CSS Tutorial

CSS is used to control the style of a web document in a simple and easy way.

CSS is the acronym for **"Cascading Style Sheet"**. This tutorial covers both the versions CSS1,CSS2 and CSS3, and gives a complete understanding of CSS, starting from its basics to advanced concepts.

Why to Learn 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 is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning CSS:

- **Create Stunning Web site** - 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, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.
- **Become a web designer** - If you want to start a carrer as a professional web designer, HTML and CSS designing is a must skill.
- **Control web** - 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.
- **Learn other languages** - Once you understands the basic of HTML and CSS then other related technologies like javascript, php, or angular are become easier to understand.

Hello World using CSS.

Just to give you a little excitement about CSS, I'm going to give you a small conventional CSS Hello World program, You can try it using Demo link.

```
!DOCTYPE html>
<html>
   <head>
      <title>This is document title</title>
      <style>
      h1 {
         color: #36CFFF;
      }
      </style>
   </head>
   <body>
      <h1>Hello World!</h1>
   </body>
</html>
```

Applications/Advantages of CSS

As mentioned before, CSS is one of the most widely used style language over the web. I'm going to list few of them here:

- **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 it 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 a far better look to your HTML page in comparison to 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.

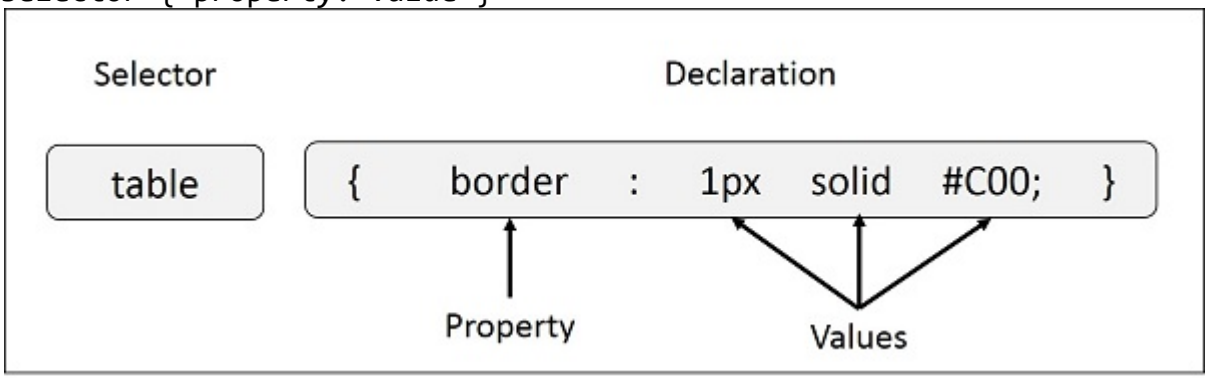
CSS - Syntax

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 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 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.

Css Selectors

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 element only when it lies inside <ul> tag.

```
ul em {
   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 when those headings will lie with in tags having *id* attribute set to *black*.

The Child Selectors

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 direct child of <body> element. Other paragraphs put inside other elements like or <td> 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 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.

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 **semicolon (;)**. You can keep them in a single line or multiple lines. For better readability, we keep them in separate lines.

For a while, don't bother about the properties mentioned in the above block. These properties will be explained in the 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 the various *id* selectors together as shown below −

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

CSS - Inclusion

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 the <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document. Here is the generic syntax −

```
<!DOCTYPE html>
<html>
   <head>
      <style type = "text/css" media = "all">
```

```
         body {
            background-color: linen;
         }
         h1 {
            color: maroon;
            margin-left: 40px;
         }
      </style>
   </head>
   <body>
      <h1>This is a heading</h1>
      <p>This is a paragraph.</p>
   </body>
</html>
```

It will produce the following result −

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 *all*. This is an optional attribute.

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 *style* attribute is a combination of style declarations separated by semicolon (;).

Example

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

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

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

It will produce the following result −

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 rel="stylesheet”  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 *all*. 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 a 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.

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 can simple 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

