



Before you begin, it's important that you know Windows or Unix. A working knowledge of Windows or Unix makes it much easier to learn HTML.

You should be familiar with:

- Basic word processing using any text editor.
- How to create directories and files.
- How to navigate through different directories.
- Basic understanding on internet browsing using a browser like Internet Explorer or Firefox etc.
- Basic understanding on developing simple Web Pages using HTML or XHTML.

If you are new to HTML and XHTML then I would suggest you to go through our [HTML Tutorial](#) or [XHTML Tutorial](#). Anyone of HTML or XHTML is enough to proceed.

## What is CSS?

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

## Advantages of CSS:

- **CSS saves time** - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance** - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** - CSS has a much wider array of attributes than HTML so you can give far better look to your HTML page in comparison of HTML attributes.

- **Multiple Device Compatibility** - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** - Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

## Who Creates and Maintains CSS?

CSS is created and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called specifications. When a specification has been discussed and officially ratified by W3C members, it becomes a recommendation.

These ratified specifications are called recommendations because the W3C has no control over the actual implementation of the language. Independent companies and organizations create that software.

**NOTE:** The World Wide Web Consortium, or W3C is a group that makes recommendations about how the Internet works and how it should evolve.

## CSS Versions:

Cascading Style Sheets, level 1 (CSS1) was came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags.

CSS2 was became a W3C recommendation in May 1998 and builds on CSS1. This version adds support for media-specific style sheets e.g. printers and aural devices, downloadable fonts, element positioning and tables.

## About this Tutorial:

This tutorial covers both versions CSS1 and CSS2 and teaches you CSS starting from basic concepts to advanced concepts. So now start from next chapter and finish it till end to become master in CSS.

## CSS Syntax – Selectors

A 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 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* or *border* etc.
- **Value:** Values are assigned to properties. For example *color* property can have 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 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 Type Selectors:

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 Universal Selectors:

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 Descendant Selectors:

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, style rule will apply to `<em>` element only when it lies inside `<ul>` tag.

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

## The Class Selectors:

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 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 only when those headings will lie with in tags having id attribute set to *black*.

### The Child Selectors:

You have seen descendant selectors. There is one more type of selectors 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 direct child of <body> element. Other paragraphs put inside other elements like <div> or <td> etc. would not have any effect of this rule.

### The Attribute Selectors:

You can also apply styles to HTML elements with particular attributes. The style rule below will match all input elements that has 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.

There are following rules applied to attribute selector.

- **p[lang]** - Selects all paragraph elements with a *lang* attribute.
- **p[lang="fr"]** - Selects all paragraph elements whose *lang* attribute has a value of exactly "fr".
- **p[lang~="fr"]** - Selects all paragraph elements whose *lang* attribute contains the word "fr".
- **p[lang|= "en"]** - Selects all paragraph elements whose *lang* attribute contains values that are exactly "en", or begin with "en-".

## Multiple Style Rules:

You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example:

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

Here all the property and value pairs are separated by a **semi colon (;)**. You can keep them in a single line or multiple lines. For better readability we keep them into separate lines.

For a while don't bother about the properties mentioned in the above block. These properties will be explained in coming chapters and you can find complete detail about properties in [CSS References](#).

## 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 various *class* selectors together as shown below:

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

## CSS Inclusion - Associating Styles

There are four ways to associate styles with your HTML document. Most commonly used methods are inline CSS and External CSS.

## Embedded CSS - The <style> Element:

You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document. Here is the generic syntax:

```
<head>
<style type="text/css" media="...">
Style Rules
.....
</style>
</head>
```

### Attributes:

Attributes associated with <style> elements are:

Attribute	Value	Description
type	text/css	Specifies the style sheet language as a content-type (MIME type). This is required attribute.
media	screen tty tv projection handheld print braille aural all	Specifies the device the document will be displayed on. Default value is <i>all</i> . This is optional attribute.

### Example:

Following is the example of embed CSS based on above syntax:

```
<head>
<style type="text/css" media="all">
h1{
color: #36C;
}
</style>
</head>
```

## Inline CSS - The *style* Attribute:

You can use *style* attribute of any HTML element to define style rules. These rules will be applied to that element only. Here is the generic syntax:

```
<element style="...style rules....">
```

### Attributes:

Attribute	Value	Description
style	style rules	The value of <i>style</i> attribute is a combination of style declarations separated by semicolon (;).

### Example:

Following is the example of inline CSS based on above syntax:

```
<h1 style ="color:#36C;"> This is inline CSS </h1>
```

This will produce following result:

**This is inline CSS**

## External CSS - The <link> Element:

The <link> element can be used to include an external stylesheet file in your HTML document.

An external style sheet is a separate text file with .css extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.



Here is the generic syntax of including external CSS file:

```
<head>
<link type="text/css" href="..." media="..." />
</head>
```

### Attributes:

Attributes associated with <style> elements are:

Attribute	Value	Description
type	text/css	Specifies the style sheet language as a content-type (MIME type). This attribute is required.
href	URL	Specifies the style sheet file having Style rules. This attribute is a required.
media	screen tty tv projection handheld print braille aural all	Specifies the device the document will be displayed on. Default value is <i>all</i> . This is optional attribute.

