CSS

Cascading Style Sheets (CSS) describe how documents are presented on screens, in print, or perhaps how they are pronounced. W3C has actively promoted the use of style sheets on the Web since the consortium was founded in 1994.

Cascading Style Sheets (CSS) provide easy and effective alternatives to specify various attributes for the HTML tags. Using CSS, you can specify a number of style properties for a given HTML element. Each property has a name and a value, separated by a colon (:). Each property declaration is separated by a semi-colon (;).

Example

First let's consider an example of HTML document which makes use of <font> tag and associated attributes to specify text color and font size:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML CSS</title>
</head>
<body>
<p><font color="green" size="5">Hello, World!</font></p>
</body>
</html>
```

We can re-write above example with the help of Style Sheet as follows:

```
<!DOCTYPE html>
<html>
<head>
<title>HTML CSS</title>
</head>
<body>
<p style="color:green;font-size:24px;">Hello, World!</p>
</body>
</html>
```

You can use CSS in three ways in your HTML document:

- **External Style Sheet**: Define style sheet rules in a separate .css file and then include that file in your HTML document using HTML <link> tag.
- **Internal Style Sheet**: Define style sheet rules in header section of the HTML document using <style> tag.
- **Inline Style Sheet**: Define style sheet rules directly along-with the HTML elements using **style** attribute.

Let's see all the three cases one by one with the help of suitable examples.

External Style Sheet

If you need to use your style sheet to various pages, then its always recommended to define a common style sheet in a separate file. A cascading style sheet file will have extension as **.css** and it will be included in HTML files using <link> tag.

Example

Consider we define a style sheet file **style.css** which has following rules:

```
.red{

    color: red;

}

.thick{

    font-size:20px;

}

.green{

    color:green;

}
```

Here we defined three CSS rules which will be applicable to three different classes defined for the HTML tags. I suggest you should not bother about how these rules are being defined because you will learn them while studying CSS. Now let's make use of the above external CSS file in our following HTML document:

```
 <!DOCTYPE html>

 <html>

 <head>

 <title>HTML External CSS</title>

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

 </head>

 <body>

<p class="red">This is red</p>

 <p class="thick">This is thick</p>

 <p class="green">This is green</p>

 <p class="thick green">This is thick and green</p>

 </body>

 </html>
```

Internal Style Sheet

If you want to apply Style Sheet rules to a single document only, then you can include those rules in header section of the HTML document using <style> tag.

Rules defined in internal style sheet overrides the rules defined in an external CSS file.

Example

Let's re-write above example once again, but here we will write style sheet rules in the same HTML document using <style> tag:

```
<!DOCTYPE html>

<html>

<head>

<title>HTML Internal CSS</title>

<style type="text/css">

.red{

    color: red;

}

.thick{

    font-size:20px;

}

    .green{ color:green;

}

</style>

</head>

<body>

<p class="red">This is red</p>

<p class="thick">This is thick</p>

<p class="green">This is green</p>

<p class="thick green">This is thick and green</p>

</body>

</html>
```

Inline Style Sheet

You can apply style sheet rules directly to any HTML element using **style** attribute of the relevant tag. This should be done only when you are interested to make a particular change in any HTML element only.

Rules defined inline with the element overrides the rules defined in an external CSS file as well as the rules defined in <style> element.

Example

Let's re-write above example once again, but here we will write style sheet rules along with the HTML elements using **style** attribute of those elements.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML Inline CSS</title>
</head>
<body>
<p style="color:red;">This is red</p>
<p style="font-size:20px;">This is thick</p>
<p style="color:green;">This is green</p>
<p style="color:green;font-size:20px;">This is thick and green</p>
</body>
</html>
```

SYNTAX

CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts:

- **Selector:** A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- **Property:** A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border*, etc.
- **Value:** Values are assigned to properties. For example, *color* property can have the value either *red* or *#F1F1F1* etc.

You can put CSS Style Rule Syntax as follows:

```
selector { property: value }
```

Example: You can define a table border as follows:

```
table{ border :1px solid #C00; }
```

Here table is a selector and border is a property and the given value *1px solid #C00* is the value of that property.

