



```
body {  
  font: x-small;  
  background: #;  
  color: black;  
  margin: 0;  
  padding: 0;  
}  
  
table {
```



Week 5 - Cascading Style Sheet (CSS)





Aim/ Objectives

- To build student awareness of style sheet mechanism available to web programmers seeking to develop and interactive web sites.
- At the same time students will develop practical expertise in CSS providing them with a solid foundation from which to develop a mastery of web programming techniques



Learning Outcome

- Use style sheets to enhance the user interface of a web site.



Cascading Style Sheet

- Cascading Style Sheet (CSS) is a style sheet language used to describe the presentation of a document written in a markup language (e.g. HTML and XHTML).
- CSS is a list of statements (rules) that define colors, fonts, layout, and other aspects to HTML elements.



Why Use Style Sheets?

- It allows much greater degree of layout and display control.
- The amount of format coding necessary to control display characteristics can be greatly reduced.
- It allows multiple styles to be attached to a document at once.
- It also allows for all the style formatting in a document to be changed at once, thus a document can be easily formatted for different purposes (online, brochures, printing, etc.).



Three ways to insert CSS

- **External style sheets**, i.e. a separate CSS-file referenced from the document
- **Embedded style (internal)**, blocks of CSS information inside the HTML document itself
- **Inline style**, attaches a style definition within the HTML element it is modifying



Inline Style

- To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property.

```
<html>
<head>
</head>
<body>

</body>
</html>
```



Embedded / internal Style Sheet

- An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, by using the <style> tag

```
<html><head>  
<style type="text/css">  
.border{  
border-bottom-width:thick;  
border-bottom-style:double;  
border-bottom-color:red;  
}  
</style></head>
```

```
<body>  
  
</body>  
</html>
```




External Style Sheet

- An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section

```
<html>
<head>
<link rel="stylesheet" type="text/css"
href="externalstyle.css"/>
</head>
<body>

</body>
</html>
```

```
.border{
border-bottom-width:thick;
border-bottom-style:double;
border-bottom-color:red;
}
```

Externalstyle.css



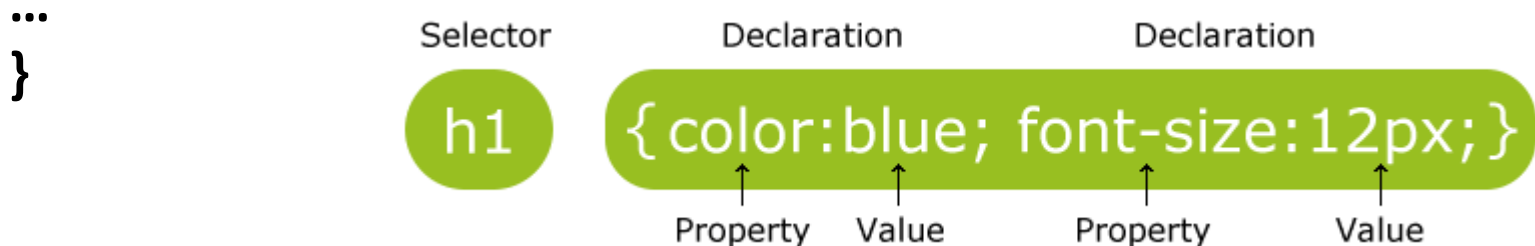
Cascading order

- What style will be used when there is more than one style specified for an HTML element? (multiple style sheets)
 1. Inline style (inside an HTML element)
 2. Internal style sheet (in the head section)
 3. External style sheet
 4. Browser default

Defining a Style

- CSS style definitions follow the format given below:

```
selector_expression {
  element_property: property_value(s);
  element_property: property_value(s);
  ...
}
```



- The ***selector_expression*** is an expression that can be used to match specific elements (HTML elements/tags) in the document; the ***element_property*** specifies which properties of the element the definition will affect, and each property can take a value. The property and value are separated by a colon, and surrounded by curly braces.