### Example:

Consider a simple style sheet file with a name *mystyle.css* having the following rules:

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

Now you can include this file *mystyle.css* in any HTML document as follows:

```
<head>
<link type="text/css" href="mystyle.css" media="all" />
</head>
```

## Imported CSS - @import Rule:

@import is used to import an external stylesheet in a manner similar to the <link> element. Here is the generic syntax of @import rule.

```
<head>
<@import "URL";
</head>
```

Here URL is the URL of the style sheet file having style rules. You can use another syntax as well:

```
<head>
<@import url("URL");
</head>
```

### Example:

Following is the example showing you how to import a style sheet file into HTML document:

```
<head>
@import "mystyle.css";
</head>
```

## CSS Rules Overriding:

We have discussed four ways to include style sheet rules in an HTML document. Here is the rule to override any Style Sheet Rule.

- Any inline style sheet takes highest priority. So it will override any rule defined in <style>...</style> tags or rules defined in any external style sheet file.
- Any rule defined in <style>...</style> tags will override rules defined in any external style sheet file.
- Any rule defined in external style sheet file takes lowest priority and rules defined in this file will be applied only when above two rules are not applicable.

## Handling old Browsers:

There are still many old browsers who do not support CSS. So we should take care while writing our Embedded CSS in an HTML document. The following snippet shows how you can use comment tags to hide CSS from older browsers:

```
<style type="text/css">
<!--
body, td {
    color: blue;
}
-->
</style>
```

## CSS Comments:

Many times you may need to put additional comments in your style sheet blocks. So it is very easy to comment any part in style sheet. You simply put your comments inside `/*.....this is a comment in style sheet.....*/`.

You can use `/* ....*/` to comment multi-line blocks in similar way you do in C and C++ programming languages.

### Example:

```
/* This is an external style sheet file */
h1, h2, h3 {
color: #36C;
font-weight: normal;
letter-spacing: .4em;
margin-bottom: 1em;
text-transform: lowercase;
}
```

## CSS - Measurement Units

Before we start actual exercise, I would like to give a brief idea about the CSS Measurement Units.

CSS supports a number of measurements including absolute units such as inches, centimeters, points, and so on, as well as relative measures such as percentages and em units. You need these values while specifying various measurements in your Style rules e.g. `border="1px solid red"`.

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

Unit	Description	Example
%	Defines a measurement as a percentage	p {font-size: 16pt; line-height: 125%;}

	relative to another value, typically an enclosing element.	
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;}
pc	Defines a measurement in picas. A pica is equivalent to 12 points; thus, there are 6 picas per inch.	p {font-size: 20pc;}
pt	Defines a measurement in points. A point is defined as 1/72nd of an inch.	body {font-size: 18pt;}
px	Defines a measurement in screen pixels.	p {padding: 25px;}

## CSS - Colors

CSS uses color values to specify a color. Typically, these are used to set a color either for the foreground of an element (i.e., its text) or else for the background of the element. They can also be used to affect the color of borders and other decorative effects.

You can specify your color values in various formats. Following table tells you all possible formats:

Format	Syntax	Example
Hex Code	#RRGGBB	p{color:#FF0000;}
Short Hex Code	#RGB	p{color:#6A7;}
RGB %	rgb(rrr%,ggg%,bbb%)	p{color:rgb(50%,50%,50%);}

RGB Absolute	rgb(rrr,ggg,bbb)	p{color:rgb(0,0,255);}
keyword	aqua, black, etc.	p{color:teal;}

These formats are explained in more detail in the following sections:

## CSS Colors - Hex Codes:

A hexadecimal is a 6 digit representation of a color. The first two digits(RR) represent a red value, the next two are a green value(GG), and the last are the blue value(BB).

A hexadecimal value can be taken from any graphics software like Adobe Photoshop, JascPaintshop Pro or even using Advanced Paint Brush.

Each hexadecimal code will be preceded by a pound or hash sign #. Following are the examples to use Hexadecimal notation.

Color	Color HEX
	#000000
	#FF0000
	#00FF00
	#0000FF
	#FFFF00
	#00FFFF
	#FF00FF
	#C0C0C0
	#FFFFFF

## CSS Colors - Short Hex Codes:

This is a shorter form of the six-digit notation. In this format, each digit is replicated to arrive at an equivalent six-digit value; For example: #6A7 becomes #66AA77.

A hexadecimal value can be taken from any graphics software like Adobe Photoshop, JascPaintshop Pro or even using Advanced Paint Brush.

Each hexadecimal code will be preceded by a pound or hash sign #. Following are the examples to use Hexadecimal notation.

Color	Color HEX
	#000
	#F00
	#0F0
	#0FF
	#FF0
	#0FF
	#F0F
	#FFF

## CSS Colors - RGB Values:

This color value is specified using the **rgb( )** property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage.

**NOTE:** All the browsers does not support rgb() property of color so it is recommended not to use it.

Following is the example to show few colors using RGB values.

