

5. CASCADING STYLE SHEETS

INTRODUCTION

CSS stands for Cascading Style Sheets. It not only extends its features in controlling colors and sizes of fonts, but also controls spaces between various elements, the color and width of a given line etc.

- CSS defines **how HTML elements are to be displayed**
- Styles were added to HTML 4.0 **to solve a problem**
- CSS saves a lot of work

CSS Solved a Big Problem

HTML was NEVER intended to contain tags for formatting a document. HTML was intended to define the content of a document, like:

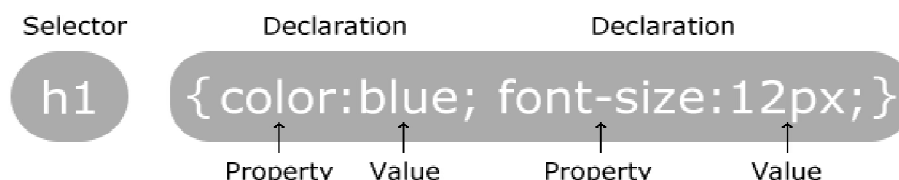
```
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

When tags like , and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS. In HTML 4.0, all formatting could (and should!) be removed from the HTML document, and stored in a separate CSS file.

CSS Syntax

A CSS rule set consists of a selector and a declaration block:



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a property name and a value, separated by a colon.

CSS Example

A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly braces:

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

To make the CSS code more readable, you can put one declaration on each line.

CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. Comments are ignored by browsers.

A CSS comment starts with `/*` and ends with `*/`. Comments can also span multiple lines:

Example

```
p {  
    color: red;  
    /* This is a single-line comment */  
    text-align: center;  
}  
  
/* This is  
a multi-line  
comment */
```

CSS Selectors

CSS selectors allow you to select and manipulate HTML elements. CSS selectors are used to "find" (or select) HTML elements based on their id, class, type, attribute, and more.

The element Selector

The element selector selects elements based on the element name. You can select all `<p>` elements on a page like this: (all `<p>` elements will be center-aligned, with a red text color)

Example

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

The id Selector

The id selector uses the id attribute of an HTML element to select a specific element. An id should be unique within a page, so the id selector is used if you want to select a single, unique element. To select an element with a specific id, write a hash character, followed by the id of the element.

The style rule below will be applied to the HTML element with `id="para1"`:

Example

```
#para1 {  
    text-align: center;  
    color: red;  
}
```

The class Selector

The class selector selects elements with a specific class attribute. To select elements with a specific class, write a period character, followed by the name of the class:

In the example below, all HTML elements with `class="center"` will be center-aligned:

Example

```
.center {  
    text-align: center;  
    color: red;  
}
```

You can also specify that only specific HTML elements should be affected by a class. In the example below, all <p> elements with class="center" will be center-aligned:

Example

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

Grouping Selectors

If you have elements with the same style definitions, like this:

```
h1 {  
    text-align: center;  
    color: red;  
}  
  
h2 {  
    text-align: center;  
    color: red;  
}  
  
p {  
    text-align: center;  
    color: red;  
}
```

you can group the selectors, to minimize the code. To group selectors, separate each selector with a comma. In the example below we have grouped the selectors from the code above:

Example

```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```

Ways to use CSS

There are three ways of inserting a style sheet:

- Inline style
- Internal style sheet
- External style sheet

Inline Style

An inline sheet applies style to a particular element in a web page. Inclusion of Style in a tag is called Inline styles. Operator colon (:) is followed by **style** property. The style attribute can contain any CSS property. To separate multiple properties we have to use operator semicolon (;).

Example: Program to illustrate Inline Style

```
<html>
  <head>
    <title>Inline Styles</title>
  </head>
  <body bgcolor="lavender">
    <h1 style="color:blue;margin-left:30px;">
      This is a heading.
    </h1>
    <p>This is a paragraph.</p>
    <p style="font-size:15pt;font-family:'Bookman Old Style'">
      Web Technologies
    </p>
    <p style="font-size:12pt;font-style:italic;">
      Web Technologies
    </p>
    <p style="font-size:20pt;font-variant:small-caps;">
      Web Technologies
    </p>
  </body>
</html>
```



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, inside the <style> tag.

