

CSS



Project Supervisor: Muzammil Hassan | Research Officer



Al-Khwarizmi Institute of Computer Science (KICS)

Office: +92-42-9250245 | Fax: +92-42-9250246

**Al-Khawarizmi Institute of Computer Science
(KICS) UET, Lahore**

<http://www.kics.edu.pk>



Table of Contents

About the Tutorial	vii
Prerequisites	vii
1. What is CSS(Cascading Style Sheet)	1
Advantages of CSS.....	1
• CSS saves time -.....	1
• Pages load faster.....	1
• Easy maintenance	1
• Superior styles to HTML.....	1
• Multiple Device Compatibility	1
• Global web standards	1
CSS Versions.....	1
Difference Between Different Versions.....	2
2. CSS Syntax	1
• Selector:.....	1
• Property:	1
• Value:	1
The Type Selectors.....	1
The Universal Selectors	1
The Descendant Selectors	2
The Class Selectors	2
The ID Selectors.....	3
The Child Selectors	4
The Attribute Selectors.....	4
Multiple Style Rules.....	5
Grouping Selectors.....	5
3. INCLUSION.....	6
Embedded CSS - The <style> Element.....	6
Attributes	6
Inline CSS - The style Attribute.....	7
Attributes	7
External CSS - The <link> Element.....	7
Attributes	8

Imported CSS - @import Rule.....	8
CSS Rules Overriding	9
Handling Old Browsers	9
CSS Comments	10
4. COLORS.....	11
Building Colors Codes.....	12
Browser Safe Colors.....	12
5. Background	15
Set the Background Color	15
Set the Background Image	16
Repeat the Background Image	16
Set the Background Image Position	17
Set the Background Attachment.....	18
Shorthand Property.....	18
6. Fonts.....	19
Set the Font Family	19
Set the Font Style	19
Set the Font Variant.....	19
Set the Font Size.....	20
Set the Font Size Adjust.....	20
Shorthand Property.....	21
7. Text.....	22
Set the Text Color	22
Set the Text Direction	22
Set the Space between Characters	23
Set the Space between Words	23
Set the Text Indent	23
Set the Text Alignment	24
Decorating the Text.....	24
Set the Text Cases	25
8. Image.....	26
The Image Border Property.....	26
The Image Height And Width Property	26

9.	Links.....	28
	Set the Color of Links	28
	Set the Color of Visited Links.....	28
	Change the Color of Links when Mouse is Over	29
	Change the Color of Active Links.....	29
10.	Tables	30
	The order-collapse Property	30
	The border-spacing Property	31
	The caption-side Property	33
	The empty-cells Property	35
	The table-layout Property.....	36
11.	Borders.....	38
	The border-style Property.....	39
	The border-width Property	41
	Border Properties Using Shorthand.....	42
12.	Margins	43
	The Margin Property	43
	The margin-bottom Property.....	44
	The margin-top Property	45
	The margin-left Property	45
	The margin-right Property.....	46
13.	List.....	47
	The list-style-type Property	47
	The list-style-position Property	50
	The list-style-image Property	51
	The list-style Property	52
	The marker-offset Property	53
14.	Padding	54
	The padding-bottom Property	54
	The padding-top Property.....	55
	The padding-left Property	55
	The padding-right Property	56
	The Padding Property.....	56

15.	Dimensions.....	58
	The Height and Width Properties.....	58
	The line-height Property.....	58
	The max-height Property.....	59
	The min-height Property	60
	The max-width Property.....	60
	The min-width Property	61
16.	Positioning	63
	Relative Positioning	63
	Absolute Positioning	63
	Fixed Positioning	64
17.	Floating	65
	What is "Float"?	65

About the Tutorial

CSS is used to control the style of a web document in a simple and easy way. CSS stands for Cascading Style Sheets. This tutorial covers both the versions CSS1 and CSS2 and gives a complete understanding of CSS, starting from its basics to advanced concepts.

Prerequisites

You should be familiar with:

- Basic word processing using any text editor.
- How to create directories and files.
- How to navigate through different directories.
- Internet browsing using popular browsers like Internet Explorer or Firefox.
- Developing simple Web Pages using HTML or XHTML.

If you are new to HTML and XHTML, then we would suggest you to go through our [HTML Tutorial](#) or [XHTML Tutorial](#) first.

1. What is CSS(Cascading Style Sheet)

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

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

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

Advantages of CSS

- **CSS saves time** - You can write CSS once and then reuse the 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 the 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 cellphones or for printing.
- **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

CSS Versions

- CSS1
- CSS2
- CSS3

Difference Between Different Versions

Cascading Style Sheets level 1 (CSS1) 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 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 is the latest evolution of the Cascading Style Sheets language and aims at extending CSS2.1. It brings a lot of long-awaited novelties, like rounded corners, shadows, gradients, transitions or animations, as well as new layouts like multi-columns, flexible box or grid layouts.

2. 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 the value either red or #F1F1F1 etc.

You can put CSS Style Rule Syntax as follows:

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

Example: You can define a table border as follows:

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

Here table is a selector and border is a property and the given value 1px solid

#C00 is the value of that property.

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

The 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, the style rule will apply to `` element only when it lies inside the `` tag.

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

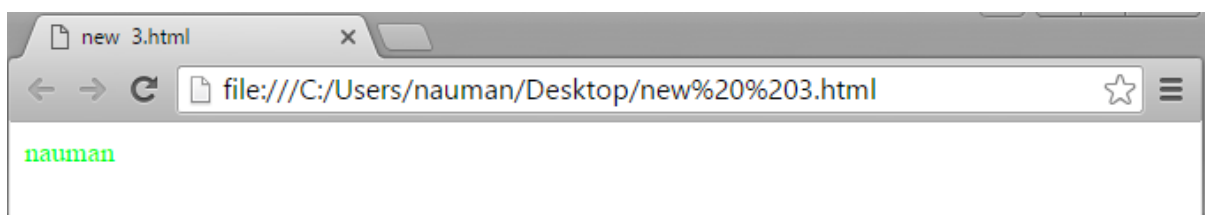
The Class Selectors

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.

Example

```
<html>  
<head>  
  <title>class</title>  
  <style type="text/css">  
    .black {  
      color: #00ff21;  
    }  
  </style>  
</head>  
<body>  
  <p class="black">nauman </p>  
</body>  
</html>
```

Result:



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.

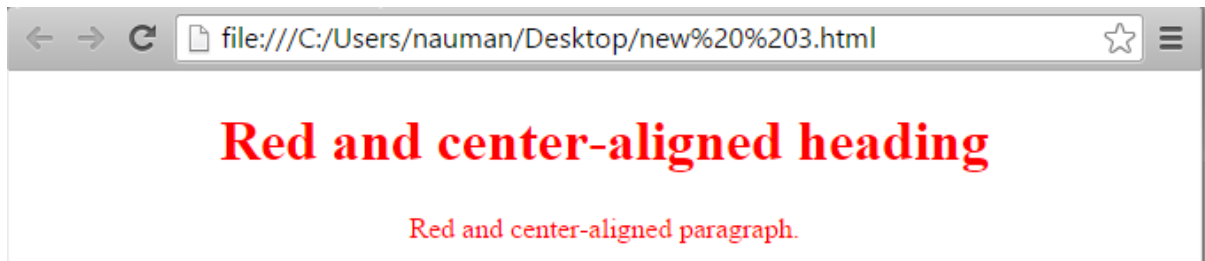
Example

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>
```

Result



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 a direct child of the `<body>` element. Other paragraphs put inside other elements like `<div>` or `<td>` would not have any effect of this rule.

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 the 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 class selectors together as shown below:

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

3. 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:

```
<head>

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

.....

</style>

</head>
```

Attributes

Attributes associated with <style> elements are:

Attribute	Value	Description
type	text/css	Specifies the style sheet language as a content-type (MIME type). This is a 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

Example

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

```
<head>

<style type="text/css" media="all">

h1{

color: #36C;

}

</style>
```



```
</head>
```

Inline CSS - The style Attribute

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

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

Attributes

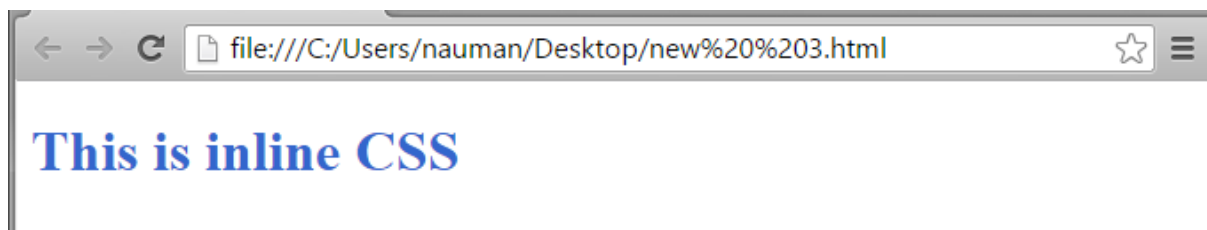
Attribute	Value	Description
style	style rules	The value of style attribute is a combination of style declarations separated by semicolon (;).

Example

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

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

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 an 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 an HTML document:

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

CSS Rules Overriding

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

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

Handling Old Browsers

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

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

CSS Comments

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

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

Example

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

4. 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 for the background of the element. They can also be used to affect the color of borders and other decorative effects.

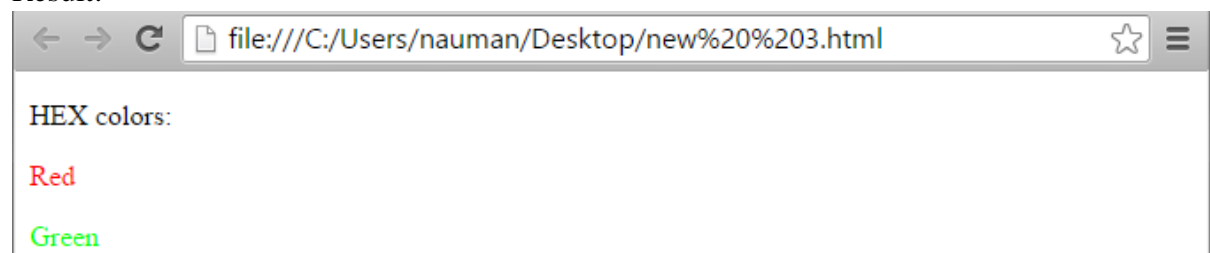
You can specify your color values in various formats. Following table lists all the possible formats:

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;}

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
#p1 {color:#ff0000;}
#p2 {color:#00ff00;}
</style>
</head>
<body>
<p>HEX colors:</p>
<p id="p1">Red</p>
<p id="p2">Green</p>
</body>
</html>
```