Example Code

```
<html>
<head>
<style type="text/css">
body
{
background-color:yellow;
}
h1
{
background-color:#00ff00;
}
p
{
background-
color:rgb(255,0,255);
}
</style>
</head>

<body>
<h1>This is heading 1</h1>
<p>This is a paragraph.</p>
</body>
</html>
```



Example -view

This is heading 1

This is a paragraph.



Property Values and Units

CSS supports the following metrics for property values :

- CSS keywords and other properties, such as thin, thick, transparent, ridge, and etc.
- Real-world measures
 - Inches (in)
 - Centimeters (cm)
 - Millimeters (mm)
 - Points (pt) – The points used by CSS2 are equal to 1/72 of an inch
 - Picas (pc) – 1 pica is equal to 12 points
- Screen measures in pixels (px)
- Relational to font size (font size (em) or x-height size (ex))
- Percentages (%)
- Color codes (#rrggbb or rgb(r,g,b))



Property Values and Units

- Angles
 - Degrees (deg)
 - Radians (rad)
- Time values (seconds (s) and milliseconds (ms)) – Used with aural style sheets
- Frequencies (hertz (Hz) and kilohertz (kHz))
 - Used with aural style sheets

Selectors

- Selectors are essentially patterns that enable a user agent to identify what elements get what styles.
 - E.g. “If it is a paragraph tag, give it this style”:
p { text-indent: 2em; }
- The ways of matching:
 - Matching elements by name
 - Matching using the universal selector
 - Matching elements by class
 - Matching elements by identifier
 - Matching elements by specific attributes

Matching Elements by Name

- Example:
- **h1 { color: green; }**
 - Causes all occurrences of the selector tags (h1) to be formatted with the property/value section of the definition (color: green)
- Multiple selectors
 - Example
 - **h1, h2, h3, h4 { color: green; }**



Matching Using the Universal Selector

- The universal selector (*) can be used to match any element in the document.
- E.g.
 - Match every tag and apply the color red:
*** { color: red; }**
 - Match any **** tag that is a descendant of a **<td>** tag, which is a descendant of a **<tr>** tag: **tr td ol { color: red; }**



Matching Elements by Class

- With the class selector you can define different styles for the same type of HTML element.
- Say that you would like to have two types of paragraphs in your document: one right-aligned paragraph, and one center-aligned paragraph. Here is how you can do it with styles:
 - **p.right {text-align: right}**
 - **p.center {text-align: center}**
- You have to use the class attribute in your HTML document:

<p class="right">This paragraph will be rightaligned.</p>

**<p class="center">This paragraph will be
centeraligned.</p>**



Matching Elements by Class

- To apply more than one class per given element, the syntax is:

<p class="center bold"> This is a paragraph. </p>

The paragraph above will be styled by the class "center" AND the class "bold".

- You can also omit the tag name in the selector to define a style that will be used by all HTML elements that have a certain class. In the example below, all HTML elements with class="center" will be center-aligned:

.center {text-align: center}

**<h1 class="center">This heading will be center-aligned.
</h1>**

<p class="center">This paragraph will be centeraligned.</p>



Matching Elements by Identifier

- You can also match element identifiers (the id attribute). To match identifiers, you use the pound sign (#) in the selector as a prefix for the id. E.g.

#comment {background-color: green}

...

**<p id="comment">This paragraph is a
comment.</p>**



Example

```
<html>
<head>
<style type="text/css">
p{
color:blue;
text-align:center;
}
#para1
{
text-align:left;
color:green;
}

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

```
<p>
Sri Lanka's documented history spans 3,000 years
</p>
<p id="para1">
Sri Lanka's documented history spans 3,000 years
<p class="left">
Sri Lanka's documented history spans 3,000 years
</p>
<p class="center">
Sri Lanka's documented history spans 3,000 years
</p>