Syntax for <style> tag

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

Advantage of Internal Style Sheet comparing with inline styles, at a time several tags can be formatted with internal style sheets, where as in inline styles only one tag at a time formatted.

Example: Program to illustrate Internal Style Sheet

```
<html>  
    <head>  
        <style type="text/css">  
            body {  
                background-color: linen;  
            }  
            h1 {  
                color: maroon;  
                margin-left: 40px;  
            }  
        </style>  
    </head>  
    <body>  
        <h1>This is a heading</h1>  
        <p>This is a paragraph.</p>  
    </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 just one file.

Each page must include a link to the style sheet with the <link> tag. The <link> tag goes inside the head section:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension. An example of a style sheet file called "myStyle.css", is shown below:

```
body {
    background-color: lightblue;
}
h1 {
    color: navy;
    margin-left: 20px;
}
```

Example: Program to illustrate External Style Sheet

ourstyles.css

```
p {color:red;font-family:Arial}
.s5 {font-size:25;}
.s10 {font-size:30;}
.s15 {font-size:50;}
h1 {color:white;background-color:blue}
```

```
<html>
<head>
    <title>External Styles Sheets</title>
<link rel="stylesheet" type="text/css" href="ourstyles.css">
</head>
<body bgcolor=tan>
    <h1 align=center>Color Header, example!</h1>
    <hr>
    <p>Normal Paragraph</p>
    <p class="s5">Web Technologies</p>
    <p class="s10">Web Technologies</p>
    <p class="s20">Web Technologies</p>
    <p class="s10">Web Technologies</p>
    <p class="s5">Web Technologies</p>
</body>
</html>
```



Multiple Style Sheets

If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.

For example, assume that an external style sheet has the following properties for the `<h1>` element:

```
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

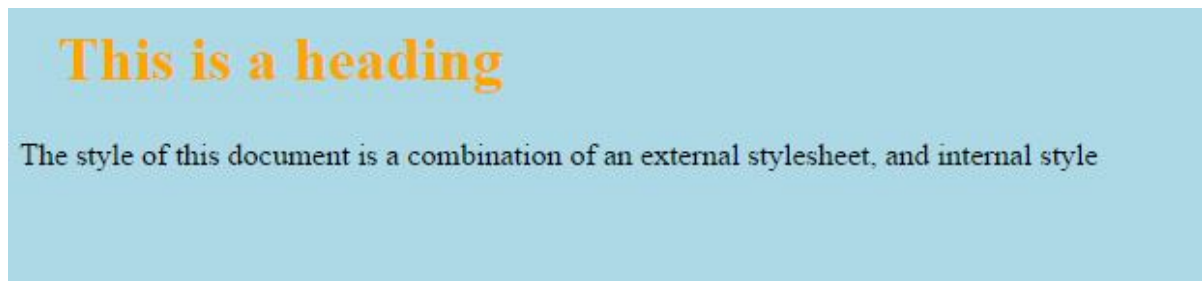
then, assume that an internal style sheet also has the following property for the `<h1>` element:

```
h1 {  
    color: orange;  
}
```

If the page with the internal style sheet also links to the external style sheet the properties for the `<h1>` element will be:

```
color: orange;  
margin-left: 20px;
```

Output:



Multiple Styles Will Cascade into One

Styles can be specified:

- inside an HTML element
- inside the <head> section of an HTML page
- in an external CSS file

Tip: Even multiple external style sheets can be referenced inside a single HTML document.

Cascading order

What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:

- Browser default
- External style sheet
- Internal style sheet (in the head section)
- Inline style (inside an HTML element)

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the <head> tag, or in an external style sheet, or in a browser (a default value).

Note: If the link to the external style sheet is placed after the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!