Result:



Building Colors 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 FFFFFFFF. 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

5. Background

This chapter teaches you how to set backgrounds of various HTML elements. You can set the following background properties of an element:

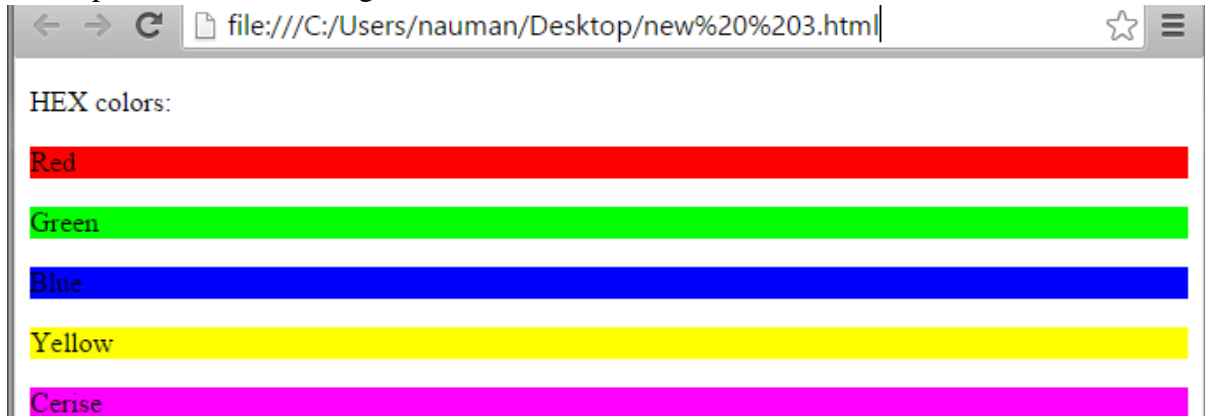
- The background-color property is used to set the background color of an element.
- The background-image property is used to set the background image of an element.
- The background-repeat property is used to control the repetition of an image in the background.
- The background-position property is 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.

```
<!DOCTYPE html>
<html>
<head>
<style>
#p1 {background-color:#ff0000;}
#p2 {background-color:#00ff00;}
#p3 {background-color:#0000ff;}
#p4 {background-color:#ffff00;}
#p5 {background-color:#ff00ff;}
</style>
</head>
<body>
<p>HEX colors:</p>
<p id="p1">Red</p>
<p id="p2">Green</p>
<p id="p3">Blue</p>
<p id="p4">Yellow</p>
<p id="p5">Cerise</p>
</body>
</html>
```

It will produce the following result:



Set the Background Image

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

```
<tr><td>
```

This table has background image set.

```
</td></tr>
```

```
</table>
```

Repeat the Background Image

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

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

```
<table style="background-image:url(/images/pattern1.gif);
```

```
background-repeat: repeat;">
```

```
<tr><td>
```

This table has background image which repeats multiple times.

```
</td></tr>
```

```
</table>
```

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

```
<table style="background-image:url(/images/pattern1.gif);
```

```
background-repeat: repeat-y;">
```

```
<tr><td>
```

This table has background image set which will repeat vertically.

```
</td></tr>
```

```
</table>
```

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

```
<table style="background-image:url(/images/pattern1.gif);
```

```
background-repeat: repeat-x;">
```

```
<tr><td>
```

This table has background image set which will repeat horizontally.

```
</td></tr>
```

```
</table>
```

Set the Background Image Position

The following example demonstrates how to set the background image position

100 pixels away from the left side.

```
<table style="background-image:url(/images/pattern1.gif);
```

```
background-position:100px;">
```

```
<tr><td>
```

Background image positioned 100 pixels away from the left.

```
</td></tr>
```

```
</table>
```

The following example demonstrates how to set the background image position

100 pixels away from the left side and 200 pixels down from the top.

```
<table style="background-image:url(/images/pattern1.gif);
```

```
background-position:100px 200px;">
```

```
<tr><td>
```

This table has background image positioned 100

pixels away from the left and 200 pixels from the top.

```
</td></tr>
```

```
</table>
```

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.

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

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

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

Shorthand Property

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

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

6. Fonts

This chapter teaches you how to set fonts of a content, available in an HTML element. You can set the following font properties of an element:

- The font-family property is used to change the face of a font.
- The font-style property is used to make a font italic or oblique.
- The font-variant property is used to create a small-caps effect.
- The font-weight property is used to increase or decrease how bold or light a font appears.
- The font-size property is used to increase or decrease the size of a font.
- The font property is used as shorthand to specify a number of other font properties.

Set the Font Family

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

```
<p style="font-family:georgia,garamond,serif;">
```

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

```
</p>
```

It will produce the 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

The following example demonstrates how to set the font style of an element. Possible values are normal, italic and oblique.

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

This text will be rendered in italic style

```
</p>
```

It will produce the following result:

This text will be rendered in italic style

Set the Font Variant

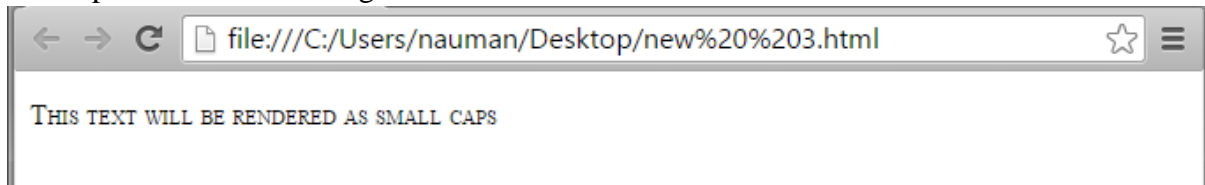
The following example demonstrates how to set the font variant of an element. Possible values are normal and small-caps.

```
<p style="font-variant:small-caps;">
```

This text will be rendered as small caps

</p>

It will produce the 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 %.

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

</p>

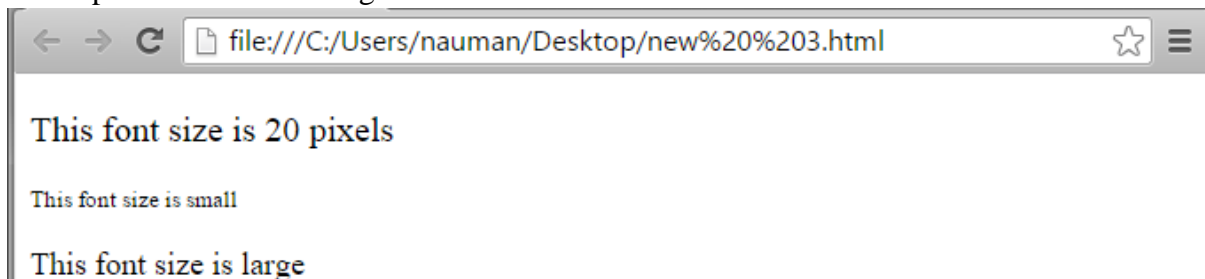
<p style="font-size:small;"> This font size is small

</p>

<p style="font-size:large;"> This font size is large

</p>

It will produce the following result:



Set the Font Size Adjust

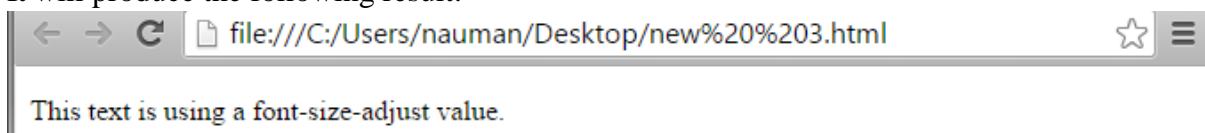
The following example demonstrates how to set the font size adjust of an element. This property enables you to adjust the x-height to make fonts more legible. Possible value could be any number.

<p style="font-size-adjust:0.61;">

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

</p>

It will produce the following result:



Shorthand Property

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

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

```
</p>
```

It will produce the following result:



7. Text

This chapter teaches you how to manipulate text using CSS properties. You can set the following text properties of an element:

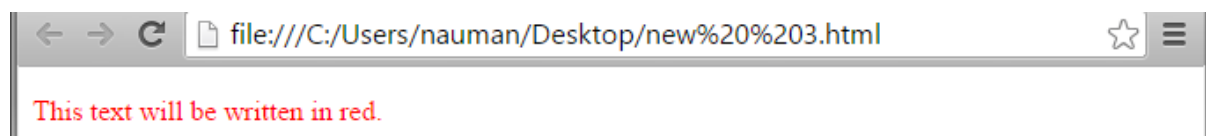
- The color property is used to set the color of a text.
- The direction property is used to set the text direction.
- The letter-spacing property is used to add or subtract space between the letters that make up a word.
- The word-spacing property is used to add or subtract space between the words of 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.