Sri Lanka's documented history spans 3,000 years
</body> </html>
```

Sri Lanka's documented history spans 3,000 years

Sri Lanka's documented history spans 3,000 years

Sri Lanka's documented history spans 3,000 years

Sri Lanka's documented history spans 3,000 years

Sri Lanka's documented history spans 3,000 years



Matching elements by Specific Attributes

- To match any attribute in elements, not just class and id, the attribute and the value(s) have to be specified at the end of the selector, offset in square brackets. E.g.
 - Match any table with a border attribute set to 3,
table[border="3"]
 - Match any table with a border attribute,
table[border]
 - Match any table elements with a class value of *datalist* and *border* value of 3,
table.datalist[border="3"]



Example

```
<html>
```

```
<head>
```

```
<style>
```

```
a[target] {
```

```
    background-color: yellow;
```

```
}
```

```
</style></head>
```

```
<body>
```

```
<p>The links with a target attribute gets a yellow background:</p>
```

```
<a href="http://www.w3schools.com">w3schools.com</a>
```

```
<a href="http://www.disney.com" target="_blank">disney.com</a>
```

```
<a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>
```

```
</body>
```

```
</html>
```

The links with a target attribute gets a yellow background:

[w3schools.com](http://www.w3schools.com) [disney.com](http://www.disney.com) [wikipedia.org](http://www.wikipedia.org)



Pseudo-classes

- Pseudo classes are identifiers that are understood by user agents and apply to elements of certain types without the elements having to be explicitly styled. E.g. Anchor styles

```
<html>
<head>
<style type="text/css">
:link{color:#FF00FF}
:visited{color:#FF00FF}
:active{color:#FF00FF}
</style>
</head>
<body>
<a href="http://www.sliate.ac.lk/"> Visit SLIATE Web </a>
</body>
</html>
```



Fonts

- Fonts are styled collection of letters and symbols. Different fonts can be used to convey different information.
- CSS supports five different font family types:
 - **Serif fonts** They are used in body text; the finishing strokes, flared or tapering ends, or serified endings, make the lines of characters flow and tend to be easier on the eyes.
 - **Sans serif fonts** They are used for headings or other large areas of emphasis.
 - **Cursive fonts** They are used in extreme cases where emphasis is on ornamentation rather than legibility.
 - **Fantasy fonts** They are used for logos and other ornamentation purposes where legibility is secondary.
 - **Monospace fonts** They are used in code listings and other approximating terminal output.



Font Characteristics

- Fonts are mapped according to a system similar to ruled paper.
 - The line that the characters or symbols sit on is called the **baseline**.
 - The distance between the baseline and the top of the highest characters (usually capital letters and lowercase letters such as l, f or t) is known as the **ascension**.
 - The distance between the baseline and the lowest point of characters that dip below it (such as p, g or q) is known as the **descension**.





Defining a Font Family

- The font-family property defines the font or fonts should be used for elements in the document.
- More than one font family names is defined with a generic family name for versatility.

Introduction

This chapter introduces what is expected from a final year project and highlights its objectives. It also describes on how to select a project and a supervisor. A brief summary of the respect of the chapters is also given.



CSS font

- `p{font-family:"Times New Roman", Times, serif;}`
- `p.normal {font-style:normal;}`
- `p.italic {font-style:italic;}`
- `p.oblique {font-style:oblique;}`
- `h1 {font-size:40px;}`
`h2 {font-size:30px;}`
`p {font-size:14px;}`
- `h1 {font-size:2.5em;} /* 40px/16=2.5em */`
`h2 {font-size:1.875em;} /* 30px/16=1.875em */`
`p {font-size:0.875em;} /* 14px/16=0.875em */`
- `H1{font-weight:normal;}`
- `H1{font-weight: bold ;}`
- `H1{font-weight:lighter;}`



Defining a Font Family

```
<head>
<title>Font Family</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
body { font-family: "Courier New", Courier, mono; }
h3 {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-weight: bold;
}
-->
</style>
</head>