Positioning at required location:

When user want to design his screen one should have control on it. Style sheets provide required facilities to position the elements in the required position. One such style element "position: absolute" means that you can place requirement absolutely at required location. This concept is known as **absolute positioning**.

Other possible value for it is "position: relative", you can give top, bottom, left or right values but it keeps elements in the general flow of elements on the page. In absolute positioning we can use left, top, right, or bottom values that give exact location of element.

When talking about absolute position, another important factor is, if we place two or more elements one on another, which one is displayed, and can we change the order. In general last element appears on the top of previous elements, and definitely we can change the order of appearance with the help of **z-index** style.

Example: Write a CSS program to position WELCOME TO WEB along with some hyphens below or above it.

```
<html>
<head>
    <title>Header Position</title>
</head>
<body bgcolor=tan>
    <h1 style="position:absolute;top:130px;left:120px;">
        WELCOME TO WEB </h1>
    <h1 style="position:absolute;top:135px;left:130px;color:blue">
        ===== </h1>
    <h1 style="position:absolute;top:125px;left:135px;color:blue">
        ===== </h1>
    <h1 style="position:absolute;top:30px;left:120px;z-index:3">
        WELCOME TO WEB </h1>
    <h1 style="position:absolute;top:25px;left:130px;color:blue;z-index:2">
        ===== </h1>
    <h1 style="position:absolute;top:35px;left:130px;color:blue;z-index:1">
        ===== </h1>
</body>
</html>
```



Spacing

You can control the spacing of your page with margins, padding, width, and height. This will be more important in the next session (and frankly, is a little complicated), so for now just remember that spacing is set in pixels. A pixel is a standard size that all browsers recognize – though it can still be somewhat flexible, depending on the size of the computer screen on which your page is being viewed.

You will get the feel for how large pixels are after you play around with your page. Pixels are written px, and some typical commands might include things like:

```
margin-left: 10px;    width: 100px;        height: 252px;
padding: 5px 5px 5px 5px;
```

Example: Write a CSS to demonstrate styles for links

```
<html>
<head>
<style type="text/css">
  a.one:link {color:#ff0000;}
  a.one:visited {color:#0000ff;}
  a.one:hover {color:#ffcc00;}
  a.two:link {color:#ff0000;}
  a.two:visited {color:#0000ff;}
  a.two:hover {font-size:150%;}
  a.three:link {color:#ff0000;}
  a.three:visited {color:#0000ff;}
  a.three:hover {background:#66ff66;}
  a.four:link {color:#ff0000;}
  a.four:visited {color:#0000ff;}
  a.four:hover {font-family:monospace;}
  a.five:link {color:#ff0000;text-decoration:none;}
  a.five:visited {color:#0000ff;text-decoration:none;}
  a.five:hover {text-decoration:underline;}
</style>
</head>
<body>
  <p>Mouse over the links to see them change layout.</p>
  <p><b><a class="one" href="default.html" target="_blank">
    This link changes color</a></b></p>
  <p><b><a class="two" href="default.html" target="_blank">
    This link changes font-size</a></b></p>
  <p><b><a class="three" href="default.html" target="_blank">
    This link changes background-color</a></b></p>
  <p><b><a class="four" href="default.html" target="_blank">
    This link changes font-family</a></b></p>
  <p><b><a class="five" href="default.html" target="_blank">
    This link changes text-decoration</a></b></p>
</body>
</html>
```

Example: Write a CSS program to display Horizontal Menu

```
<html>
<head>
<style type="text/css">
  ul {
    list-style-type:none;
    margin:0;
    padding:0;
    overflow:hidden;
```

```
    }
    li {
        float:left;
    }
    a:link,a:visited {
        display:block;
        width:120px;
        font-weight:bold;
        color:#FFFFFF;
        background-color:#98bf21;
        text-align:center;
        padding:4px;
        text-decoration:none;
        text-transform:uppercase;
    }
    a:hover,a:active {
        background-color:#7A991A;
    }
}
</style></head>
<body>
    <ul>
        <li><a href="#home">Home</a></li>
        <li><a href="#news">News</a></li>
        <li><a href="#contact">Contact</a></li>
        <li><a href="#about">About</a></li>
    </ul>
</body>
</html>
```