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

It will produce the following result:

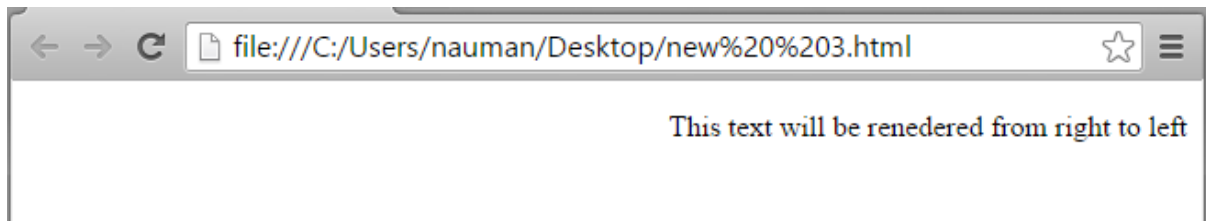


Set the Text Direction

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

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


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.

```
<p style="letter-spacing:5px;">
```

This text is having space between letters.

```
</p>
```

It will produce the following result:

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

Set the Space between Words

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

```
<p style="word-spacing:5px;">
```

This text is having space between words.

```
</p>
```

It will produce the following result:

This text is having space between words.

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.

```
<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>
```

It will produce the following result:

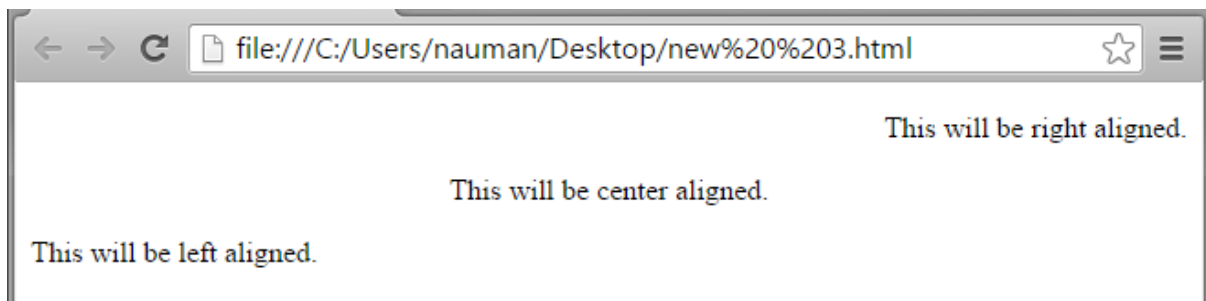
This text will have first line indented by 1cm
and this line will remain at its actual
position this is done by CSS text-indent property.

Set the Text Alignment

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

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

It will produce the following result:

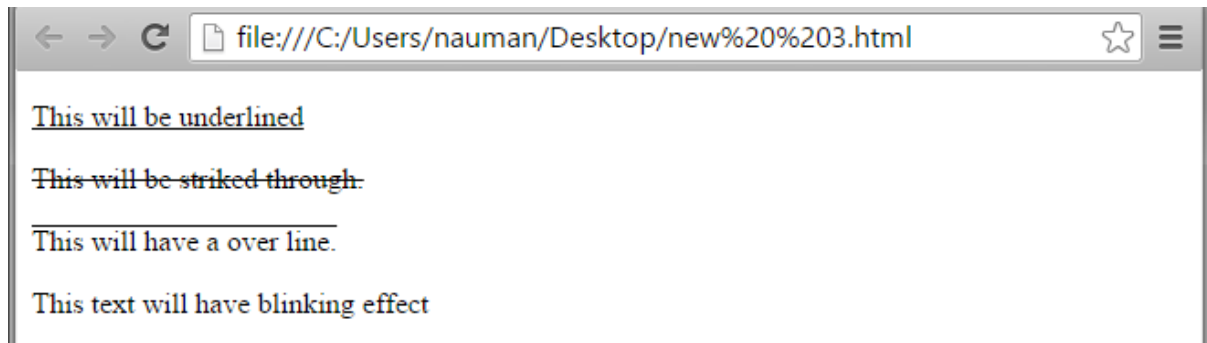


Decorating the Text

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

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

It will produce the 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.

```
<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>
```

It will produce the following result:

This Will Be Capitalized

THIS WILL BE IN UPPERCASE

this will be in lowercase

8. Image

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

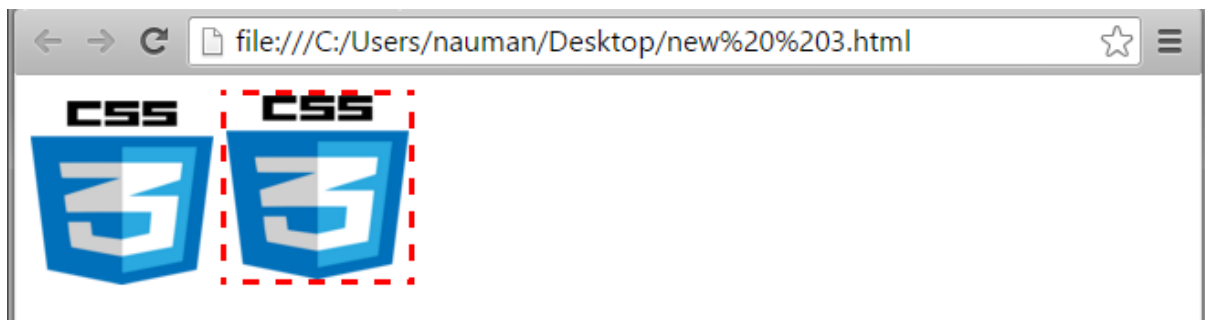
The border property of an image is used to set the width of an image border. This property can have a value in length or in %.

A width of zero pixels means no border. Here is an example:

```
  
  

```

It will produce the following result:



The Image Height And Width Property

The height property of an image is used to set the height and 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:

```
  
  

```

It will produce the following result:



9. Links

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

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

- The `:link` signifies unvisited hyperlinks.
- The `:visited` signifies visited hyperlinks.
- The `:hover` signifies an element that currently has the user's mouse pointer hovering over it.
- The `:active` signifies an element on which the user is currently clicking. Usually, all these properties are kept in the header part of the HTML document.

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

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

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.

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

It will produce the following black link:

Black Link

Set the Color of Visited Links

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

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

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

Click this link

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.

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

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.

Bring Mouse Here

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.

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

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

Click This Link

10. Tables

This chapter teaches you how to set different properties of an HTML table using CSS. You can set the following properties of a table:

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

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:

```
<style type="text/css">
table.one { border-collapse:collapse;}
table.two { border-collapse:separate;}
td.a {
border-style:dotted; border-width:3px; border-color:#000000; padding: 10px;
}
td.b {border-style:solid; border-width:3px; border-color:#333333;
padding:10px;
}
</style>
<table class="one">
<caption>Collapse Border Example</caption>
<tr><td class="a"> Cell A Collapse Example</td></tr>
<tr><td class="b"> Cell B Collapse Example</td></tr>
</table>
```

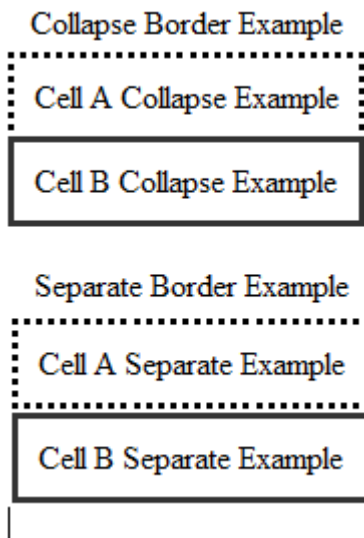


```

<br />
<table class="two">
<caption>Separate Border Example</caption>
<tr><td class="a"> Cell A Separate Example</td></tr>
<tr><td class="b"> Cell B Separate Example</td></tr>
</table>

```

It will produce the following result:



Collapse Border Example
 Cell A Collapse Example
 Cell B Collapse Example
 Separate Border Example
 Cell A Separate Example
 Cell B Separate Example

The border-spacing Property

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

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

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

```

<style type="text/css">
/* 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:

<style type="text/css">
table.one {
border-collapse:separate;
width:400px;
border-spacing:10px;
}
table.two {
border-collapse:separate;
width:400px;
border-spacing:10px 50px;
}
</style>

<table class="one" border="1">
<caption>Separate Border Example with border-spacing</caption>
<tr><td> Cell A Collapse Example</td></tr>
<tr><td> Cell B Collapse Example</td></tr>
46
</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>

```

It will produce the following result:

Now let's modify the previous example and see the effect:
 Separate Border Example with border-spacing

Cell A Collapse Example

Cell B Collapse Example

Separate Border Example with border-spacing

Cell A Separate Example

Cell B Separate Example

Separate Border Example with border-spacing

Cell A Collapse Example

Cell B Collapse Example

Separate Border Example with border-spacing

Cell A Separate Example

Cell B Separate Example

The caption-side Property

The caption-side property allows you to specify where the content of a

<caption> element should be placed in relationship to the table. The table that follows lists the possible values.

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

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

```
<style type="text/css"> caption.top {caption-side:top} caption.bottom {caption-side:bottom}
caption.left {caption-side:left}
caption.right {caption-side:right}
</style>
<table style="width:400px; border:1px solid black;">
<caption class="top">
This caption will appear at the top
</caption>
<tr><td > Cell A</td></tr>
<tr><td > Cell B</td></tr>
```

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

It will produce the following result:

Cell A
Cell B

This caption will appear at the bottom

Cell A
Cell B

This caption will appear at the left

Cell A
Cell B

This caption will appear at the right

Cell A
Cell B

The empty-cells Property

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

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

Here is the empty-cells property used to hide borders of empty cells in the

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

```

<th></th>
<th>Title one</th>
<th>Title two</th>
</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>

```

It will produce the following result:

element.