<body>
<h3>Introduction</h3>
<p>This chapter introduces what is expected from a final year project and
highlights its objectives. It also describes on how to select a project and
a supervisor. A brief summary of the respect of the chapters is also
given.</p>
</body>
```



Some Other Font Properties

- Font sizing
 - Two properties can be used to control font sizing: **font-size** and **font-size-adjust**. Both properties can adjust a font absolutely or relative to the current font size.
- Font styling
 - Four properties can be used to affect font styling: **font-style**, **font-variant**, **font-weight**, and **font-stretch**.
- Line spacing
 - The **line-height** property controls the line height of text, i.e. the distance between the baseline of two vertically stacked lines of text.

Example -Code

```
<head>
<title>Font Properties</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
p.heading { font-size: x-large; font-weight: bold; }
p.italic { font-style: italic; font-variant: small-caps; }
p.spaced { line-height: 200%; }
-->
</style>
</head>

<body>
<p class="heading">Introduction</p>
<p class="italic">This chapter introduces what is expected from a final
year project and highlights its objectives.</p>
<p class="spaced">The individual project is by far the most important
single piece of work in the BIT degree programme. It provides the
opportunity for a candidate to demonstrate independence and originality</p>
</body>
```




Example - View

Introduction

THIS CHAPTER INTRODUCES WHAT IS EXPECTED FROM A FINAL YEAR PROJECT AND HIGHLIGHTS ITS OBJECTIVES.

The individual project is by far the most important single piece of work in the BIT degree programme. It provides the opportunity for a candidate to demonstrate independence and originality



Colors

- CSS options for defining colors can entirely replace the color attributes in plain HTML. In addition CSS has the following advantages:
 - In plain HTML, when you wanted to create an area with a specified color, you were forced to create a table. However, with CSS, you can define an area to have a specific color without that area being part of a table.
 - In plain HTML, specially when working with tables, you had to specify font attributes and colors etc. for each and every table cell. However, with CSS you can simply refer to a certain class.



Specifying a Color

- The color value can be expressed using one of three methods:
 - Color keywords
 - blue, black, green, and so on
 - Color hexadecimal values
 - values specified in the form #rrggbb
 - •E.g. #FF0000 for red, #000000 for black
 - – Color decimal or percentage values
 - Values specified using the rgb() function. The value can be an integer between 0 and 255 or a percentage.
 - E.g. rgb(100, 0, 100), rgb(50%, 0, 50%)



Specifying a Color

- Most elements in a HTML document have two color properties: a foreground color and a background color.
 - Setting the foreground color for contents
color: <color_value>;
 - Setting the background color for an area
background-color: <color_value>;
 - Setting a background image to fill out an area
background-image: url("<url_to_image>");

Example- Code

```
<head>
<title>Colors</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
.redtext { color: #FF0000; }
.filled { background-color: #0000FF; color: #00FF00; }
-->
</style>
</head>