Cascading Style Sheets (CSS 2)

BACKGROUND	
background	background-color background-image background-repeat background-attachment background-position
background-attachment	scroll fixed
background-color	color-rgb color-hex color-name transparent
background-image	url none
background-position	top left top center top right center left center center center right bottom left bottom center bottom right x-% y-% x-pos y-pos
background-repeat	repeat repeat-x repeat-y no-repeat
BORDER	
border	border-width border-style border-color
border-bottom	border-bottom-width border-style border-color
border-bottom-color	border-color
border-bottom-style	border-style
border-bottom-width	thin medium thick length
border-color	color
border-left	border-left-width border-style border-color
border-left-color	border-color
border-left-style	border-style
border-left-width	thin medium thick length
border-right	border-right-width border-style border-color
border-right-color	border-color
border-right-style	border-style
border-right-width	thin medium thick length
border-style	none hidden dotted dashed solid double groove ridge inset outset
border-top	border-top-width border-style border-color
border-top-color	border-color
border-top-style	border-style
border-top-width	thin medium thick length
border-width	thin medium thick length

CLASSIFICATION	
clear	left right both none
cursor	url auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help
display	none inline block list-item run-in compact marker table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption
float	left right none
position	static relative absolute fixed
visibility	visible hidden collapse
DIMENSION	
height	auto length %
line-height	normal number length %
max-height	none length %
max-width	none length %
min-height	length %
min-width	length %
width	auto % length
FONT	
font	font-style font-variant font-weight font-size/line-height font-family caption icon menu message-box small-caption status-bar
font-family	family-name generic-family
font-size	xx-small x-small small medium large x-large xx-large smaller larger length %
font-size-adjust	none number
font-stretch	normal wider narrower ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded
font-style	normal italic oblique
font-variant	normal small-caps
font-weight	normal bold bolder lighter 100 200 300 400 500 600 700 800 900

GENERATED CONTENT	
content	string url counter(name) counter(name, list-style-type) counters(name, string) counters(name, string, list-style-type) attr(X) open-quote close-quote no-open-quote no-close-quote
counter-increment	none identifier number
counter-reset	none identifier number
quotes	none string string string string
LIST & MARKERS	
list-style	list-style-type list-style-position list-style-image
list-style-image	none url
list-style-position	Inside outside
list-style-type	none disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana katakana hiragana-iroha katakana-iroha
marker-offset	auto length
MARGIN	
margin	margin-top margin-right margin-bottom margin-left
margin-bottom	auto length %
margin-left	auto length %
margin-right	auto length %
margin-top	auto length %
OUTLINE	
outline	outline-color outline-style outline-width
outline-color	color invert
outline-style	None dotted dashed solid double groove ridge inset outset
outline-width	thin medium thick length

Values in italics are place holders for an actual value (like 1px, 1em, 1%), values in normal text are values that can be used as the actual value

CSS Properties in **Dark Red** are shorthand properties and each value must be defined. The exception is where the property can define from one to four of the sides of a box element property (Top-Right-Bottom-Left) - i.e. border-width

Values in italics are place holders for an actual value (like 1px, 1em, 1%), values in normal text are values that can be used as the actual value

PADDING	
padding	<i>padding-top padding-right padding-bottom padding-left</i>
padding-bottom	<i>length %</i>
padding-left	<i>length %</i>
padding-right	<i>length %</i>
padding-top	<i>length %</i>
POSITIONING	
bottom	<i>auto % length</i>
clip	<i>shape auto</i>
left	<i>auto % length</i>
overflow	<i>visible hidden scroll auto</i>
position	<i>static relative absolute fixed</i>
right	<i>auto % length</i>
top	<i>auto % length</i>
vertical-align	<i>Baseline sub super top text-top middle bottom text-bottom length %</i>
z-index	<i>auto</i>