	Title one	Title two
Row Title	value	value
Row Title	value	

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.

```

<style type="text/css">
table.auto
{

```

```

table-layout: auto
}
table.fixed
{
table-layout: fixed
}
</style>
<table class="auto" border="1" width="100%">
<tr>
<td width="20%">100000000000000000000000000000000</td>
<td width="40%">100000000</td>
<td width="40%">100</td>
</tr>
</table>
<br />
<table class="fixed" border="1" width="100%">
<tr>
<td width="20%">100000000000000000000000000000000</td>
<td width="40%">100000000</td>
<td width="40%">100</td>
</tr>
</table>

```

It will produce the following result:

100000000000000000000000000000000	100000000	100
100000000000000000000000000000000	100000000	100

11. 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:

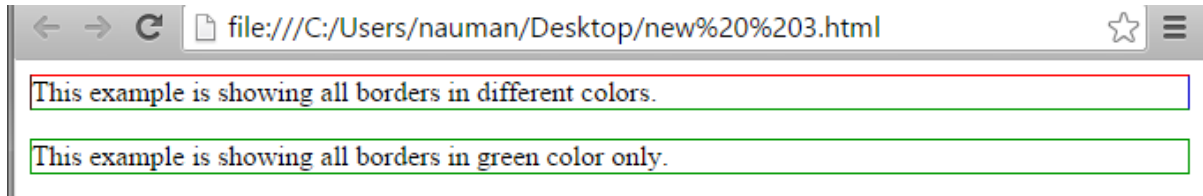
- border-bottom-color changes the color of bottom border.
- border-top-color changes the color of top border.
- border-left-color changes the color of left border.
- border-right-color changes the color of right border. The following example shows the effect of all these properties:

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


This example is showing all borders in green color only.

</p>

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:

<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 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;">

55

This is a hidden border.

</p>

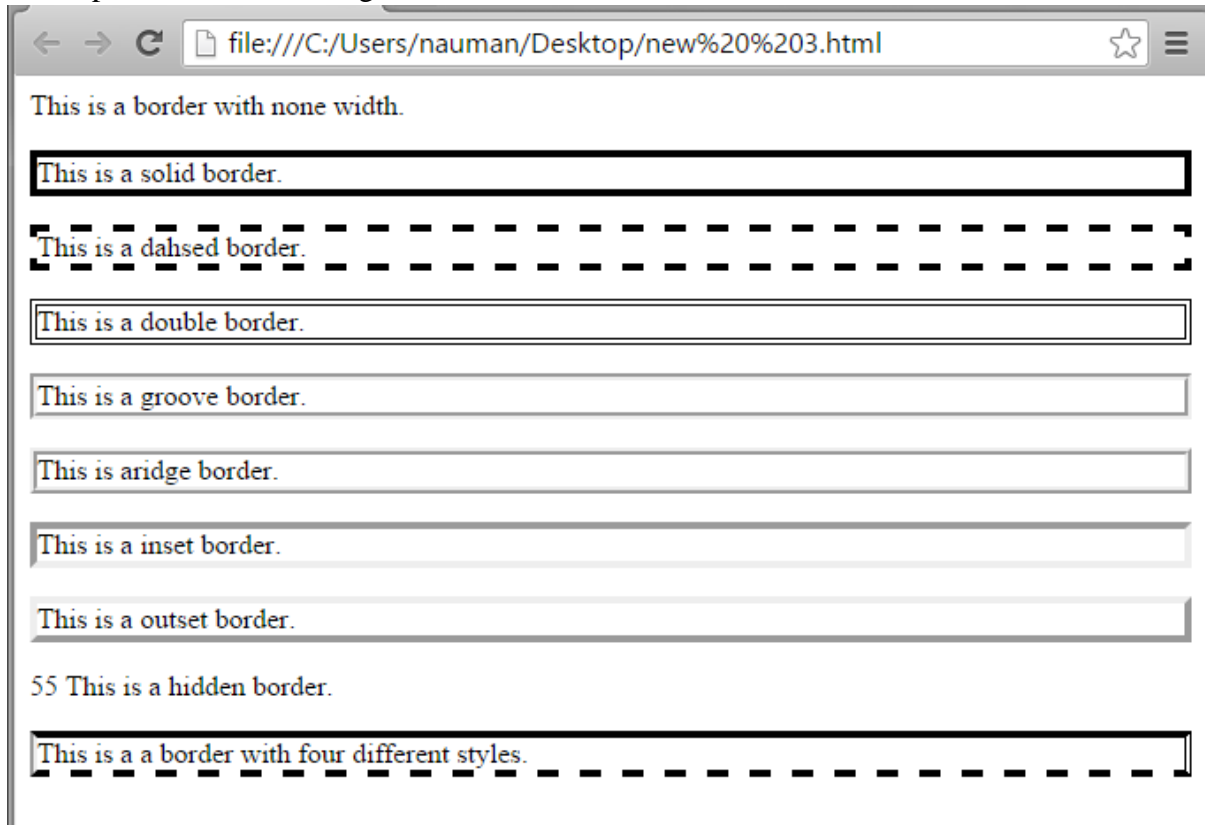
<p style="border-width:4px;

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>

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:

```
<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.
```

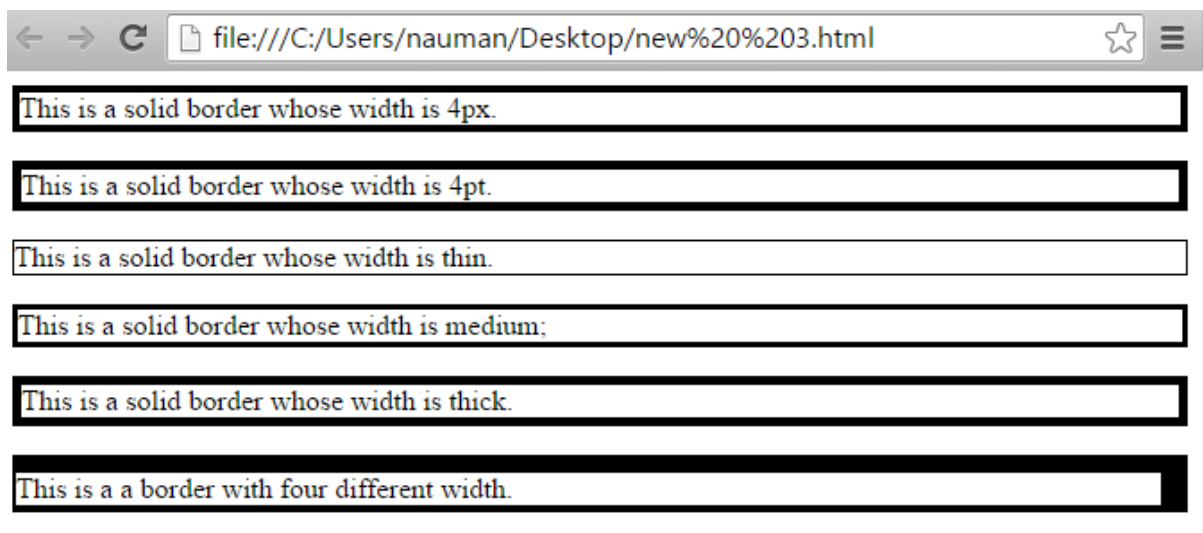
```
</p>
```

```
<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>
```

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:

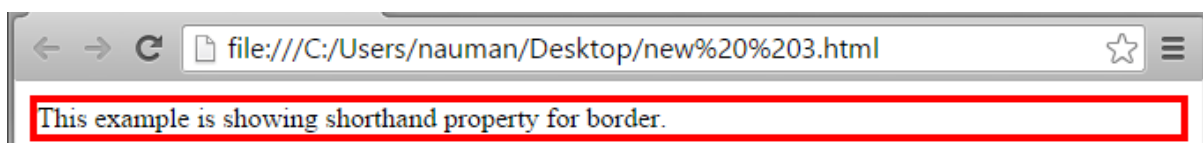
The following example shows how to use all the three properties into a single property. This is the most frequently used property to set border around any element.

```
<p style="border:4px solid red;">
```

This example is showing shorthand property for border.

```
</p>
```

It will produce the following result:



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

```
<style type="text/css">
```

```
p {margin: 15px }
```

all four margins will be 15px

```
p {margin: 10px 2% }
```

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

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

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

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

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

```
</style>
```

Here is an example:

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

all four margins will be 15px

</p>

<p style="margin:10px 2%; border:1px solid black;">

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

</p>

<p style="margin: 10px 2% -10px; border:1px solid black;">

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

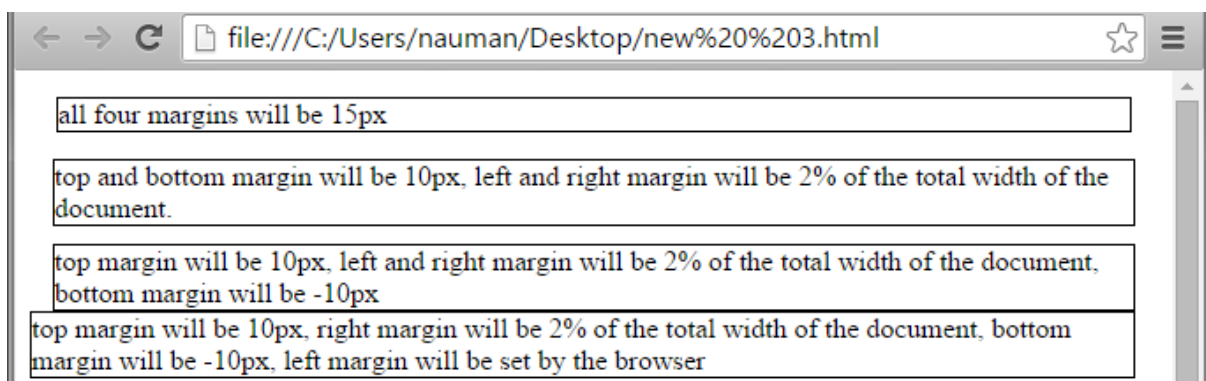
</p>

<p style="margin: 10px 2% -10px auto; border:1px solid black;">

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

</p>

It will produce the following result:



The margin-bottom Property

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

Here is an example:

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

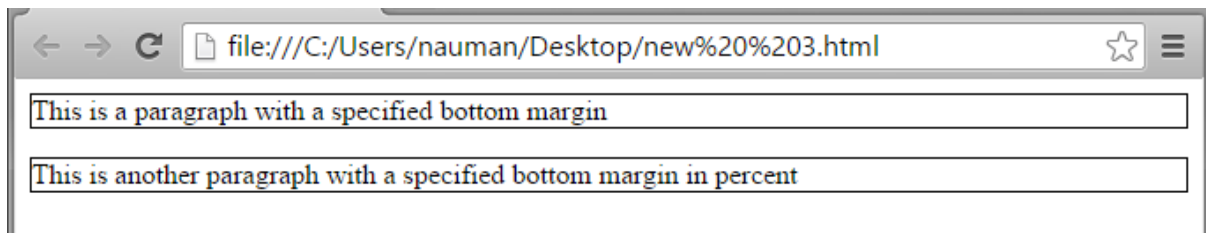
</p>

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

This is another paragraph with a specified bottom margin in percent

</p>

It will produce the following result:



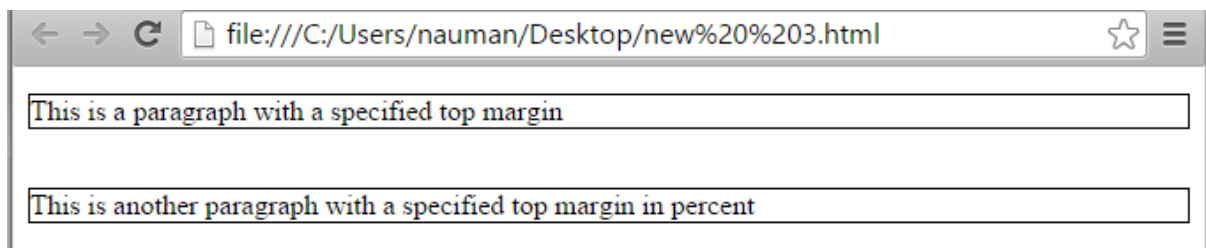
The margin-top Property

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

Here is an example:

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

It will produce the following result:



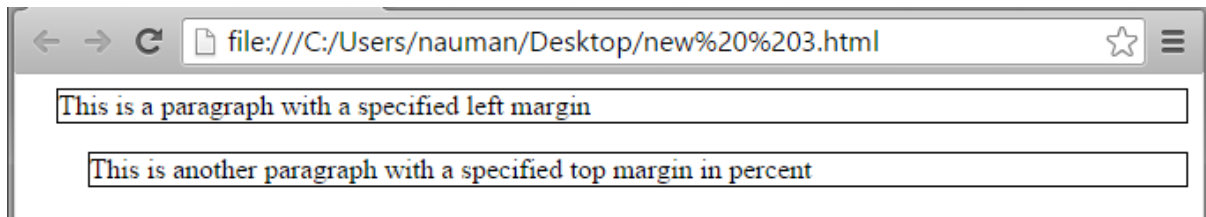
The margin-left Property

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

Here is an example:

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

It will produce the following result:



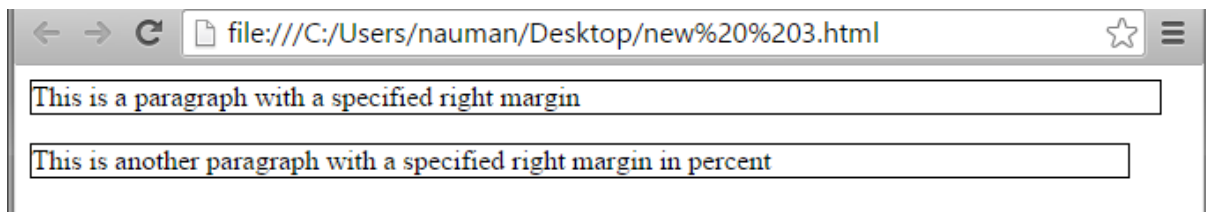
The margin-right Property

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

Here is an example:

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

It will produce the following result:



13. List

Lists are very helpful in conveying a set of either numbered or bulleted 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 a bullet point (also known as a marker) in 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

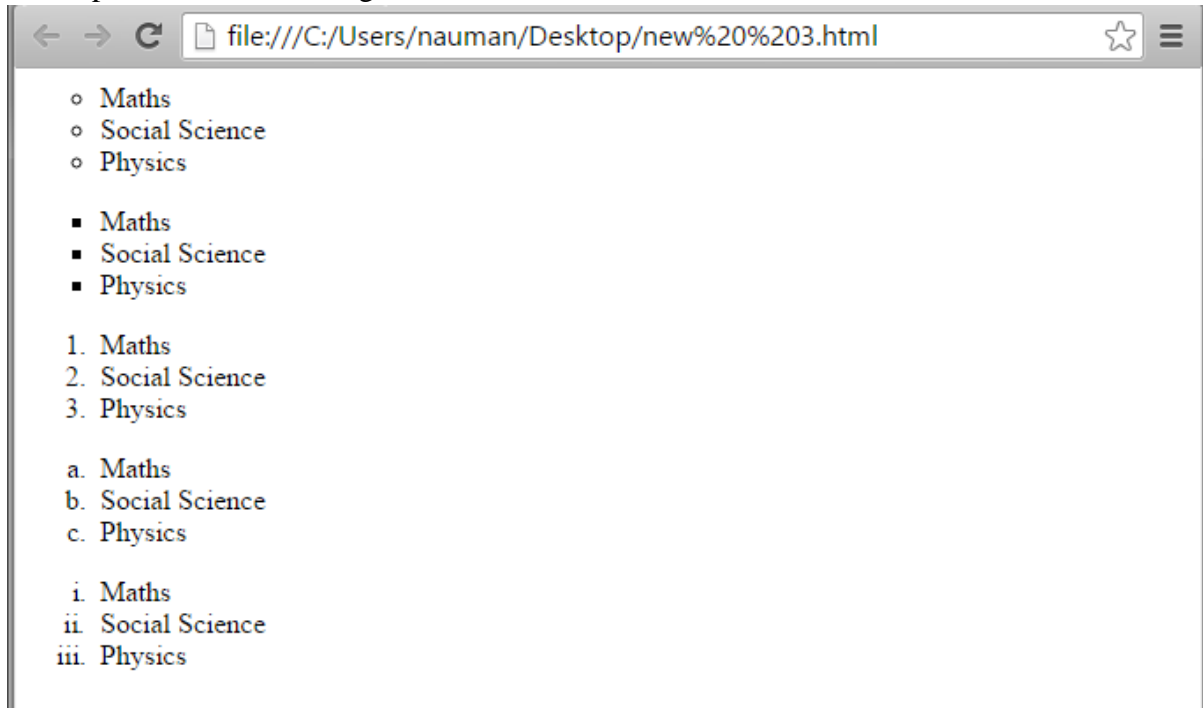
decimal-leading- zero	0 before the number	01, 02, 03, 04, 05
lower-alpha	Lowercase alphanumeric characters	a, b, c, d, e
upper-alpha	Uppercase alphanumeric characters	A, B, C, D, E
lower-roman	Lowercase Roman numerals	i, ii, iii, iv, v
upper-roman	Uppercase Roman numerals	I, II, III, IV, V
lower-greek	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:

```
<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>
```

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 of 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:

```
<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>
```

It will produce the following result:



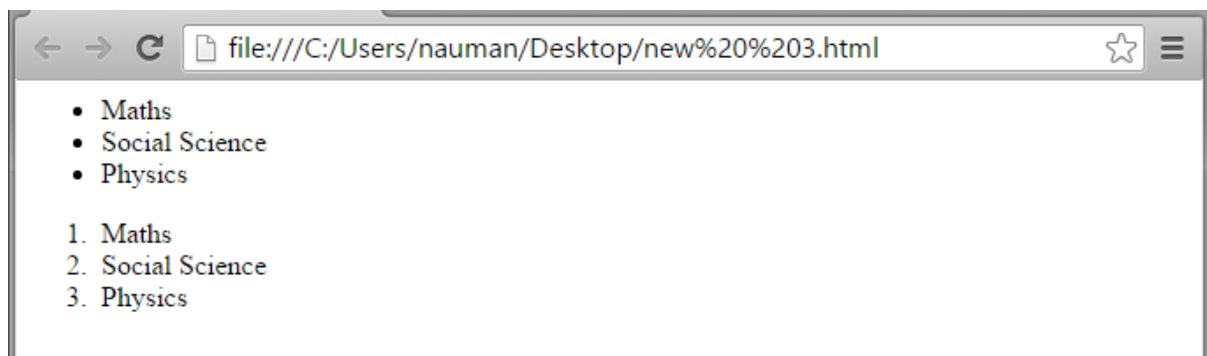
The list-style-image Property

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:

```
<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>
```

It will produce the following result:



The list-style Property

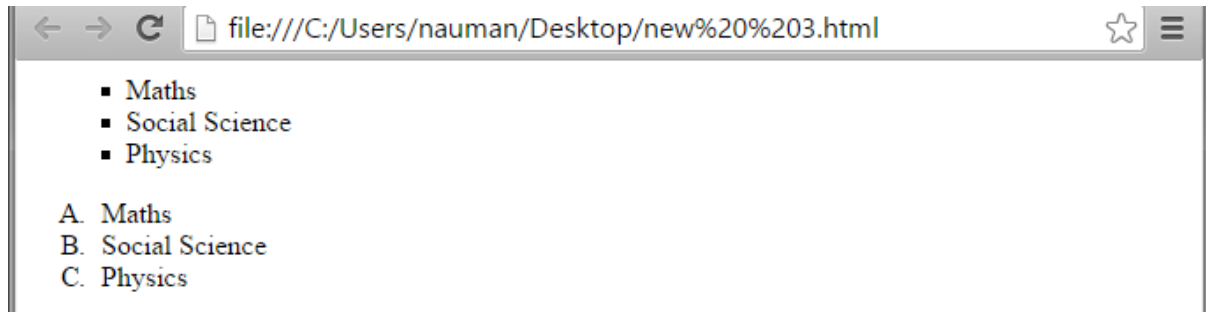
The list-style allows you to specify all the list properties into a single expression. These properties can appear in any order.