<body>
<p class="redtext">Although the project is done for a client a candidate
should remember that the purpose of the project is to fulfil an examination
requirement of the BIT degree programme. </p>
<p class="filled">It is the responsibility of the candidate to identify a
suitable project. The project should comprise a substantial amount of
individual work to satisfy the PEB that the project objectives have been m
et as well as the time spent on the project is justified. </p>
</body>
```



CSS Background image -Example

- `Body { background-image:url('paper.gif');}`
- `Body { background-image:url('paper.gif');
 background-repeat:repeat-x;
 }`
- `Body { background-image:url('paper.gif');
 background-repeat:no-repeat;
 background-position:right top;
 }`



CSS Background color-Example

- `Body { background-color:#b0c4de;}`
- `h1 { background-color:red;}`
- `div { background-color:rgb(255,255,0);}`



Text color- Example

- `body {color:blue;}`
- `h1 {color:#00ff00;}`
- `h2 {color:rgb(255,0,0);}`



Text alignment- Example

- `h1 {text-align:center;}`
`p.date {text-align:right;}`
`p.main {text-align:justify;}`



Text decoration -Example

- `h1 {text-decoration:overline;}`
- `h2 {text-decoration:line-through;}`
- `h3 {text-decoration:underline;}`
- `h4 {text-decoration:blink;}`
- `a{text-decoration:none;}`



Text Transformation

- `p.uppercase {text-transform:uppercase;}`
- `p.lowercase {text-transform:lowercase;}`
- `p.capitalize {text-transform:capitalize;}`



Text Indentation

- `P{text-indent:50px;}`



CSS Lists -Example

- `ul.a {list-style-type: circle;}`
- `ul.b {list-style-type: square;}`
- `ol.c {list-style-type: upper-roman;}`
- `ol.d {list-style-type: lower-alpha;}`
- `ul`
`{`
`list-style-image: url('sqpurple.gif');`
`}`



Values for Unordered Lists

Value	Description
none	No marker
disc	Default. The marker is a filled circle
circle	The marker is a circle
square	The marker is a square

Values for Ordered Lists

Value	Description
armenian	The marker is traditional Armenian numbering
decimal	The marker is a number
decimal-leading-zero	The marker is a number padded by initial zeros (01, 02, 03, etc.)
georgian	The marker is traditional Georgian numbering (an, ban, gan, etc.)
lower-alpha	The marker is lower-alpha (a, b, c, d, e, etc.)
lower-greek	The marker is lower-greek (alpha, beta, gamma, etc.)
lower-latin	The marker is lower-latin (a, b, c, d, e, etc.)
lower-roman	The marker is lower-roman (i, ii, iii, iv, v, etc.)
upper-alpha	The marker is upper-alpha (A, B, C, D, E, etc.)
upper-latin	The marker is upper-latin (A, B, C, D, E, etc.)
upper-roman	The marker is upper-roman (I, II, III, IV, V, etc.)



Tables

CSS properties available for table attributes:

Purpose	Table Attribute	CSS Property(ies)
Borders	border	border properties
Spacing inside cell	cellpadding	padding properties
Spacing between cells	cellspacing	border-spacing properties
Width of a table	width	width and table-layout properties
Table framing	frame	border properties
Alignment	align, valign	text-align, vertical-alignment

Table-Examples

- `table, th, td`
`{`
`border: 1px solid black;`
`}`
- `td`
`{`
`text-align:right;`
`}`
- `td`
`{`
`height:50px;`
`vertical-align:bottom;`
`}`
- `td`
`{`
`padding:15px;`
`}`
- `th`
`{`
`background-color:green;`
`color:white;`
`}`

CSS Border- Example

- ```
p.one
{
border-style:solid;
border-width:5px;
}
```
- ```
p.two
{
border-style:solid;
border-width:medium;
}
```
- ```
p.one
{
border-style:solid;
border-color:red;
}
```
- ```
p.two
{
border-style:solid;
border-color:#98bf21;
}
```
- ```
p
{
border-top-style:dotted;
border-right-style:solid;
border-bottom-style:dotted;
border-left-style:solid;
}
```



# CSS Margin- Examples

- `margin-top:100px;`  
`margin-bottom:100px;`  
`margin-right:50px;`  
`margin-left:50px;`



# CSS Margin- Examples

- The margin property can have from one to four values..
- **margin:25px 50px 75px 100px;**
  - top margin is 25px
  - right margin is 50px
  - bottom margin is 75px
  - left margin is 100px
- **margin:25px 50px 75px;**
  - top margin is 25px
  - right and left margins are 50px
  - bottom margin is 75px
- **margin:25px 50px;**
  - top and bottom margins are 25px
  - right and left margins are 50px
- **margin:25px;**
  - all four margins are 25px



# CSS Padding - Examples

- `padding-top:25px;`
- `padding-bottom:25px;`
- `padding-right:50px;`
- `padding-left:50px;`



# CSS Padding - Examples

- The padding property can have from one to four values.
  - right and left paddings are 50px
  - bottom padding is 75px
- **padding:25px 50px 75px 100px;**
  - top padding is 25px
  - right padding is 50px
  - bottom padding is 75px
  - left padding is 100px
- **padding:25px 50px;**
  - top and bottom paddings are 25px
  - right and left paddings are 50px
- **padding:25px 50px 75px;**
  - top padding is 25px
- **padding:25px;**
  - all four paddings are 25px