```
<!DOCTYPE html>
<html>
   <head>
      <style>
         p {
            color: red;
            /* This is a single-line comment */
            text-align: center;
         }
         /* This is a multi-line comment */
      </style>
   </head>

   <body>
      <p>Hello World!</p>
   </body>
</html>
```

It will produce the following result −

Hello World!

CSS - Measurement Units

Before we start the actual exercise, we 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 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	An em is a CSS unit that measures the size of a font, from the top of a font's cap height to the bottom of its lowest descender. Originally, the em was equal to the width of the capital letter M, which is where its name originated. 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 lists all the 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	rgba(rrr,ggg,bbb,0 to 1)	p{color:rgb(0,0,255,0.4);}
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, Jasc Paintshop 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, Jasc Paintshop 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)

Building Color Codes

You can build millions of color codes using our Color Code Builder. Check our **HTML Color Code Builder**. To use this tool, you would need a Java Enabled Browser.

Browser Safe Colors

Here is the list of 216 colors which are supposed to be most safe and computer independent colors. These colors vary from hexa code 000000 to FFFFFF. These colors are safe to use because they ensure that all computers would display the colors correctly when running a 256 color palette −

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF

669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF
999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF

FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

CSS - Backgrounds

This chapter teaches you how to set backgrounds of various HTML elements. 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.

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

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

This will produce following result –

This text has a yellow background color.

Set the Background Image

We can set the background image by calling local stored images as shown below −

```
<html>
   <head>
      <style>
         body  {
            background-image: url("/css/images/css.jpg");
            background-color: #cccccc;
         }
      </style>
   </head>

   <body>
      <h1>Hello World!</h1>
```

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

It will produce the following result −

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 *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.

```
<html>
   <head>
      <style>
         body {
            background-image: url("/css/images/css.jpg");
            background-repeat: repeat;
         }
      </style>
   </head>

   <body>
      <p>Tutorials point</p>
   </body>
</html>
```

It will produce the following result −

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

```
<html>
   <head>
      <style>
         body {
            background-image: url("/css/images/css.jpg");
            background-repeat: repeat-y;
         }
      </style>
   </head>

   <body>
      <p>Tutorials point</p>
   </body>
</html>
```

It will produce the following result −

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

```
<html>
   <head>
      <style>
         body {
            background-image: url("/css/images/css.jpg");
            background-repeat: repeat-x;
         }
      </style>
   </head>

   <body>
      <p>Tutorials point</p>
```

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

It will produce the following result −

Set the Background Image Position

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

```
<html>
   <head>
      <style>
         body {
            background-image: url("/css/images/css.jpg");
            background-position:100px;
         }
      </style>
   </head>

   <body>
      <p>Tutorials point</p>
   </body>
</html>
```

It will produce the following result −

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.

```
<html>
   <head>
      <style>
         body {
            background-image: url("/css/images/css.jpg");
            background-position:100px 200px;
         }
      </style>
   </head>

   <body>
      <p>Tutorials point</p>
   </body>
</html>
```

It will produce the following result −

Set the Background Attachment

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

The following example demonstrates how to set the fixed background image.

```
<!DOCTYPE html>
<html>
   <head>
      <style>
         body {
            background-image: url('/css/images/css.jpg');
            background-repeat: no-repeat;
            background-attachment: fixed;
         }
      </style>
   </head>
```

```
   <body>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
   </body>
</html>
```

It will produce the following result −

The following example demonstrates how to set the scrolling background image.

```
<!DOCTYPE html>
<html>
   <head>
      <style>
         body {
            background-image: url('/css/images/css.jpg');
            background-repeat: no-repeat;
            background-attachment: fixed;
            background-attachment:scroll;
         }
      </style>
   </head>

   <body>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
      <p>The background-image is fixed. Try to scroll down the page.</p>
   </body>
</html>
```

It will produce the following result −

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 parapgraph has fixed repeated background image.
</p>
```

CSS - Fonts

This chapter teaches 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.

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

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

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*.

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

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

This will produce following result –

This text will be rendered in italic style

Set the Font Variant

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

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

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

```
</html>
```

This will produce following result −

This text will be rendered as small caps

Set the Font Weight

The following example 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*.

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

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

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

      <p style = "font-weight:500;">
         This font is 500 weight.
      </p>
   </body>
</html>
```

This will produce following result −

This font is bold.

This font is bolder.

This font is 500 weight.

Set the Font Size

The following example 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 %*.

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

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

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

The following example 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.

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

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

This will produce following result –

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

Set the Font Stretch

The following example 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*.

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

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

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 −

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

   <body>
      <p style = "font:italic small-caps bold 15px georgia;">
         Applying all the properties on the text at once.
      </p>
   </body>
</html>
```

This will produce following result −

Applying all the properties on the text at once.

CSS - Text

This chapter teaches 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

The following example demonstrates how to set the text color. Possible value could be any color name in any valid format.

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

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

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

It will produce the following result −

Set the Text Direction

The following example demonstrates how to set the direction of a text. Possible values are *ltr or rtl*.

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

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

It will produce the following result −

Set the Space between Characters

The following example demonstrates how to set the space between characters. Possible values are *normal or a number specifying space.*.

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

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

It will produce the following result −

Set the Space between Words

The following example demonstrates how to set the space between words. Possible values are *normal or a number specifying space*.

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

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

This will produce following result −

Set the Text Indent

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

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

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

It will produce the following result −

Set the Text Alignment

The following example demonstrates how to align a text. Possible values are *left, right, center, justify*.

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

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

This will produce following result −

Decorating the Text

The following example demonstrates how to decorate a text. Possible values are *none, underline, overline, line-through, blink*.

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

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

This will produce following result −

Set the Text Cases

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

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

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

This will produce following result −

Set the White Space between Text

The following example demonstrates how white space inside an element is handled. Possible values are *normal, pre, nowrap*.

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

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

This will produce following result −

Set the Text Shadow

The following example demonstrates how to set the shadow around a text. This may not be supported by all the browsers.

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

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

It will produce the following result −

CSS - Using Images (not in syllabus)

Images play an important role in any webpage. Though it is not recommended to include a lot of images, but it is still important to use good images wherever required.

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.

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 −

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

   <body>
      <img style = "border:0px;" src = "/css/images/logo.png" />
      <br />
      <img style = "border:3px dashed red;" src = "/css/images/logo.png" />
   </body>
</html>
```

It will produce the 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 an example −

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

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

It will produce the 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 an example −

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

   <body>
      <img style = "border:1px solid red; width:150px;" src =
"/css/images/logo.png" />
      <br />
      <img style = "border:1px solid red; width:100%;" src =
"/css/images/logo.png" />
   </body>
</html>
```

It will produce the following result −

The -moz-opacity Property

The *-moz-opacity* property of an image is used to set the opacity of an image. This property is used to create a transparent image in Mozilla. IE uses **filter:alpha(opacity=x)** to create transparent images.

In Mozilla (-moz-opacity:x) x can be a value from 0.0 - 1.0. A lower value makes the element more transparent (The same things goes for the CSS3-valid syntax opacity:x).

In IE (filter:alpha(opacity=x)) x can be a value from 0 - 100. A lower value makes the element more transparent.

Here is an example −

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

   <body>
      <img style = "border:1px solid red; -moz-opacity:0.4;
filter:alpha(opacity=40);" src = "/css/images/logo.png" />
   </body>
</html>
```

It will produce the following result −

CSS - Links(not in syllabus)

This chapter teaches you how to set different properties of a hyper link using CSS. You can set following properties of a hyper link −

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>
```

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

Set the Color of Links

The following example demonstrates how to set the link color. Possible values could be any color name in any valid format.

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

   <body>
      <a href = "">Link</a>
   </body>
</html>
```

It will produce the following black link −

Set the Color of Visited Links

The following example demonstrates how to set the color of visited links. Possible values could be any color name in any valid format.

```
<html>
   <head>
      <style type = "text/css">
         a:visited {color: #006600}
      </style>
   </head>

   <body>
      <a href = ""> link</a>
   </body>
</html>
```

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

Change the Color of Links when Mouse is Over

The following example demonstrates how to change the color of links when we bring a mouse pointer over that link. Possible values could be any color name in any valid format.

```
<html>
   <head>
      <style type = "text/css">
         a:hover {color: #FFCC00}
      </style>
   </head>

   <body>
      <a href = "">Link</a>
   </body>
</html>
```

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

Change the Color of Active Links

The following example demonstrates how to change the color of active links. Possible values could be any color name in any valid format.

```
<html>
   <head>
      <style type = "text/css">
         a:active {color: #FF00CC}
      </style>
   </head>

   <body>
      <a href = "">Link</a>
   </body>
</html>
```

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

CSS – Tables (not in syllabus)

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 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 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*. The following example uses both the values −

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

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

It will produce the following result −

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 the previous example and see the effect −

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

   <body>

      <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>

   </body>
</html>
```

It will produce the following result −

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*. The following example uses each value.

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

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

   <body>

      <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>

   </body>
</html>
```

It will produce the following result −

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.

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

   <body>

      <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>

   </body>
</html>
```

It will produce the following result −

The 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.

```
<html>
   <head>
      <style type = "text/css">
         table.auto {
            table-layout: auto
         }
         table.fixed {
            table-layout: fixed
         }
      </style>
   </head>

   <body>

      <table class = "auto" border = "1" width = "100%">
         <tr>
            <td width = "20%">10000000000000000000000000000</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%">10000000000000000000000000000</td>
            <td width = "40%">10000000</td>
            <td width = "40%">100</td>
         </tr>
      </table>

   </body>
</html>
```

It will produce the following result −

CSS - Borders

The *border* properties allow you to specify how the border of the box representing an element should look. 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.

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

The border-color Property

The border-color property allows you to change the color of the border surrounding an element. You can individually change the color of the bottom, left, top and right sides of an element's border using the properties −

- **border-bottom-color** changes the color of bottom border.
- **border-top-color** changes the color of top border.
- **border-left-color** changes the color of left border.

- **border-right-color** changes the color of right border.

The following example shows the effect of all these properties −

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

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

It will produce the following result −

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 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 −

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

   <body>
      <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 dashed 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 a ridge  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 a a border with four different styles.
      </p>
   </body>
</html>
```

It will produce the following result −

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.

The following example shows all these border width −

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

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

It will produce the following result −

Border Properties Using Shorthand

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

The following example shows how to use all the three properties into a single property. This is the most frequently used property to set border around any element.

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

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

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

It will produce the following result −

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 the 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.

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.

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 −

Here is an example −

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

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

```
</html>
```

It will produce the following result −

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 an example −

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

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

It will produce the following result −

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 an example −

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

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

It will produce the following result −

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 an example −

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

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

It will produce the following result −

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 an example −

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

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

It will produce the following result −

CSS - Lists

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

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.

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 −

Sr.No.	Value & Description
1	**none** NA
2	**disc (default)** A filled-in circle
3	**circle** An empty circle
4	**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 an example −

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

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

It will produce the following result −

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 −

Sr.No.	Value & Description
1	**none** NA
2	**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.
3	**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 an example −

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

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

It will produce the following result −

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 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 the given image then default bullets are used.

Here is an example −

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

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

It will produce the following result −

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 an example −

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

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

It will produce the 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, this property is not supported in IE 6 or Netscape 7.

Here is an example −

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

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

It will produce the following result −

- Maths
- Social Science
- Physics

A. Maths
B. Social Science
C. Physics

CSS – Paddings (Not in syllabus)

The *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 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 an example −

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

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

It will produce the following result −

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 an example −

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

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

It will produce the following result −

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 an example −

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

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

It will produce the following result −

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 an example −

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

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

It will produce the following result −

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 an example −

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

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

It will produce the following result −

all four padding will be 15px

top and bottom padding will be 10px, left and right padding 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 10px

CSS - Cursors (Not in syllabus)

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 changes from a pointer to a hand. However, it does not change

form for a submit button on a form. Therefore, whenever someone hovers over an image that is a submit button, it provides a visual clue that the image is clickable.

The following table shows the possible values for the cursor property −

Sr.No.	Value & Description
1	**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...
2	**Crosshair** A crosshair or plus sign
3	**Default** An arrow
4	**Pointer** A pointing hand (in IE 4 this value is hand)
5	**Move** The I bar
6	**e-resize** The cursor indicates that an edge of a box is to be moved right (east)
7	**ne-resize** The cursor indicates that an edge of a box is to be moved up and right (north/east)
8	**nw-resize** The cursor indicates that an edge of a box is to be moved up and left (north/west)
9	**n-resize** The cursor indicates that an edge of a box is to be moved up (north)
10	**se-resize** The cursor indicates that an edge of a box is to be moved down and right (south/east)
11	**sw-resize**

	The cursor indicates that an edge of a box is to be moved down and left (south/west)
12	**s-resize** The cursor indicates that an edge of a box is to be moved down (south)
13	**w-resize** The cursor indicates that an edge of a box is to be moved left (west)
14	**Text** The I bar
15	**Wait** An hour glass
16	**Help** A question mark or balloon, ideal for use over help buttons
17	**<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 an example −

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

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

It will produce the 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 (Not in syllabus)

Outlines are very similar to borders, but there are few major differences as well −

- 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 an element.

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

You can set the 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 an example –

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

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

It will produce the following result −

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 an example −

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

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

It will produce the following result −

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 an example −

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

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

It will produce the following result −

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 an example −

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

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

It will produce the following result −

CSS - Scrollbars (Not in syllabus)

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 do 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 −

Sr.No.	Value & Description
1	**Visible** Allows the content to overflow the borders of its containing element.
2	**Hidden** The content of the nested element is simply cut off at the border of the containing element and no scrollbars is visible.
3	**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.
4	**Auto** The purpose is the same as scroll, but the scrollbar will be shown only if the content does overflow.

Here is an example −

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

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

It will produce the following result −

CSS - Visibility (Not in syllabus)

A property called *visibility* allows you to hide an element from view. You can use this property along with JavaScript to create very complex menu and very complex webpage layouts.

You may choose to use the visibility property to hide error messages that are only displayed if the user needs to see them, or to hide answers to a quiz until the user selects an option.

NOTE − Remember that the source code will still contain whatever is in the invisible paragraph, so you should not use this to hide sensitive information such as credit card details or passwords.

The *visibility* property can take the values listed in the table that follows −

Sr.No.	Value & Description
1	**Visible** The box and its contents are shown to the user.
2	**Hidden** The box and its content are made invisible, although they still affect the layout of the page.
3	**Collapse** This is for use only with dynamic table columns and row effects.

Here is an example −

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

   <body>
      <p>
         This paragraph should be visible in normal way.
      </p>

      <p style = "visibility:hidden;">
         This paragraph should not be visible.
      </p>
   </body>
</html>
```

It will produce the following result −

This paragraph should be visible in normal way.

CSS - Positioning (Not in syllabus)

CSS helps you to position your HTML element. You can put any HTML element at whatever location you like. You can specify whether you want the element positioned relative to its natural position in the page or absolute based on its parent element.

Now, we will see all the CSS positioning related properties with examples −

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 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*.

Here is the example −

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

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

It will produce the following result −

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 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.

Here is an example −

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

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

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*.

Here is an example −

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

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

This div has fixed positioning.

CSS - Layers (Not in syllabus)

CSS gives you opportunity to create layers of various divisions. The CSS layers refer to applying the *z-index* property to elements that overlap with each other.

The z-index property is used along with the *position* property to create an effect of layers. You can specify which element should come on top and which element should come at bottom.