Color	Color RGB
	rgb(0,0,0)
	rgb(255,0,0)
	rgb(0,255,0)
	rgb(0,0,255)
	rgb(255,255,0)
	rgb(0,255,255)
	rgb(255,0,255)
	rgb(192,192,192)
	rgb(255,255,255)

## Setting Backgrounds using CSS

This tutorial will teach you how to set backgrounds of various HTML elements. You can set 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 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>
```

This will produce following result:

This text has a yellow background color.

## Set the background image:

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

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

## Repeat the background image:

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

By default *background-repeat* property will have *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>
```

Following is the 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>
```



Following is the example which 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:

Following is the example which 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>
```

Following is the example which 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>
```

## Set the background attachment:

Background attachment determines whether a background image is fixed or scrolls with the rest of the page.

Following is the example which demonstrates how to set the fixed background image.

```
<p style="background-image:url (/images/pattern1.gif) ;
background-attachment:fixed;">
This paragraph has fixed background image.
</p>
```

Following is the example which demonstrates how to set the scrolling background image.

```
<p style="background-image:url (/images/pattern1.gif) ;  
background-attachment:scroll;">  
This paragraph has scrolling background image.  
</p>
```

## Shorthand property :

You can use the *background* property to set all the background properties at once. For example:

```
<p style="background:url (/images/pattern1.gif) repeat fixed;">  
This paragraph has fixed repeated background image.  
</p>
```

## Setting Fonts using CSS

This tutorial will teach you how to set fonts of a content available in an HTML element. You can set following font properties of an element:

- The **font-family** property is used to change the face of a font.
- 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.

## Set the font family:

Following is the example which demonstrates how to set the font family of an element. Possible value could be any font family name.

```
<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>
```

This will produce following result:

This text is rendered in either georgia, garamond, or the default serif font depending on which font you have at your system.

## Set the font style:

Following is the example which demonstrates how to set the font style of an element. Possible values are *normal*, *italic* and *oblique*.

```
<p style="font-style:italic;">  
This text will be rendered in italic style  
</p>
```

This will produce following result:

*This text will be rendered in italic style*

## Set the font variant:

Following is the example which demonstrates how to set the font variant of an element. Possible values are *normal* and *small-caps*.

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

This will produce following result:

THIS TEXT WILL BE RENEDED AS SMALL CAPS

## Set the font weight:

Following is the example which demonstrates how to set the font weight of an element. The font-weight property provides the functionality to specify how bold a font is. Possible values could be *normal*, *bold*, *bolder*, *lighter*, *100*, *200*, *300*, *400*, *500*, *600*, *700*, *800*, *900*.

```
<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>
```

This will produce following result:

**This font is bold.**

**This font is bolder.**

**This font is 900 weight.**

## Set the font size:

Following is the example which demonstrates how to set the font size of an element. The font-size property is used to control the size of fonts. Possible values could be *xx-small*, *x-small*, *small*, *medium*, *large*, *x-large*, *xx-large*, *smaller*, *larger*, *size in pixels* or *in %*

```
<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>
```

This will produce following result:

This font size is 20 pixels

This font size is small

This font size is large

## Set the font size adjust:

Following is the example which demonstrates how to set the font size adjust of an element. This property enables you to adjust the x-height to make fonts more legible. Possible value could be any number.

```
<p style="font-size-adjust:0.61;">
This text is using a font-size-adjust value.
</p>
```

This will produce following result:

This text is using a font-size-adjust value.

## Set the font stretch:

Following is the example which demonstrates how to set the font stretch of an element. This property relies on the user's computer to have an expanded or condensed version of the font being used.

Possible values could be *normal*, *wider*, *narrower*, *ultra-condensed*, *extra-condensed*, *condensed*, *semi-condensed*, *semi-expanded*, *expanded*, *extra-expanded*, *ultra-expanded*.

```
<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>
```

This will produce following result:

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.

## Shorthand property :

You can use the *font* property to set all the font properties at once. For example:

```
<p style="font:italic small-caps bold 15px georgia;">
```

Applying all the properties on the text at once.

```
</p>
```

This will produce following result:

***APPLYING ALL THE PROPERTIES ON THE TEXT AT ONCE.***

## Manipulating Text using CSS

This tutorial will teach you how to manipulate text using CSS properties. You can set following text properties of an element:

- 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.

## Set the text color:

Following is the example which demonstrates how to set the text color. Possible value could be any color name in any valid format.

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

This will produce following result:

This text will be written in red.

## Set the text direction :

Following is the example which demonstrates how to set the direction of a text. Possible values are *ltr* or *rtl*.

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

This will produce following result:

This text will be rendered from right to left

## Set the space between characters:

Following is the example which demonstrates how to set the space between characters. Possible values are *normal* or a number specifying space..

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

This will produce following result:

T h i s t e x t i s h a v i n g s p a c e b e t w e e n l e t t e r s .

## Set the space between words:

Following is the example which demonstrates how to set the space between words. Possible values are *normal* or a number specifying space..

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

This will produce following result:

This text is having space between words.

## Set the text indent:

Following is the example which demonstrates how to indent the first line of a paragraph. Possible values are *%* or a number specifying indent space..

```
<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>
```

This will produce following result:

    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.

## Set the text alignment:

Following is the example which demonstrates how to align a text. Possible values are *left*, *right*, *center*, *justify*..