UNITS	
MEASUREMENT	
%	percentage
cm	centimeter
em	1em = current font size of current element
ex	1ex = ~1/2 current size of current element
in	inch
mm	millimeter
pc	pica (= 12 points)
pt	point (= 1/72 inch)
px	pixel
COLORS	
Color name	Red, blue, green, dark green
rgb(x,y,z)	Red = rgb(255,0,0)
rgb(x%,y%,z%)	Red = rgb(100%,0,0)
#rrggbb	Red = #ff0000 (or shorthand = #f00)

TABLE	
border-collapse	<i>collapse separate</i>
border-spacing	<i>length length</i>
caption-side	<i>top bottom left right</i>
empty-cells	<i>show hide</i>
table-layout	<i>auto fixed</i>
TEXT	
color	<i>color</i>
direction	<i>ltr rtl</i>
letter-spacing	<i>normal length</i>
text-align	<i>left right center justify</i>
text-decoration	<i>none underline overline line-through blink</i>
text-indent	<i>length %</i>
text-shadow	<i>none color length</i>
text-transform	<i>none capitalize uppercase lowercase</i>
unicode-bidi	<i>normal embed bidi-override</i>
white-space	<i>normal pre nowrap</i>
word-spacing	<i>normal length</i>

SELECTOR TYPES		
Name	Info	Example
Universal	Any element	* { font: 10px Arial; }
Type	Any element of that type	h1 { text-decoration: underline; }
Grouping	Multiple elements of different types	h1, h2, h3 { font-family: Verdana; }
Class	Multiple elements of different types when you don't want to affect all instances	.sampleclass { text-decoration: underline; }
Id	A single element type when you don't want to affect all instances of	#sampleid { text-decoration: underline; }
Descendant	An element that is below (in the document tree) another element—no matter how many levels below	#gallery h1 { text-decoration: underline; }
Child	An element that is directly below (in the document tree) another	#title > p { font-weight: bold; }
Sibling	All elements of a type that share the same parent	h1 + p { font-style: italic; }
Attribute	An element with that matches the attribute listed	option[selected] { color: #ff0000; } input[type="pass"] { color: #ccc; } img[src="sm.gif"] { border: 1px solid #000; } a[rel~="next"] { color: #fff; } *[lang="en"] { color: red; }

PSEUDO-CLASS	
:active	Adds special style to an activated element
:focus	Adds special style to an element while the element has focus
:hover	Adds special style to an element when you mouse over it
:link	Adds special style to an unvisited link
:visited	Adds special style to a visited link
:first-child	Adds special style to an element that is the first child of some other element
:lang	Allows the author to specify a language to use in a specified
PSEUDO-ELEMENT	
:first-letter	Adds special style to the first letter of a text
:first-line	Adds special style to the first line of a text
:before	Inserts some content before an element
:after	Inserts some content after an element

SELECTOR PATTERNS	
Name	Example
*	any element
E	an element of type E
E[foo]	an E element with a "foo" attribute
E[foo="bar"]	an E element whose "foo" attribute value is exactly equal to "bar"
E[foo~="bar"]	an E element whose "foo" attribute value is a list of space-separated values, one of which is exactly equal to "bar"
E[hreflang="en"]	an E element whose "hreflang" attribute has a hyphen-separated list of values beginning (from the left) with "en"
E:first-child	an E element, first child of its parent
E:link E:visited	an E element being the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited (:visited)
E:active E:hover E:focus	an E element during certain user actions
E:lang(fr)	an element of type E in language "fr" (the document language specifies how language is determined)
E::first-line	the first formatted line of an E element
E::first-letter	the first formatted letter of an E element
E::before	generated content before an E element
E::after	generated content after an E element
E.warning	an E element whose class is "warning" (the document language specifies how class is determined).
E#myid	an E element with ID equal to "myid".
E F	an F element descendant of an E element
E > F	an F element child of an E element
E + F	an F element immediately preceded by an E element