A z-index property can help you to create more complex webpage layouts. Following is the example which shows how to create layers in CSS.

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

   <body>
      <div style = "background-color:red;
         width:300px;
         height:100px;
         position:relative;
         top:10px;
         left:80px;
         z-index:2">
      </div>

      <div style = "background-color:yellow;
         width:300px;
         height:100px;
         position:relative;
         top:-60px;
         left:35px;
         z-index:1;">
      </div>

      <div style = "background-color:green;
         width:300px;
         height:100px;
         position:relative;
         top:-220px;
         left:120px;
         z-index:3;">
      </div>
   </body>
</html>
```

It will produce the following result −

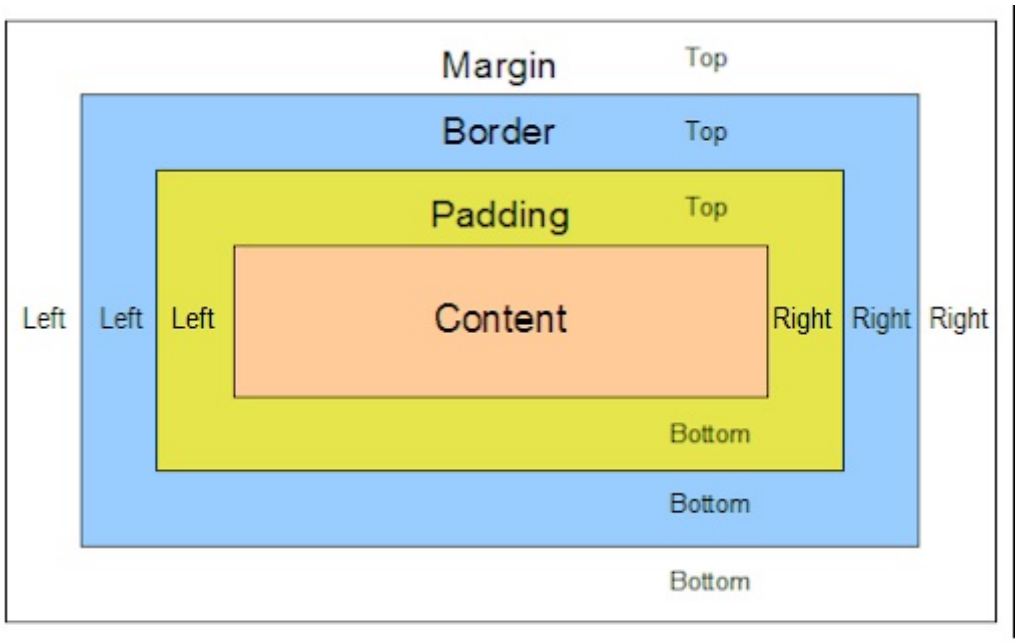
CSS Box Model

All HTML elements can be considered as boxes.

The CSS Box Model

In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

Example

Demonstration of the box model:

```
div {
  width: 300px;
  border: 15px solid green;
```

```
  padding: 50px;
  margin: 20px;
}
```

CSS – Box 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 learn how we can change the dimensions of boxes.

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 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 an example −

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

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

It will produce the following result −

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 an example −

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

   <body>
```

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

It will produce the following result −

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 an example −

```
<html>
   <head>
   </head>
   <body>
      <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>
      <br>
      <br>
      <br>
      <img alt = "logo" src = "/css/images/logo.png" width = "195" height =
"84" />
   </body>
</html>
```

It will produce the following result −

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 an example −

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

   <body>
      <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>
      <img alt = "logo" src = "/css/images/logo.png" width = "95" height = "84"
/>
   </body>
</html>
```

It will produce the following result −

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 an example −

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

   <body>
      <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>
      <img alt = "logo" src = "/images/css.gif" width = "95" height = "84" />
   </body>
</html>
```

This will produce following result −

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 an example −

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

   <body>
      <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
      </p>
      <img alt = "logo" src = "/css/images/css.gif" width = "95" height = "84"
/>
   </body>
</html>
```