You can define selectors in various simple ways based on your comfort. Let me put these selectors one by one.

The TypeSelectors

This is the same selector we have seen above. Again, one more example to give a color to all level 1 headings:

```
h1 {
   color: #36CFFF;
}
```

The UniversalSelectors

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type:

```
* {
  color: #000000;
}
```

This rule renders the content of every element in our document in black.

The DescendantSelectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, the style rule will apply to <em> element only when it lies inside the <ul> tag.

```
ul em {
  color: #000000;
}
```

The ClassSelectors

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

```
.black {
  color: #000000;
}
```

This rule renders the content in black for every element with class attribute set to *black* in our document. You can make it a bit more particular. For example:

```
h1.black {
  color: #000000;
}
```

This rule renders the content in black for only <h1> elements with class attribute set to *black*.

You can apply more than one class selectors to a given element. Consider the following example:

```
<p class="center bold">
This para will be styled by the classes center and bold.
</p>
```

The ID Selectors

You can define style rules based on the *id* attribute of the elements. All the elements having that *id* will be formatted according to the defined rule.

```
#black {
  color: #000000;
}
```

This rule renders the content in black for every element with *id* attribute set to *black* in our document. You can make it a bit more particular. For example:

```
h1#black {
  color: #000000;
}
```

This rule renders the content in black for only <h1> elements with *id* attribute set to *black*.

The true power of *id* selectors is when they are used as the foundation for descendant selectors. For example:

```
#black h2 {
  color: #000000;
}
```

In this example, all level 2 headings will be displayed in black color when those headings will lie within tags having *id* attribute set to *black*.

The ChildSelectors

You have seen the descendant selectors. There is one more type of selector, which is very similar to descendants but have different functionality. Consider the following example:

```
body > p { color:
  #000000;
}
```

This rule will render all the paragraphs in black if they are a direct child of the <body> element. Other paragraphs put inside other elements like <div> or <td> would not have any effect of this rule.

TheAttribute Selectors

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of *text*:

```
input[type="text"]{
  color: #000000;
}
```

The advantage to this method is that the <input type="submit" /> element is unaffected, and the color applied only to the desired text fields.

Grouping Selectors

You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example:

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

This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

You can combine the various *class* selectors together as shown below:

```
#content, #footer
 { position: absolute;
left: 510px;
width: 200px;
}
```

MEASUREMENTS:

We have listed out all the CSS Measurement Units along with proper Examples:

Unit	Description	Example
%	Defines a measurement as a percentage relative to another value, typically an enclosing element.	p {font-size: 16pt; line-height: 125%;}
Cm	Defines a measurement in centimeters.	div {margin-bottom: 2cm;}

em	A relative measurement for the height of a font in em spaces. Because an em unit is equivalent to the size of a given font, if you assign a font to 12pt, each "em" unit would be 12pt; thus, 2em would be 24pt.	p {letter-spacing: 7em;}
Ex	This value defines a measurement relative to a font's x-height. The x-height is determined by the height of the font's lowercase letter x.	p {font-size: 24pt; line-height: 3ex;}
In	Defines a measurement in inches.	p {word-spacing: .15in;}
mm	Defines a measurement in millimeters.	p {word-spacing: 15mm;}

BACKGROUND PROPERTIES:

You can set the following background properties of an element:The **background-color** property is used to set the background color of an element.

- The **background-image** property is used to set the background image of an element.
- The **background-repeat** property is used to control the repetition of an image in the background.
- The **background-position** property is used to control the position of an image in the background.
- The **background-attachment** property is used to control the scrolling of an image in the background.
- The **background** property is used as a shorthand to specify a number of other background properties.

Set the Background Color

Following is the example, which demonstrates how to set the background color for an element.

```
<p style="background-color:yellow;">
This text has a yellow background color.
</p>
```

Set the Background Image

```
<table style="background-image:url(/images/pattern1.gif);">
<tr><td>
This table has background image set.
</td></tr>
</table>
```

Repeat the Background Image

The following example demonstrates how to repeat the background image if an image is small. You can use *no-repeat* value for the *background-repeat* property if you don't want to repeat an image. In this case, the image will display only once.

