CSS Introduction

What is CSS?

- **CSS** stands for **C**ascading **S**tyle **S**heets
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External style sheets are stored in **CSS files**

Why Use CSS?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

Advantages of CSS

- **CSS saves time −** You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster −** If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply 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.
- **Offline Browsing −** CSS can store web applications locally with the help of an offline cache. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.
- **Platform Independence −** The Script offer consistent platform independence and can support latest browsers as well.

Who Creates and Maintains CSS?

CSS was invented by **Håkon Wium Lie** on October 10, 1994 and maintained through a group of people within the W3C called the CSS Working Group. The CSS Working Group creates documents called **specifications**. When a specification has been discussed and officially ratified by W3C members, it becomes a recommendation.

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

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

CSS Versions:

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

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

CSS3 was became a W3C recommendation in June 1999 and builds on older versions CSS. It has divided into documentations is called as Modules and here each module having new extension featured defined in CSS2.

CSS3 Modules

CSS3 Modules are having old CSS specifications as well as extension features.

- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface

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 bg*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 <em> 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 within 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 <div> 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 **semi colon (;)**. You can keep them in a single line or multiple lines. For better readability we keep them into separate lines.

For a while, don't bother about the properties mentioned in the above block. These properties will be explained in the coming chapters and you can find complete detail about properties in https://www.tutorialspoint.com/css/css_references.htm

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 - 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.

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

Check the result?

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

Check the 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>
```

Check the 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>
```

Check the 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>
```

Check the 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</>
</body>
</html>
```

Check the 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</>
</body>
</html>
```

Check the 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>
```

Check the 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>
```

Check the 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>
```

Check the 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*.

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

Check the 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*.

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

Check the 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*.

```
<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>
```

Check the 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 %*.

```
<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>
```

Check the 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.

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

Check the 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*.

```
<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
condensed or expanded version of the font being used.
</p>
</body>
</html>
```

Check the result?

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>
```

Check the result?

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>
```

Check the 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 renedered from right to left
</p>
</body>
</html>
```

Check the 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>
```

Check the 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>
```

Check the 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>
```

Check the 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>
```

Check the 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>
```

Check the 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>
```

Check the 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>
```

Check the 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>
```

Check the result?

CSS - Using Images

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>
```

Check the 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>
```

Check the 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>
```

Check the 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>
```

Check the result?

CSS – Tables

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

- The **border-collapse** specifies whether the browser should control the appearance of 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>
```

Check the 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>
```

Check the 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>
```

Check the 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>
```

Check the 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%">1000000000000000000000000000</td>
<td width="40%">10000000</td>
<td width="40%">100</td>
</tr>
</table>
<br />
```

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

Check the result?

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 −

```
<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>
```

Check the 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 dahsed border.
</p>
<p style="border-width:4px; border-style:double;">
This is a double border.
</p>
<p style="border-width:4px; border-style:groove;">
This is a groove border.
</p>
<p style="border-width:4px; border-style:ridge">
This is aridge border.
</p>
<p style="border-width:4px; border-style:inset;">
```

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

Check the 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>
```

Check the 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>
```

Check the 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 –

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

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

Value	Description	Example
decimal	Number	1,2,3,4,5
decimal-leading-zero	0 before the number	01, 02, 03, 04, 05
lower-alpha	Lowercase alphanumeric characters	a, b, c, d, e
upper-alpha	Uppercase alphanumeric characters	A, B, C, D, E
lower-roman	Lowercase Roman numerals	i, ii, iii, iv, v
upper-roman	Uppercase Roman numerals	I, II, III, IV, V
lower-greek	The marker is lower-greek	alpha, beta, gamma
lower-latin	The marker is lower-latin	a, b, c, d, e
upper-latin	The marker is upper-latin	A, B, C, D, E
hebrew	The marker is traditional Hebrew numbering	
armenian	The marker is traditional Armenian numbering	
georgian	The marker is traditional Georgian numbering	
cjk-ideographic	The marker is plain ideographic numbers	
hiragana	The marker is hiragana	a, i, u, e, o, ka, ki
katakana	The marker is katakana	A, I, U, E, O, KA, KI
hiragana-iroha	The marker is hiragana-iroha	i, ro, ha, ni, ho, he, to
katakana-iroha	The marker is katakana-iroha	I, RO, HA, NI, HO, HE, TO

Here is 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>
```

Check the 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

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

Here is an example −

```
<html>
<head>
</head>
<body>
<ul style="list-style-type:circle; list-style-position:outside;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ul>
<ul style="list-style-type:square;list-style-position:inside;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ul>
<ol style="list-style-type:decimal;list-style-position:outside;">
<li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
```

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

Check the 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>
```

Check the 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>
```

Check the result?

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>
```

Check the result?