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.

Applications 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:

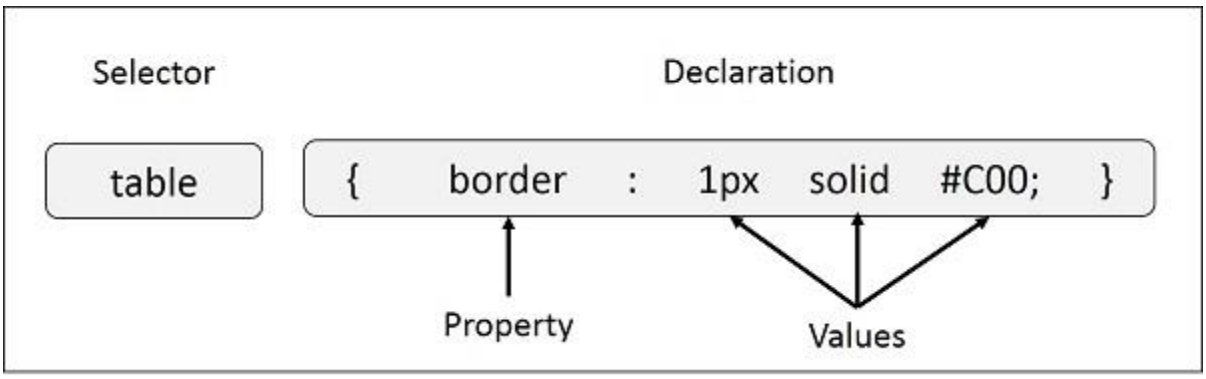
- **CSS saves time** - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write 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.

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.

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;
}
```

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 −

Live Demo

Live Demo

```
<!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	Specifies the device the document will be displayed on. Default value is *all*. This is an optional attribute.

	braille aural all	

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 −

Live Demo

```
<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 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 Background

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.

Live Demo

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

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

This will produce following result −

Set the Background Image

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

Live Demo

```
<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.

Live Demo

```
<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.

Live Demo

```
<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.

Live Demo

```
<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.

Live Demo

```
<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.

Live Demo

```
<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.

Live Demo

```
<!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.

Live Demo

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

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

CSS Colors - Hex Codes

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

A hexadecimal value can be taken from any graphics software like Adobe Photoshop, 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 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.

Live Demo

```
<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*.

Live Demo

```
<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.*.

Live Demo

```
<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*.

Live Demo

```
<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*.

Live Demo

```
<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*.

Live Demo

```
<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*.

Live Demo

```
<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*.

Live Demo

```
<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*.

Live Demo

```
<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>
```

CSS Font

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.

Live Demo

```
<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 −

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*.

Live Demo

```
<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 −

Set the Font Variant

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

Live Demo

```
<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 −

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*.

Live Demo

```
<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 −

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 %*.

Live Demo

```
<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 −

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.

Live Demo

```
<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 −

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*.

Live Demo

```
<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 −

Shorthand Property

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

Live Demo

```
<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>
```

CSS border

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 −

Live Demo

```
<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 −

Live Demo

```
<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 −

Live Demo

```
<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.

Live Demo

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

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

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.

CSS Margin

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 −

Live Demo

```
<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 −

Live Demo

```
<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 −

Live Demo

```
<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 −

Live Demo

```
<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 −

Live Demo

```
<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>
```

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.

CSS padding

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 −

Live Demo

```
<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 −

Live Demo

```
<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 −

Live Demo

```
<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 −

Live Demo

```
<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 −

Live Demo

```
<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>
```