By default, the *background-repeat* property will have a *repeat* value.

```
<table style="background-image:url(/images/pattern1.gif);
              background-repeat: repeat;">
<tr><td>
This table has background image which repeats multiple times.
</td></tr>
</table>
```

The following example which demonstrates how to repeat the background image vertically.

```
<table style="background-image:url(/images/pattern1.gif);
              background-repeat: repeat-y;">
<tr><td>
This table has background image set which will repeat vertically.
</td></tr>
</table>
```

The following example demonstrates how to repeat the background image horizontally.

```
<table style="background-image:url(/images/pattern1.gif);
              background-repeat: repeat-x;">
<tr><td>
This table has background image set which will repeat horizontally.
</td></tr>
</table>
```

Set the Background Image Position

The following example demonstrates how to set the background image position 100 pixels away from the left side.

```
<table style="background-image:url(/images/pattern1.gif);
              background-position:100px;">
<tr><td>
Background image positioned 100 pixels away from the left.
</td></tr>
</table>
```

The following example demonstrates how to set the background image position 100 pixels away from the left side and 200 pixels down from the top.

```
<table style="background-image:url(/images/pattern1.gif);
              background-position:100px 200px;">
<tr><td>
This table has background image positioned 100
pixels away from the left and 200 pixels from the top.
</td></tr>
</table>
```

FONT PROPERTIES:

You can set the following font properties of an element:

- The **font-family** property is used to change the face of afont.
- The **font-style** property is used to make a font italic or oblique.

- The **font-variant** property is used to create a small-caps effect.
- The **font-weight** property is used to increase or decrease how bold or light a font appears.
- The **font-size**property is used to increase or decrease the size of a font.
- The **font** property is used as shorthand to specify a number of other font properties.

Font family:

```
<p style="font-family:georgia,garamond,serif;">
This text is rendered in either georgia, garamond, or the default
serif font depending on which font  you have at your system.
</p>
```

Font style:

```
<p style="font-style:italic;">
```

Font Variant:

```
<p style="font-variant:small-caps;"> This
text will be rendered as small caps
</p>
```

Font weight:

```
<p style="font-weight:bold;">
This font is bold.
</p>
<p style="font-weight:bolder;">
This font is bolder.
</p>
<p style="font-weight:900;">
This font is 900 weight.
```

```
</p>
```

Font size:

```
<p style="font-size:20px;">
This font size is 20 pixels
</p>
<p style="font-size:small;">
This font size is small
</p>
<p style="font-size:large;">
This font size is large
</p>
```

Font stretch:

```
<p style="font-stretch:ultra-expanded;">
If this doesn't appear to work, it is likely that
your computer doesn't have a condensed or expanded
version of the font being used.
</p>
```

TEXT PROPERTIES:

- The **color** property is used to set the color of a text.
- The **direction** property is used to set the text direction.
- The **letter-spacing** property is used to add or subtract space between the letters that make up a word.
- The **word-spacing** property is used to add or subtract space between the words of a sentence.
- The **text-indent** property is used to indent the text of a paragraph.
- The **text-align** property is used to align the text of a document.
- The **text-decoration** property is used to underline, overline, and strikethrough text.

- The **text-transform** property is used to capitalize text or convert text to uppercase or lowercase letters.
- The **white-space** property is used to control the flow and formatting of text.
- The **text-shadow**property is used to set the text shadow around a text.