```
<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>
```

This will produce following result:

This will be right aligned.

This will be center aligned.

This will be left aligned.

## Decorating the text:

Following is the example which demonstrates how to decorate a text. Possible values are *none*, *underline*, *overline*, *line-through*, *blink*..

```
<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>
```

This will produce following result:

This will be underlined



~~This will be striked through.~~

This will have a over line.

This text will have blinking effect

## Set the text cases:

Following is the example which demonstrates how to set the cases for a text. Possible values are *none*, *capitalize*, *uppercase*, *lowercase*..

```
<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>
```

This will produce following result:

This will be capitalized

THIS WILL BE IN UPPERCASE

This will be in lowercase

## Set the white space between text:

Following is the example which demonstrates how white space inside an element is handled. Possible values are *normal*, *pre*, *nowrap*.

```
<p style="white-space:pre;">This text has a line break
and the white-space pre setting tells the browser to honor it
just like the HTML pre tag.</p>
```

This will produce following result:

This text has a line break

and the white-space pre setting tells the browser to honor it just like the HTML pre tag.

## Set the text shadow:

Following is the example which demonstrates how to set the shadow around a text. This may not be supported by all the browsers.

```
<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>
```

This will produce following result:

If your browser supports the CSS text-shadow property, this text will have a blue shadow.

## CSS - Images

Images are very important part of any Web Page. Though it is not recommended to include lot of images but it is still important to use good images wherever it is required.

CSS plays a good role to control image display. You can set 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.

## The image border Property:

The *border* property of an image is used to set the width of an image border. This property can have a value in length or in %.

A width of zero pixels means no border.

Here is the example:

```

<br />

```

This will produce following result:



## The image height Property:

The *height* property of an image is used to set the height of an image. This property can have a value in length or in %. While giving value in %, it applies it in respect of the box in which an image is available.

Here is the example:

```

<br />

```

This will produce following result:





## The image width Property:

The *width* property of an image is used to set the width of an image. This property can have a value in length or in %. While giving value in %, it applies it in respect of the box in which an image is available.

Here is the example:

```

<br />

```

This will produce following result:



## CSS - Links

This tutorial will teach you how to set different properties of a hyper link using CSS. You can set following properties of a hyper link:

We will revisit 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 these all properties are kept in the header part of 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>
```

Now we will see how to use these properties to give different effects to hyperlinks.

## Set the color of Links:

Following is the example which demonstrates how to set the link color. Possible value could be any color name in any valid format.

```
<style type="text/css">
a:link {color:#000000}
</style>
<a href="/html/index.htm">Black Link</a>
```

This will produce following black link:

[Black Link](#)

## Set the color of Visited Links:

Following is the example which demonstrates how to set the color of visited links. Possible value could be any color name in any valid format.

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

This will produce following link. Once you will click this link, it will change its color to green.

[Click this link](#)

## Change the color of links when mouse is over:

Following is the example which demonstrates how to change the color of links when we bring a mouse pointer over that link. Possible value could be any color name in any valid format.

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

This will produce following link. Now you bring your mouse over this link and you will see that it changes its color to yellow.

[Bring Mouse Here](#)

## Change the color of active links:

Following is the example which demonstrates how to change the color of active links. Possible value could be any color name in any valid format.

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

This will produce following link. This will change its color to pink when user clicks it.

## CSS - Tables

This tutorial will teach you how to set different properties of an HTML table using CSS. You can set following properties of a table:

- The **border-collapse** Specifies whether the browser should control the appearance of 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 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.

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

## The border-collapse Property:

This property can have two values *collapse* and *separate*. Following is the example to show both values:

```
<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>
```

This will produce following result:

Collapse Border Example

Cell A Collapse Example

Cell B Collapse Example

Separate Border Example

Cell A Separate Example

Cell B Separate Example

## The border-spacing Property:

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

If you provide one value it will 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">
/* If you provide one value */
table.example {border-spacing:10px;}
/* This is how you can provide two values */
table.example {border-spacing:10px; 15px;}
</style>
```

Now let's modify previous example and see the effect:

```
<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>
```

This will produce following result:

Separate Border Example with border-spacing
Cell A Collapse Example
Cell B Collapse Example

Separate Border Example with border-spacing
Cell A Separate Example
Cell B Separate Example

## The caption-side 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*. Let us see following example to show 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>
```

This will produce following result:

This caption will appear at the top	
Cell A	
Cell B	

This caption will appear at the bottom	
Cell A	

Cell B
--------

This caption will appear at the left	
Cell A	
Cell B	

This caption will appear at the right	
Cell A	
Cell B	

## The empty-cells 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>
```

	Title one	Title two
Row Title	value	value
Row Title	value	

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

Here is the example to show the difference between these properties.

```
<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%">10000000000000000000000000000</td>  
<td width="40%">10000000</td>  
<td width="40%">100</td>
```

[illegible]

```
<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;        /* Green */
}
</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>
```

This will produce following result:

This example is showing all borders in different colors.

This example is showing all borders in green color only.

## The border-style Property:

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 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.

Following is the example to show 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 aridge border.
</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 aa border with four different styles.
</p>
```

This will produce following result:

This is a border with none width.