Here is an example:

```
<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>
```

It will produce the following result:



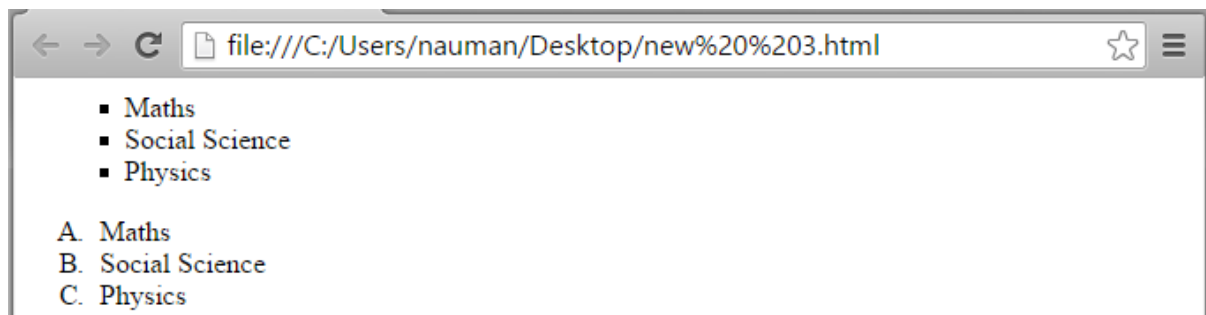
The marker-offset Property

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:

```
<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>
```

It will produce the following result:



14. Padding

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.

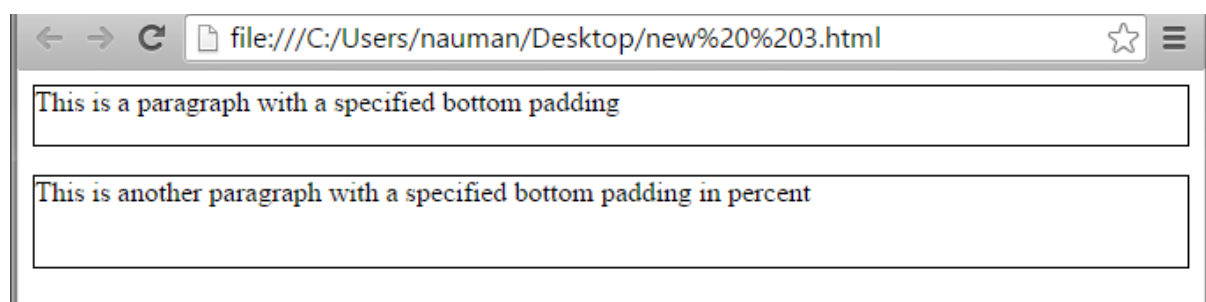
The padding-bottom Property

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:

```
<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>
```

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 or %.

Here is an example:

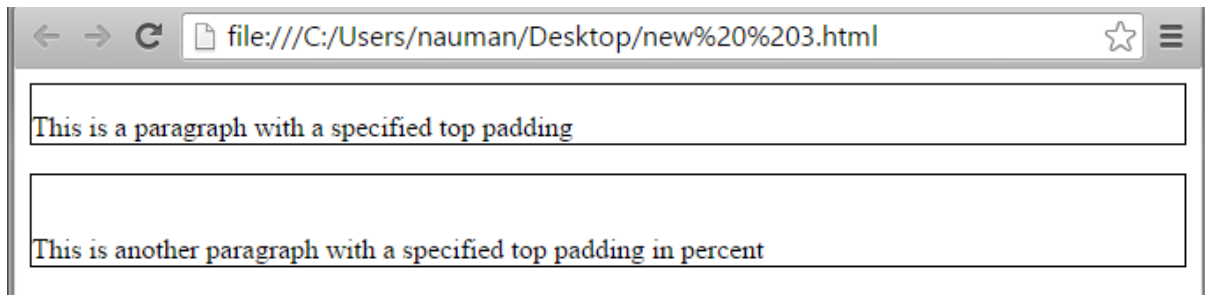
```
<p style="padding-top: 15px; border: 1px solid black;"> This is a paragraph with a specified top padding</p>

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

This is another paragraph with a specified top padding in percent

</p>
```

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 or %.

Here is an example:

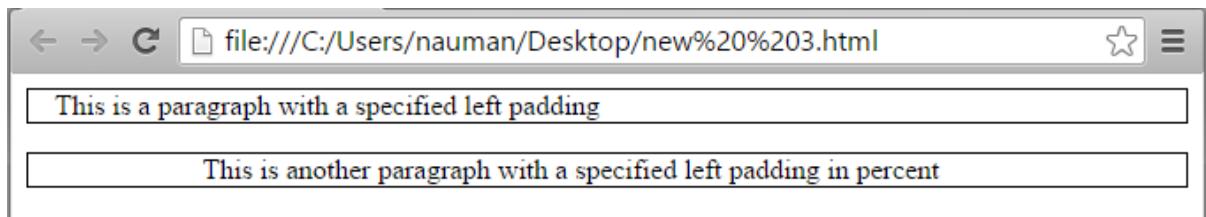
```
<p style="padding-left: 15px; border: 1px solid black;"> This is a paragraph with a specified left padding</p>

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

This is another paragraph with a specified left padding in percent

</p>
```

It will produce the following result:



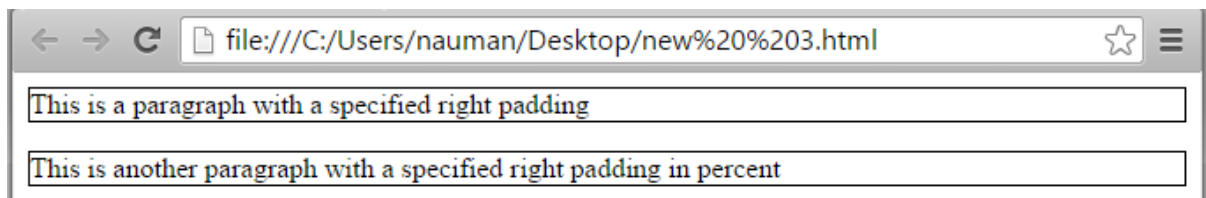
The padding-right Property

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

Here is an example:

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

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 or %.

Here is an example:

```
<p style="padding: 15px; border: 1px solid black;">
```

all four padding will be 15px

```
</p>
```

```
<p style="padding: 10px 2%; border: 1px solid black;">
```

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

```
</p>
```

```
<p style="padding: 10px 2% 10px; border:1px solid black;">
```

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

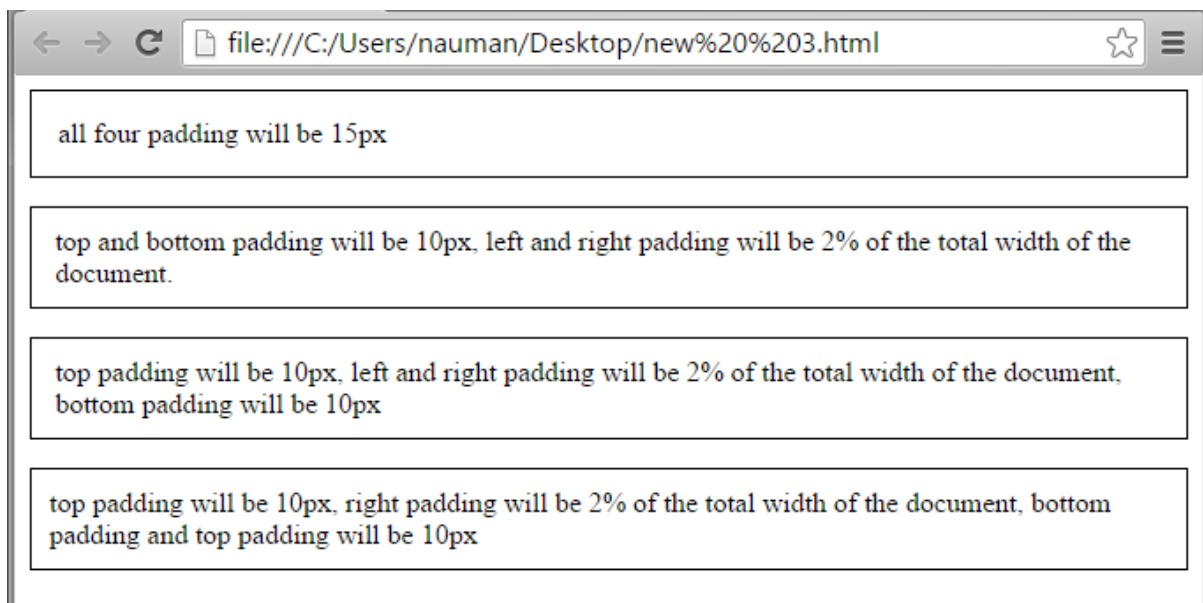
```
</p>
```

```
<p style="padding: 10px 2% 10px 10px; border:1px solid black;">
```

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

```
</p>
```

It will produce the following result:



15. Dimensions

You have seen the border that surrounds every box i.e. element, the padding that can appear inside each box, and the margin that can go around them. In this chapter, we will learn how to 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.

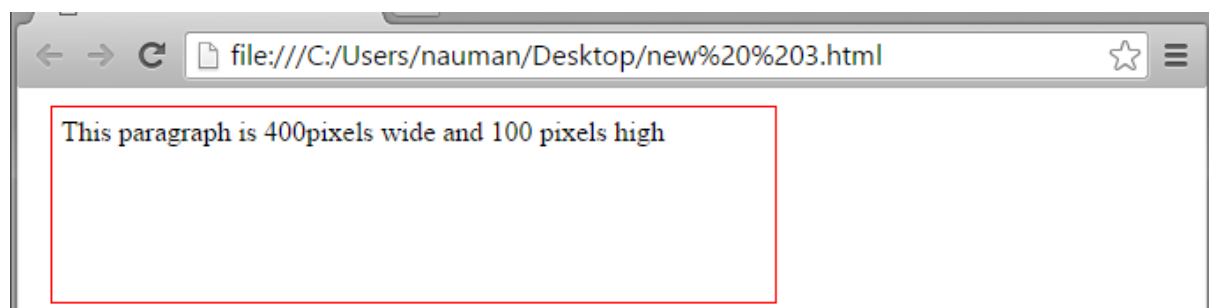
The Height and Width Properties

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:

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

It will produce the following result:



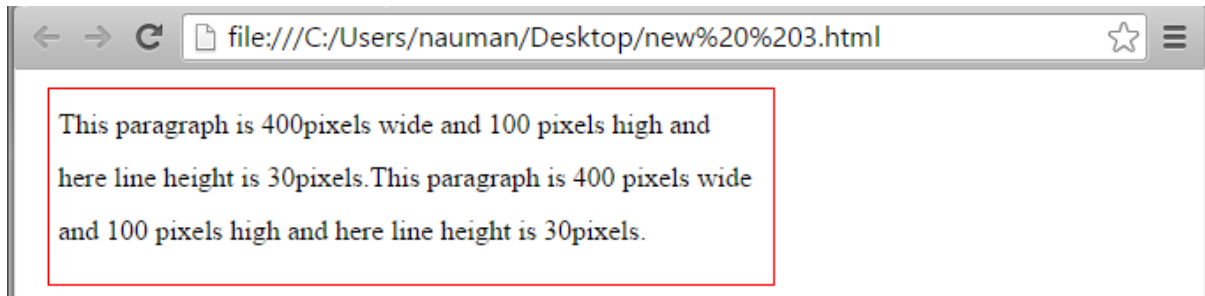
The line-height Property

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:

```
<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>
```

It will produce the following result:



The max-height Property

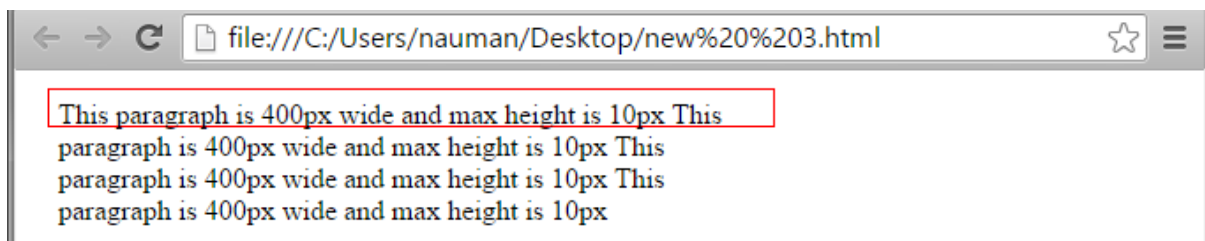
The max-height property allows you to specify the maximum height of a box.

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:

```
<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>
```

It will produce the following result:



The min-height Property

The min-height property allows you to specify the 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:

```
<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

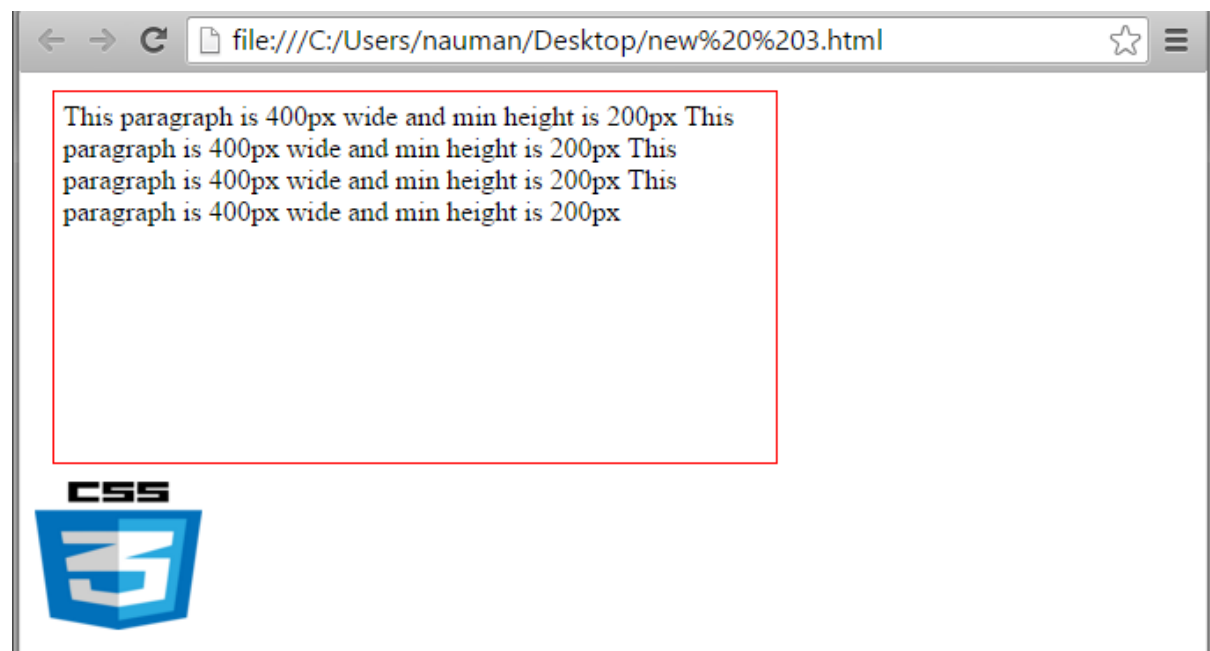
This paragraph is 400px wide and min height is 200px This paragraph is 400px wide and min height is 200px This paragraph is 400px wide and min height is 200px

```
</p>
```

```

```

It will produce the following result:



The max-width Property

The max-width property allows you to specify the 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:

```
<p style="max-width:100px; height:400px;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>

It will produce the following result:



The min-width Property

The min-width property allows you to specify the 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:

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

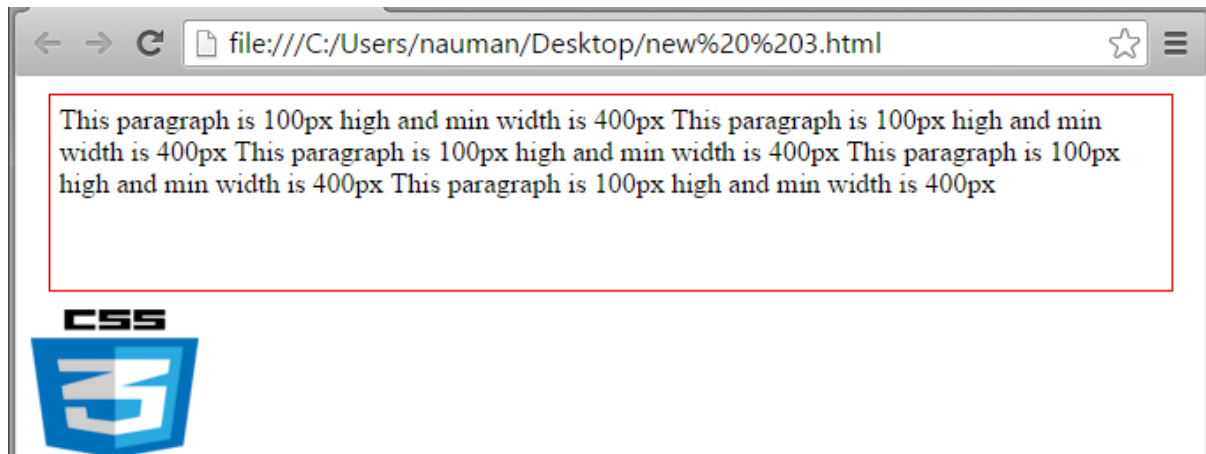
This paragraph is 100px high and min width is 400px This paragraph is 100px high and min width is 400px This paragraph is 100px high and min width is 400px

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

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

</p>

It will produce the following result:



16. Positioning

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 an HTML document.

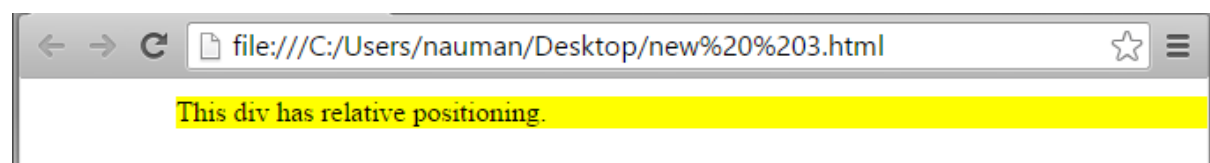
- Move Left - Use a negative value for left.
- Move Right - Use a positive value for left.
- Move Up - Use a negative value for top.
- Move Down - Use a positive value for top.

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

Here is an example:

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

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 HTML document.

- Move Left - Use a negative value for left.
- Move Right - Use a positive value for left.

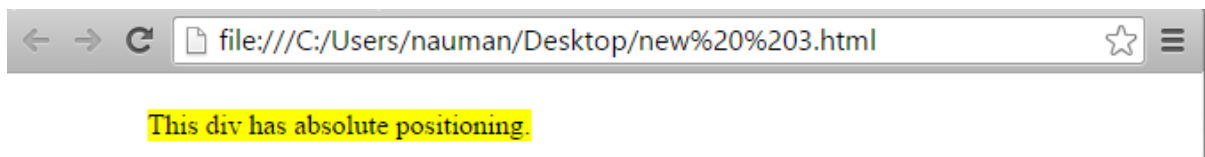
- Move Up - Use a negative value for top.
- Move Down - Use a positive value for top.

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

Here is an example:

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

Here is Result:



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.

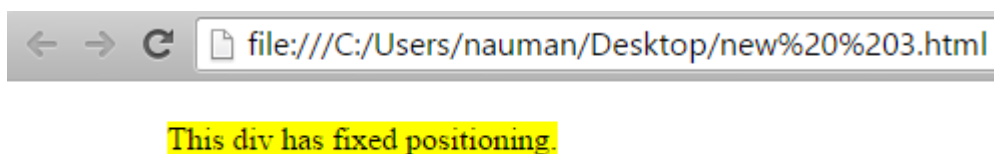
- Move Left - Use a negative value for left.
- Move Right - Use a positive value for left.
- Move Up - Use a negative value for top.
- Move Down - Use a positive value for top.

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

Here is an example:

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

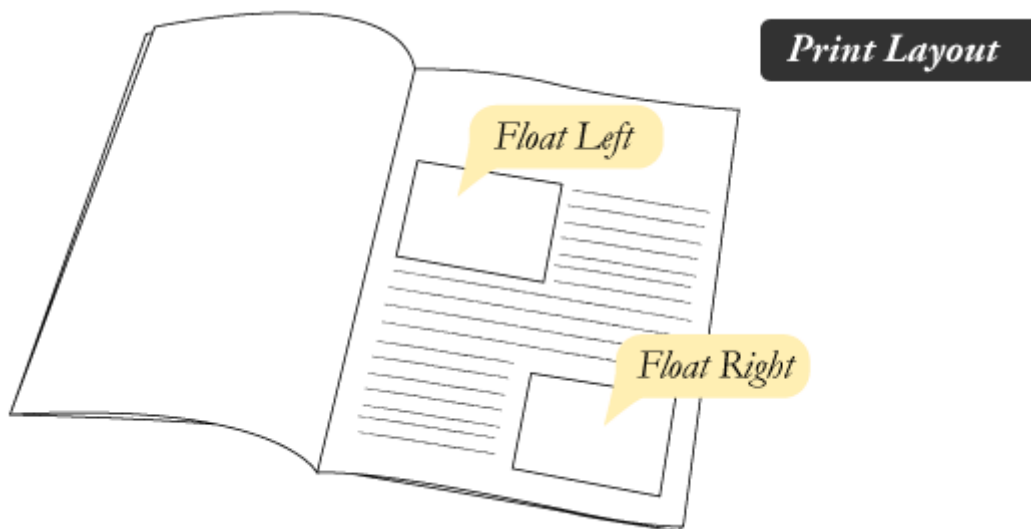
Here is Result :



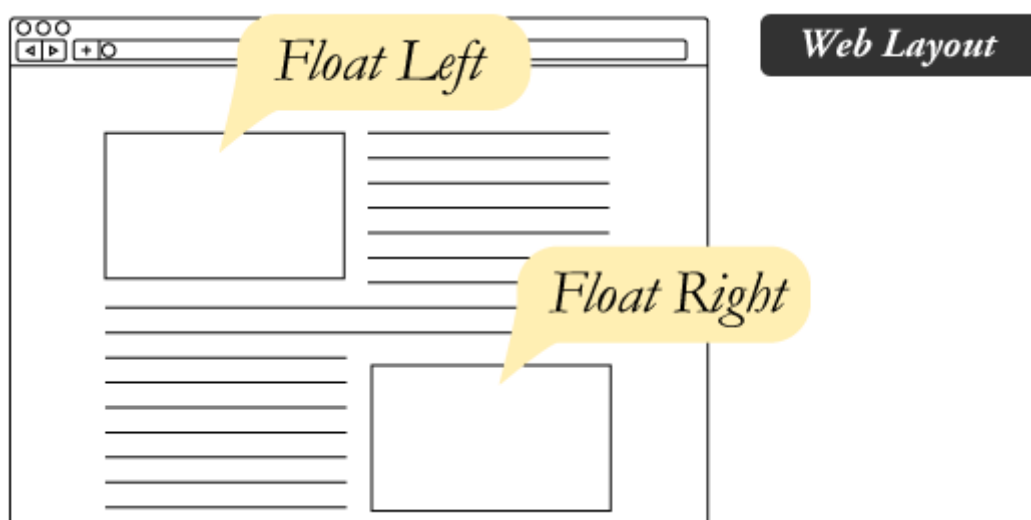
17. Floating

What is "Float"?

Float is a CSS positioning property. To understand its purpose and origin, we can look to print design. In a print layout, images may be set into the page such that text wraps around them as needed. This is commonly and appropriately called "text wrap". Here is an example of that.



In page layout programs, the boxes that hold the text can be told to honor the text wrap, or to ignore it. Ignoring the text wrap will allow the words to flow right over the image like it wasn't even there. This is the difference between that image being part of the flow of the page (or not). Web design is very similar.



In web design, page elements with the CSS float property applied to them are just like the images in the print layout where the text flows around them. Floated elements remain a part of the flow of the web page. This is distinctly different than page elements that use absolute positioning. Absolutely positioned page elements are removed from the flow of the webpage,

like when the text box in the print layout was told to ignore the page wrap. Absolutely positioned page elements will not affect the position of other elements and other elements will not affect them, whether they touch each other or not.

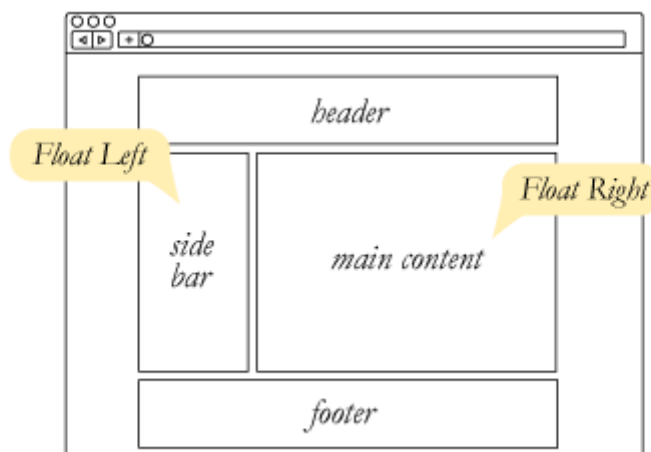
Setting the float on an element with CSS happens like this:

```
#sidebar {  
  float: right;  
}
```

There are four valid values for the float property. Left and Right float elements those directions respectively. None (the default) ensures the element will not float and Inherit which will assume the float value from that elements parent element.

What are floats used for?

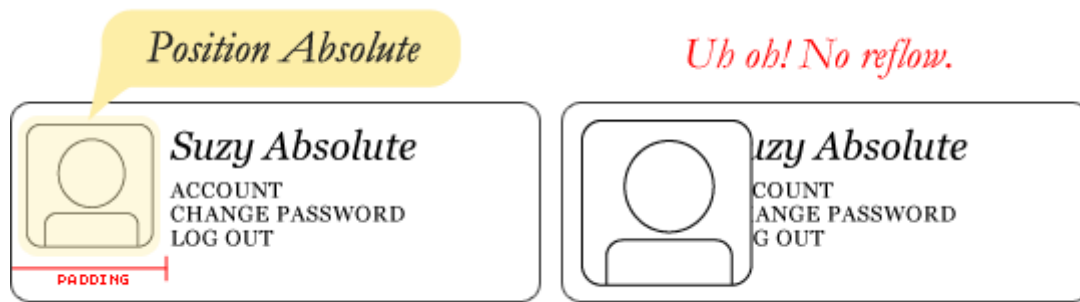
Aside from the simple example of wrapping text around images, floats can be used to create entire web layouts.



Floats are also helpful for layout in smaller instances. Take for example this little area of a web page. If we use float for our little avatar image, when that image changes size the text in the box will reflow to accommodate:



This same layout could be accomplished using relative positioning on container and absolute positioning on the avatar as well. In doing it this way, the text would be unaffected by the avatar and not be able to reflow on a size change.



Example

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
  float: right;
  margin: 0 0 10px 10px;
}
</style>
</head>
<body>

<p>
hi this is a float example.<br/>In this example, the image will float to the right in the
paragraph, and the text in the paragraph will wrap around the image. In this example, the
image will float to the right in the paragraph, and the text in the paragraph will wrap
around the image. In this example, the image will float to the right in the paragraph, and
the text in the paragraph will wrap around the image.<br />
</p>

</body>
</html>
```

Result:

