applied** instead of the more general
- An additional factor in applying a style sheet is that properties are passed from a parent element to its children in a process known as **style inheritance**

```
body {color: blue;}
```

```
h1 {text-align: center;}
```


STYLE CASCADE EXAMPLE

- For example, an external style sheet has these properties for the h3 selector

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

- And an embedded style sheet has these properties for the h3 selector:

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

More Specific



- Then the color is inherited from the external style sheet and the text-alignment and the font-size are replaced by the embedded style sheet

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

FORMATTING TEXT WITH CSS

Property	Effect / Value
color	Specify the color of the text
text-indent	Specify the indentation
text-align	Specify the alignment left, right, justify
text-decoration	Value can be e.g.: underline, blink, line-through
text-transform	Value can be e.g.: uppercase, lowercase, capitalise
vertical-align	Value can be eg: bottom, middle, sub, super, top

FORMATTING TEXT WITH CSS

Property	Effect / Value
text-shadow	Specify the horizontal shadow, the vertical shadow, the blur distance, and the color of the shadow
word-wrap	Force the text to wrap to fit within an area

FORMATTING TEXT WITH CSS

- **color** → set the text-color
 - e.g. `h1 {color : blue;}`
- Color value can be specified in the following ways:
 - **Hexadecimal colors**
 - format : #RRGGBB, where the RR (red), GG (green) and BB (blue)
 - e.g. #ff0000
 - **RGB colors**
 - `rgb(red, green, blue)`
 - e.g. `rgb(0,0,255)`, `rgb(0%,0%,100%)`

FORMATTING TEXT WITH CSS

–RGBA colors

- format : rgba(red, green, blue, alpha)
- The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (fully opaque)
- e.g rgba(0,0,255,0.5)

–Predefined/Cross-browser color names

- e.g. red, black, white

FORMATTING TEXT WITH CSS

- **text-indent** → specifies the indentation of the first line in a text-block
 - e.g. `p {text-indent : 50px;}`
- **text-align** → specifies the horizontal alignment of text in an element
 - Values are left, right, center, justify
 - e.g. `p {text-align : center;}`

FORMATTING TEXT WITH CSS

- `text-decoration` → specifies the decoration added to text
 - e.g. `.important {text-decoration:underline;}`

This line of text is very IMPORTANT!!!

This line of text is very IMPORTANT!!!

- e.g. `a {text-decoration:none;}`

Please don't underline this link!

Please don't underline this link!

FORMATTING TEXT WITH CSS

- **text-shadow** → specifies the horizontal shadow, vertical shadow, blur distance and the color of the shadow

–e.g. `h1 {text-shadow: 5px 5px 5px #FF0000;}`

Text shadow effect!

- **word-wrap** → Allows long, unbreakable words to be broken and wrap to the next line

–e.g. `h1 {word-wrap: break-word;}`

This paragraph contains a very long word:
thisisaveryveryveryveryveryverylongword.
The long word will break and wrap to the next line.



This paragraph contains a very long word:
thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.

FONT FORMATTING WITH CSS

Property	Description
font-family	Specifies the font family for text
font-size	Specifies the font size of text
font-style	Specifies the font style for text
font-weight	Specifies the weight of a font

FONT FORMATTING WITH CSS

- **font-family**

Font	Generic Font Family
Arial	sans-serif
Times New Roman	serif
Courier New	monospace
Comic Sans	cursive

FONT FAMILY

```
body {  
    font-family: Verdana, Geneva, Arial, sans-serif;  
}
```



If the first font type is available on the computer, use it as the font , otherwise, look for the next font type .

FONT FORMATTING WITH CSS

- **font-size**

unit	size
px	pixel
%	percentage relative to the current default setting
pt	1/72cm
em	unit of measure that refers to the current font

FONT FORMATTING WITH CSS

- **font-style**

Value	Effect
italic	Uses an italic font
normal	Uses an non-oblique, not-italic font
oblique	Uses an oblique or slanted font

FONT FORMATTING WITH CSS

- **font-weight**

Value	Effect
100 to 900	Lightest (100) to heaviest(900)
bold	Same as 700
bolder	+100 heavier than the current
lighter	-100 lighter than the current
Normal	Same as 400

FONT FORMATTING WITH CSS

- Setting spacing in between

Property	Effect / Value
line-height	Specify the amount of space between lines. May use e.g. em, px, % as unit
letter-spacing	Specify the amount of space between letters. Usually using px; can be positive or negative.

FORMATTING BACKGROUND WITH CSS

Property	Description
background-color	Sets the background color of an element
background-image	Sets the background image for an element
background-size	Specifies the size of the background images
background-repeat	Sets how a background image will be repeated

FORMATTING BACKGROUND WITH CSS

- To set the **background color** of an element, use the property

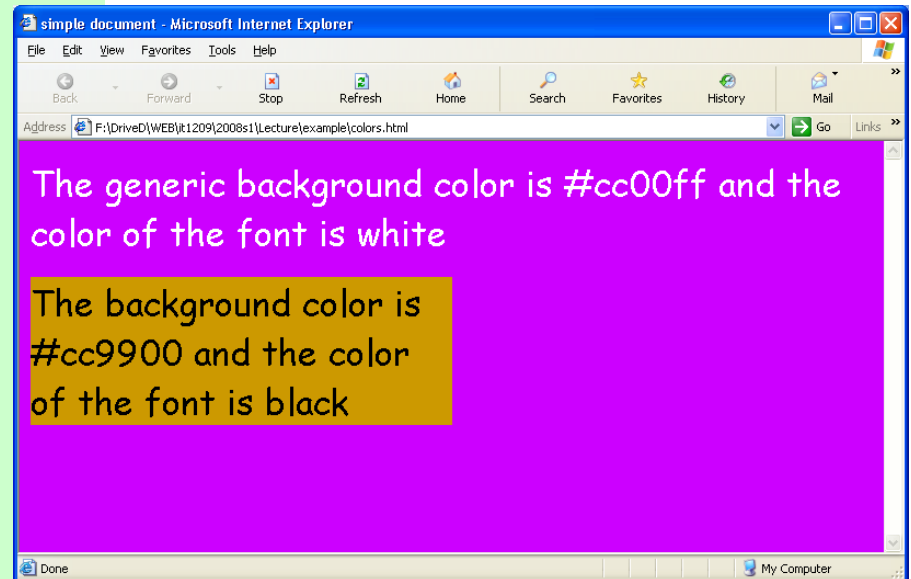
`background-color: color;`

where *color* is a color name or a color value.

FORMATTING BACKGROUND COLOR

Setting font colour

```
<style type="text/css">
body {
    font-size: 200%;
    font-family: comic sans ms;
    color: #FFFFFF;
    background-color: #cc00ff;
}
#p1 {
    color: black;
    background-color: #cc9900;
    width: 50%;
}
</style>
```



Setting background colour

FORMATTING BACKGROUND WITH CSS

- To set the **background image** of an element, use the property

`background-image: url(file);`

where *file* is the filename of the image.

- To set the **size of the background image** of an element, use the property

`background-size: width height;`

FORMATTING BACKGROUND WITH CSS

- To set how the **background image should repeat**, use the property

`background-repeat: value;`

where *value* can be repeat| repeat-x| repeat-y| no-repeat.

FORMATTING BACKGROUND IMAGE

```
<style type="text/css">
```

```
#div1 {
```

```
background: url(img_tree.gif) top left;
```

```
background-size: 20% 100%;
```

```
background-repeat: no-repeat;
```

```
background-color: lightblue;
```

```
width: 100%;
```

```
height: 50px;
```

```
}
```

```
#div2 {
```

```
background: url(img_flwr.gif);
```

```
background-size: 20% 100%;
```

```
background-repeat: repeat-x;
```

```
background-color: lightgreen;
```

```
width: 100%;
```

```
height: 50px;
```

```
}
```

```
</style>
```