This is a solid border.

This is a dahsed border.

This is a double border.

This is a groove border.

This is aridge border.

This is a inset border.

This is a outset border.

This is a hidden border.

This is aa border with four different styles.

## The border-width Property:

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.

Following is the example to show 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 aa border with four different width.
</p>
```

This will produce following result:

This is a solid border whose width is 4px.

This is a solid border whose width is 4pt.

This is a solid border whose width is thin.

This is a solid border whose width is medium;

This is a solid border whose width is thick.

This is aa border with four different width.

## Border Properties Using Shorthand:

The border property allows you to specify color, style, and width of lines in one property:

Following is the example to show to use all the three properties into a single property. This is the most frequently used property to set border around any element.

```
<p style="border:4px solid red;">
This example is showing shorthand property for border.
</p>
```

This will produce following result:

This example is showing shorthand property for border.

## CSS - Margins

The *margin* property defines the space around an HTML element. It is possible to use negative values to overlap content.

The values of the margin property are not inherited by child elements. Remember that the adjacent vertical margins (top and bottom margins) will collapse into each other so that the distance between the blocks is not the sum of the margins, but only the greater of the two margins or the same size as one margin if both are equal.

There are following four properties to set an element margin.

- The **margin** 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.

## The margin Property:

The margin property allows you set all of the properties for the four margins in one declaration. Here is the syntax to set margin around a paragraph:

```
<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 {margin: 10px 2% -10px}
```

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 {margin: 10px 2% -10px auto}
```

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

```
</style>
```

Here is the example:

```
<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>

This will produce following result:

all four margins will be 10px
-------------------------------

top and bottom margin will be 10px, left and right margin will be 2% of the total width of the document.
--

top margin will be 10px, left and right margin will be 2% of the total width of the document, bottom margin will be -10px
---

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
--

## The margin-bottom Property:

The margin-bottom property allows you set bottom margin of an element. It can have a value in length, % or auto.

Here is the example:

<p style="margin-bottom: 15px; border:1px solid black;">

This is a paragraph with a specified bottom margin

</p>

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

This is another paragraph with a specified bottom margin in percent

</p>

This will produce following result:

This is a paragraph with a specified bottom margin

This is another paragraph with a specified bottom margin in percent

## The margin-top Property:

The margin-top property allows you set top margin of an element. It can have a value in length, % or auto.

Here is the example:

```
<p style="margin-top: 15px; border:1px solid black;">
```

This is a paragraph with a specified top margin

```
</p>
```

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

This is another paragraph with a specified top margin in percent

```
</p>
```

This will produce following result:

This is a paragraph with a specified top margin

This is another paragraph with a specified top margin in percent

## The margin-left Property:

The margin-left property allows you set left margin of an element. It can have a value in length, % or auto.

Here is the example:

```
<p style="margin-left: 15px; border:1px solid black;">
```

This is a paragraph with a specified left margin  
</p>

<p style="margin-left: 5%; border:1px solid black;">  
This is another paragraph with a specified top margin in percent  
</p>

This will produce following result:

This is a paragraph with a specified left margin

This is another paragraph with a specified top margin in percent

## The margin-right Property:

The margin-right property allows you set right margin of an element. It can have a value in length, % or auto.

Here is the example:

<p style="margin-right: 15px; border:1px solid black;">  
This is a paragraph with a specified right margin  
</p>  
  
<p style="margin-right: 5%; border:1px solid black;">  
This is another paragraph with a specified right margin in percent  
</p>

This will produce following result:

This is a paragraph with a specified right margin

This is another paragraph with a specified right margin in percent

## CSS - Lists

Lists are very helpful in conveying a set of either numbered or bulleted points. This tutorial teaches you how to control list type, position, style etc. using CSS

There are 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.

## The list-style-type Property:

The *list-style-type* property allows you to control the shape or style of bullet point (also known as a marker) in the case of unordered lists, and the style of numbering characters in ordered lists.

Here are the values which can be used for an unordered list:

Value	Description
none	NA
disc (default)	A filled-in circle
circle	An empty circle
square	A filled-in square

Here are the values which can be used for an ordered list:

Value	Description	Example
decimal	Number	1,2,3,4,5
decimal-leading-zero	0 before the number	01, 02, 03, 04, 05
lower-alpha	Lowercase alphanumeric characters	a, b, c, d, e

upper-alpha	Uppercase alphanumeric characters	A, B, C, D, E
lower-roman	Lowercase Roman numerals	i, ii, iii, iv, v
upper-roman	Uppercase Roman numerals	I, II, III, IV, V
lower-greek	The marker is lower-greek	alpha, beta, gamma
lower-latin	The marker is lower-latin	a, b, c, d, e
upper-latin	The marker is upper-latin	A, B, C, D, E
hebrew	The marker is traditional Hebrew numbering	
armenian	The marker is traditional Armenian numbering	
georgian	The marker is traditional Georgian numbering	
cjk-ideographic	The marker is plain ideographic numbers	
hiragana	The marker is hiragana	a, i, u, e, o, ka, ki
katakana	The marker is katakana	A, I, U, E, O, KA, KI
hiragana-iroha	The marker is hiragana-iroha	i, ro, ha, ni, ho, he, to
katakana-iroha	The marker is katakana-iroha	I, RO, HA, NI, HO, HE, TO

Here is the example:

```
<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>
```

This will produce following result:

- Maths
- Social Science
- Physics
  
- Maths
- Social Science
- Physics
  
- 1. Maths
- 2. Social Science
- 3. Physics
  
- a. Maths
- b. Social Science
- c. Physics
  
- i. Maths
- ii. Social Science
- iii. Physics

## The list-style-position Property:



The *list-style-position* property indicates whether the marker should appear inside or outside of the box containing the bullet points. It can have one the two values:

Value	Description
none	NA
inside	If the text goes onto a second line, the text will wrap underneath the marker. It will also appear indented to where the text would have started if the list had a value of outside.
outside	If the text goes onto a second line, the text will be aligned with the start of the first line (to the right of the bullet).

Here is the example:

```
<ul style="list-style-type:circle; list-style-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-style-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>
```

This will produce following result:

- Maths
- Social Science

- Physics
- Maths
- Social Science
- Physics
- 1. Maths
- 2. Social Science
- 3. Physics
- a. Maths
- b. Social Science
- c. Physics

## The list-style-image Property:

The *list-style-image* allows you to specify an image so that you can use your own bullet style. The syntax is as follows, similar to the background-image property with the letters url starting the value of the property followed by the URL in brackets. If it does not find given image then default bullets are used.

Here is the example:

```
<ul>
<li style="list-style-image: url(/images/bullet.gif);">Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ul>

<ol>
<li style="list-style-image: url(/images/bullet.gif);">Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ol>
```

This will produce following result:

- Maths
- Social Science
- Physics
- 1. Maths
- 2. Social Science

### 3. Physics

## The list-style Property:

The *list-style* allows you to specify all the list properties into a single expression. These properties can appear in any order.

Here is the example:

```
<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>
```

This will produce following result:

- Maths
  - Social Science
  - Physics
- 
- A. Maths
  - B. Social Science
  - C. Physics

## The marker-offset Property:

The *marker-offset* property allows you to specify the distance between the marker and the text relating to that marker. Its value should be a length as shown in the following example:

Unfortunately, however, this property is not supported in IE 6 or Netscape 7.

Here is the example:

```
<ul style="list-style: inside square; marker-offset:2em;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ul>
```

```
<ol style="list-style: outside upper-alpha; marker-offset:2cm;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ol>
```

This will produce following result:

- Maths
  - Social Science
  - Physics
- 
- A. Maths
  - B. Social Science
  - C. Physics

## CSS - Paddings

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

There are following five CSS properties which can be used to control lists:

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.

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 an element.
- 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 an element.
- The **padding** Serves as shorthand for the preceding properties.

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

## The padding-bottom Property:

The *padding-bottom* property sets the bottom padding (space) of an element. This can take a value in terms of length of %.

Here is the example:

```
<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>
```

This will produce following result:

This is a paragraph with a specified bottom padding

This is another paragraph with a specified bottom padding in percent

## The padding-top Property:

The *padding-top* property sets the top padding (space) of an element. This can take a value in terms of length of %.

Here is the example:

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

<p style="padding-top: 5%; border:1px solid black;">
This is another paragraph with a specified top padding in percent
</p>
```

This will produce following result:

This is a paragraph with a specified top padding

This is another paragraph with a specified top padding in percent

## The padding-left Property:

The *padding-left* property sets the left padding (space) of an element. This can take a value in terms of length of %.

Here is the example:

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

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

This will produce following result:

This is a paragraph with a specified left padding

This is another paragraph with a specified left padding in percent

## The padding-right Property:

The *padding-right* property sets the right padding (space) of an element. This can take a value in terms of length of %.

Here is the example:

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

```
<p style="padding-right: 5%; border:1px solid black;">  
This is another paragraph with a specified right padding in percent  
</p>
```

This will produce following result:

This is a paragraph with a specified right padding

This is another paragraph with a specified right padding in percent

## The padding Property:

The *padding* property sets the left, right, top and bottom padding (space) of an element. This can take a value in terms of length of %.

Here is the example:

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

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

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

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

This will produce following result:

---

all four paddings will be 15px

---



---

top and bottom paddings will be 10px, left and right paddings will be 2% of the total width of the document.

---

top padding will be 10px, left and right padding will be 2% of the total width of the document,

bottom padding will be 10px

top padding will be 10px, right padding will be 2% of the total width of the document, bottom padding and top padding will be 10p

## CSS - Cursors

The *cursor* property of CSS allows you to specify the type of cursor that should be displayed to the user.

One good usage of this property is in using images for submit buttons on forms. By default, when a cursor hovers over a link, the cursor changed from a pointer to a hand. For a submit button on a form this does not happen. Therefore, using the cursor property to change the cursor to a hand whenever someone hovers over an image that is a submit button. This provides a visual clue that they can click it.

The table that follows shows possible values for the cursor property:

Value	Description
auto	Shape of the cursor depends on the context area it is over. For example an I over text, a hand over a link, and so on...
crosshair	A crosshair or plus sign
default	An arrow
pointer	A pointing hand (in IE 4 this value is hand)
move	The I bar
e-resize	The cursor indicates that an edge of a box is to be moved right (east)
ne-resize	The cursor indicates that an edge of a box is to be moved up and right (north/east)
nw-resize	The cursor indicates that an edge of a box is to be moved up and left (north/west)
n-resize	The cursor indicates that an edge of a box is to be moved up (north)
se-resize	The cursor indicates that an edge of a box is to be moved down and right (south/east)
sw-resize	The cursor indicates that an edge of a box is to be moved down and left (south/west)
s-resize	The cursor indicates that an edge of a box is to be moved down (south)



w-resize	The cursor indicates that an edge of a box is to be moved left (west)
text	The I bar
wait	An hour glass
help	A question mark or balloon, ideal for use over help buttons
<url>	The source of a cursor image file

**NOTE:** You should try to use only these values to add helpful information for users, and in places they would expect to see that cursor. For example, using the crosshair when someone hovers over a link can confuse visitors.

Here is the example:

```
<p>Move the mouse over the words to see the cursor change:</p>
<div style="cursor:auto">Auto</div>
<div style="cursor:crosshair">Crosshair</div>
<div style="cursor:default">Default</div>
<div style="cursor:pointer">Pointer</div>
<div style="cursor:move">Move</div>
<div style="cursor:e-resize">e-resize</div>
<div style="cursor:ne-resize">ne-resize</div>
<div style="cursor:nw-resize">nw-resize</div>
<div style="cursor:n-resize">n-resize</div>
<div style="cursor:se-resize">se-resize</div>
<div style="cursor:sw-resize">sw-resize</div>
<div style="cursor:s-resize">s-resize</div>
<div style="cursor:w-resize">w-resize</div>
<div style="cursor:text">text</div>
<div style="cursor:wait">wait</div>
<div style="cursor:help">help</div>
```

This will produce following result:

Move the mouse over the words to see the cursor change:

Auto  
Crosshair  
Default  
Pointer  
Move  
e-resize  
ne-resize  
nw-resize  
n-resize  
se-resize  
sw-resize

s-resize  
w-resize  
text  
wait  
help

## CSS - Outlines

Outlines are very similar to the borders but there are few major differences in borders and outlines:

- An outline does not take up space.
- Outlines do not have to be rectangular.
- Outline is always the same on all sides; you cannot specify different values for different sides of the element.

**NOTE:** The outline properties are not supported by IE 6 or Netscape 7.

You can set following outline properties using CSS.

- The **outline-width** property is used to set the width of the outline.
- The **outline-style** property is used to set the line style for the outline.
- The **outline-color** property is used to set the color of the outline.
- The **outline** property is used to set all the above three properties in a single statement.

## The outline-width Property:

The *outline-width* property specifies the width of the outline to be added to the box. Its value should be a length or one of the values *thin*, *medium*, or *thick* .just like the border-width attribute

A width of zero pixels means no outline.

Here is the example:

```
<p style="outline-width:thin; outline-style:solid;">
This text is having thin outline.
</p>
<br />
<p style="outline-width:thick; outline-style:solid;">
This text is having thick outline.
</p>
<br />
<p style="outline-width:5px; outline-style:solid;">
This text is having 5x outline.
</p>
```

This will produce following result:

This text is having thin outline.

This text is having thick outline.

This text is having 5x outline.

## The outline-style Property:

The *outline-style* property specifies the style for the line (solid, dotted, or dashed) that goes around an element. It can take one of the following values:

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

Here is the example:

```
<p style="outline-width:thin; outline-style:solid;">
This text is having thin solid outline.
</p>
<br />
<p style="outline-width:thick; outline-style:dashed;">
This text is having thick dashed outline.
</p>
<br />
<p style="outline-width:5px;outline-style:dotted;">
This text is having 5x dotted outline.
</p>
```

This will produce following result:

This text is having thin solid outline.

This text is having thick dashed outline.

This text is having 5x dotted outline.

## The outline-color Property:

The *outline-color* property allows you to specify the color of the outline. Its value should either be a color name, a hex color, or an RGB value, as with the color and border-color properties.

Here is the example:

```
<p style="outline-width:thin; outline-style:solid;
outline-color:red">
This text is having thin solid red outline.
</p>
<br />
<p style="outline-width:thick; outline-style:dashed;
outline-color:#009900">
This text is having thick dashed green outline.
</p>
<br />
<p style="outline-width:5px;outline-style:dotted;
outline-color:rgb(13,33,232)">
This text is having 5x dotted blue outline.
</p>
```

This will produce following result:

This text is having thin solid red outline.

This text is having thick dashed green outline.

This text is having 5x dotted blue outline.

## The outline Property:

The *outline* property is a shorthand property that allows you to specify values for any of the three properties discussed previously in any order but in a single statement.

Here is the example:

```
<p style="outline:thin solid red;">
This text is having thin solid red outline.
</p>
<br />
<p style="outline:thick dashed #009900;">
This text is having thick dashed green outline.
</p>
<br />
<p style="outline:5px dotted rgb(13,33,232);">
This text is having 5x dotted blue outline.
</p>
```

This will produce following result:

This text is having thin solid red outline.

This text is having thick dashed green outline.

This text is having 5x dotted blue outline.

## CSS - Dimension

You have seen the border that surrounds every box ie. element, the padding that can appear inside each box and the margin that can go around them. In this tutorial we will how we can change the dimensions of boxes.

There are 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 a box.
- The **line-height** property is used to set the height of a line of text.
- 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.

## The height and width Properties:

The *height* and *width* properties allow you to set the height and width for boxes. They can take values of a length, a percentage, or the keyword auto.

Here is the example:

```
<p style="width:400px; height:100px;border:1px solid red;
padding:5px; margin:10px;">
This paragraph is 400pixels wide and 100 pixels high
</p>
```

This will produce following result:

This paragraph is 400pixels wide and 100 pixels high

## The line-height Property:

The *line-height* property allows you to increase the space between lines of text. The value of the line-height property can be a number, a length, or a percentage.

Here is the example:

```
<p style="width:400px; height:100px;border:1px solid red;
padding:5px; margin:10px;line-height:30px;">
This paragraph is 400pixels wide and 100 pixels high
and here line height is 30pixels.This paragraph is 400 pixels
wide and 100 pixels high and here line height is 30pixels.
</p>
```

This will produce following result:

This paragraph is 400pixels wide and 100 pixels high and here line height is 30pixels.This paragraph is 400 pixels wide and 100 pixels high and here line height is 30pixels.

## The max-height Property:

The *max-height* property allows you to specify maximum height of a box. The value of the max-height property can be a number, a length, or a percentage.

**NOTE:** This property does not work in either Netscape 7 or IE 6.

Here is the example:

```
<p style="width:400px; max-height:10px;border:1px solid red;
padding:5px; margin:10px;">
This paragraph is 400px wide and max height is 10px
This paragraph is 400px wide and max height is 10px
This paragraph is 400px wide and max height is 10px
This paragraph is 400px wide and max height is 10px
</p>

```

This will produce following result:

This paragraph is 400px wide and max height is 10px This paragraph is 400px wide and max height is 10px This paragraph is 400px wide and max height is 10px This paragraph is 400px wide and max height is 10px



## The min-height Property:

The *min-height* property allows you to specify minimum height of a box. The value of the min-height property can be a number, a length, or a percentage.

**NOTE:** This property does not work in either Netscape 7 or IE 6.

Here is the example:

```
<p style="width:400px; min-height:200px;border:1px solid red;
padding:5px; margin:10px;">
```

```
This paragraph is 400px wide and min height is 200px
This paragraph is 400px wide and min height is 200px
This paragraph is 400px wide and min height is 200px
This paragraph is 400px wide and min height is 200px
</p>

```

This will produce following result:

This paragraph is 400px wide and min height is 200px This paragraph is 400px wide and min height is 200px This paragraph is 400px wide and min height is 200px This paragraph is 400px wide and min height is 200px



## The max-width Property:

The *max-width* property allows you to specify maximum width of a box. The value of the max-width property can be a number, a length, or a percentage.

**NOTE:** This property does not work in either Netscape 7 or IE 6.

Here is the example:

```
<p style="max-width:100px; height:200px;border:1px solid red;
padding:5px; margin:10px;">
This paragraph is 200px high and max width is 100px
This paragraph is 200px high and max width is 100px
This paragraph is 200px high and max width is 100px
This paragraph is 200px high and max width is 100px
This paragraph is 200px high and max width is 100px
</p>

```

This will produce following result:

This paragraph is 200px high and max width is 100px This paragraph is 200px high and max width is 100px This paragraph is 200px high and max width is 100px This paragraph is 200px high and max width is 100px



200px high and max width is 100px This paragraph is 200px high and max width is 100px



## The min-width Property:

The *min-width* property allows you to specify minimum width of a box. The value of the min-width property can be a number, a length, or a percentage.

**NOTE:** This property does not work in either Netscape 7 or IE 6.

Here is the example:

```
<p style="min-width:400px; height:100px;border:1px solid red;
padding:5px; margin:10px;">
This paragraph is 100px high and min width is 400px
This paragraph is 100px high and min width is 400px
This paragraph is 100px high and min width is 400px
This paragraph is 100px high and min width is 400px
This paragraph is 100px high and min width is 400px
</p>

```

This will produce following result:

This paragraph is 100px high and min width is 400px This paragraph is 100px high and min width is 400px This paragraph is 100px high and min width is 400px This paragraph is 100px high and min width is 400px

## CSS - Scrollbars

There may be a case when an element's content might be larger than the amount of space allocated to it. For example given width and height properties did not allow enough room to accommodate the content of the element.

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:

Value	Description
visible	Allows the content to overflow the borders of its containing element.
hidden	The content of the nested element is simply cut off at the border of the containing element and no scrollbars is visible.
scroll	The size of the containing element does not change, but the scrollbars are added to allow the user to scroll to see the content.
auto	The purpose is the same as scroll, but the scrollbar will be shown only if the content does overflow.

Here is the example:

```
<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>
```

This will produce following result:

Example of scroll value:

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.

Example of auto value:

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.

*All the best for your bright career*

By :Infocampus