**CSS**

**2.1 Overview Of Style Sheets**

It is time to take your web designing skills to the next level with **C**ascading **S**tyle **S**heets (CSS). They are a way to control the look and feel of your HTML documents in an organized and efficient manner.

With CSS you will be able to:

* Add new looks to your old HTML
* Completely restyle a web site with only a few changes to your CSS code
* Use the "style" you create on any webpage you wish!

**2.1.1 CSS Syntax**

The CSS syntax consists of a set of rules. These rules have 3 parts: a selector, a property, and a value.

Syntax:

selector { property: value }

The selector is often the HTML element that you want to style.

For example:

|  |
| --- |
| h1 { color: blue } |

This code tells the browser to render all occurrences of the HTML *h1* element in blue.

To apply more than one property separate each declaration with a semi-colon.

|  |
| --- |
| h1 { color:blue; font-family:arial,helvetica,"sans serif" } |

Example

|  |
| --- |
| <html>  <head>  <style type=”text/css”>  p{color: red;}  </style>  </head>  <body>  <p> CSS Tutorial </p>  </body>  </html> |

Output :

CSS Tutorial

**2.1.2 Class Selecto**

The class selector is used to specify a style for a group of elements. It is most often used on several elements. This allows you to set a particular style for many HTML elements with the same class. The class selector uses the HTML class attribute, and is defined with a "."

Syntax :

**.** [Class Name] {   
  property : value;   
  ...   
}

Example

|  |
| --- |
| . sample { color : blue;} |

To apply this style to the HTML, use the following code:

|  |
| --- |
| <head>  <style type="text/css">  h1 .sample {color:blue;}  </style>  </head>  <body>  <h1 class="sample">CSS Class</h1>  </body>  </html> |

The above HTML code renders:

# CSS Class

**Multiple Classes**

It is possible to include multiple classes at the same time.

For example, assuming we have the following CSS declaration,

.applylarge{ font-size:20px; }

.applyred{ color:#FF0000; }

the following HTML,

would render,

<p class="applylarge applyred">This is an example of multiple classes.</p>

|  |
| --- |
| This is an example of multiple classes. |

**2.2 Creating Rules**

You can create styles either within the HTML of your Web page or within an external style sheet that is linked to your Web page. Internal styles come in two types: inline and embedded. External style sheets are either linked or imported.

The DIV are used to divide the content into different sections. In this session we are going to discuss about all these tags in detail.

**2.2.1 Implementing CSS**

There are 3 ways of implementing CSS.

They are:

* Inline CSS
* Embedded CSS
* External CSS

## Inline CSS

Style sheet information is applied to the **current** element. Instead of defining the style once, then applying the style against all instances of an element , you only apply the style to the instance you want the style to apply to.

Example:

<p style="color:#ff9900"> CSS tutorial.</p>

## Embedded CSS

You embed CSS information into an HTML document using the 'style' element. You do this by embedding the CSS information within **<style>...</style>**tags in the head of your document.

Example

**Place the following code between the <head>...</head> tags of your HTML document:**

<style type="text/css" >

p {color:#ff9900;}

</style>

Now, whenever any of those elements are used within the body of the document, they will be formatted as instructed in the above style sheet.

Example program with **Inline** and **Embedded** Style Sheets:

|  |
| --- |
| <html>  <head>  <style type="text/css" >  p {color : #ff9900;}  </style>  </head>  <h1 style="color :blue "> WELCOME </h1>  <body>  <p> This is some text.</p>  </body>  </html> |

Output :

# WELCOME

This is some text.

## External CSS

An external style sheet is a separate file where you can declare all the styles that you want to use throughout your website. You then link to the external style sheet from all your HTML pages.

This means you only need to set the styles for each element once. If you want to update the style of your website, you only need to do it in one place.

For example:

* Type the following into a plain text file, and save with a .css extension.(Example external.css)

p {font-family: Georgia, serif; font-size: x-small;}

h1 {color: #000099; }

* Add the following between the **<head>...</head>** tags of all HTML documents that you want to reference the external style sheet.