Text color:

```
<p style="color:red;">
This text will be written in red.
</p>
```

Text direction:

```
<p style="direction:rtl;">
This text will be renedered from right to left
</p>
```

Space between the characters:

```
<p style="letter-spacing:5px;">
This text is having space between letters.
</p>
```

Space between the words:

```
<p style="word-spacing:5px;">
This text is having space between words.
</p>
```

Text Indentation:

```
<p style="text-indent:1cm;">
```

```
This text will have first line indented by 1cm and
this line will remain at its actual position this
is done by CSS text-indent property.
</p>
```

Text alignment:

```
<p style="text-align:right;">
This will be right aligned.
</p>
<p style="text-align:center;">
This will be center aligned.
</p>
<p style="text-align:left;">
This will be left aligned.
</p>
```

Text Decoration:

```
<p style="text-decoration:underline;">
This will be underlined
</p>
<p style="text-decoration:line-through;"> This
will be striked through.
</p>
<p style="text-decoration:overline;"> This
will have a over line.
</p>
<p style="text-decoration:blink;">
This text will have blinking effect
</p>
```

Text transform:

```
<p style="text-transform:capitalize;">
This will be capitalized
</p>
<p style="text-transform:uppercase;"> This
will be in uppercase
</p>
<p style="text-transform:lowercase;"> This
will be in lowercase
</p>
```

Text Shadowing:

```
<p style="text-shadow:4px 4px 8px blue;">
If your browser supports the CSS text-shadow property,
this text will have a  blue shadow.</p>
```

IMAGE PROPERTIES:

CSS plays a good role to control image display. You can set the following image properties using CSS.

- The **border**property is used to set the width of an image border.
- The **height**property is used to set the height of an image.
- The **width** property is used to set the width of an image.
- The **-moz-opacity** property is used to set the opacity of an image.

Image border:

```
<img style="border:0px;" src="/images/css.gif" />
<br />
```

```
<img style="border:3px dashed red;" src="/images/css.gif" />
```

Image height:

```
<img style="border:1px solid red; height:100px;"
        src="/images/css.gif" />
<br />
<img style="border:1px solid red; height:50%;"
        src="/images/css.gif" />
```

Image Width:

```
<img style="border:1px solid red; width:100px;"
        src="/images/css.gif" />
<br />
<img style="border:1px solid red; width:100%;"
        src="/images/css.gif" />
```

LINK PROPERTIES:

To set different properties of a hyper link using CSS. You can set the following properties of a hyperlink:

We will revisit the same properties when we will discuss Pseudo-Classes of CSS.

- The **:link** signifies unvisited hyperlinks.
- The **:visited** signifies visited hyperlinks.
- The **:hover** signifies an element that currently has the user's mouse pointer hovering over it.
- The **:active** signifies an element on which the

user is currently clicking. Usually, all these properties are kept in the header part of the HTML document.

Remember a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective. Also, a:active MUST come after a:hover in the CSS definition as follows:

```
<style type="text/css">
a:link {color: #000000}
a:visited {color: #006600}
a:hover {color: #FFCC00}
a:active {color: #FF00CC}
</style>
```

Set the color of the link:

```
<style type="text/css">
a:link {color:#000000}
</style>
```

Set color for visited link:

```
<style type="text/css">
a:visited {color: #006600}
</style>
<a href="/html/index.htm">Click this link</a>
```

Set color when hover:

```
<style type="text/css">
a:hover {color: #FFCC00}
</style>
<a href="/html/index.htm">Bring Mouse Here</a>
```

Set color for active link:

```
<style type="text/css">
a:active {color: #FF00CC}
</style>
<a href="/html/index.htm">Click This Link</a>
```

TABLE PROPERTIES:

You can set the following properties of a table:

- The **border-collapse** specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.
- The **border-spacing** specifies the width that should appear between table cells.
- The **caption-side** captions are presented in the <caption> element. By default, these are rendered above the table in the document. You use the *caption-side* property to control the placement of the table caption.
- The **empty-cells** specifies whether the border should be shown if a cell is empty.
- The **table-layout** allows browsers to speed up the layout of a table by using the first width properties it comes across for the rest of a column rather than having to load the whole table before rendering it.

Border collapse:

```
<style type="text/css">
table.one    {border-collapse:collapse;}
table.two    {border-collapse:separate;}
td.a {
      border-style:dotted;
      border-width:3px;
      border-color:#000000;
      padding: 10px;
}
```

```
td.b {border-style:solid;
      border-width:3px;
      border-color:#333333;
      padding:10px;
}
</style>
<table class="one">
<caption>Collapse Border Example</caption>
<tr><td class="a"> Cell A Collapse Example</td></tr>
<tr><td class="b"> Cell B Collapse Example</td></tr>
</table>
<br />
<table class="two">
<caption>Separate Border Example</caption>
<tr><td class="a"> Cell A Separate Example</td></tr>
<tr><td class="b"> Cell B Separate Example</td></tr>
</table>
```

Border spacing:

The border-spacing property specifies the distance that separates the adjacent cells' borders. It can take either one or two values; these should be units of length.

If you provide one value, it applies to both vertical and horizontal borders. Or you can specify two values, in which case, the first refers to the horizontal spacing and the second to the vertical spacing:

NOTE: Unfortunately, this property does not work in Netscape 7 or IE 6.

```
<style type="text/css">
table.one {
   border-collapse:separate;
   width:400px;
   border-spacing:10px;
}
table.two {
    border-collapse:separate;
    width:400px;
    border-spacing:10px 50px;
}
</style>
<table class="one" border="1">
<caption>Separate Border Example with border-spacing</caption>
<tr><td> Cell A Collapse Example</td></tr>
<tr><td> Cell B Collapse Example</td></tr>
</table>
<br />
<table class="two" border="1">
<caption>Separate Border Example with border-spacing</caption>
<tr><td> Cell A Separate Example</td></tr>
<tr><td> Cell B Separate Example</td></tr>
</table>
```

Caption property:

The caption-side property allows you to specify where the content of a <caption> element should be placed in relationship to the table. The table that follows lists the possible values.

This property can have one of the four values *top, bottom, left*, or *right*. The following example uses each value.

NOTE: These properties may not work with your IE Browser.

```
<style type="text/css">
caption.top {caption-side:top}
caption.bottom {caption-side:bottom}
caption.left {caption-side:left}
            caption.right {caption-side:right}
            </style>

            <table style="width:400px; border:1px solid black;">
            <caption class="top">
```

```
This caption will appear at the top
</caption>
<tr><td > Cell A</td></tr>
<tr><td > Cell B</td></tr>
</table>
<br />

<table style="width:400px; border:1px solid black;">
<caption class="bottom">
This caption will appear at the bottom
</caption>
<tr><td > Cell A</td></tr>
<tr><td > Cell B</td></tr>
</table>
<br />

<table style="width:400px; border:1px solid black;">
<caption class="left">
This caption will appear at the left
</caption>
<tr><td > Cell A</td></tr>
<tr><td > Cell B</td></tr>
</table>
<br />

<table style="width:400px; border:1px solid black;">
<caption class="right">
 This caption will appear at the right
 </caption>
 <tr><td > Cell A</td></tr>
 <tr><td > Cell B</td></tr>
 </table>
```

Empty cell property:

The empty-cells property indicates whether a cell without any content should have a border displayed.

This property can have one of the three values - *show, hide*,or *inherit*.

Here is the empty-cells property used to hide borders of empty cells in the <table> element.

```
<style type="text/css">
table.empty{
    width:350px;
    border-collapse:separate;
    empty-cells:hide;
}
td.empty{
    padding:5px;
    border-style:solid;
    border-width:1px;
    border-color:#999999;
}
</style>
<table class="empty">
<tr>
<th></th>
<th>Title one</th>
<th>Title two</th>
</tr>
<tr>
<th>Row Title</th>
<td class="empty">value</td>
<td class="empty">value</td>
</tr>
```

```
<tr>
<th>Row Title</th>
<td class="empty">value</td>
<td class="empty"></td>
</tr>
</table>
```

Table layout property:

The table-layout property is supposed to help you control how a browser should render or lay out a table.

This property can have one of the three values: *fixed, auto*, or *inherit*. The following example shows the difference between these properties.

NOTE: This property is not supported by many browsers, so do not rely on this property.

```
<style type="text/css">
table.auto
{
table-layout: auto
}
table.fixed
{
table-layout: fixed
}
</style>
<table class="auto" border="1" width="100%">
<tr>
<td width="20%">1000000000000000000000000000</td>
<td width="40%">10000000</td>
<td width="40%">100</td>
</tr>
</table>
<br />
```

```
<table class="fixed" border="1" width="100%">
<tr>
<td width="20%">1000000000000000000000000000</td>
<td width="40%">10000000</td>
<td width="40%">100</td>
</tr>
</table>
```

Border property:

There are three properties of a border you can change:

- The **border-color** specifies the color of a border.
- The **border-style** specifies whether a border should be solid, dashed line, double line, or one of the other possible values.
- The **border-width** specifies the width of a border.

Border Property:

```
<style type="text/css">
p.example1{
   border:1px solid;
   border-bottom-color:#009900; /* Green */
   border-top-color:#FF0000;     /* Red */
   border-left-color:#330000;    /* Black */
   border-right-color:#0000CC;   /* Blue */
}
p.example2{ border:1px
   solid;
border-color:#009900;
}
</style>
<p class="example1">
```

```
This example is showing all borders in different colors.

</p>

<p class="example2">

This example is showing all borders in green color only.

</p>
```

Border style properties:

The border-style property allows you to select one of the following styles of border:

- **none:** No border. (Equivalent of border-width:0;)
- **solid:** Border is a single solid line.
- **dotted:** Border is a series of dots.
- **dashed:** Border is a series of short lines.
- **double:** Border is two solid lines.
- **groove:** Border looks as though it is carved into the page.
- **ridge:** Border looks the opposite of groove.
- **inset:** Border makes the box look like it is embedded in the page.
- **outset:** Border makes the box look like it is coming out of the canvas.
- **hidden:** Same as none, except in terms of border-conflict resolution for table elements.

You can individually change the style of the bottom, left, top, and right borders of an element using the following properties:

- **border-bottom-style** changes the style of bottom border.
- **border-top-style** changes the style of top border.
- **border-left-style** changes the style of left border.
- **border-right-style** changes the style of right border. The following example shows all these border styles:

```
<p style="border-width:4px; border-
style:none;"> This is a border with
```

```
                    none width.
                    </p>
                    <p style="border-width:4px; border-
                    style:solid;"> This is a solid
                    border.
                    </p>
                    <p style="border-width:4px; border-
                    style:dashed;"> This is a dahsed
                    border.
                    </p>
                    <p style="border-width:4px; border-
                    style:double;"> This is a double
                    border.
                    </p>
                    <p style="border-width:4px; border-
                    style:groove;"> This is a groove
                    border.
                    </p>
                    <p style="border-width:4px;
                    border-style:ridge"> This is
                    aridgeborder.
                    </p>
                    <p style="border-width:4px; border-
                    style:inset;"> This is a inset
                    border.
                    </p>
                    <p style="border-width:4px; border-
                    style:outset;"> This is a outset
                    border.
                    </p>
                    <p style="border-width:4px; border-style:hidden;">
                     This is a hidden border.
</p>
```

```
<p style="border-width:4px;
           border-top-style:solid; border-
           bottom-style:dashed; border-
           left-style:groove; border-
           right-style:double;">
This is a a border with four different styles.
</p>
```

Border width:

The border-width property allows you to set the width of an element borders. The value of this property could be either a length in px, pt, or cm, or it should be set to *thin, medium*, *or thick.*

You can individually change the width of the bottom, top, left, and right borders of an element using the following properties:

- **border-bottom-width** changes the width of bottom border.
- **border-top-width** changes the width of top border.
- **border-left-width** changes the width of left border.
- **border-right-width** changes the width of right border. The following example shows all these border width:

```
<p style="border-width:4px; border-style:solid;">
This is a solid border whose width is 4px.
</p>
<p style="border-width:4pt; border-style:solid;">
This is a solid border whose width is 4pt.
</p>
<p style="border-width:thin; border-style:solid;"> This
is a solid border whose width is thin.
</p>
<p style="border-width:medium; border-style:solid;">
This is a solid border whose width is medium;
</p>
```

```
 <p style="border-width:thick; border-style:solid;"> This
 is a solid border whose width is thick.
 </p>
 <p style="border-bottom-width:4px;
              border-top-width:10px;
              border-left-width: 2px;
              border-right-width:15px;
              border-style:solid;">
 This is a a border with four different width.
 </p>
```

MARGINS:

We have the following properties to set an element margin.

- The **margin** specifies a shorthand property for setting the margin properties in one declaration.
- The **margin-bottom**specifies the bottom margin of an element.
- The **margin-top**specifies the top margin of an element.
- The**margin-left** specifies the left margin of an element.
- The **margin-right** specifies the right margin of an element. Now, we will see how to use these properties with examples.

MARGIN PROPERTY:

```
 <style type="text/css"> p
 {margin: 15px}
 all four margins will be 15px

 p {margin: 10px 2%}
top and bottom margin will be 10px, left and right
margin will be 2% of the total width of the
document
```

```
<p style="margin: 15px; border:1px solid black;"> all
four margins will be 15px
</p>


<p style="margin:10px 2%; border:1px solid black;">
top and bottom margin will be 10px, left and right margin will be 2% of the
total width of the document.
</p>


<p style="margin: 10px 2% –10px; border:1px solid black;">
top margin will be 10px, left and right margin will be 2% of the total width
of the document, bottom margin will be –10px
</p>


<p style="margin: 10px 2% –10px auto; border:1px solid black;">
top margin will be 10px, right margin will be 2% of the total width of the
document, bottom margin will be –10px, left margin will be set by the
browser
</p>
```

LISTS:

We have the following five CSS properties, which can be used to control lists:

- The **list-style-type** allows you to control the shape or appearance of the marker.
- The **list-style-position** specifies whether a long point that wraps to a second line should align with the first line or start underneath the start of the marker.
- The **list-style-image** specifies an image for the marker rather than a bullet point or number.
- The **list-style** serves as shorthand for the preceding properties.
- The **marker-offset** specifies the distance between a marker and the text in the list.

Now we will see how to use these properties with examples.

List style type:

```
<ul style="list-style-type:circle;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ul>

<ul style="list-style-type:square;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ul>

<ol style="list-style-type:decimal;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ol>

<ol style="list-style-type:lower-alpha;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ol>

<ol style="list-style-type:lower-roman;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ol>
```

List type position:

```
<ul style="list-style-type:circle; list-stlye-position:outside;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ul>

<ul style="list-style-type:square;list-style-position:inside;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ul>

<ol style="list-style-type:decimal;list-stlye-position:outside;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ol>

<ol style="list-style-type:lower-alpha;list-style-position:inside;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ol>
```

List style property:

```
<ul style="list-style: inside square;">
<li>Maths</li>
<li>Social Science</li>
```

```
<li>Physics</li>
</ul>

<ol style="list-style: outside upper-alpha;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ol>
```

PADDING:

Padding property allows you to specify how much space should appear between the content of an element and its border:

The value of this attribute should be either a length, a percentage, or the word *inherit*. If the value is *inherit*, it will have the same padding as its parent element. If a percentage is used, the percentage is of the containing box.

The following CSS properties can be used to control lists. You can also set different values for the padding on each side of the box using the following properties:

- The **padding-bottom** specifies the bottom padding of anelement.
- The **padding-top**specifies the top padding of an element.
- The**padding-left** specifies the left padding of an element.
- The **padding-right** specifies the right padding of anelement.
- The **padding**serves as shorthand for the preceding properties. Now, we will see how to use these properties with examples.

```
<p style="padding-bottom: 15px; border:1px solid black;"> This
is a paragraph with a specified bottom padding
</p>

<p style="padding-bottom: 5%; border:1px solid black;">
```

```
This is another paragraph with a specified bottom padding in percent
</p>
```

NOTE: Similarly it will be for all the other sides.

DIMENSONS:

We have the following properties that allow you to control the dimensions of a box.

- The **height**property is used to set the height of a box.
- The **width** property is used to set the width of abox.
- The **line-height** property is used to set the height of a line oftext.
- The **max-height** property is used to set a maximum height that a box can be.
- The **min-height** property is used to set the minimum height that a box can be.
- The **max-width** property is used to set the maximum width that a box can be.
- The **min-width** property is used to set the minimum width that a box can be.

SCROLLBARS:

CSS provides a property called *overflow*, which tells the browser what to do if the box's contents is larger than the box itself. This property can take one of the following values

```
<style type="text/css">
 .scroll{
     display:block;  border:
     1px solid red;
     padding:5px;
     margin-top:5px;
     width:300px;
```

```
    height:50px;
    overflow:scroll;
    }
.auto{
    display:block;  border:
    1px solid red;
    padding:5px;
    margin-top:5px;
    width:300px;
    height:50px;
    overflow:auto;
    }
</style>
<p>Example of scroll value:</p>
<div class="scroll">
I am going to keep lot of content here just to show you
how scrollbars works if there is an overflow in an
element box. This provides your horizontal as well as
vertical scrollbars.
</div>
<br />
<p>Example of auto value:</p>
<div class="auto">
I am going to keep lot of content here just to show you
how scrollbars works if there is an overflow in an
element box. This provides your horizontal as well as
vertical scrollbars.
</div>
```

Relative positioning:

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.

You can use two values *top* and *left* along with the *position* property to move an HTML element anywhere in an HTML document.

- Move Left - Use a negative value for *left*.
- Move Right - Use a positive value for *left*.
- Move Up - Use a negative value for *top*.
- Move Down - Use a positive value for *top*.

NOTE: You can use the *bottom* or *right* values as well in the same way as *top* and *left*.

Here is an example:

```
<div style="position:relative;left:80px;top:2px;
            background-color:yellow;">
This div has relative positioning.
</div>
```

Absolute Positioning:

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

You can use two values *top* and *left* along with the *position* property to move an HTML element anywhere in HTML document.

- Move Left - Use a negative value for *left*.
- Move Right - Use a positive value for *left*.
- Move Up - Use a negative value for *top*.
- Move Down - Use a positive value for *top*.

NOTE: You can use *bottom* or *right* values as well in the same way as top and left.

Here is an example:

```
<div style="position:absolute;left:80px;top:20px;
            background-color:yellow;">
This div has absolute positioning.
</div>
```

Fixed Positioning:

Fixed positioning allows you to fix the position of an element to a particular spot on the page, regardless of scrolling. Specified coordinates will be relative to the browser window.

You can use two values *top* and *left* along with the *position* property to move an HTML element anywhere in the HTML document.

- Move Left - Use a negative value for *left*.
- Move Right - Use a positive value for *left*.
- Move Up - Use a negative value for *top*.
- Move Down - Use a positive value for *top*.

NOTE: You can use *bottom* or *right* values as well in the same way as *top* and *left*.

```
<div style="position:fixed;left:80px;top:20px;
            background-color:yellow;">
This div has fixed positioning.
</div>
```

PSEUDO CLASSES:

Pseudo-classes are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-classes is as follows:

```
selector:pseudo-class {property: value}
```

CSS classes can also be used with pseudo-classes:

```
selector.class:pseudo-class {property: value}
```

The most commonly used pseudo-classes are as follows:

Value	Description
:link	Use this class to add special style to an unvisited link.

:visited	Use this class to add special style to a visited link.
:hover	Use this class to add special style to an element when you mouse over it.
:active	Use this class to add special style to an active element.
:focus	Use this class to add special style to an element while the element has focus.
:first-child	Use this class to add special style to an element that is the first child of some other element.
:lang	Use this class to specify a language to use in a specified element.

PSEUDO ELEMENTS:

pseudo-elements are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-element is as follows:

```
selector:pseudo-element {property: value}
```

CSS classes can also be used with the pseudo-elements:

```
selector.class:pseudo-element {property: value}
```

The most commonly used pseudo-elements are as follows:

Value	Description
:first-line	Use this element to add special styles to the first line of the text in a selector.
:first-letter	Use this element to add special style to the first letter of the text in a selector.

:before	Use this element to insert some content before an element.
:after	Use this element to insert some content after an element.