<link rel="stylesheet" href="external.css" type="text/css">

Example

|  |
| --- |
| p {font-family: georgia, serif; font-size: x-small;}  h1 {color: #009900; }  Note : Save these code as external.css |
| <html>  <head>  <link rel="stylesheet" href="external.css" type="text/css">  </head>  <h1 > WELCOME </h1>  <body>  <p> This is CSS text.</p>  </body>  </html> |

Output :

# WELCOME

Div (short for division) divides the content into individual sections. Each section can then have its own formatting, as specified by the CSS. Div is a block-level container, meaning that there is a line feed after the </div> tag.

Example

|  |
| --- |
| <html>  <head>  <style type = “text/css”>  .large {  color: #00FF00;  font-family:arial;  font-size: 20pt;  }  </style>  </head>  <body>  <div class="large"> This is a DIV sample. </div>This line printed in new line  <div class="large"> This is a DIV sample. </div>  </body>  </html> |

Output

This is a DIV sample.

This line printed in new line

This is a DIV sample.

**2.3 Text and Font Properties**

CSS gives  you  a range of style options to enable formatting of text. These style options can be applied easily on your document. They are explained in this session.

The background of any website is very important. Wrong use of colors in the background of a page won't lend an attractive sight to the text. If you choose a dark color for background and a light color for the text, then the text won't be really legible. Professional websites tend to apply light background with dark text. The CSS background properties allow you to control the background color of an element. Besides, you can also place images in the background. Images can be placed either horizontally or vertically. Different style options are explained below.

**2.3.1 Text**

The most commonly-used CSS properties related to text are :

* [**direction**](http://www.1keydata.com/css-tutorial/text.php#direction)
* [**letter-spacing**](http://www.1keydata.com/css-tutorial/text.php#letterspacing)
* [**line-height**](http://www.1keydata.com/css-tutorial/text.php#lineheight)
* [**text-align**](http://www.1keydata.com/css-tutorial/text.php#textalign)
* [**text-decoration**](http://www.1keydata.com/css-tutorial/text.php#textdecoration)
* [**text-indent**](http://www.1keydata.com/css-tutorial/text.php#textindent)
* [**text-transform**](http://www.1keydata.com/css-tutorial/text.php#texttransform)
* [**word-spacing**](http://www.1keydata.com/css-tutorial/text.php#wordspacing)

**direction**

The **direction** property specifies the text direction. Possible values are :

ltr - left to right

rtl - right to left

For example, with a CSS declaration of,

p { direction:ltr; }

The following HTML,

<p>**LTR Direction**</p>

renders

|  |
| --- |
| **LTR Direction** |

With a CSS declaration of,

p { direction:rtl; }

The following HTML,

<p>**RTL Direction**</p>

renders

|  |
| --- |
| **RTL Direction** |

**letter-spacing**

The **letter-spacing** property specifies the amount of space between characters.

For example, with a CSS declaration of,

p { letter-spacing:8px; }

The following HTML,

<p>8px between letters</p>

renders

|  |
| --- |
| 8px between letters |

**line-height**

The **line-height** property specifies the amount of space between lines.

For example, with a CSS declaration of,

p { line-height:30px; }

The following HTML,

<p>30px between line 1<br>and line 2.</p>

renders

|  |
| --- |
| 30px between line 1 and line 2. |

**text-align**

The **text-align** property specifies how text is justified.

Possible values are:

* **left**: left-justified
* **right**: right-justified
* **center**: text is centered
* **justified**: text is both right- and left-justified

Examples below:

|  |  |
| --- | --- |
| **CSS Declaration** | **Output** |
| p { text-align:left; } | This sentence illustrates what it looks like to be left-justified. |
| p { text-align:right; } | This sentence illustrates what it looks like to be right-justified. |
| p { text-align:center;} | This sentence illustrates what it looks like to be centered. |
| p { text-align:justify; } | This sentence illustrates what it looks like to be fully-justified. |

**text-decoration**

The **text-decoration** property specifies how text is decorated.  
Possible values are:

* **underline**: adds an underline to the text
* **overline**: adds a line on top of the text
* **line-through**: adds a line through the middle of the text.
* **blink**: causes the text to blink.

Examples below:

|  |  |
| --- | --- |
| **CSS Declaration** | **Output** |
| p {   text-decoration:underline; } | An underline example |
| p {   text-decoration:overline; } | An overline example |
| p {  text-decoration:line-through; } | ~~A strikethrough (line-through) example~~ |

**text-indent**

The **text-indent** property specifies how much space to indent before the first line of the text in a block. Both length and percentage can be used.

For example, with a CSS declaration of,

p {   text-indent:150px; }

The following HTML,

<p>This text is indented by 15px at the beginning of the paragraph. Subsequent lines are not indented.</p>

renders

|  |
| --- |
| This text is indented by 150px at the beginning of the paragraph. Subsequent lines are not indented. |

**text-transform**

The **text-transform** property controls how upper and lower cases are displayed.

Possible values are:

* **capitalize**: capitalizes the first letter in a word
* **uppercase**: makes the entire word upper case
* **lowercase**: makes the entire word lower case
* **none**: no transform is performed

For example, if we apply each of the following CSS style to the text "this is a TEXT", we get the following:

|  |  |
| --- | --- |
| **CSS Declaration** | **Output** |
| p {   text-transform:capitalize; } | This is a TEXT |
| p {   text-transform:uppercase; } | this is a text |
| p {   text-transform:lowercase; } | this is a text |

**word-spacing**

The **word-spacing** property controls the amount of space between words.

For example, with a CSS declaration of,

p {   word-spacing:5px; }

The following HTML,

<p>Words here are separated by 5px.</p>

renders

|  |
| --- |
| Words here are separated by 5px. |

**2.3.2 Font**

Common ways of manipulating font properties in CSS include the following:

* [**font-family**](http://www.1keydata.com/css-tutorial/font.php#family)
* [**font-size**](http://www.1keydata.com/css-tutorial/font.php#size)
* [**font-weight**](http://www.1keydata.com/css-tutorial/font.php#weight)
* [**font-style**](http://www.1keydata.com/css-tutorial/font.php#style)
* [**font-variant**](http://www.1keydata.com/css-tutorial/font.php#variant)

Each is described in detail below:

**font-family**

The **font-family** property specifies the type of the font.

Example :

|  |  |
| --- | --- |
| **CSS Declaration** | **Output** |
| p {font-family: verdana;} | Font Family Verdana. |
| p {font-family: arial;} | Font Family Arial. |
| p {font-family: impact;} | Font Family Impact. |

**font-size**

The **font-size** property specifies the size of the font. The size can be numerical (length or percentage), or in text (possible values are "xx-large", "x-large", "large", "medium", "small", "x-small", and "xx-small").

Example :

|  |  |
| --- | --- |
| **CSS Declaration** | **Output** |
| p {font-size:9px;} | Font Size 9px. |
| p {font-size:150%;} | Font Size 150%. |
| p {font-size:0.8cm;} | Font Size 0.8cm. |
| p {font-size:small;} | Font Size small. |
| p {font-size:large;} | Font Size Large. |

**font-weight**

The **font-weight** property specifies the thickness of the font.

Example :

|  |  |
| --- | --- |
| **CSS Declaration** | **Output** |
| p {font-weight: 100;} | This is font weight 100. |
| p {font-weight: 900;} | **This is font weight 900.** |
| p {font-weight: bold;} | **This is bold font weight.** |

**font-style**

The **font-style** property specifies whether the font is italic or oblique.

|  |  |
| --- | --- |
| **CSS Declaration** | **Output** |
| p {font-style: italic;} | *This is font style italics.* |

**2.3.3 Color**

The **color** property allows webmasters to define the color of an element in a CSS style sheet.

This property takes values in 3 forms:

* **Hexadecimal code**
* **RGB**
* **Color name**

The general syntax for the **color** property is as follows:

**Hexadecimal code:**

{color : #XXXXXX;}where X is a hexadecimal code.

**RGB**:

{color:rgb(X,Y,Z); }   where X, Y, and Z are numbers between 0 and 255   
    OR   
 {color:rgb(X%,Y%,Z%); }   where X, Y, and Z are numbers between 0 and 100.

**Color name**:

{color:[color\_name];}

Example :

|  |  |
| --- | --- |
| **CSS Declaration** | **Output** |
| p {color:#FF0000;} | FF0000 is red. |
| p {color:rgb(255,0,255);} | rgb(255,0,255) is pink. |
| p {color:green;} | This is green. |

2.3.4 Background

The background of any website is very important. Wrong use of colors in the background of a page won't lend an attractive sight to the text.

The CSS background properties allow you to control the background color of an element. Besides, you can also place images in the background. Images can be placed either horizontally

or vertically.  
Different style options in context to background are explained below:

**Background Color**

* The background color property adds a background color for an element.
* This property can have the values- color-name, color-RGB, color-hex, and transparent.
* By default background color property value is transparent.

Example :

|  |
| --- |
|  |
| p { background-color : yellow;}                      Lists are a fundamental way of organizing data in a linear way, one item after another.   p { background-color : #887799;}                      Lists are a fundamental way of organizing data in a linear way, one item after another.   p { background-color : rgb(150,200,250);}                       Lists are a fundamental way of organizing data in a linear way, one item after another. |

**Background Images**

* CSS lets you set a background image for both the page and single elements on the page.
* It can have the values- none or url.

In addition, CSS offers several positioning methods for background images. The background-repeat property determines how a specified background image is repeated.

Background repeat property can have the values- repeat, repeat-x, repeat-y, no-repeat.

**repeat** - This will tile the image until the entire page is filled. The background image will be repeated horizontally and vertically.   
**repeat-x** - This will repeat the background image horizontally.   
**repeat-y** - This will repeat the background image vertically.   
**no-repeat**- This will display the background image only once.

Example:

|  |
| --- |
| p{ background-image : url ("C:\Administrator\adminpanel.gif");background-repeat : repeat-x ;} |
| j0299587j0299587  HELLO WORLD |

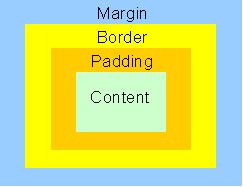
**2.4 Box Properties**

The box model is an important concept in CSS. It dictates how elements are laid out. Box Model Contains different elements, they are Margin, Border and Padding.CSS Margins are nearly identical to the CSS Padding attribute except for one important difference: a margin defines the white space around an HTML element's border, while padding refers to the white space within the border. Setting the actual value of margin is just the same as with padding. CSS Border, allow you to completely customize the borders that appear around HTML elements.

Floating is often used to push an image to one side or another, while having the text of a paragraph wrap around it. This type of usage is often referred to as text wrapping and resembles what you might see in many magazines that have articles which wrap around images of various shapes and sizes. These tags you can probably zip right through this lesson.

**2.4.1 Box Model**

Each element in the document is considered to be a rectangular area, surrounded by a padding, a border and a margin. This is shown in the image given below:



* The innermost component is the content. Padding is applied outside the content area. Border is then applied outside of the padding area. Finally, margin is applied outside of the padding area. Margins specify the relationship between different elements.
* The relevant CSS commands are [margin](http://www.1keydata.com/css-tutorial/margin.php), [border](http://www.1keydata.com/css-tutorial/border.php), and [padding](http://www.1keydata.com/css-tutorial/padding.php). Each one will be descried in more detail below.

**2.4.2 Margin**

As we saw from the box model, margin is the space outside of the border, and is used to determine spacing among the different elements.

In a box, there are four sides. So, we can specify margins up to the 4 sides:

* **margin-top**
* **margin-right**
* **margin-bottom**
* **margin-left**

Example :

|  |
| --- |
| #container {  margin-top:5px;  margin-left:10%;  margin-right:auto;  margin-bottom:20px;  border: 1px solid 000000;  } |

The following HTML

|  |
| --- |
| <div id="container"> This is an example for the margin  </div> |

renders the following:

|  |
| --- |
| This is an example for the margin |

Notice the margin between the box and the top, left, and bottom of the light green area are 5px, 10%, and 20px, respectively.

**2.4.3 Border**

Common ways of specifying border properties in CSS include the following:

* [**border-style**](http://www.1keydata.com/css-tutorial/border.php#borderstyle)
* [**border-width**](http://www.1keydata.com/css-tutorial/border.php#borderwidth)
* [**border-color**](http://www.1keydata.com/css-tutorial/border.php#bordercolor)

**border-style**

The **border-style** property defines the format of the border.

|  |  |
| --- | --- |
| **CSS Declaration** | **Output** |
| p {border-style:solid;} | Solid Border |
| p {border-style:dashed;} | Dashed Border |
| p {border-style:double;} | Double Border |
| p {border-style:dotted;} | Dotted Border |
| p {border-style:groove;} | Groove Border |
| p {border-style:ridge;} | Ridge Border |
| p {border-style:inset;} | Inset Border |
| p {border-style:outset;} | Outset Border |

**border-width**

The **border-width** property specifies the width of the border. The value can be "thin", "medium", "thick", or a numerical width.

|  |  |
| --- | --- |
| **CSS Declaration** | **Output** |
| p {border-width:9px; border-style:solid;} | Border Width 9px |
| p {border-width:medium; border-style:dashed;} | Border Width Medium |

**border-color**

The **border-color** property specifies the color of the border.

Examples below:

|  |  |
| --- | --- |
| **CSS Declaration** | **Output** |
| p {border-color:#0000FF; border-style:solid;} | Border Color Blue |
| p {border-color:red; border-style:dotted;} | Border Color Red |

**2.4.4 Padding**

As we saw from the box model, padding is the space just outside the content area, and just inside the border area. In a box, there are four sides. So, we can specify paddings up to the 4 sides:

* **padding-top**
* **padding-right**
* **padding-bottom**
* **padding-left**

paddings can be specified in 3 ways: length, percentage, or auto.

Example:

|  |
| --- |
| #container {    padding-top:15px;    padding-left:5px;    padding-right:30px;    padding-bottom:40px;    border: 1px solid 000000;  } |

The following HTML

|  |
| --- |
| <div id="container"> This is an example for the padding  </div> |

renders the following:

|  |
| --- |
| This is an example for the padding |

Notice the padding between the box and the top, left, bottom, and right of the light green area are 15px, 5px, 40px, and 30px, respectively.

**2.4.5 Floating**

One of the commonly-seen layout, especially in large websites displaying ads, is wrapping the text around an advertising block. This is accomplished using the **float** property.

The **float** property has three possible values: 'left', 'right', and 'none'. Let's take a look at the following examples:

Given the CSS declaration,

|  |
| --- |
| #leftfloat{ **float:left;**} |

the following HTML,

|  |
| --- |
| <span id="**leftfloat"><**img src="yp.jpg"></span>This example illustrates how float:left affects the appearance of a block. Notice how the image "floats" to the left. |

renders,

|  |
| --- |
| j0157763This example illustrates how float:left affects the appearance of a block. Notice how the image "floats" to the left. |

**2.4.6 Styling List**

Common ways of manipulating font properties in CSS include the following:

* [**list-style-type**](http://www.1keydata.com/css-tutorial/list.php#styletype)
* [**list-style-position**](http://www.1keydata.com/css-tutorial/list.php#styleposition)
* [**list-style-image**](http://www.1keydata.com/css-tutorial/list.php#styleimage)

Each is described in detail below:

**list-style-type**

The **list-style-type** property lets you specify a different type of marker than the default disc. The most commonly used list-style-types are:

* none
* disc
* circle
* square

One may also wish to use ordered character sets. Common ones are:

* upper-latin
* lower-latin
* upper-roman
* lower-roman
* upper-alpha
* lower-alpha

Example:

|  |
| --- |
| <ul style='list-style-type:upper-roman;>   <li>item 1</li>   <li>item 2</li> </ul> |

Output:

1. item 1
2. item 2

**list-style-position**

The **list-style-position** property offers a way for the user to specify whether the marker should be treated as part of the regular text when it comes to formatting. The possible values are 'inside' and 'outside'. 'Outside' is the default value.

Below are examples to illustrate the difference between the two (notice the indentation and how the lines align):

Example:

|  |
| --- |
| <ul style='list-style-position:inside;'>   <li>First one<br>second line    <li>Second one  </ul> |

Output:

* First one

second line

* Second one

Example:

|  |
| --- |
| <ul style='list-style-position:outside;'>   <li>First one<br>second line    <li>Second one  </ul> |

Output:

* First one  
  second line
* Second one

**list-style-image**

The **list-style-image** property is used to specify an image to use for the marker. The syntax is

list-style-image:url([image\_url]);

For example, if our CSS code is

|  |
| --- |
| ul {    list-style-image:url("pumpkin.gif");  } |

the following HTML code

|  |
| --- |
| <ul>   <li>First list for custom marker.    <li>Second list for custom marker.  </ul> |

renders

j0305493First list for custom marker.

j0305493Second list for custom marker.

**2.5 CSS Positioning**

With the knowledge of CSS Positioning you will be able to manipulate the exact position of your HTML elements. Designs that previously required the use of JavaScript or HTML image maps may now be done entirely in CSS. Not only is it easier to code, but it also loads much quicker!

In CSS, each element is given a priority. HTML elements that appear later in the source code than others will have a higher priority by default. If there are two overlapping CSS positioned elements, the element with the higher priority will appear on top of the other.

To manually define a priority, set the *z-index* value. The larger the value, the higher the priority the element will have.

# 2.5.1 Positioning

The term "CSS positioning" refers to using CSS to position elements on your HTML page. CSS allows you to position any element precisely where you want it. You can specify whether you want the element positioned ***relative***to its natural position in the page or ***absolute***based on its parent element.

Absolute positioning can be very useful for creating advanced layouts and cool visual effects such as overlapping elements to present a layered effect.

## Relative Positioning

To perform relative positioning in CSS use **position:relative;** followed by the desired offset from either top, right, bottom or left

Relative positioning changes the position of the HTML element relative to where it normally appears. So "left:20" adds 20 pixels to the element's LEFT position

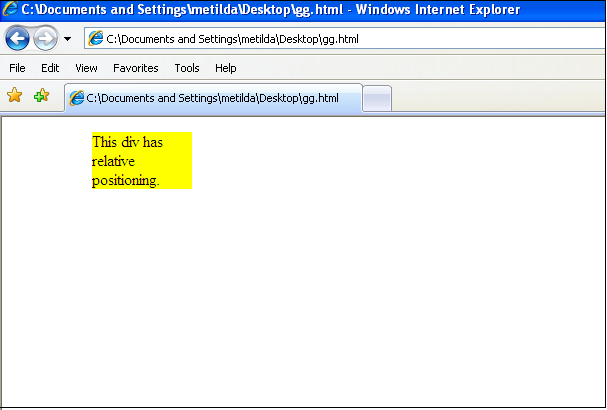
Example:

<div style="**position:relative;left:80px;**background-color:yellow;width:100px;">

This div has relative positioning.

</div>

Output:



This example offsets the element 80 pixels from the left of where it would have been. If we had specified top, it would appear 80 pixels below where it would have been. It's important to note that other elements are not affected by this element's offset..

## Absolute Positioning

An element with **position: absolute** is positioned at the specified coordinates relative to your screen top-left corner.

To perform absolute positioning in CSS use **position:absolute;** followed by the desired offset.

Example :

<div style="**position:absolute;top:100px;left:60px;**background-color:yellow;">

This div is absolutely positioned 100 pixels from the top and 60 pixels from the left of its containing block.

</div>

Output:

