# CASCADING STYLE SHEETS (CSS)

## Getting Started With CSS

**Learning objectives:**

After the Completion of this unit you should be able to know

- What a style sheet is and how it actually styles a web page
- The importance of CSS.
- How to create a style sheet and link an html document to the style sheet.
- The basic building blocks of any style sheet: rules, selectors, properties and values.
- How External Style Sheets are stored in CSS files
- Three ways of inserting a style sheet
- Different selectors using CSS
- Define background properties ,Background colour & images handling
- Font family, Style and Size using CSS
- Designing Tables using CSS

**Structure**

Introduction of CSS
- Definition
- Advantages
- Parts of CSS

CSS Syntax
- Rules / Principle of CSS
- Parts of style sheet

CSS Selectors
- The element selectors

The ID selectors
The Class selectors
Universal Selector
Attribute Selector
Ways to Insert CSS
- External style sheet
- Internal style sheet
- Inline style

Background image handling
- Repeat the Background Image
- Set the background image position

Background colour management using CSS
- CSS color hex code
- CSS color -short hex code
- CSS color RGB value

Text management using CSS

Set the text color

## Introduction of CSS

CSS stands for Cascading Style Sheets. It 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 works with HTML and other Markup Languages (such as XHTML and XML) to control the way the content is presented. Cascading Style Sheets is a means to separate the appearance of a webpage from the content of a webpage.

### Definition

**Cascading Style Sheets** (**CSS**) is a simple mechanism used to format the layout of Web Pages and adding **style** (e.g., fonts, colors, spacing...) to web documents that previously could only be defined in a page's HTML. **CSS** describes how HTML elements are to be displayed on screen, paper, or in other media. It can control the layout of multiple web pages all at once.

### Advantages

The advantages of CSS are:

- **CSS saves time** - You can write CSS once and then reuse the same sheet in multiple HTML pages.
- **Pages load faster** – Increases Download Speed
- **Easy maintenance** - To make a global change, all the elements in all the web pages will be updated automatically.
- **Superior styles to HTML** – It is 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.
- **Global web standards** - Now HTML attributes are being deprecated and it is being recommended to use CSS

### What is the "Cascade" part of CSS?

The cascade part of CSS means that more than one style sheet can be attached to a document, and all of them can influence the presentation. For example, a designer can have a global style sheet for the whole site, but a local one for say, controlling the link color and background of a specific page. Or, a user can use own style sheet if s/he has problems seeing the page, or if s/he just prefers a certain look.

**CSS Syntax**

A CSS style rule is made of three parts:

1. **Selector:** A selector is an HTML tag at which a style will be applied. This could be any tag like <h1>, <p> or <table> etc.
2. **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*, bgcolor etc.
3. **Value:** Values are assigned to properties. For example, *color* property can have the value either *red* or *#F1F1F1* etc.

The format or syntax of CSS is:

**Example:** You can `Selector {property:` define a heading as follows:

Here **h1** is a selector , **color and font-size** are properties and the given value ***red, and 15px are*** the value of that property.

**h1 {color: red; font-size :15px}**

Selector | Property | Value | Property | Value

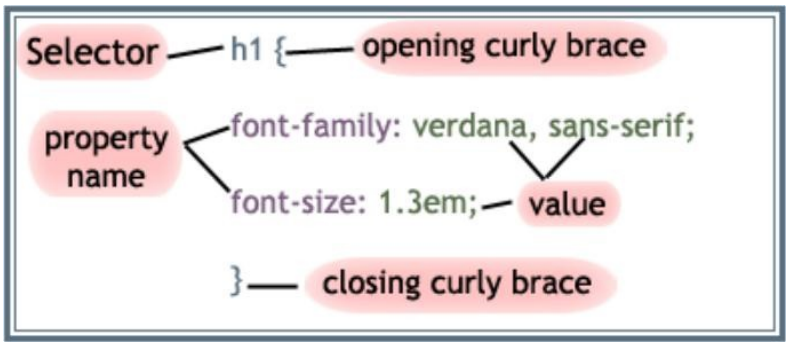
- The selector is normally the HTML element you want to style.
- Each declaration consists of a property and a value.
- The property is the style attribute you want to change. Each property has a value.

**Rules/ Principle of CSS**

1. Every statement must have a selector and a declaration. The declaration comes immediately after the selector and is contained in a pair of curly braces.
2. The declaration is one or more properties separated by semicolons.
3. Each property has a property name followed by a colon and then the value for that property. There are many different types of values, but any given property can only take certain values as set down in the specification.
4. Sometimes a property can take a number of values, as in the font-family. The values in the list should be separated by a comma and a space.
5. Sometimes a value will have a unit as well as the actual value, as in the 1.3em. You must not put a space between the value and its unit.
6. As with HTML, white space can be used to make your style sheet easier to read and write.

**Parts of style sheet**

A style sheet consists of one or more rules that describe how document elements should be displayed. A rule in CSS has two parts: the selector and the declaration. The declaration also has two parts, the property and the value. Let's take a look at a rule for a heading 1 style: h1 { font-family: verdana, "sans serif"; font-size: 1.3em } This expression is a rule that says every h1 tag will be verdana or other sans-serif font and the fontsize will be 1.3em. Let's take a look at the different parts of this rule.




```
Selector
{
        property1: some value;
        property2: some value;
}
```

The declaration contains the property and value for the selector. The property is the attribute you wish to change and each property can take a value. The property and value are separated by a colon and surrounded by curly braces:

```
body { background-color: black}
```

If the value of a property is more than one word, put quotes around that value: body { font-family: "sans serif"; } If you wish to specify more than one property, you must use a semi-colon to separate each property. This rule defines a paragraph that will have blue text that is centered.

```
p { text-align: center; color: blue }
```

You can group selectors. Separate each selector with a comma. The example below groups headers 1, 2, and 3 and makes them all yellow. h1, h2, h3 { color: yellow}

**Check your progress 1**

**Q1. Write the abbreviation of CSS?**
**Answer:**_____

**Q2. Write the three parts of CSS syntax?**
**Answer:**_____

## CSS Selectors

You can define selectors in various simple ways based on your comfort. Let me put these selectors one by one. Three types of CSS Selectors

1. The Element selectors
2. The ID Selectors
3. The Class Selectors

### The Element selectors

A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets: example -

```
p {color:red;text-align:center;}
```

To make the CSS more readable, you can put one declaration on each line, like this:

```
p
{
color:red;
text-align:center;
}
```

### The ID selectors

The id selector is used to specify a style for a single, unique element.The id selector uses the id attribute of the HTML element, and is defined with a "#". Example –

Imagine within the body element of our html page, we have the following paragraph element

```
<p id="welcome">Welcome to the 1st CSSDocument
</p>
```

We can then create a CSS rule with the id selector:

```
#welcome
{
color:red;
text-align:center;
}
```

### The Class selectors

The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements. This allows you to set a particular style for many HTML elements with the same class. The class selector uses the HTML class attribute, and is defined with a ".". In the example below, all HTML elements with class="center" will be center-aligned:

Imagine within the body element of our html page, we have the following header element

```
<h2 class="center">Summary</h2>
```

We can then create a CSS rule with the class selector:

```
.center {text-align:center;}
```

You can also specify that only specific HTML elements should be affected by a class. In the example below, all p elements with class="center" will be center-aligned: example

```
p.center {text-align:center;}
```

Some of the other selectors are used in CSS, they are :

3.3.3.1 **Universal selector**

An asterisk (*) is the universal selector for CSS. It matches a single element of any type. Omitting the asterisk with simple selectors has the same effect. For instance, *.warning and. warning are considered equal. Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type: Example-

```
*
{ Color:#000000;
}
```

This rule renders the content of every element in our document in black.

3.3.3.2 **Attribute Selector**

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-".

**Check your progress 2**

**Q1. What are the 3 types of selectors?**.
**Answer:**_____

_____

**Q2. What is universal selector ?**.
**Answer:**_____

_____

## Ways to insert CSS

There are three ways of inserting a style sheet:

1. External style sheet
2. Internal style sheet
3. Inline style

**External style sheet**

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
```

An external style sheet can be written in any text editor. The file should not contain any html tags. Your style sheet should be saved with a .css extension. An example of a style sheet file is shown below:

```
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
```

Notes : Do not leave spaces between the property value and the units! "margin-left:20 px" (instead of "margin- left:20px") will work in IE, but not in Firefox or Opera.

**Internal style sheet**

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, by using the <style> tag, like this:

```
<head>
    <style type="text/css">
        hr {color:sienna;}
        p {margin-left:20px;}
        body {background-image:url("images/back1.gif");}
    </style>
</head>
```

**Inline style sheet**

An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly! To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:

```
<p style="color:sienna;margin-left:20px">This is a paragraph.</p>
```

**Background image handling**

The background-image property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element. The background image for a page can be set like this:body {background-image:url('paper.gif');}

Example

```
<html>
<head>
	<Title>This is my Internal css page</Title>
	<style type="text/css">body
		{
		background-image:url
		("C:/Users/SAI/Desktop/Desktop/100MSDCF/11jpg");
		}
	</style>
</head>
<body>
	Background Image
</body>
</html>
```

The output of the above example is :

You can set the following background properties of an element:

- The **background-image** property is used to set the background image of an element.
- The **background-repeat** property is used to control the repetition of an image in the background.
- The **background-position** property is used to control the position of an image in the background.
- The **background-attachment** property is used to control the scrolling of an image in the background.
- The background property is used as shorthand to specify a number of other background properties.

**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>
```

**Check your progress 3**

**Q1. How many ways to build style sheet ? what are they ?**
**Answer:**_____

_____

**Q2. Write the background properties of CSS.**

**Answer:**_____

_____

## Background colour Management using CSS

The background-color property specifies the background color of an element. The background color of a page is defined in the body selector: Example

```
body {background-color:#b0c4de;}
<p style="background-color:yellow;">
This text has a yellow background color. </p>
```

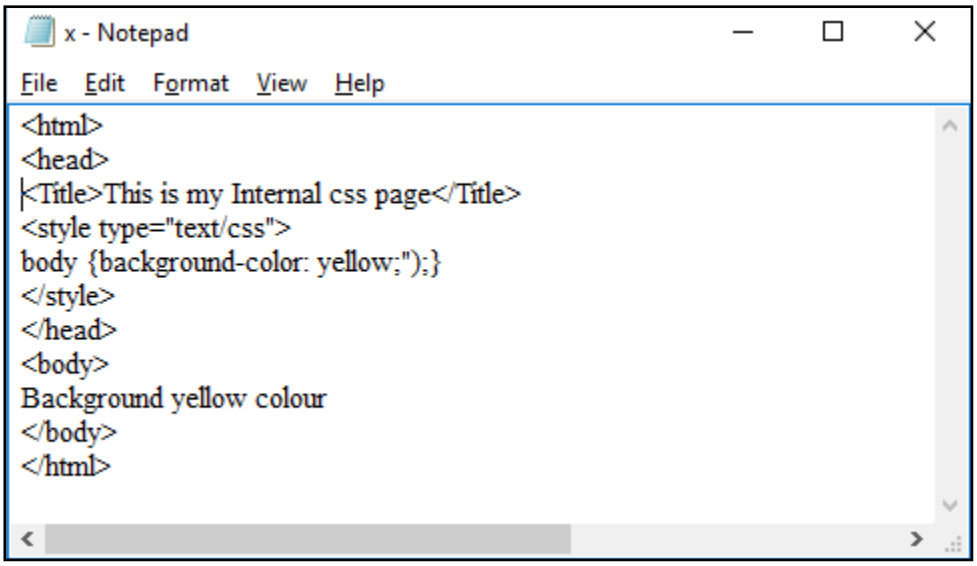
With CSS, a color is most often specified by:

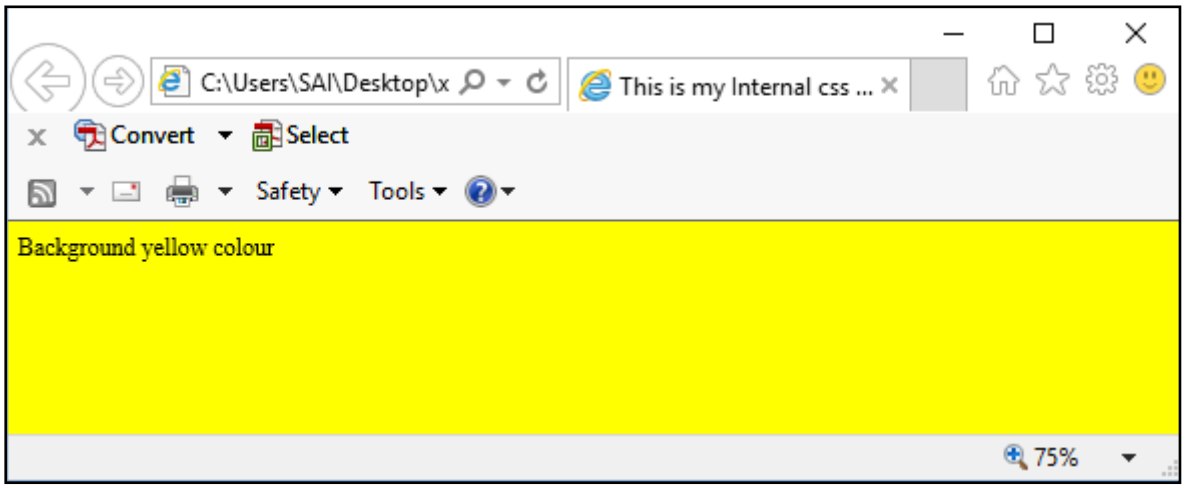
1. a HEX value - like "#ff0000"
2. an RGB value - like "rgb(255,0,0)"
3. a color name - like "red"

In the example below, the h1, p, and div elements have different background colors:

```
h1 {background-color:#6495ed;}
p  {background-color:#e0ffff;}
div {background-color:#b0c4de;}
```

Example






## CSS Colors - Hex Codes

A hexadecimal is a 6 digit representation of a color. The first two digits (RR) represent a red value, the next two are a green value (GG), and the last are the blue value (BB). Each hexadecimal code will be preceded by a pound or hash sign '#'. Following are the examples to use Hexadecimal notation.

## CSS Colors - Short Hex Codes

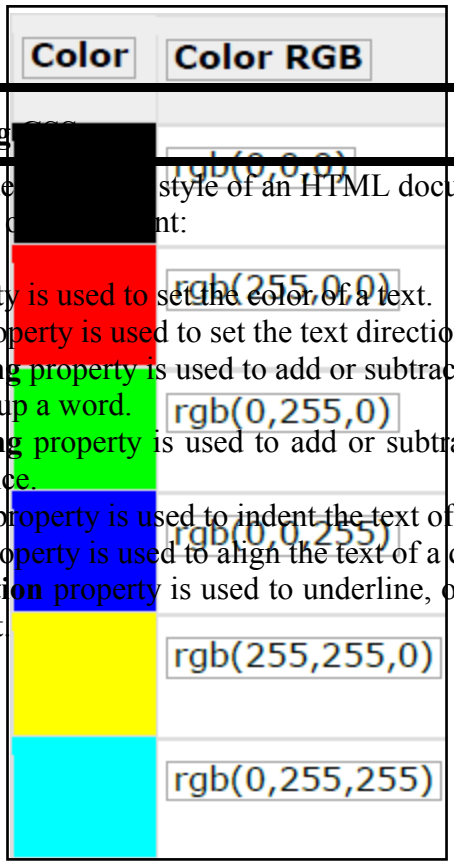
This is a shorter form of the six-digit notation. In this format, each digit is replicated to arrive at an equivalent six-digit value. For example: #6A7 becomes #66AA77.

| Color | Color HEX |
|---|---|
| | #000000 |
| | #FF0000 |
| | #00FF00 |
| | #0000FF |
| | #FFFF00 |
| | #00FFFF |
| | #FF00FF |

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

## CSS Colors - RGB Values

This color value is specified using the rgb( ) property. This property takes three values, one each for red, green, and blue. The value can be an integer between 0 and 255 or a percentage. NOTE: All the browsers does not support rgb() property of color, so it is recommended not to use it.




## Text Management using CSS

**CSS** is a language that describes style of an HTML document. You can set the following text properties of the 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>
```

**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>
```

### 3.7.3. 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>
```

**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>
```

**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>
```

**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>
```

## Decorating the Text

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

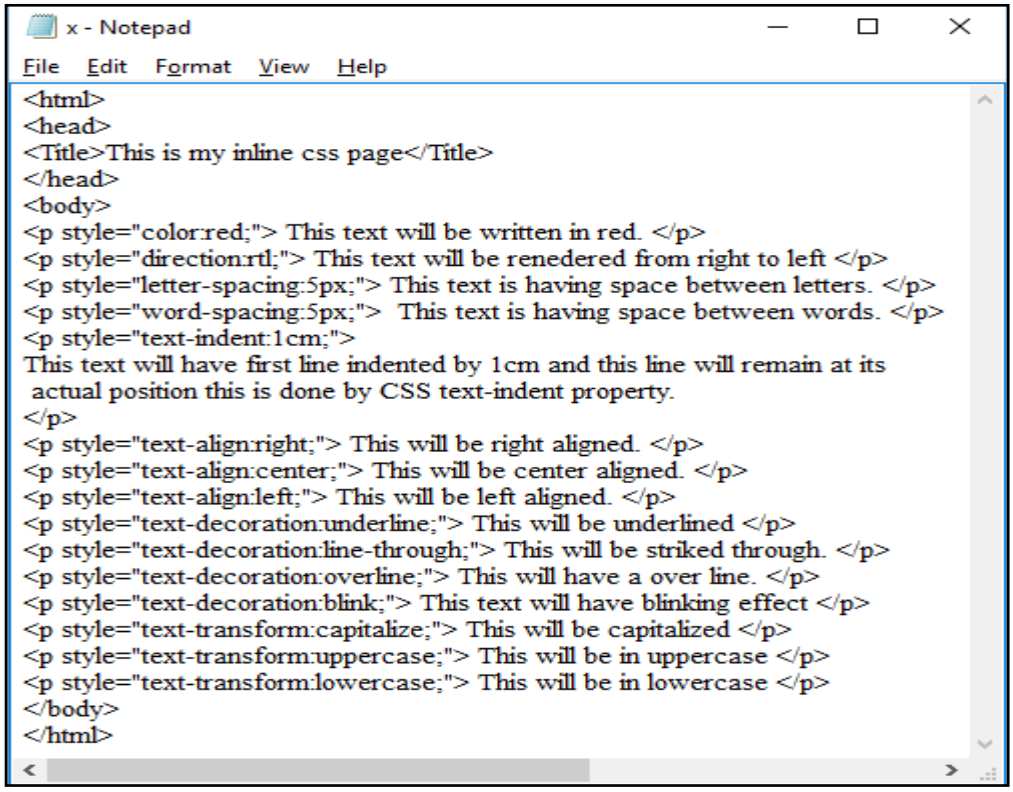
```
<p style="text-decoration:underline;"> This will be underlined </p>
<p style="text-decoration:line-through;"> This will be striked through.
</p>
<p style="text-decoration:overline;"> This will have a over line. </p>
<p style="text-decoration:blink;">This text will have blinking effect </p>
```

## Set the Text Cases

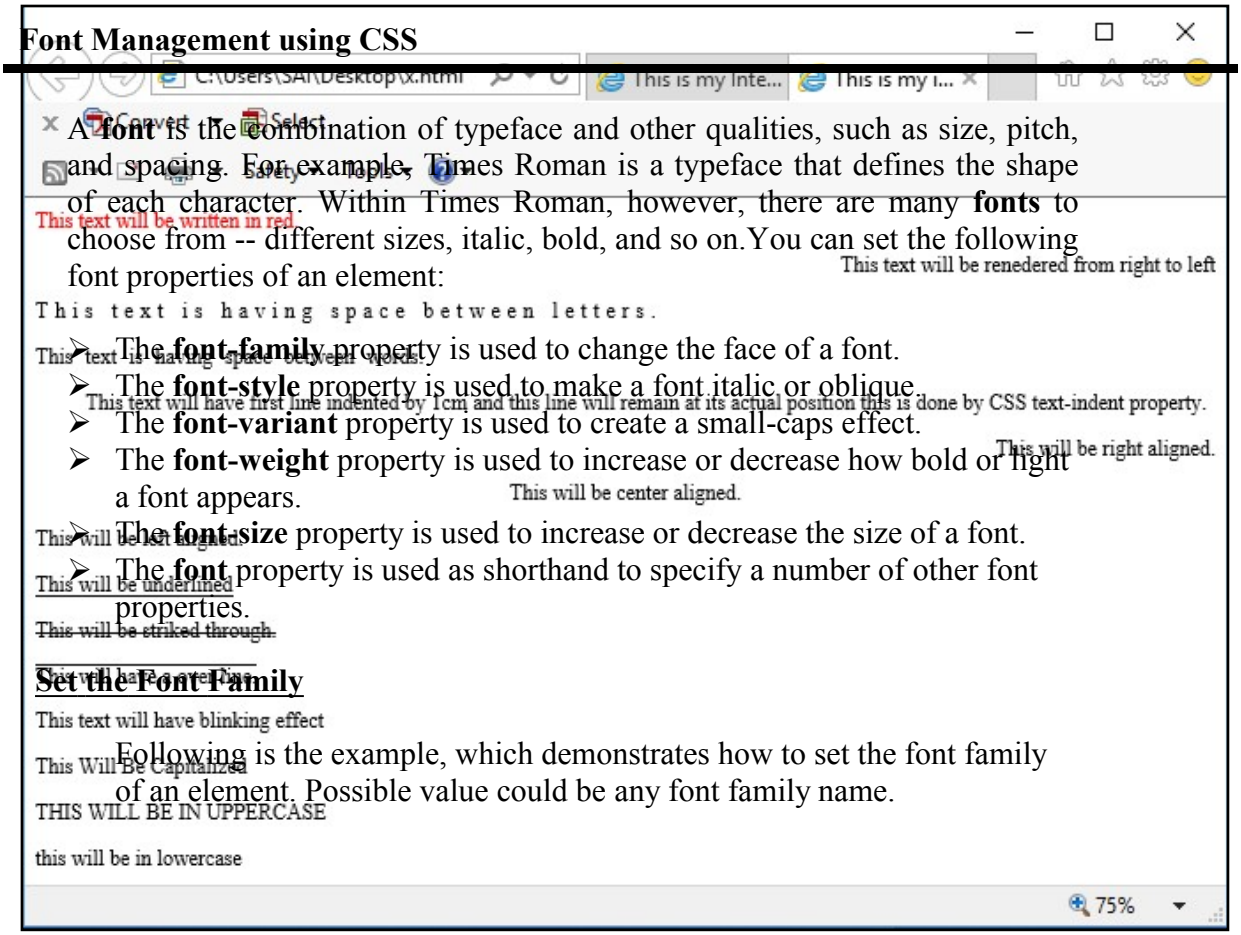
The following example demonstrates how to set the cases for a text. Possible values are none, capitalize, uppercase, lowercase.

```
<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>
```

Example :



The output of the above program is :




A **font** is the combination of typeface and other qualities, such as size, pitch, and spacing. For example, Times Roman is a typeface that defines the shape of each character. Within Times Roman, however, there are many **fonts** to choose from -- different sizes, italic, bold, and so on.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.

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.

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.

 This text will be rendered as small caps 

**Set the Font Weight**

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

 This font is bold. 
 This font is bolder. 
 This font is 900 weight. 

**Set the Font Size**

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

 This font size is 20 pixels 
 This font size is small 
 This font size is large 

**Set the Font Size Adjust**

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

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

**Set the Font Stretch**

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

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

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. 

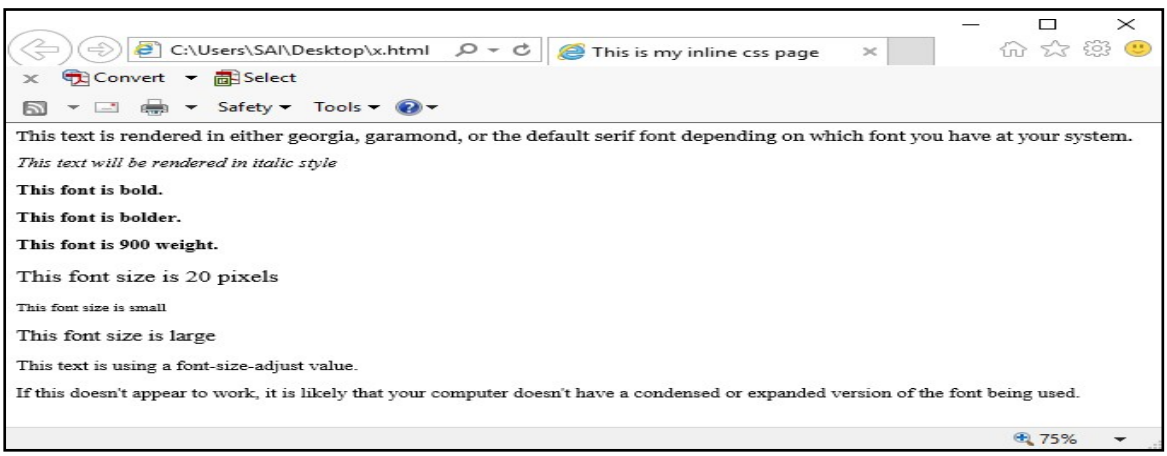
Example:-

x - Notepad

File Edit Format View Help

```
<html>
<head>
<Title>This is my inline css page</Title>
</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>

<p style="font-style:italic;"> This text will be rendered in italic style </p>
<p style="font-weight:bold;"> This font is bold.  </p>
<p style="font-weight:bolder;"> This font is bolder.  </p>
<p style="font-weight:900;"> This font is 900 weight.  </p>

<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>

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

The output of the above program is :



## Managing hyperlinks using CSS

An element in an electronic document that links to another place in the same document or to an entirely different document. A **hyperlink**, or simply a **link**, is a reference to data that the reader can directly follow either by clicking, tapping, or hovering. A **hyperlink** points to a whole document or to a specific element within a document. Hypertext is text with **hyperlinks**. The text that is linked is called anchor text. You can set the following properties of a hyperlink:

- 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>
```

### 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>
```

### 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>
```

**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>
```

**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>
```

**Check your progress 4**

**Q1. What is the representation of 6 digit Hexa code ?**
**Answer:**_____

_____

**Q2. What are the different properties of Font ?**
**Answer:**_____

_____

**Q3. Write the different properties of Hyperlink ?**
**Answer:**_____

_____

**Managing List using CSS**

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.

**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 |

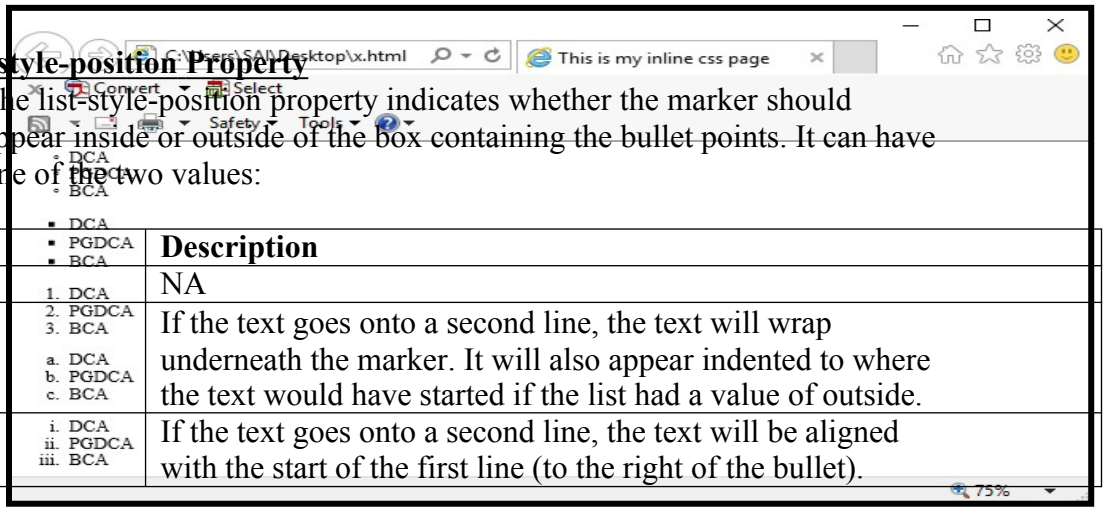
Here are the values, which can be used for an **ordered list**:

| Value | Description | Example |
|---|---|---|
| Decimal | Number | 1,2,3,4….. |
| decimal-leading-zero | 0 before the number | 01, 02, ….. |
| lower-alpha | Lowercase alphanumeric characters | a, b, c, …. |
| 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 |

Here is an example:

```
<html>
<head>
<Title>This is my inline css page</Title>
</head>
<body>
<ul style="list-style-type:circle;"> <li>DCA</li><li>PGDCA</li><li>BCA</li></ul>
<ul style="list-style-type:square;"><li>DCA</li><li>PGDCA</li><li>BCA</li></ul>
<ol style="list-style-type:decimal;"> <li>DCA</li><li>PGDCA</li><li>BCA</li></ol>
<ol style="list-style-type:lower-alpha;"> <li>DCA</li><li>PGDCA</li><li>BCA</li></ol>
<ol style="list-style-type:lower-roman;"> <li>DCA</li><li>PGDCA</li><li>BCA</li></ol>
</body>
</html>
```

The output of the above program is :




**The list-style-position Property**

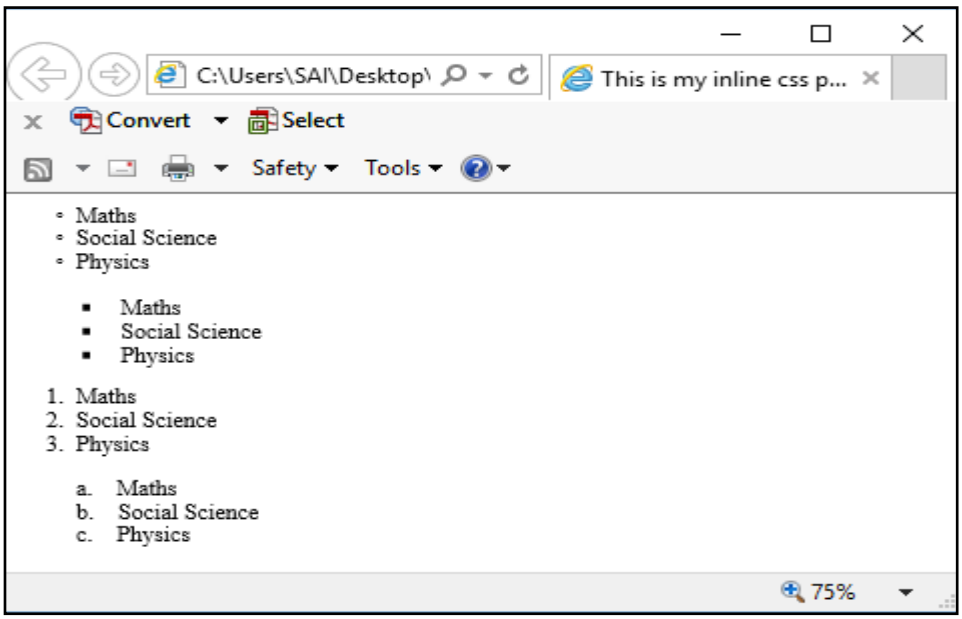
The list-style-position property indicates whether the marker should appear inside or outside of the box containing the bullet points. It can have one of the two values:

| 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). |

Example :

```
<html>
<head>
<Title>This is my inline css page</Title>
</head>
<body>
<ul style="list-style-type:circle; list-stlye-
position:outside;"><li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ul>
<ul style="list-style-type:square;list-style-
position:inside;"><li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ul>
<ol style="list-style-type:decimal;list-stlye-
position:outside;"><li>Maths</li>
<li>Social Science</li>
<li>Physics</li>
</ol>
<ol style="list-style-type:lower-alpha;list-style-position:inside;">

<li>Maths</li>
<li>Social Science</li><li>Physics</li>
</ol>
</body>
</html>
```

The output of the above program is :



## Designing Tables using CSS

Tables are an excellent way to organize and display information on a page. 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** specify 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.

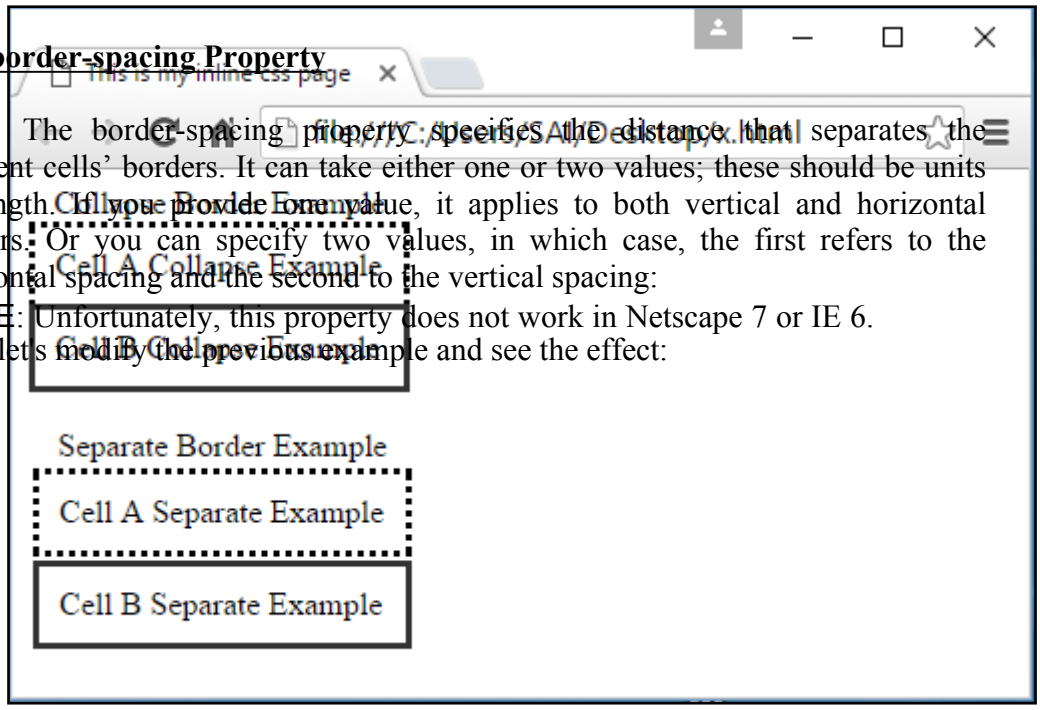
### The order-collapse Property

This property can have two values collapse and separate. The following example uses both the values:

```
<html>
<head>
<Title>This is my inline css page</Title>
</head>
<body>
<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>
</body>
</html>
```

The output of the above program is :




**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.

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>
        </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>
```

**The empty-cells Property**

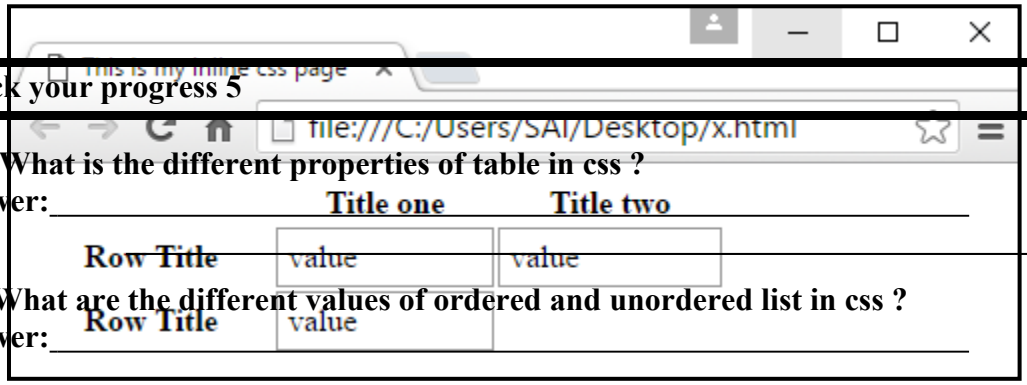
The empty-cells property indicates whether a cell without any content should have a border displayed. This property can have one of the three values - show, hide, or inherit. Here is the empty-cells property used to hide borders of empty cells in the <table> element.

**Example:**

```
<html>
<head>
<Title>This is my inline css page</Title>
</head>
<body>
<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>
</body>
</html>
```

The output of the above program is :




**Check your progress 5**

**Q1. What is the different properties of table in css ?**

**Answer:**_____

**Q2. What are the different values of ordered and unordered list in css ?**

**Answer:**_____

## Working with Box Model

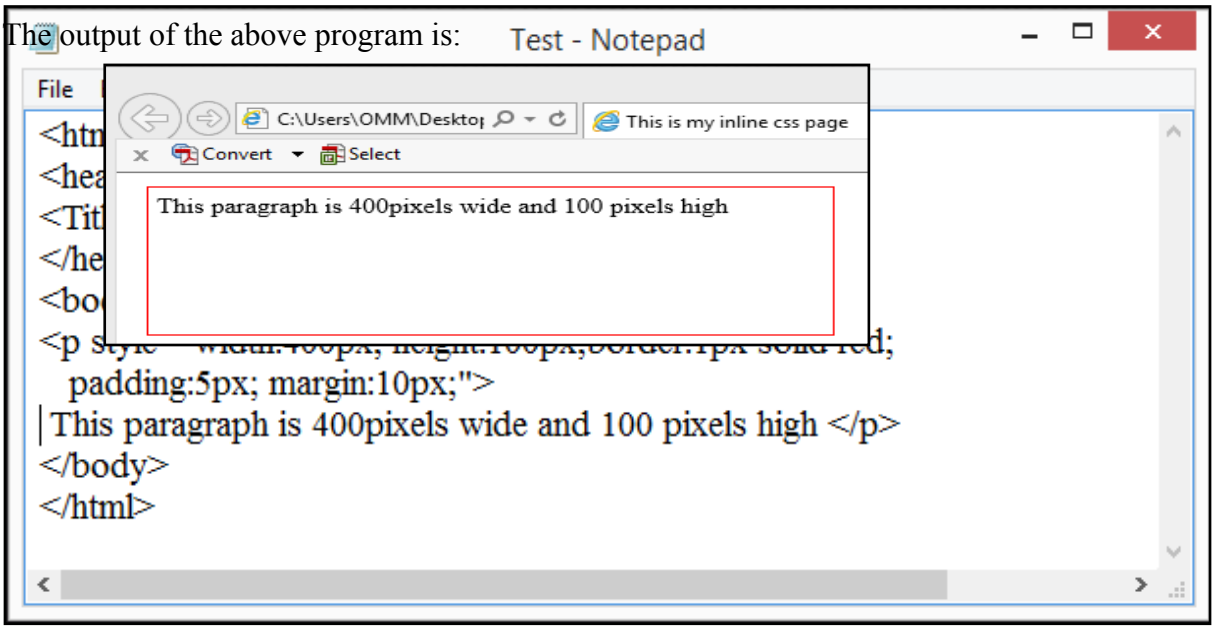
You have seen the border that surrounds every box i.e. element, the padding that can appear inside each box, and the margin that can go around them. In this chapter, we will learn how to change the dimensions of boxes. We have the following properties that allow you to control the dimensions of a box.

- 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:

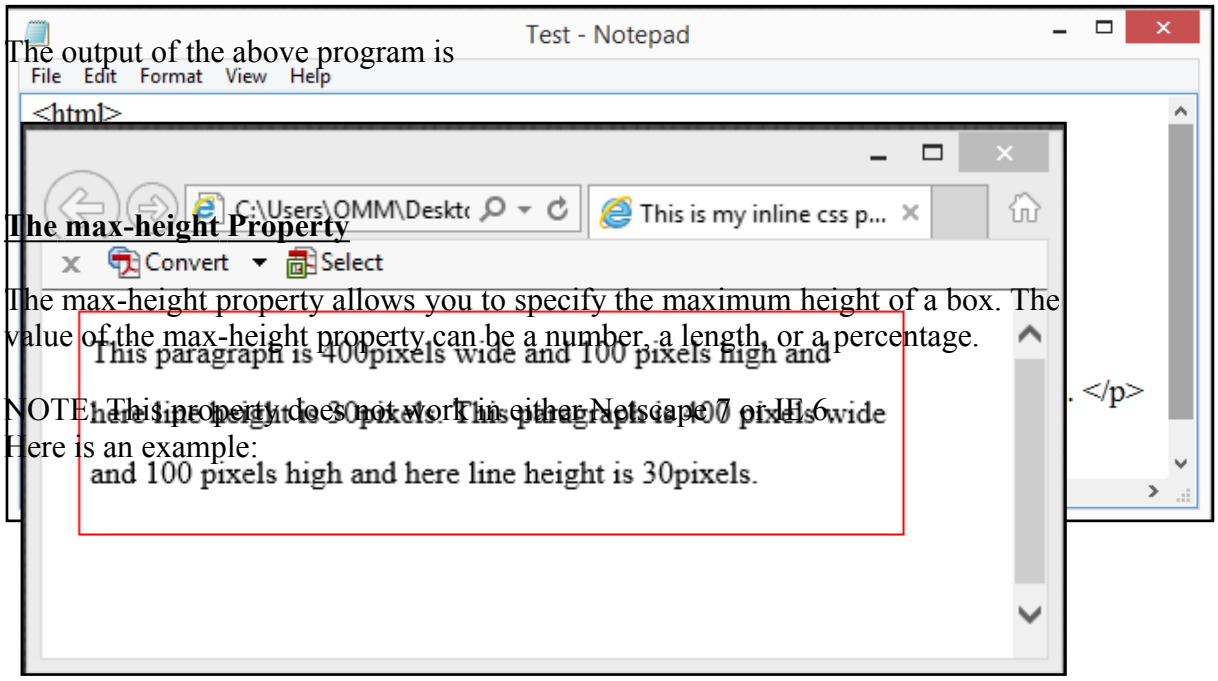
The output of the above program is:



## 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:

The output of the above program is




## 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.

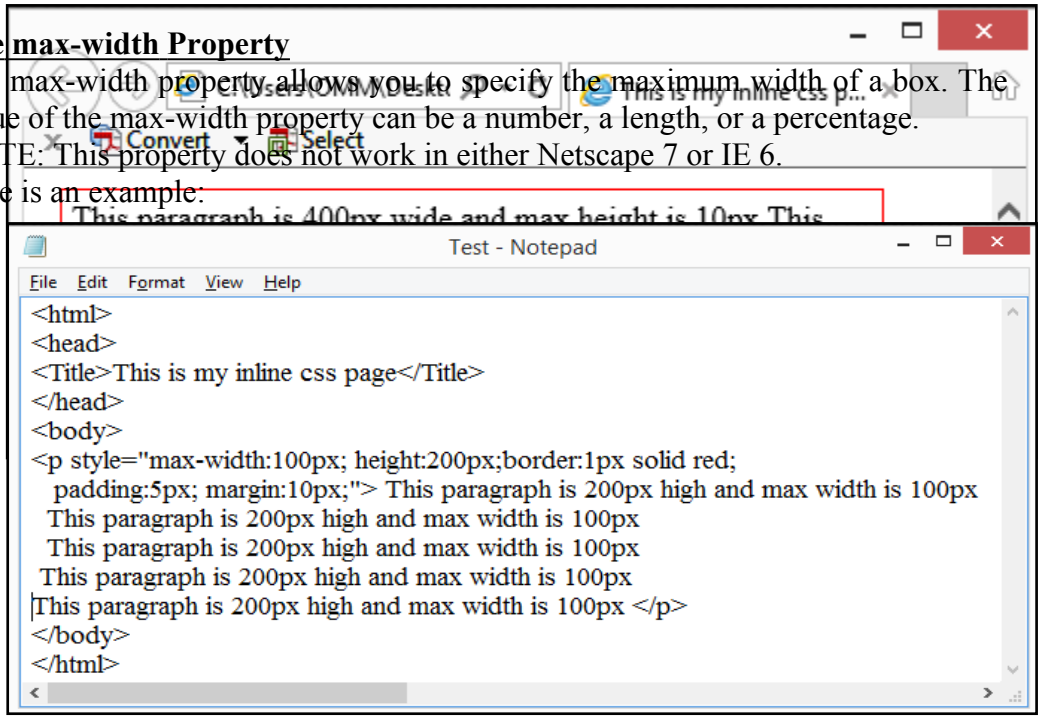
Here is an example:

Test - Notepad

File Edit Format View Help

```
<html>
<head>
<Title>This is my inline css page</Title>
</head>
<body>
<p style="width:400px; max-height:10px;border:1px solid red;
  padding:5px; margin:10px;"> This paragraph is 400px wide and max height is 10px
This paragraph is 400px wide and max height is 10px
This paragraph is 400px wide and max height is 10px
This paragraph is 400px wide and max height is 10px
</p>
</body>
</html>
```

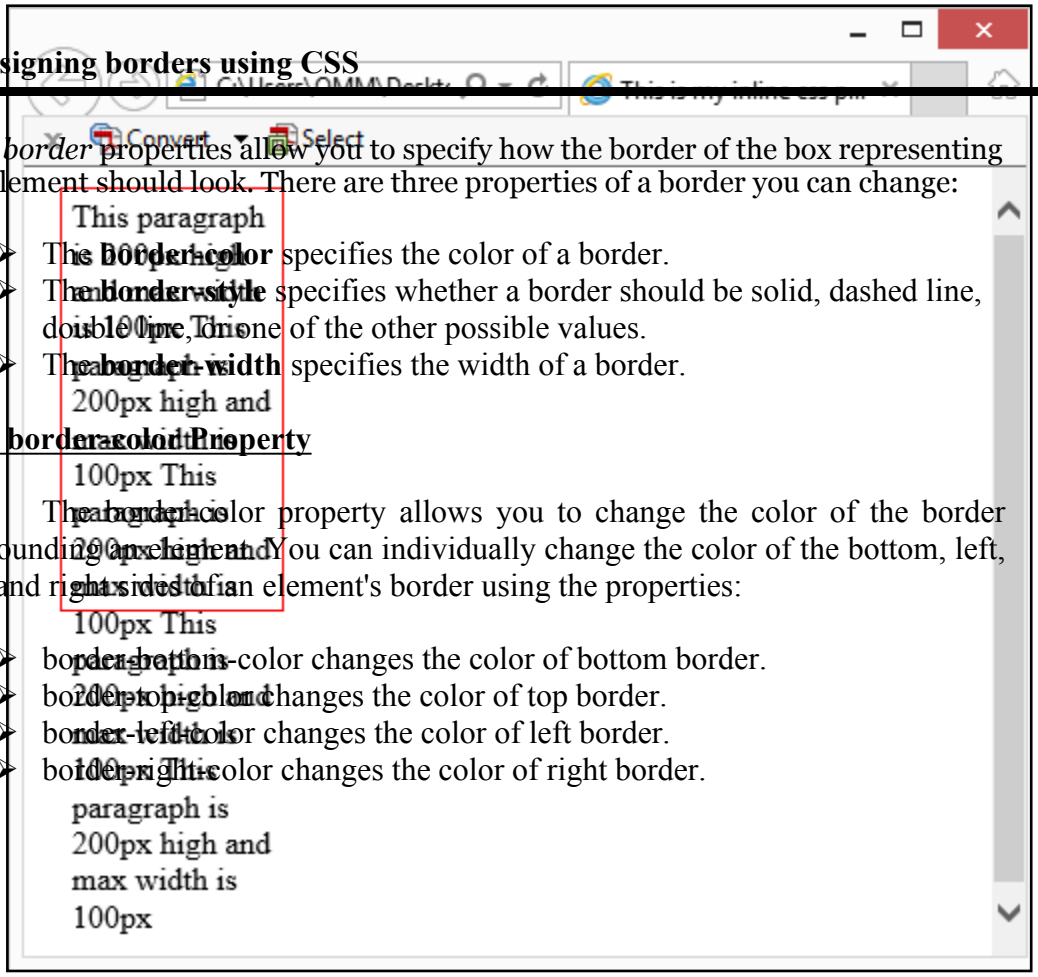
The output of the above program is




**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:

The output of the above program is




**Designing borders using CSS**

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:

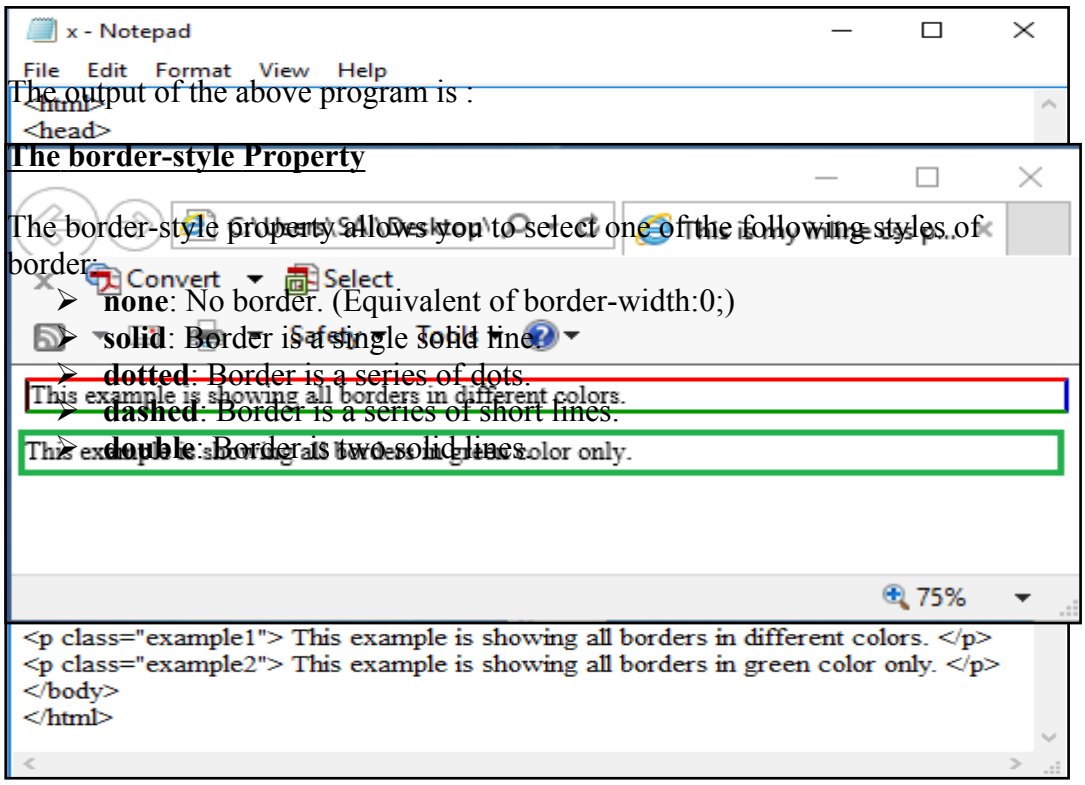
- The **border-color** specifies the color of a border.
- The **border-style** specifies whether a border should be solid, dashed line, double line, or one of the other possible values.
- The **border-width** specifies the width of a border.

**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:




The output of the above program is :

**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.

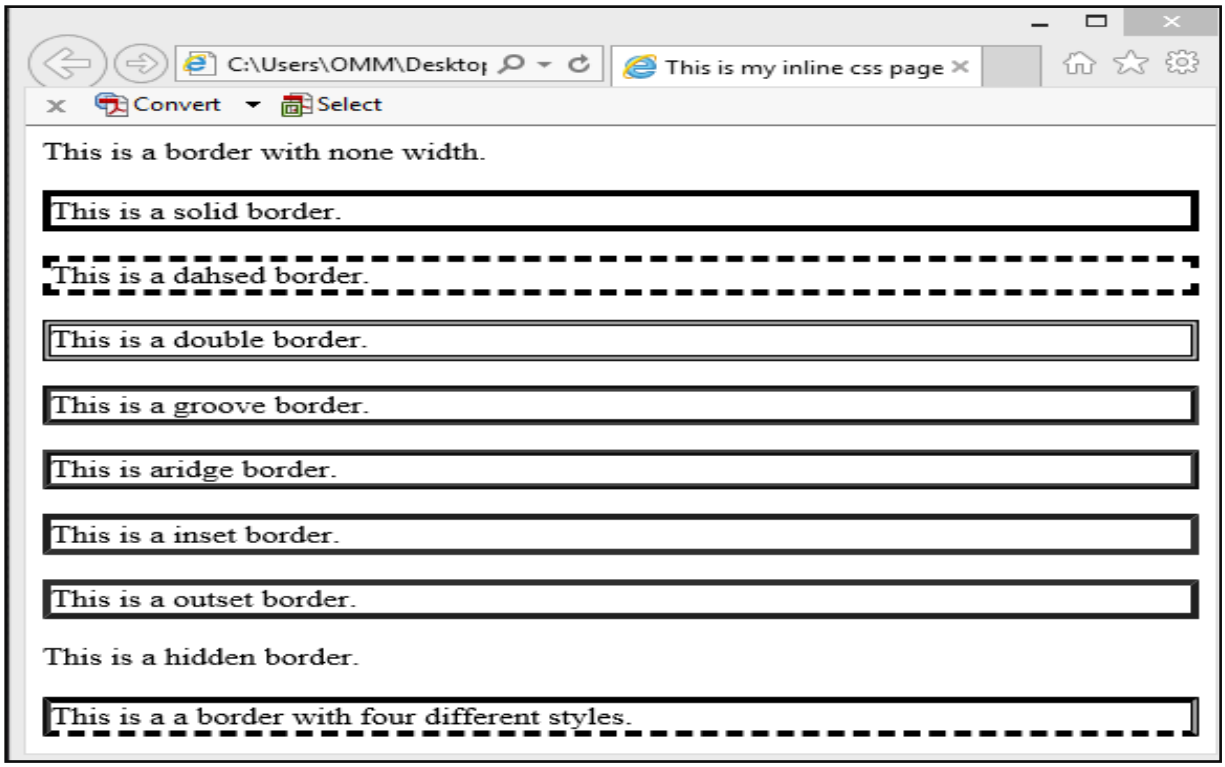
The following example shows all these border styles:

x - Notepad

File Edit Format View Help

```
<html>
<head>
<Title>This is my inline css page</Title>
</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>
```

It will produce the following result:




## Designing outline using CSS

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

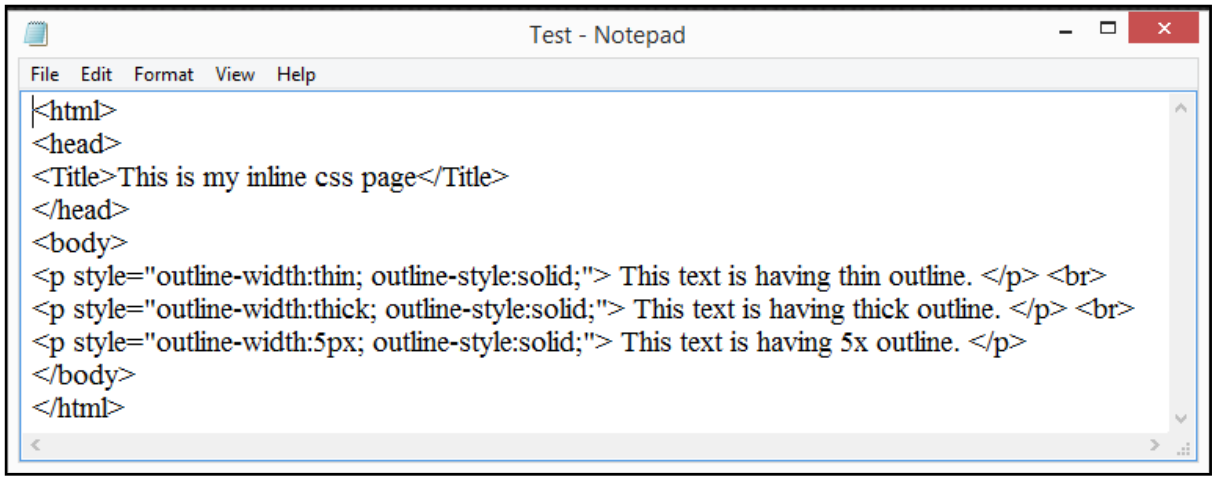
- An outline does not take up space.
- Outlines do not have to be rectangular.
- Outline is always the same on all sides; you cannot specify different values for different sides of an element.

***NOTE:*** The outline properties are not supported by IE 6 or Netscape 7. You can set the following outline properties using CSS.

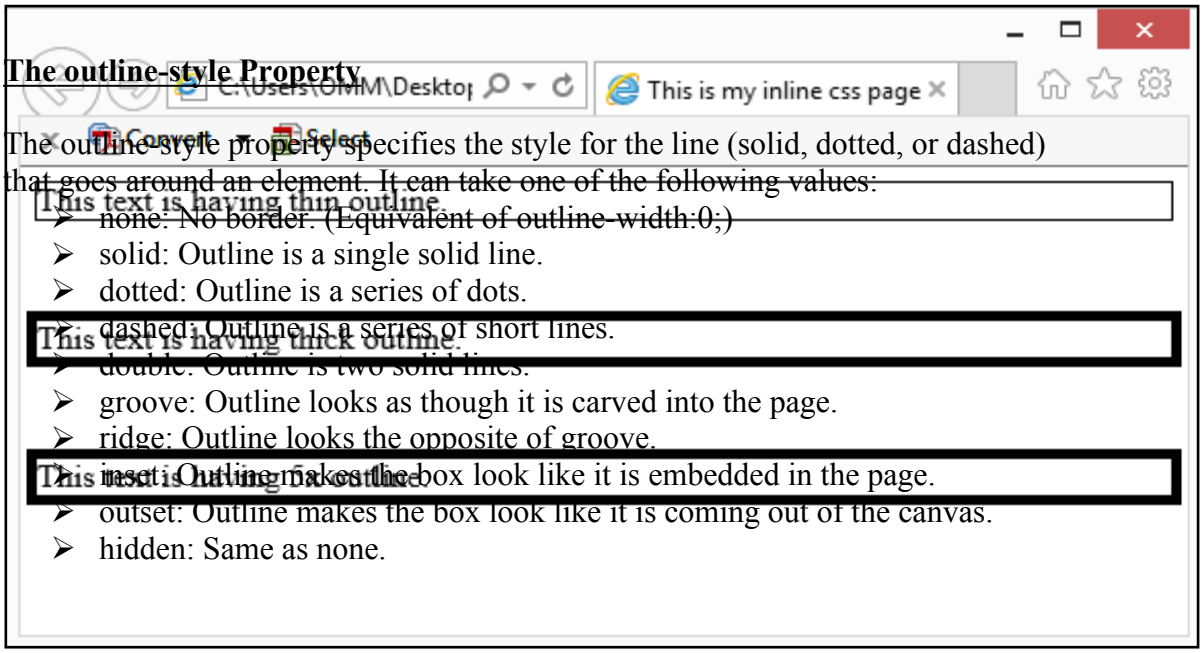
- The outline-width property is used to set the width of the outline.
- The outline-style property is used to set the line style for the outline.
- The outline-color property is used to set the color of the outline.
- The outline property is used to set all the above three properties in a single statement.

### The outline-width Property

The outline-width property specifies the width of the outline to be added to the box. Its value should be a length or one of the values thin, medium, or thick, just like the border-width attribute. A width of zero pixels means no outline. Here is an example:




The above program will produce the following result:

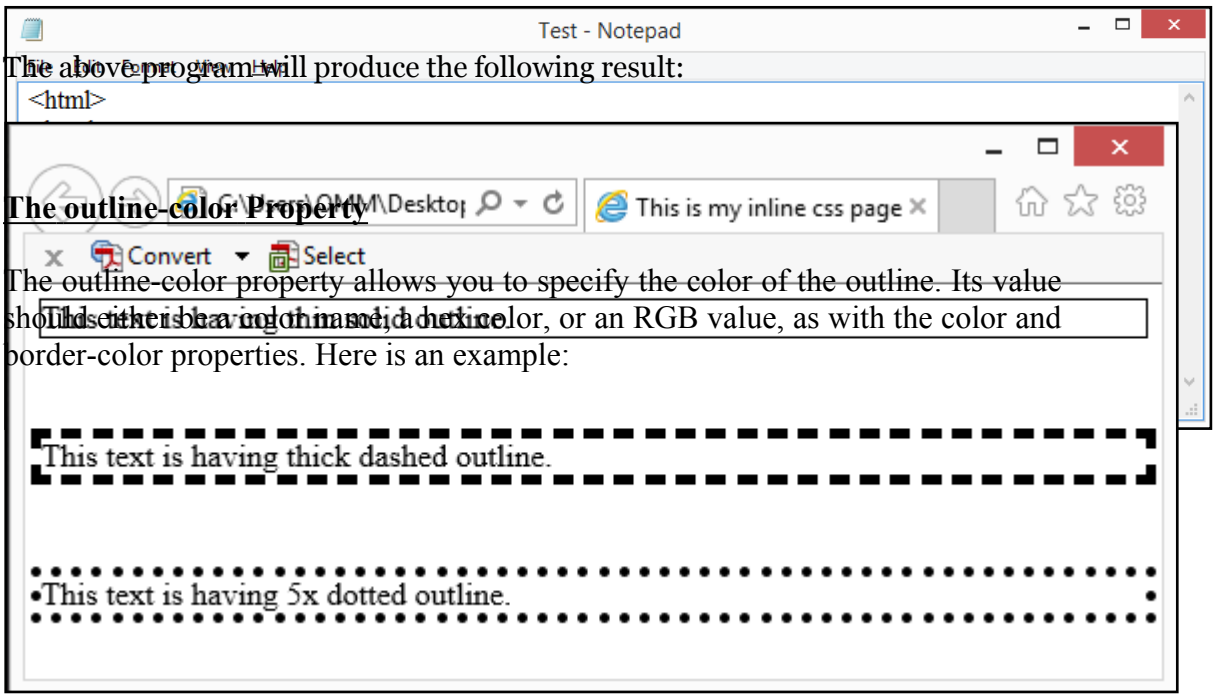



**The outline-style Property**

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

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

Here is an example:



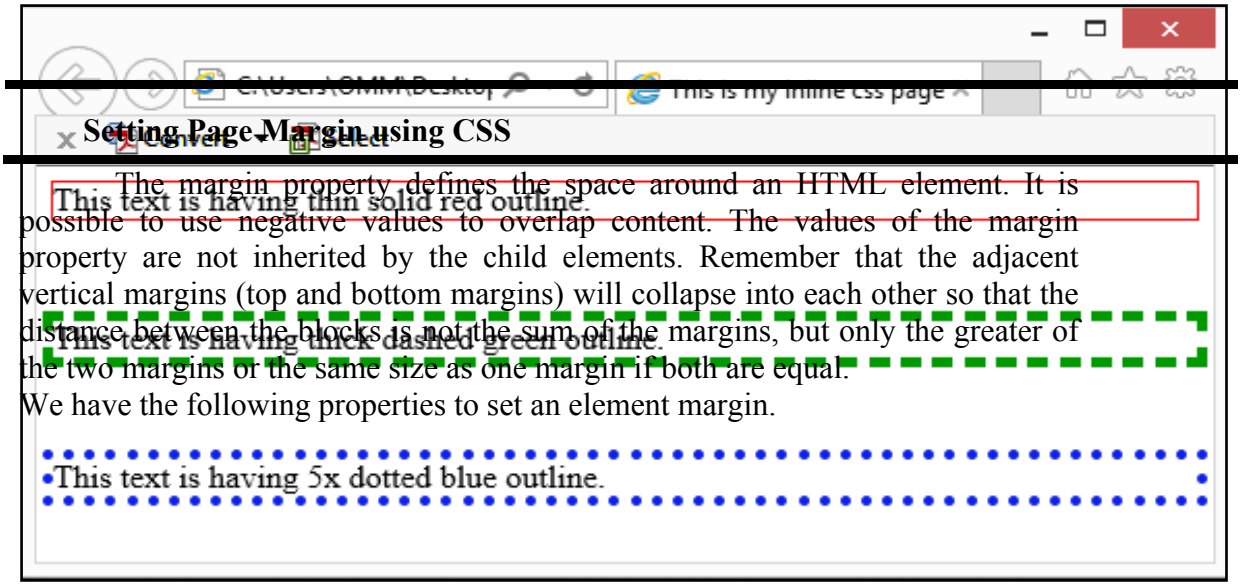

The above program will produce the following result:

**The outline-color Property**

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

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

The above program will produce the following result:




**Setting Page Margin using CSS**

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.

## 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:

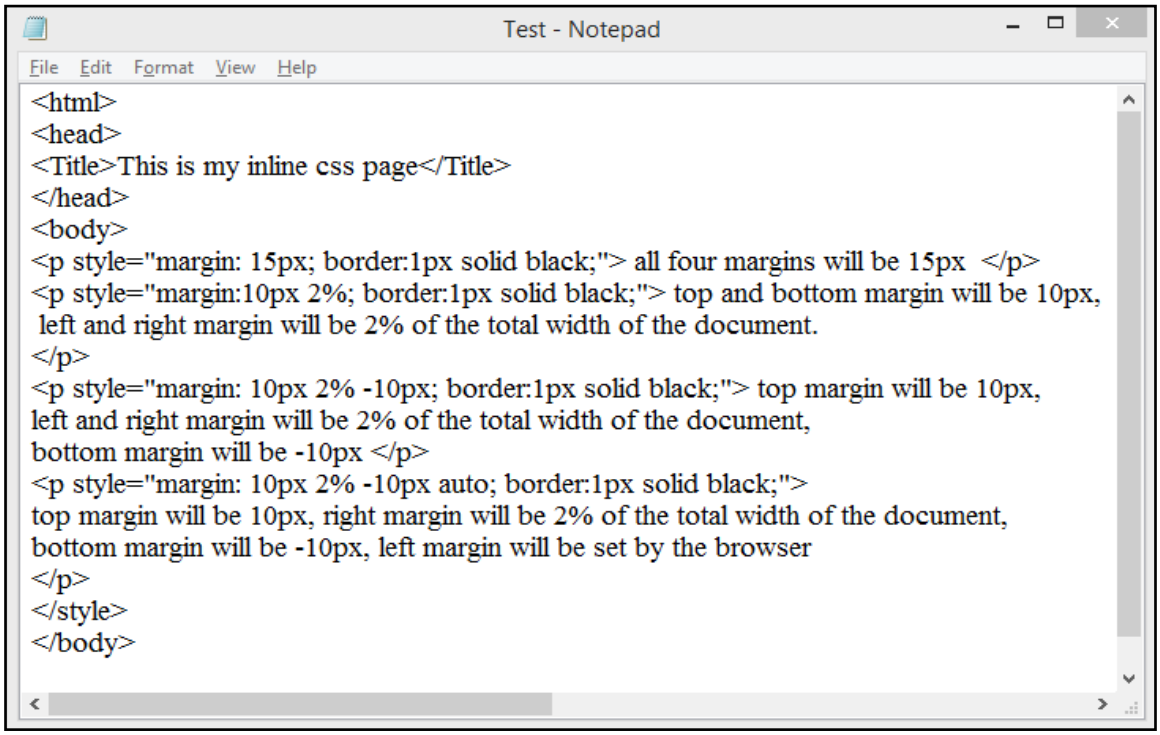
*<style type="text/css">*
*p {margin: 15px}*
*all four margins will be 15px*

*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.*

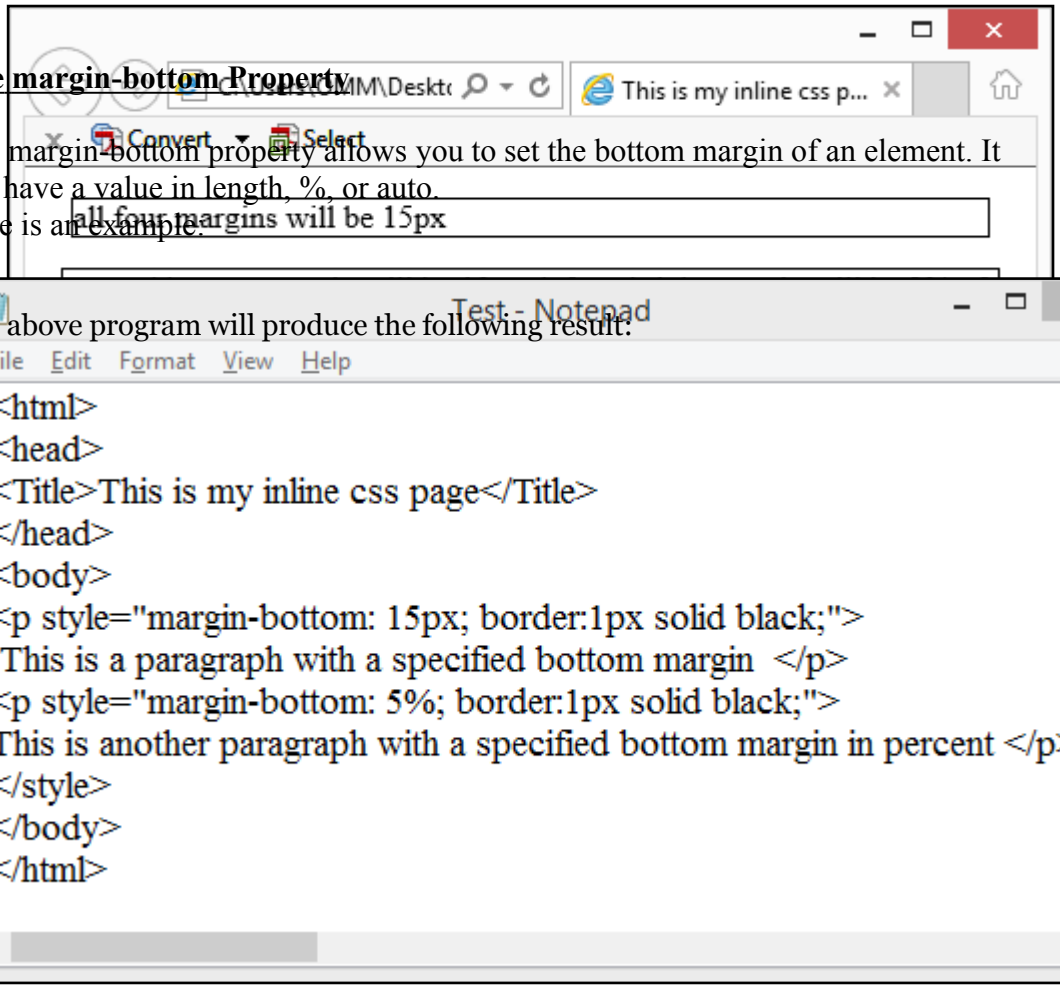
*p {margin: 10px 2% -10px}*
*top margin will be 10px, left and right margin will be 2% of the total width of the document, bottom margin will be -10px*

*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:



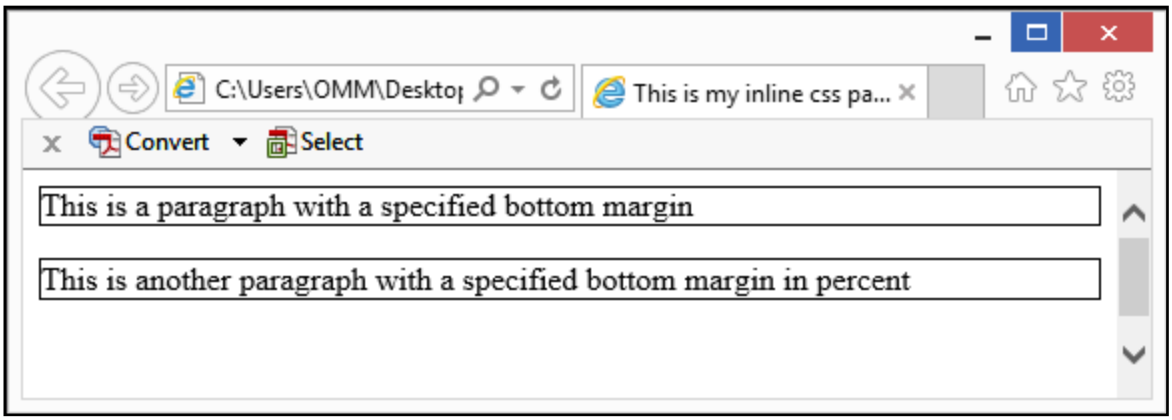
The above program will produce the following result:




**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:

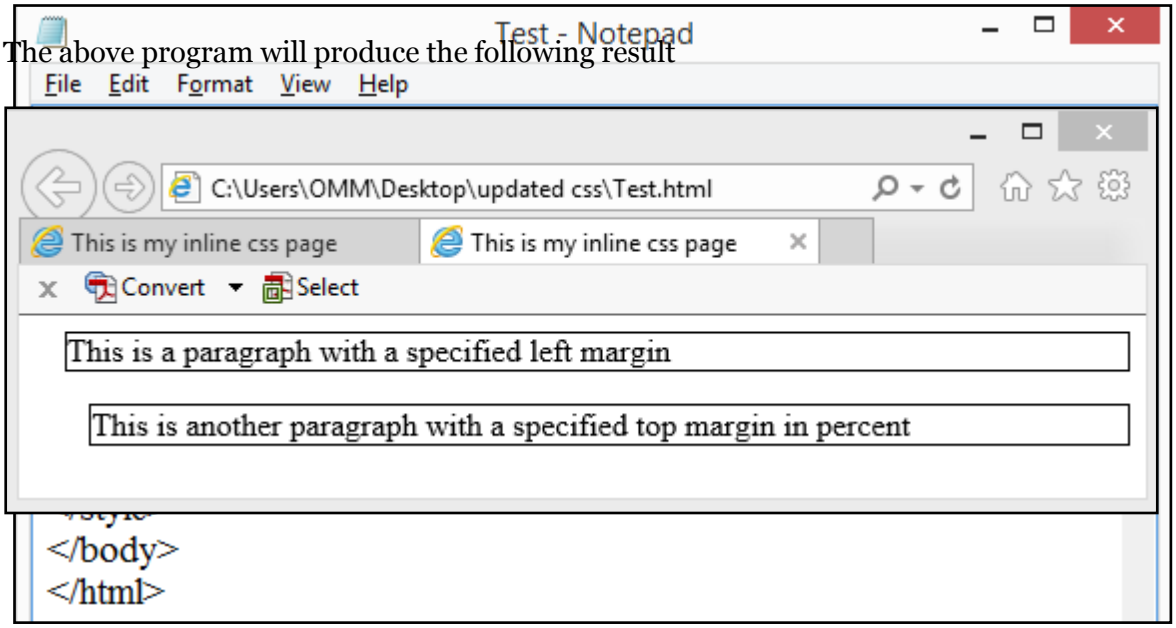
The above program will produce the following result:




**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:

The above program will produce the following result



**Check your progress 6**

**Q1. Write the different properties of the dimensions of a box.**
**Answer:**_____

_____

**Q2. Write the three properties of border.**
**Answer:**_____

_____

**Q3. Write the few major difference of outline in CSS.**
**Answer:**_____

_____

**Q4. What is a margin property?**
**Answer:**_____

_____

**Q5. Write the border-style property in CSS.**
**Answer:**_____

_____

**Let us sum up**

In this unit we have understood what is CSS, advantages of CSS, Parts of CSS, CSS syntax, CSS selectors, ways to insert CSS, background image handling, background colour management, text management, font management, managing hyperlinks, managing lists, designing tables, working with box model, designing borders, designing outline, setting page margin

**Reference**


1. W3Schools.com
2. Google.com
3. Tutorialpoints.com


**Check your progress possible answers**

**Check your progress 1**

**Q1. Write the abbreviation of CSS?**
**Answer**

CSS stands for Cascading Style Sheets. It is a simple design language intended to simplify the process of making web pages presentable.

**Q2. Write the three parts of CSS syntax**?

**Answer**

CSS style is made of three parts:

1. **Selector:** A selector is an HTML tag at which a style will be applied. This could be any tag like <h1>, <p> or <table> etc.
2. **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*, bgcolor etc.
3. **Value:** Values are assigned to properties. For example, *color* property can have the value either *red* or *#F1F1F1* etc.

**Check your progress 2**

**Q1. What are the 3 types of selectors.**

**Answer**

Three types of CSS Selectors

1. The Element selectors
2. The ID Selectors
3. The Class Selectors

**Q2. What is universal selector ?**.

**Answer**

**Universal selector**

An asterisk (*) is the universal selector for CSS. It matches a single element of any type. Omitting the asterisk with simple selectors has the same effect. For instance, *.warning and.warning are considered equal. Rather than selecting elements of a specific type

**Check your progress 3**

**Q1. How many ways to build style sheet ?what are they ?**

**Answer**

There are three ways of inserting a style sheet:

1. External style sheet
2. Internal style sheet
3. Inline style

**Q2. Write the background properties of CSS**
**Answer**

You can set the following background properties of an element:

- The **background-image** property is used to set the background image of an element.
- The **background-repeat** property is used to control the repetition of an image in the background.
- The **background-position** property is used to control the position of an image in the background.
- The **background-attachment** property is used to control the scrolling of an image in the background.
- The background property is used as shorthand to specify a number of other background properties.

**Check your progress 4**

**Q1. What is the representation of 6 digit Hexa code ?**
**Answer**
A hexadecimal is a 6 digit representation of a color. The first two digits (RR) represent a red value, the next two are a green value (GG), and the last are the blue value (BB). Each hexadecimal code will be preceded by a pound or hash sign '#'. Following are the example to use Hexadecimal notation.
#66AA77

**Q2. What are the different properties of Font ?**
**Answer**
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.

**Q3. Write the different properties of Hyperlink ?**
**Answer**
Properties of a hyperlink:

- 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.

**Check your progress 5**

**Q1. What is the different properties of table in css ?**
**Answer**

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** specify 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.

**Q2. What are the different values of ordered and unordered list in CSS ?**
**Answer**

The values, for an **unordered list**:

disc (default),Circle,Square.

The values, for an **ordered list**:

Decimal,decimal-leading-zero,lower-alpha,upper-alpha,lower-roman,upper-roman.

**Check your progress 6**

**Q1. Write the different properties of the dimensions of a box.**

**Answer**

Properties of 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.

**Q2. Write the three properties of border.**

**Answer**

There are three properties of a border:

- 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

**Q3. Write the few major difference of outline in CSS.**

**Answer**

There are few major differences are :

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

**Q4. What is a margin property?**

**Answer**

Following properties to set an element margin.

1. The **margin** specifies a shorthand property for setting the margin properties in one declaration.
2. The **margin-bottom** specifies the bottom margin of an element.
3. The **margin-top** specifies the top margin of an element.
4. The **margin-left** specifies the left margin of an element.
5. The **margin-right** specifies the right margin of an element

**Q5. Write the border-style property in CSS.**

**Answer**

The border-style property :

- **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.

# CSS Advanced

**Structure**

Padding using CSS
The padding-bottom Property
The padding-top Property
Setting Display Using CSS
Block-level Elements
Inline element
Setting Width And Max Width using CSS
Setting Position using CSS
Relative Positioning
Absolute Positioning
Fixed Positioning
Setting the Float Property using CSS
Element float
Turning off Float - Using Clear
All CSS float
Inline Block property
Horizontal Alignment in CSS
Center Align Elements
Center alignment Text
Center an image
Left and Right Align - Using position
Working With Combinatory
Descendant Selector
Child Selector
Adjacent Sibling Selector
General sibling selector

Working with Pseudo-class
Anchor Pseudo-classes
Pseudo classes and CSS classes
Simple Tooltip Hover
The first child pseudo classes
Match the first <p> element
working with Pseudo-elements
The: first-line pseudo-element
The: first-letter pseudo-element
The: before pseudo-element
The: after pseudo-element
The: selection pseudo-element
Creating a navigation bar
Navigation Bar = List of Links
4.11.2Vertical navigation bar
Center Links & Add Borders
Horizontal Navigation Bar
Working with images
The Image Border Property
The Image width Property
Working with Attribute selectors
4.13.1Multiple style rules
CSS [attribute] Selector
CSS [attribute="value"] Selector
CSS [attribute ~ ="value"] Selector
CSS [attribute |="value"] Selector
CSS [attribute ^="value"] Selector
CSS [attribute $="value"] Selector
CSS [attribute *="value"] Selector
Check Your Progress Possible answer
Reference

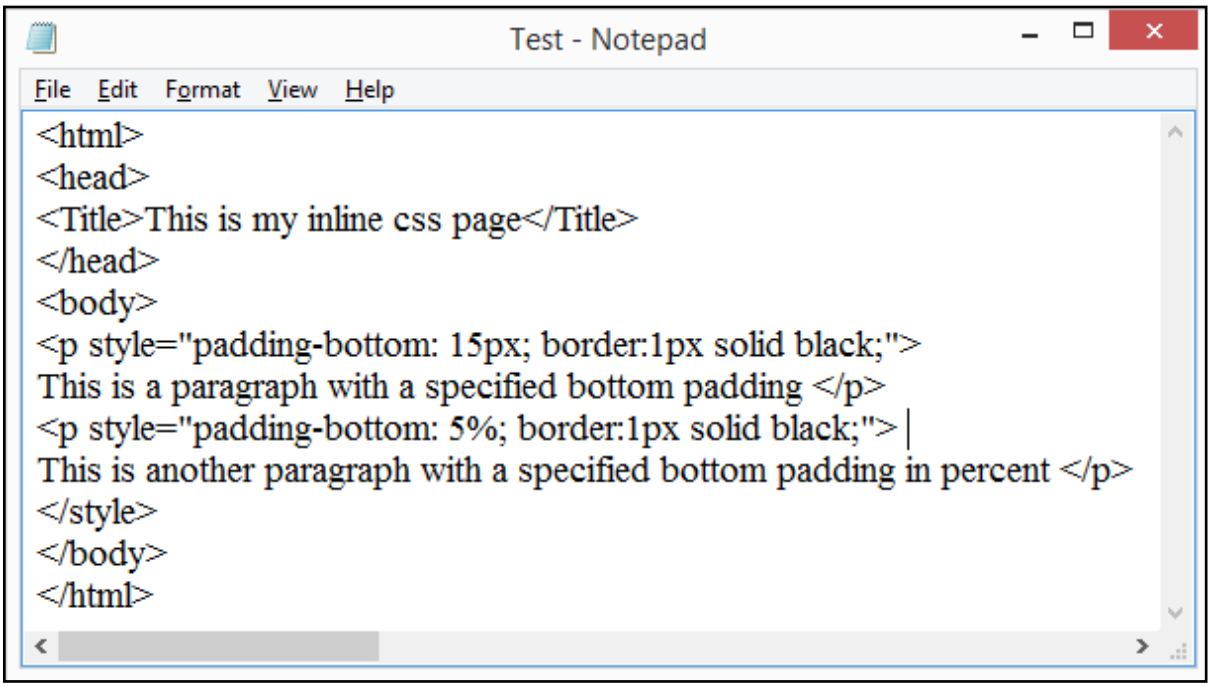
## Padding using CSS

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 inherits. 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:

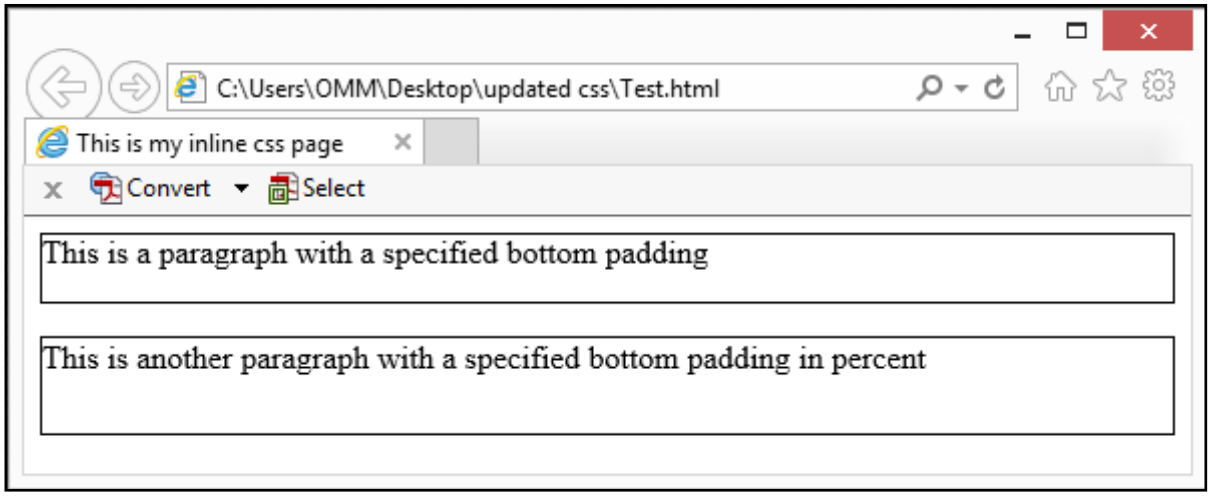
- The padding-bottom specifies the bottom padding of an element.
- The padding-top specifies the top padding of an element.
- The padding-left specifies the left padding of an element.
- The padding-right specifies the right padding of an element.
- The padding serves as shorthand for the preceding properties.

### 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:

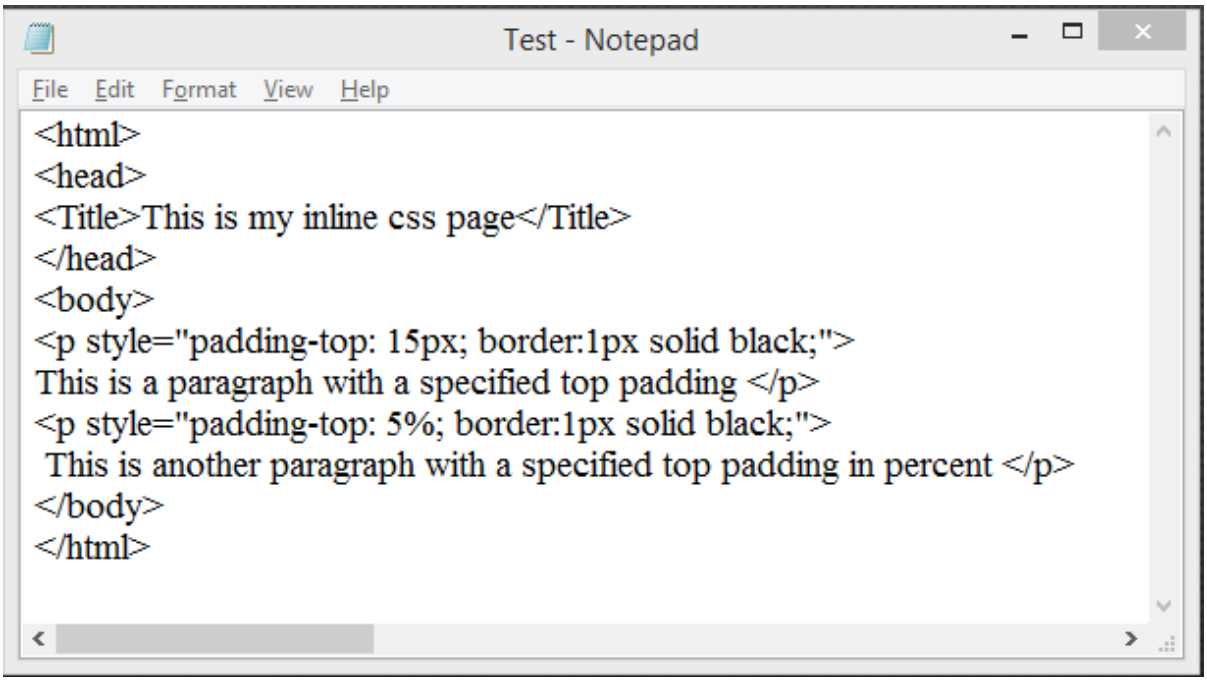



The above program will produce the following result

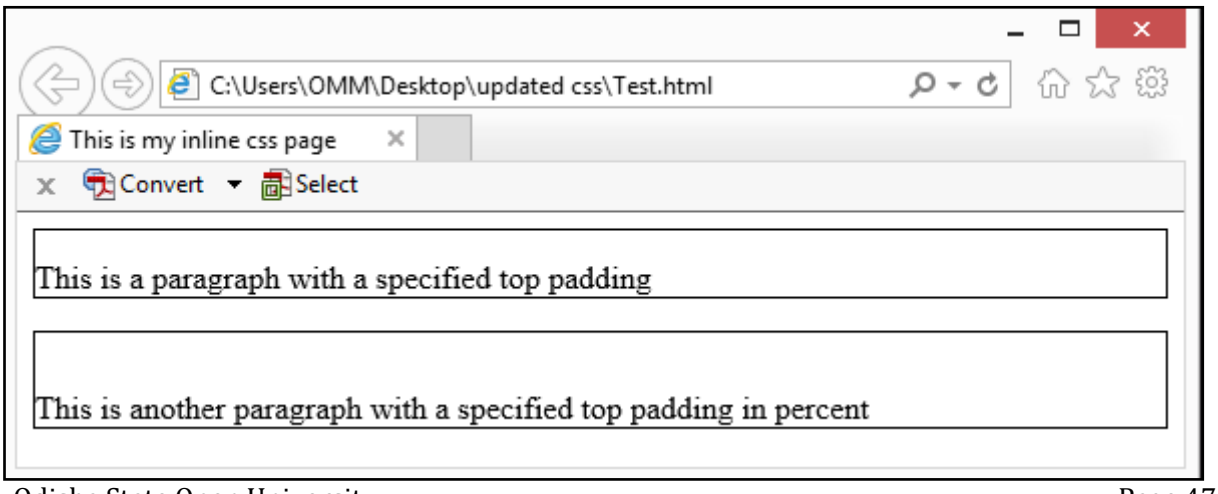



**The padding-top Property**

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




The above program will produce the following result:



## Setting Display Using CSS

The display property is the most important CSS property for controlling layout. The display property specifies if/how an element is displayed. Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline. A block element is often called a block-level element. An inline element is always just called an inline element.

### Block-level Elements

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The  element is a block-level element.

Examples of block-level elements:

- 
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

div is the standard block-level element. A block-level element starts on a new line and stretches out to the left and right as far as it can. Other common block-level elements are ***p*** and ***form***, and new in HTML5 are header, footer, section, and more.

### Inline Elements

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline  element inside a paragraph.

Span is the standard inline element. An inline element can wrap some text inside a paragraph  like this  without disrupting the flow of that paragraph. The ***a*** element is the most common inline element, since you use them for links.

Examples of inline elements:

- 
- <a>
- <img>

**CSS Syntax**

*Display: value*;

The values of ***display property*** have the following meanings:

- **Block** -This value causes an element to generate a block box.
- **Inline-block** - This value causes an element to generate an inline-level block container. The inside of an inline-block is formatted as a block box, and the element itself is formatted as an atomic inline-level box.
- **Inline** - This value causes an element to generate one or more inline boxes.
- **list-item** - This value causes an element (e.g., LI in HTML) to generate a principal block box and a marker box.
- **None** - This value causes an element to not appear in the formatting structure

Here are some examples of the display property:

```
p    { display: block }
em {display: inline }
li  { display: list-item }
```

**Setting width and Maxwidth using CSS**

Setting the width of a block-level element will prevent it from stretching out to the edges of its container. Then, you can set the margins to auto, to horizontally center the element within its container. The element will take up the specified width, and the remaining space will be split equally between the two margins:

This  element has a width of 500px, and margin set toauto.

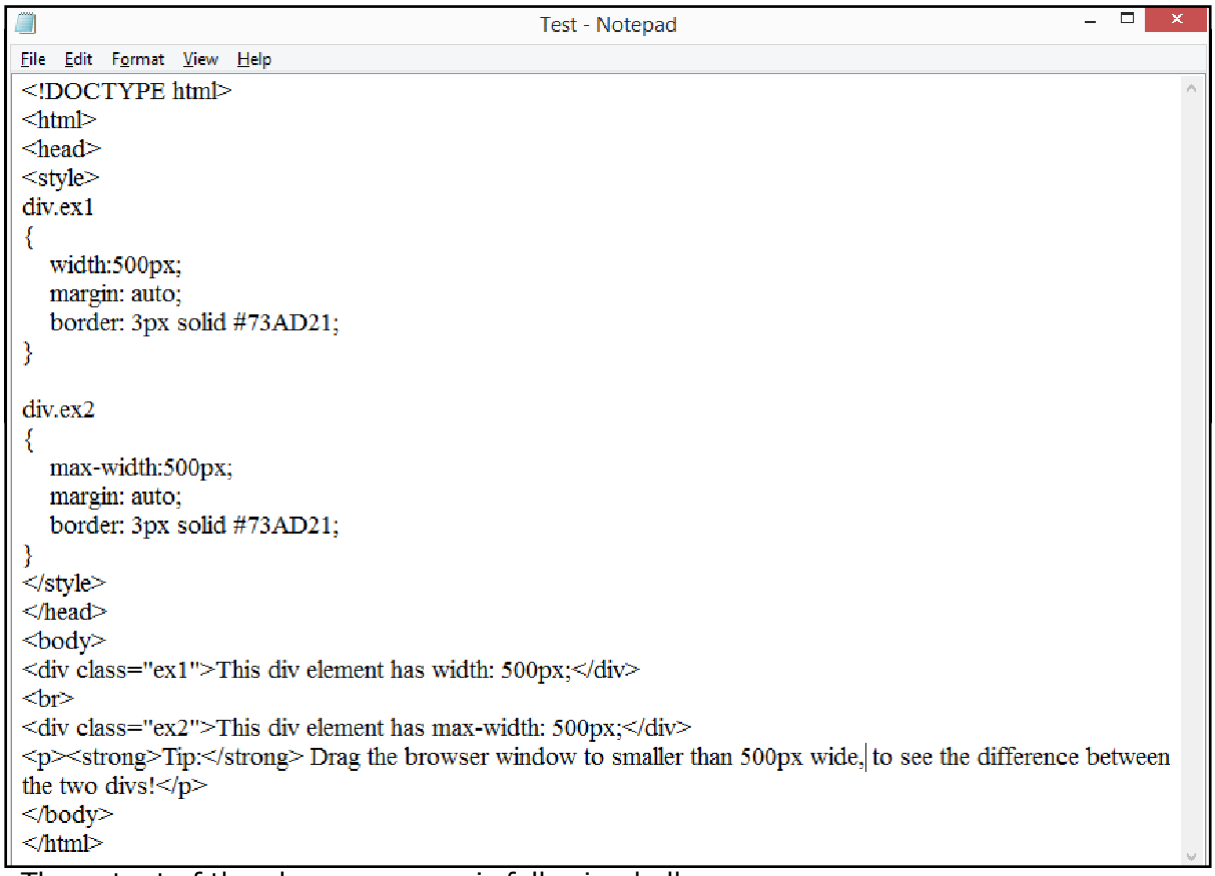
**Note:** The problem with the  above occurs when the browser window is smaller than the width of the element. The browser then adds a horizontal scrollbar to the page.

Using max-width instead, in this situation, will improve the browser's handling of small windows. This is important when making a site usable on small devices:

This  element has a max-width of 500px, and margin set to auto.

**Tip:** Resize the browser window to less than 500px wide, to see the difference between the two divs!

Here is an example of the two divs above:




The output of the above program is following bellow:

## Setting Position using CSS

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

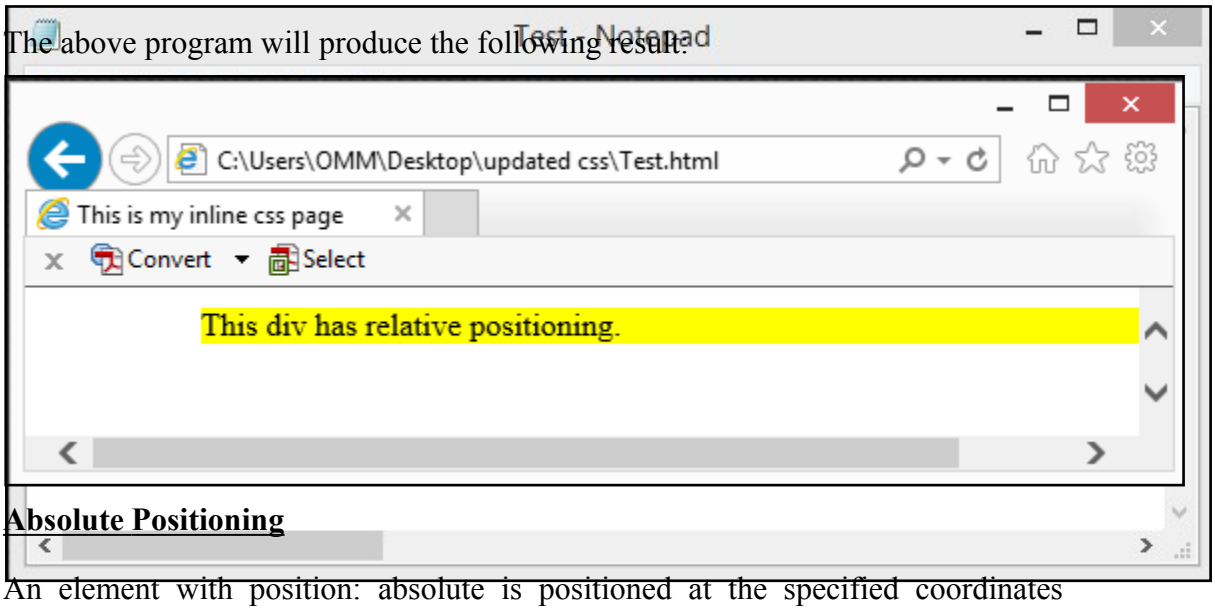
Relative positioning changes the position of the HTML element relative to where it normally appears. So "left:20" adds 20 pixels to the element's LEFT position. You can use two values top and left along with the position property to move an HTML element anywhere in an HTML document.

- 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:

The above program 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:

* *
*This div has absolute positioning.*

* *

**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:

* *
*This div has fixed positioning.*
* *

---

**Setting the float property using CSS**

---

A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the property).

With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it. Float is very often used for images, but it is also useful when working with layouts.

A line box is next to a float when there exists a vertical position that satisfies all of these four conditions:
(a) At or below the top of the line box,
(b) At or above the bottom of the line box
(c) Below the top margin edge of the float, and
(d) Above the bottom margin edge of the float.

**Elements Float**

Elements are floated horizontally; this means that an element can only be floated left or right, not up or down. A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element. The elements after the floating element will flow around it. The elements before the floating element will not be affected. If an image is floated to the right, a following text flows around it, to the left:

**Example:**

```
img { float:right; }
```

If you place several floating elements after each other, they will float next to each other if there is room. Here we have made an image gallery using the float property:

**Example**

```
.thumbnail { float:left; width:110px; height:90px; margin:5px; }
```

## Turning off Float - Using Clear

Elements after the floating element will flow around it. To avoid this, use the clear property. The clear property specifies which sides of an element other floating elements are not allowed. Add a text line into the image gallery, using the clear property:

**Example:**

```
.text_line { clear:both; }
```

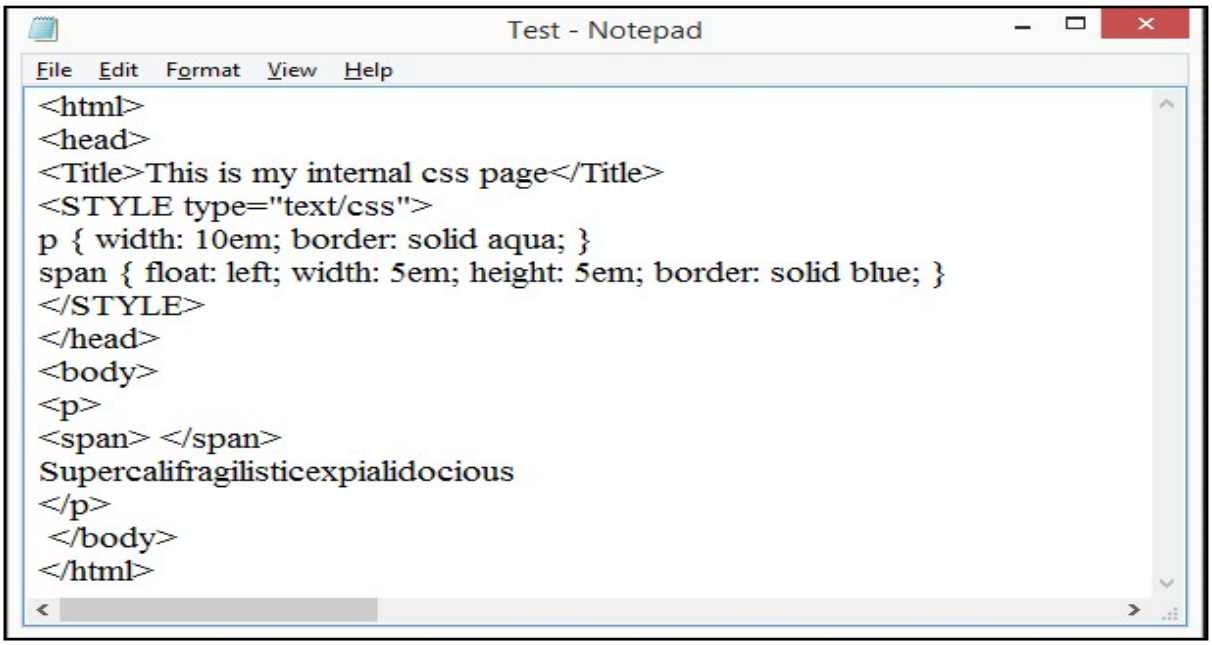
## All CSS Float Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

| Property | Description | Values | CSS |
|---|---|---|---|
| clear | Specifies which sides of an element where other floating elements are not allowed | left right both none inherit | 1 |
| float | Specifies whether or not a box should float | left right none inherit | 1 |

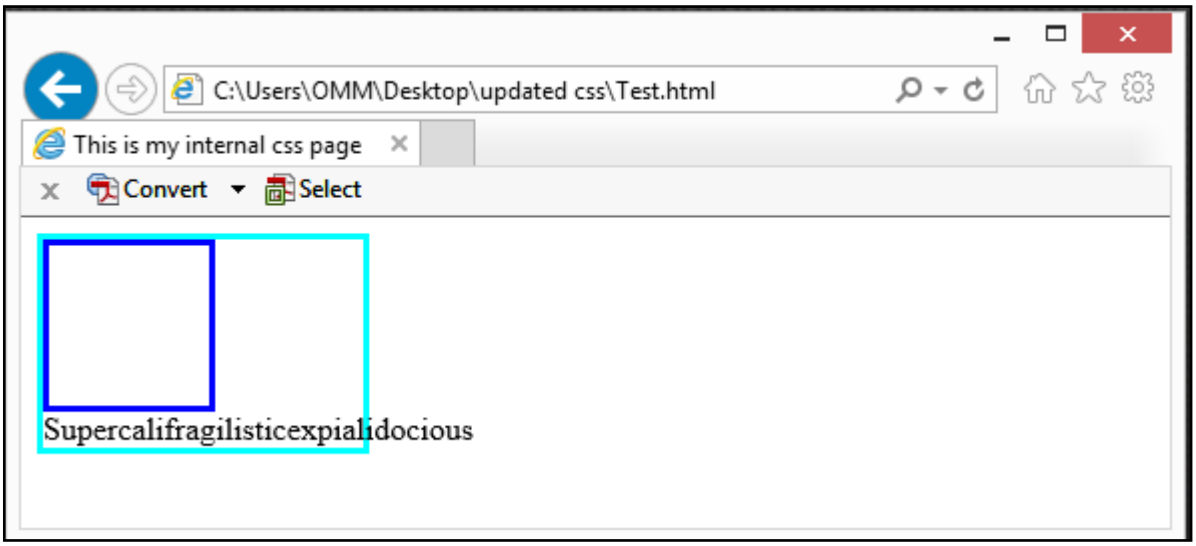
Example:

The above program will produce the following result:


Test - Notepad
File Edit Format View Help
<html>
<head>
<Title>This is my internal css page</Title>
<STYLE type="text/css">
p { width: 10em; border: solid aqua; }
span { float: left; width: 5em; height: 5em; border: solid blue; }
</STYLE>
</head>
<body>

Supercalifragilisticexpialidocious

</body>
</html>


C:\Users\OMM\Desktop\updated css\Test.html
This is my internal css page
Convert
Select
Supercalifragilisticexpialidocious

## Check your progress 1

**Q1. What is Float ?**
**Answer:**_____

_____

**Q2. Write the different properties of position?**
**Answer:**_____

_____

**Q3. What is padding?**

**Answer:**_____

_____

**Q4. Write the use of Display properties?**

**Answer:**_____

_____

## Inline Block Properties

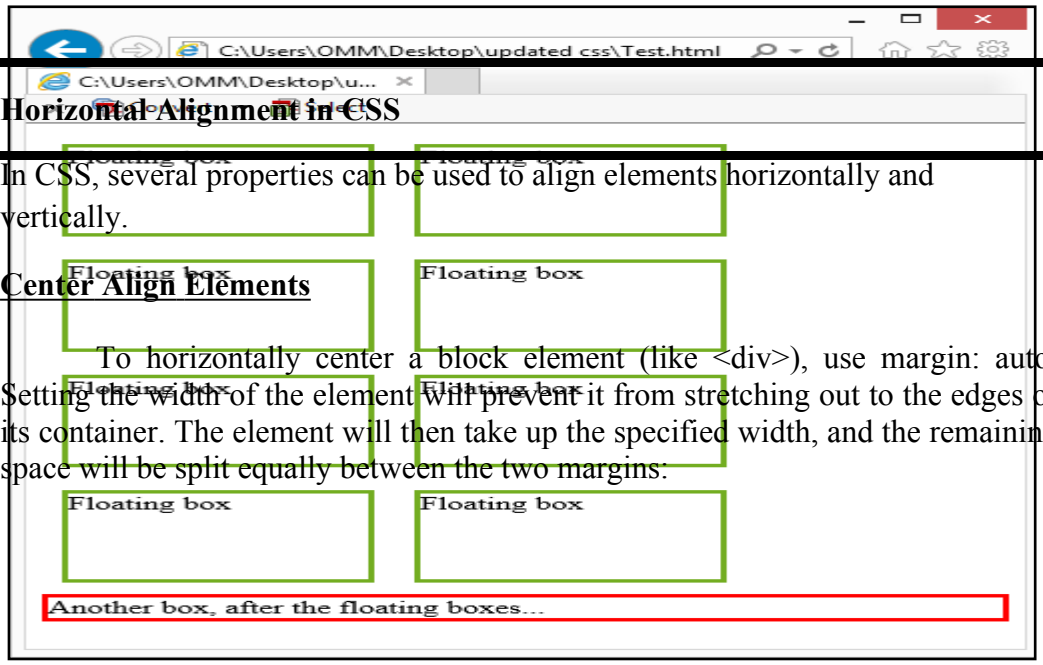
It has been possible for a long time to create a grid of boxes that fills the browser width and wraps nicely (when the browser is resized), by using the float property. However, the inline-block value of the display property makes this even easier. Inline-block elements are like inline elements but they can have a width and a height.

**Examples**
The old way - using float (notice that we also need to specify a clear property for the element after the floating boxes):

```
<!DOCTYPE html>
<html>
<head>
<style>
.floating-box
        {
          float: left;
        width: 150px;
        height: 75px;
        margin: 10px;
         border: 3px solid #73AD21;
        }

.after-box
         {
         clear: left;
         border: 3px solid red;
        }
```

```
</style>
</head>
<body>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="after-box">Another box, after the floating boxes...</div>
</body>
</html>
```

The above program will produce the following result




## Horizontal Alignment in CSS

In CSS, several properties can be used to align elements horizontally and vertically.

### Center Align Elements

To horizontally center a block element (like ), use margin: auto; Setting the width of the element will prevent it from stretching out to the edges of its container. The element will then take up the specified width, and the remaining space will be split equally between the two margins:

Example:

```
.center
        {
        margin: auto;
        width: 50%;
        border: 3px solid green;
        padding: 10px;
        }
```

**Note**: Center aligning has no effect if the width property is not set (or set to 100%).

## Center Align Text

To just center the text inside an element, use text-align: center;

Example:

```
.center
        {
           text-align: center;
           border: 3px solid green;
        }
```

## Center an Image

To center an image, use margin: auto; and make it into a block element:

```
 img
{
display: block;
margin: auto;
width: 40%;
}
```

## Left and Right Align - Using position

One method for aligning elements is to use position: absolute;:

Example:

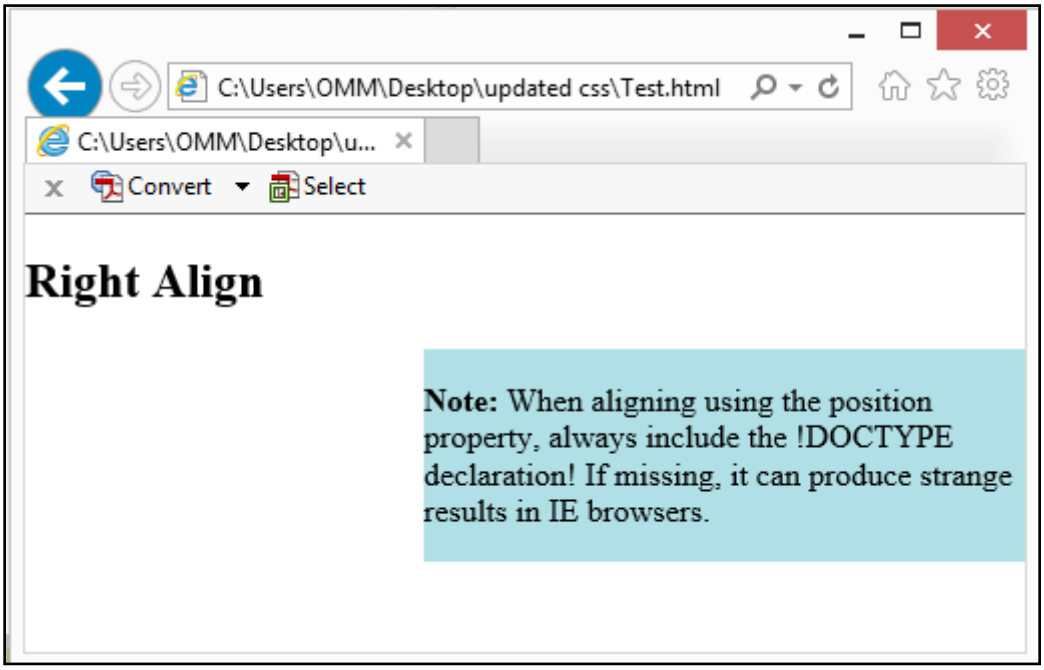
```
.right {
        position: absolute;
         right: 0px;
        width: 300px;
         border: 3px solid #73AD21;
        padding: 10px;
}
```

**Note:** Absolute positioned elements are removed from the normal flow, and can overlap elements.

**Tip:** When aligning elements with position, always define margin and padding for the <body> element. This is to avoid visual differences in different browsers. There is also a problem with IE8 and earlier, when using position. If a container element (in our case ) has a specified width, and the !DOCTYPE declaration is missing, IE8 and earlier versions will add a 17px margin on the right side. This seems to be space reserved for a scrollbar. So, always set the !DOCTYPE declaration when using position:

```
<!DOCTYPE html>
<html>
<head>
<style>
body
        {
        margin: 0;
         padding: 0;
        }
.container
        {
         position: relative;
         width: 100%;
        }
.right
        {
         position: absolute;
        right: 0px;
        width: 300px;
         background-color: #b0e0e6;
        }
        </style>
</head>
<body>

<h2>Right Align</h2>

<div class="container">
  <div class="right">
        <p><b>Note: </b>When aligning using the position property,
        always include the !DOCTYPE declaration! If missing, it can
        produce strange results in IE browsers.</p>
  </div>
</div>
</body>
</html>
```

The above program will produce the following result




## Working with Combinatory

A combinatory is something that explains the relationship between the selectors. A CSS selector can contain more than one simple selector. Between the simple selectors, we can include a combinatory.

There are four different combinatory in CSS:

- descendant selector (space)
- child selector (>)
- adjacent sibling selector (+)
- general sibling selector (~)

### Descendant Selector

The descendant selector matches all elements that are descendants of a specified element.

The following example selects all <p> elements inside  elements:

Example

```
div p
{       background-color: yellow;

}
```

**Child Selector**

The child selector selects all elements that are the immediate children of a specified element.

The following example selects all <p> elements that are immediate children of a  element:

Example

```
<div > p {
        background-color: yellow;
        }
```

**Adjacent Sibling Selector**

The adjacent sibling selector selects all elements that are the adjacent siblings of a specified element. Sibling elements must have the same parent element, and "adjacent" means "immediately following". The following example selects all <p> elements that are placed immediately after  elements:

Example:

```
div + p {
   background-color: yellow;
}
```

**General Sibling Selector**

The general sibling selector selects all elements that are siblings of a specified element. The following example selects all <p> elements that are siblings of  elements:

Example:

```
div ~ p {
   background-color: yellow;
}
```

---

**Working with Pseudo-class**

A pseudo-class is used to define a special state of an element. For example, it can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

**Syntax**

The syntax of pseudo-classes:

selector: pseudo-class

```
{
        property: value;
}
```

**Anchor Pseudo-classes**

Links can be displayed in different ways:

Example

```
/* unvisited link */
a:link {
   color: #FF0000;
}

/* visited link */
a:visited {
   color: #00FF00;
}

/* mouse over link */
a:hover {
   color: #FF00FF;
}

/* selected link */
a:active {
   color: #0000FF;
}
```

**Note:** a: hover MUST come after a: link and a:visited in the CSS definition in order to be effective!a:active MUST come after a:hover in the CSS definition in order to be effective! Pseudo-class names are not case-sensitive.

**Pseudo-classes and CSS Classes**

Pseudo-classes can be combined with CSS classes: When you hover over the link in the example, it will change color:

Example

```
a.highlight:hover
 {
   color: #ff0000;
}
```

Hover on 

An example of using the: hover pseudo-class on a  element:

Example

```
div:hover
{
   background-color: blue;
}
```

**Simple Tooltip Hover**

Hover over a  element to show a <p> element (like a tooltip):

Hover over me to show the <p> element.

Example

```
p
{
        display: none;
        background-color: yellow;
       padding: 20px;
}

div:hover p
{
   display: block;
}
```

**The: first-child Pseudo-class**

The: first-child pseudo-class matches a specified element that is the first child of another element.

**Match the first <p> element**

In the following example, the selector matches any <p> element that is the first child of any element:

Example

```
p: first-child

{
        color: blue;
}
```

Match the first <i> element in all <p> elements
In the following example, the selector matches the first <i> element in all <p> elements:

Example

```
p i:first-child
{
   color: blue;
}
```

| Selector | Example | Example description |
|---|---|---|
| :active | a:active | Selects the active link |
| :checked | input:checked | Selects every checked <input> element |
| :disabled | input:disabled | Selects every disabled <input> element |

| | | |
|---|---|---|
| :empty | p:empty | Selects every <p> element that has no children |
| :enabled | input:enabled | Selects every enabled <input> element |
| :first-child | p:first-child | Selects every <p> elements that is the first child of its parent |
| :first-of-type | p:first-of-type | Selects every <p> element that is the first <p> element of its parent |
| :focus | input:focus | Selects the <input> element that has focus |
| :hover | a:hover | Selects links on mouse over |
| :in-range | input:in-range | Selects <input> elements with a value within a specified range |
| :invalid | input:invalid | Selects all <input> elements with an invalid value |
| :lang(language) | p:lang(it) | Selects every <p> element with a lang attribute value starting with "it" |
| :last-child | p:last-child | Selects every <p> elements that is the last child of its parent |
| :last-of-type | p:last-of-type | Selects every <p> element that is the last <p> element of its parent |
| :link | a:link | Selects all unvisited links |
| :not(selector) | :not(p) | Selects every element that is not a <p> element |
| :nth-child(n) | p:nth-child(2) | Selects every <p> element that is the second child of its parent |
| :nth-last-child(n) | p:nth-last-child(2) | Selects every <p> element that is the second child of its parent, counting from the last child |
| :nth-last-of-type(n) | p:nth-last-of-type(2) | Selects every <p> element that is the second <p> element of its parent, counting from the last child |
| :nth-of-type(n) | p:nth-of-type(2) | Selects every <p> element that is the second <p> element of its parent |
| :only-of-type | p:only-of-type | Selects every <p> element that is the only <p> element of its parent |
| :only-child | p:only-child | Selects every <p> element that is the only child of its parent |
| :optional | input:optional | Selects <input> elements with no |

| | | |
|---|---|---|
| | | "required" attribute |
| :out-of-range | input:out-of-range | Selects <input> elements with a value outside a specified range |
| :read-only | input:read-only | Selects <input> elements with a "readonly" attribute specified |
| :read-write | input:read-write | Selects <input> elements with no "readonly" attribute |
| :required | input:required | Selects <input> elements with a "required" attribute specified |
| :root | root | Selects the document's root element |
| :target | #news:target | Selects the current active #news element (clicked on a URL containing that anchor name) |
| :valid | input:valid | Selects all <input> elements with a valid value |
| :visited | a:visited | Selects all visited links |

## Working with Pseudo-elements

CSS pseudo-elements are used to add special effects to some selectors. You do not need to use JavaScript or any other script to use those effects. A simple syntax of pseudo-element is as follows:

*selector: pseudo-element {property: value}*

CSS classes can also be used with the pseudo-elements:

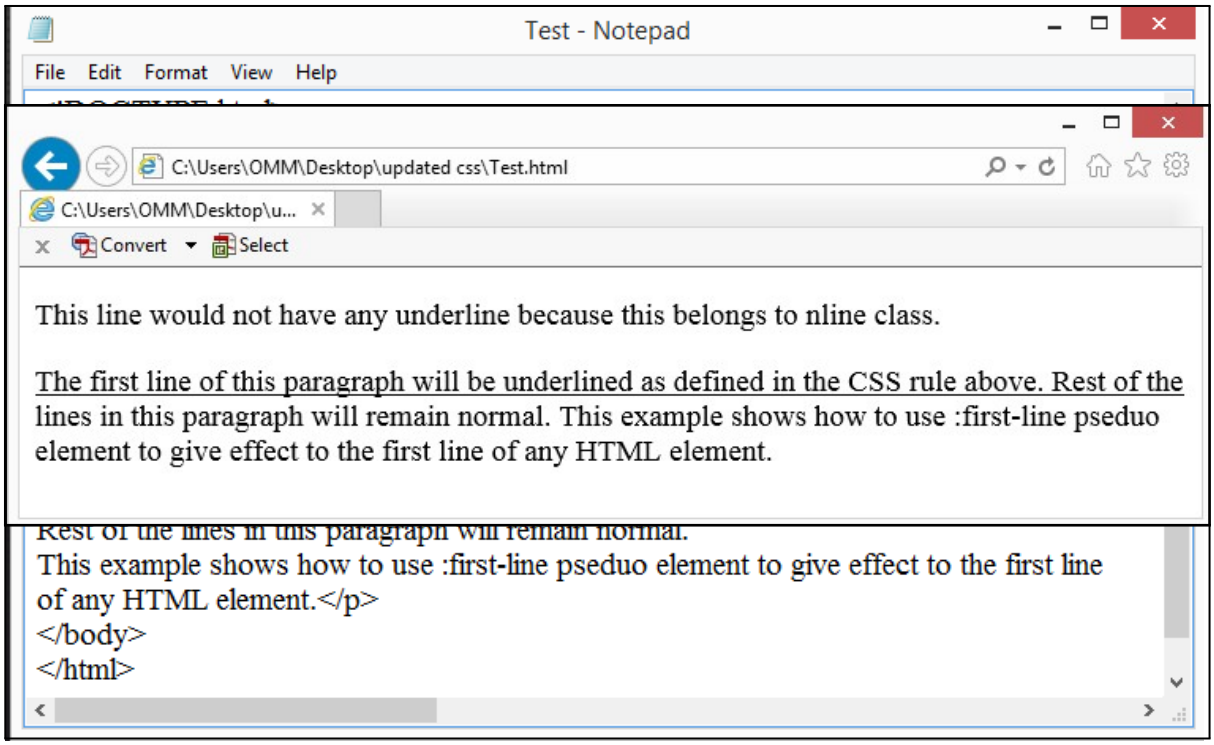
*selector. class: pseudo-element {property: value}*

The most commonly used pseudo-elements are as follows:

| Value | Description |
|---|---|
| :first-line | Use this element to add special styles to the first line of the text in a selector. |
| :first-letter | Use this element to add special style to the first letter of the text in a selector. |
| :before | Use this element to insert some content before an element. |
| :after | Use this element to insert some content after an element. |

## The: first-line pseudo-element

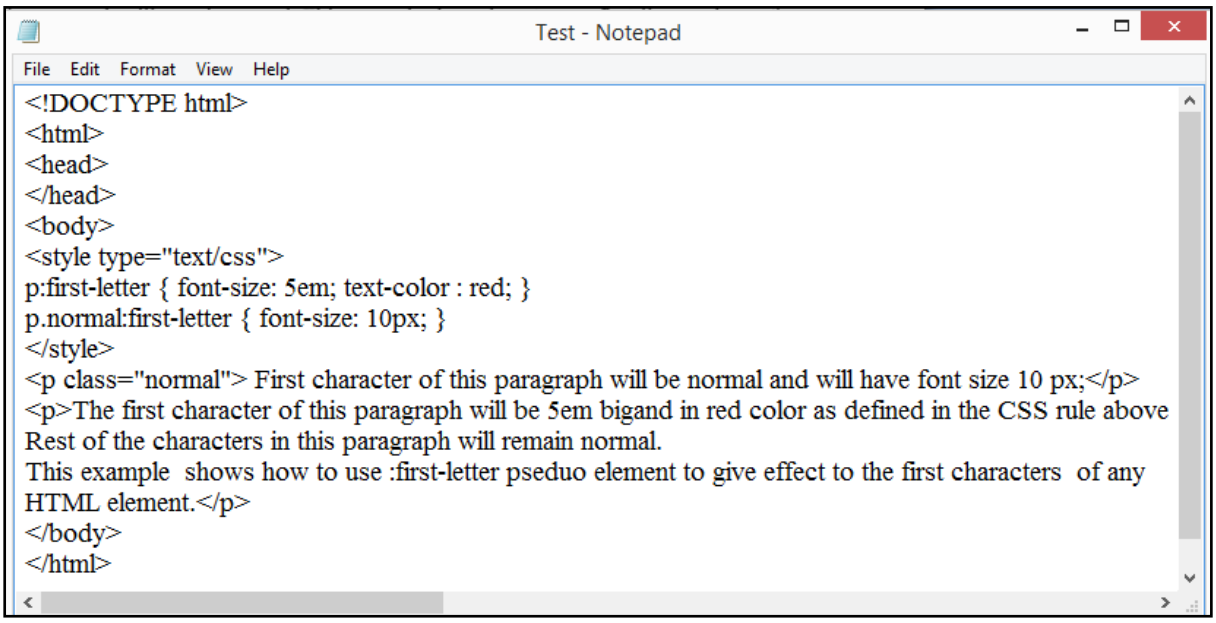
The following example demonstrates how to use the: first-line element to add special effects to the first line of elements in a document.



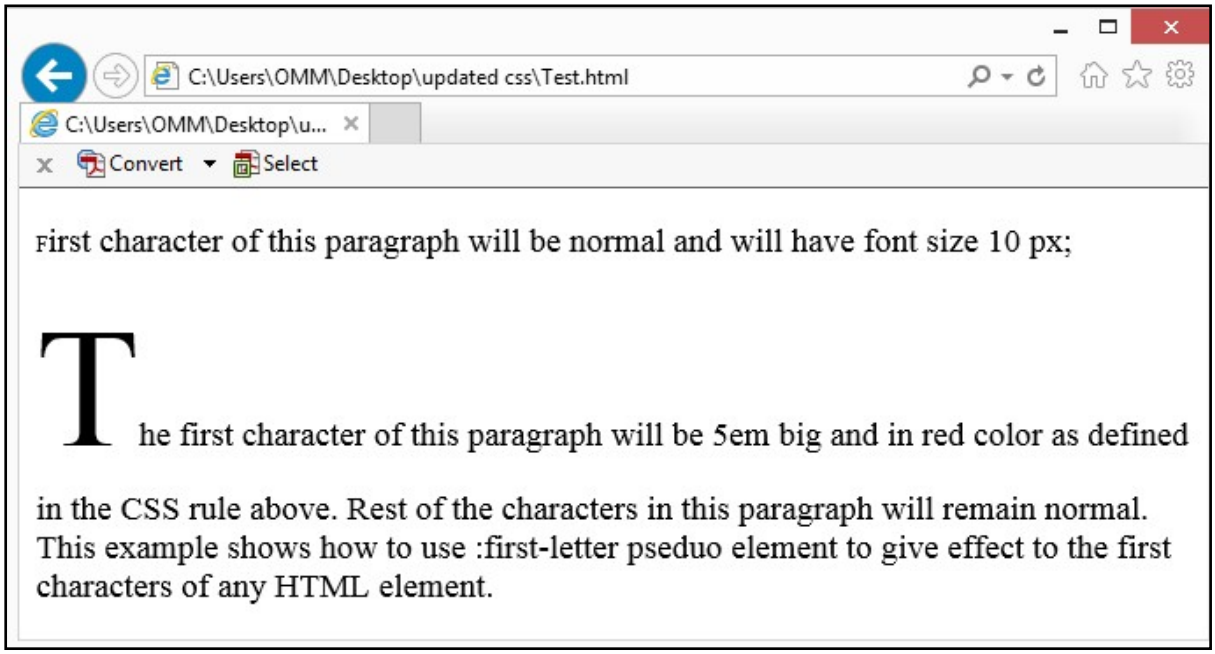

The above program will produce the following result :

## The: first-letter pseudo-element

The following example demonstrates how to use the: first-letter element to add special effect to the first letter of elements in the document.




The above program will produce the following result:



**The: before pseudo-element**

The ***: before pseudo-element*** can be used to insert some content before the content of an element. The following example demonstrates how to use :before element to add some content before any element.

```
<style type="text/css">
p:before
{
 content: url(/images/bullet.gif)
}
</style>
<p> This line will be preceded by a bullet.</p>
<p> This line will be preceded by a bullet.</p>
<p> This line will be preceded by a bullet.</p>
```

**The: after pseudo-element**

The :after pseudo-element can be used to insert some content after the content of an element. The following example demonstrates how to use: after element to add some content after any element.

```
<style type="text/css">
p:after
{
content: url(/images/bullet.gif)
}
</style>
<p> This line will be succeeded by a bullet.</p>
<p> This line will be succeeded by a bullet.</p>
<p> This line will be succeeded by a bullet.</p>
```

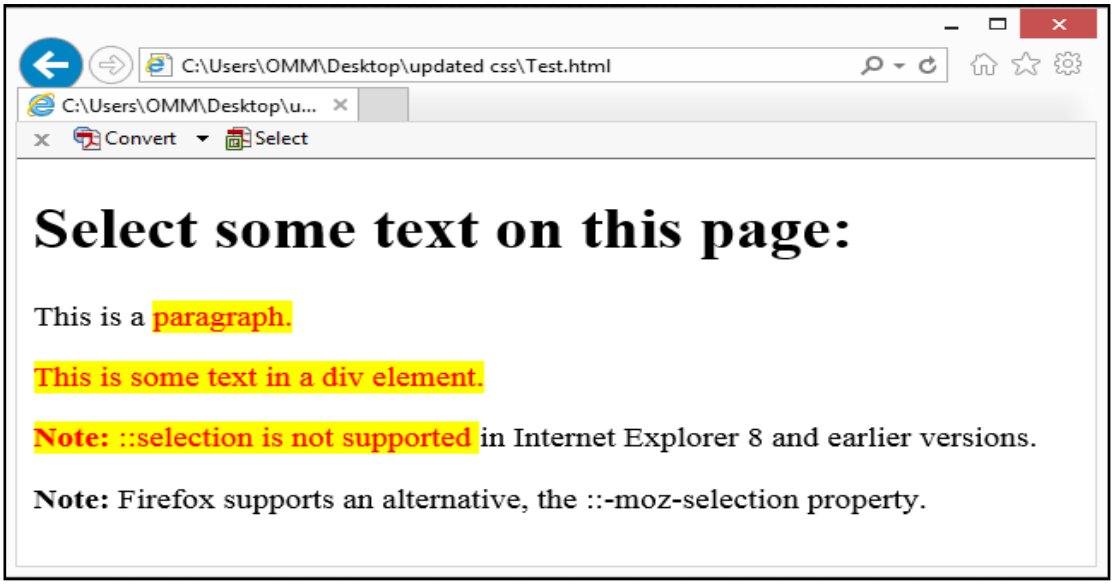
**The ::selection Pseudo-element**

The***: selection pseudo-element*** matches the portion of an element that is selected by a user. The following CSS properties can be applied to:
selection: color, background, cursor, and outline. The following example makes the selected text red on a yellow background: Here is an example:

```
<!DOCTYPE html>
<html>
<head>
<style>
::-moz-selection { /* Code for Firefox */
   color: red;
   background: yellow;
```

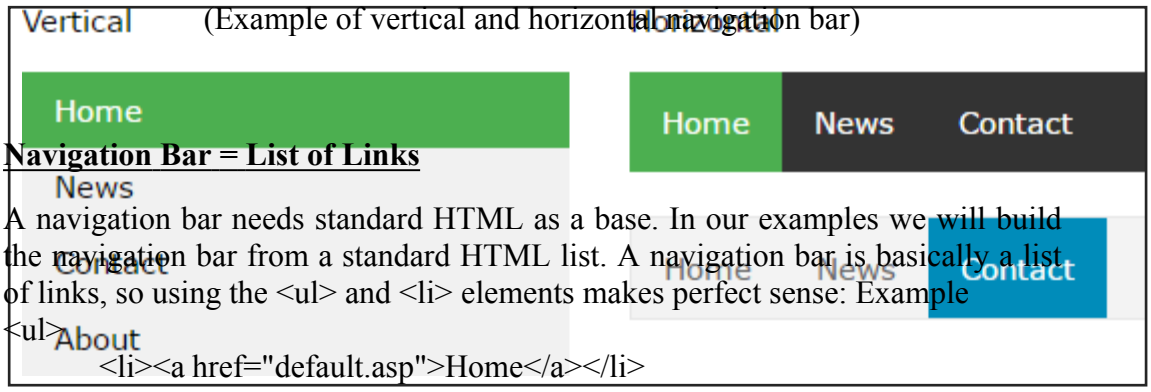
```
}
::selection {
   color: red;
   background: yellow;
}
</style>
</head>
<body>
<h1>Select some text on this page:</h1>
<p>This is a paragraph.</p>
<div>This is some text in a div element.</div>
<p><strong>Note:</strong> ::selection is not supported in Internet
Explorer 8 and earlier versions.</p>
<p><strong>Note:</strong> Firefox supports an alternative, the ::-moz-
selection property.</p>
</body>
</html>
```

The above program will produce the following result:



## Creating a navigation bar

A **navigation bar** (or **navigation** system) is a section of a graphical user interface intended to aid visitors in accessing information. **Navigation bars** are implemented in file browsers, web browsers and as a design element of some web sites. A set of buttons or images in a row or column that serves as a control point to link the user to sections on a Web site. The **navigation bar** may also be a single graphic image with multiple selections.




(Example of vertical and horizontal navigation bar)

**Navigation Bar = List of Links**

A navigation bar needs standard HTML as a base. In our examples we will build the navigation bar from a standard HTML list. A navigation bar is basically a list of links, so using the <ul> and <li> elements makes perfect sense: Example

```
<ul>
        <li><a href="default.asp">Home</a></li>
        <li><a href="news.asp">News</a></li>
        <li><a href="contact.asp">Contact</a></li>
        <li><a href="about.asp">About</a></li>
</ul>
```

Now let's remove the bullets and the margins and padding from the list:

```
ul
        {
                list-style-type:none;
                margin:0;
                padding:0;
        }
```

Example explained:

- list-style-type:none - Removes the bullets. A navigation bar does not need list markers
- Setting margins and padding to 0 to remove browser default settings

The code in the example above is the standard code used in both vertical, and horizontal navigation bars.

## Vertical Navigation Bar

To build a vertical navigation bar we only need to style the <a> elements, in addition to the code above:

Example

```
a { display:block; width:60px; }
```

Example explained:

- display:block - Displaying the links as block elements makes the whole link area clickable (not just the text), and it allows us to specify the width
- width:60px - Block elements take up the full width available by default. We want to specify a 60 px width .

**Note:** Always specify the width for <a> elements in a vertical navigation bar. If you omit the width, IE6 can produce unexpected results. You can also set the width of <ul>, and remove the width of <a>, as they will take up the full width available when displayed as block elements. This will produce the same result as our previous example:

```
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    width: 60px;
}

li a {
    display: block;
}
```

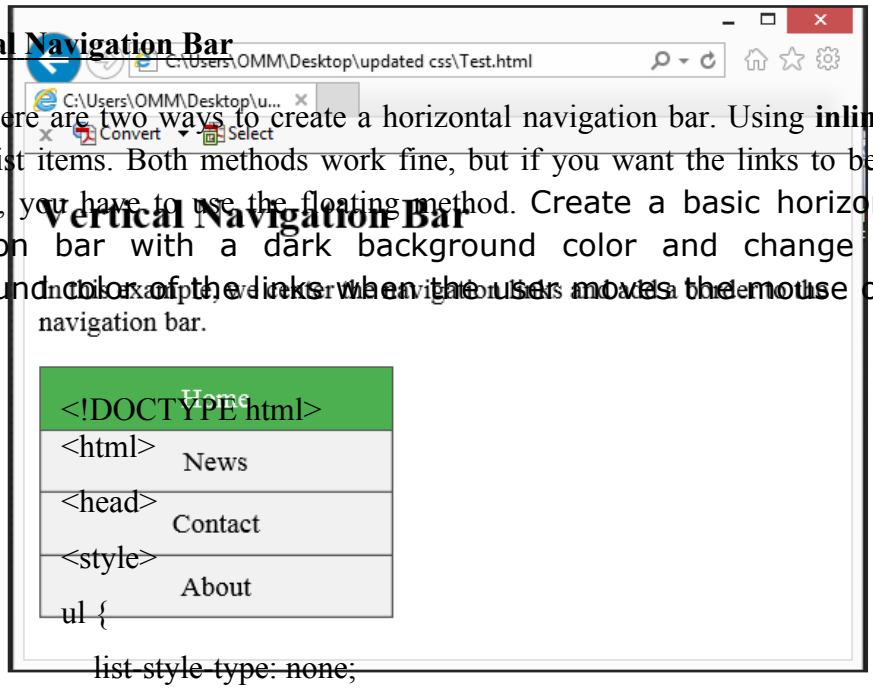
## Center Links & Add Borders

Add text-align:center to <li> or <a> to center the links. Add the border property to <ul> add a border around the navbar. If you also want borders inside the navbar, add aborder-bottom to all <li> elements, except for the last one:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
```

```
    padding: 0;
    width: 200px;
    background-color: #f1f1f1;
    border: 1px solid #555;
}

li a {
    display: block;
    color: #000;
    padding: 8px 16px;
    text-decoration: none;
}

li {
    text-align: center;
    border-bottom: 1px solid #555;
}
li:last-child {
    border-bottom: none;
}

li a.active {
    background-color: #4CAF50;
    color: white;
}

li a:hover:not(.active) {
    background-color: #555;
    color: white;
}
</style>
</head>
<body>

<h2>Vertical Navigation Bar</h2>
<p>In this example, we center the navigation links and add a border to the navigation bar.</p>

<ul>
  <li><a class="active" href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
</body>
</html>
```

The above program will produce the following result

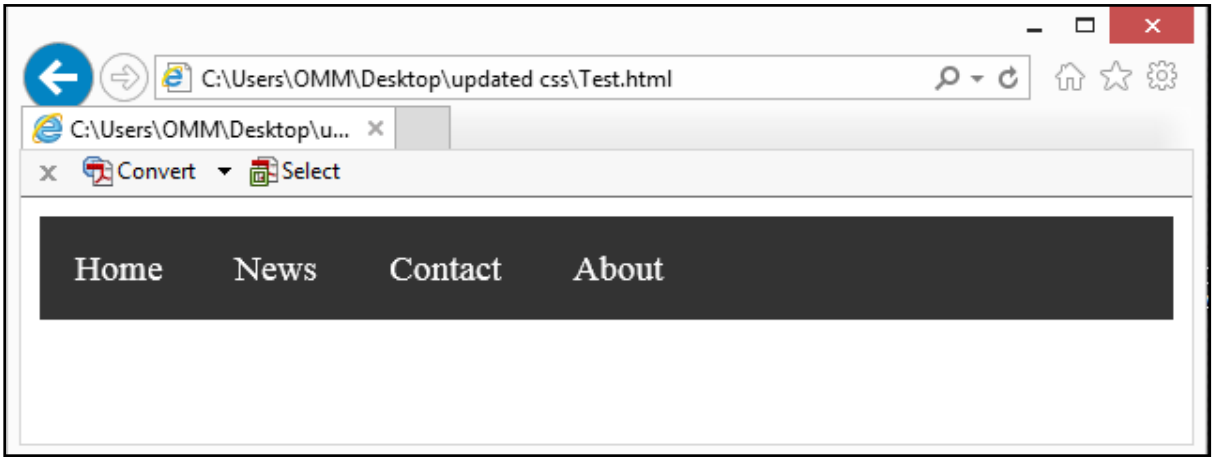



**Horizontal Navigation Bar**

There are two ways to create a horizontal navigation bar. Using **inline** or **floating** list items. Both methods work fine, but if you want the links to be the same size, you have to use the floating method. Create a basic horizontal navigation bar with a dark background color and change the background color of the links when the user moves the mouse over them:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #333;
}

li {
```

```
    float: left;
}

li a {
    display: block;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

li a:hover {
    background-color: #111;
}
</style>
</head>
<body>

<ul>
  <li><a class="active" href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>

</body>
</html>
```




## Working with 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.

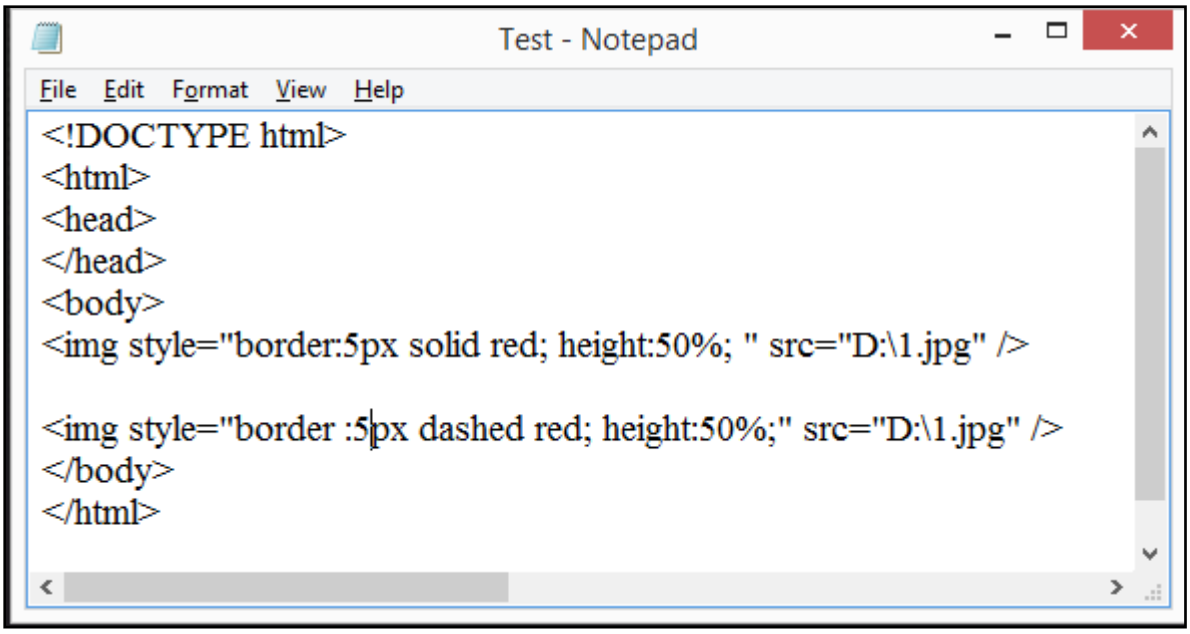
The **border** property is used to set the width of an image border.

- 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 an example:

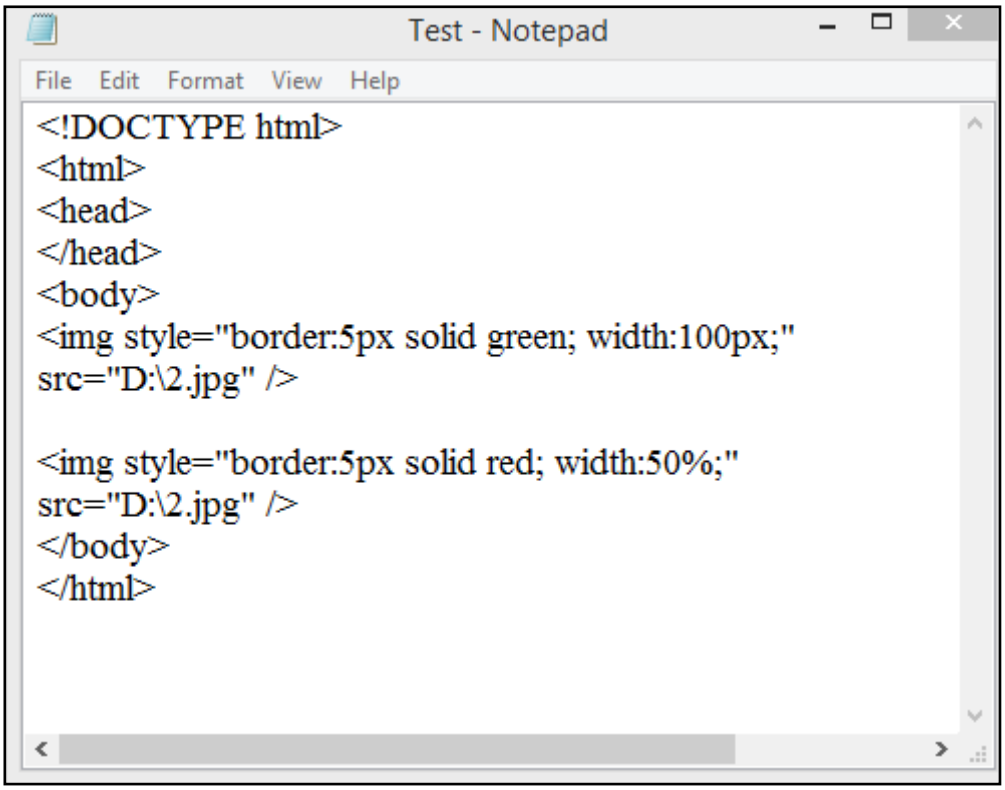



The above program will produce the following result:

## The Image Width Property

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

Here is an example:




The above program will produce the following result:

## Working with 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.

### CSS [attribute] Selector

The [attribute] selector is used to select elements with a specified attribute. The following example selects all <a> elements with a target attribute:
Example

```
a[target] {
   background-color: yellow;
}
```

## CSS [attribute="value"] Selector

The [attribute="value"] selector is used to select elements with a specified attribute and value.
The following example selects all <a> elements with a target="_blank" attribute:
Example

```
a[target="_blank"] {
  background-color: yellow;
}
```

## CSS [attribute~="value"] Selector

The [attribute~="value"] selector is used to select elements with an attribute value containing a specified word. The following example selects all elements with a title attribute that contains a space-separated list of words, one of which is "flower":
Example

```
[title~="flower"] {
  border: 5px solid yellow;
}
```

The example above will match elements with title="flower", title="summer flower", and title="flower new", but not title="my-flower" or title="flowers".

## 4.13.5 CSS [attribute|="value"] Selector

The [attribute|="value"] selector is used to select elements with the specified attribute starting with the specified value. The following example selects all elements with a class attribute value that begins with "top": **Note:** The value has to be a whole word, either alone, like class="top", or followed by a hyphen( - ), like class="top-text"!
Example

```
[class|="top"] {
  background: yellow;
}
```

## CSS [attribute^="value"] Selector

The [attribute^="value"] selector is used to select elements whose attribute value begins with a specified value.
The following example selects all elements with a class attribute value that begins with "top":
**Note:** The value does not have to be a whole word!
Example

```
[class^="top"] {
  background: yellow;
}
```

## CSS [attribute$="value"] Selector

The [attribute$="value"] selector is used to select elements whose attribute value ends with a specified value.
The following example selects all elements with a class attribute value that ends with "test":
**Note:** The value does not have to be a whole word!

Example

```
[class$="test"] {
   background: yellow;
}
```

## CSS [attribute*="value"] Selector

The [attribute*="value"] selector is used to select elements whose attribute value contains a specified value. The following example selects all elements with a class attribute value that contains "te":
**Note:** The value does not have to be a whole word!
Example:

```
[class*="te"] {
  background: yellow;
}
```

---

**Check your progress 2**

---

**Q1. Write the different properties of image ?**

**Answer:**_____

_____

**Q2. What is Pseudo Class ?**
**Answer:**_____

_____

**Q3. What is navigation bar?**

**Answer:**_____

_____

## Let us sum up

In this unit we have understood padding using CSS,setting display

using CSS,setting width and max width using CSS,setting float property

using CSS,inline block propertry,horizontal allignment in CSS,working

with combinatory,pseudo-class,pseudo-elements,creating a navigation bar,

working with images,attribute selectors.

**Check Your Progress Possible answer**

**Check your progress 1**

**Q1. What is Float ?**

**Answer**

A float is a box that is shifted to the left or right on the current line. The most interesting characteristic of a float (or "floated" or "floating" box) is that content may flow along its side (or be prohibited from doing so by the property).

**Q2. Write the different properties of position?**

**Answer**

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.

**Q3. What is padding?**

**Answer**

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

**Q4. Write the use of Display properties?**

**Answer**

The display property is the most important CSS property for controlling layout.
The display property specifies if/how an element is displayed

**Check your progress 2**

**Q1. Write the different properties of image ?**

**Answer** 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.

**Q2. What is Pseudo Class ?**
**Answer**

A pseudo-class is used to define a special state of an element

**Q3. What is navigation bar?**
**Answer**

A **navigation bar** (or **navigation** system) is a section of a graphical user interface intended to aid visitors in accessing information. **Navigation bars** are implemented in file browsers, web browsers and as a design element of some web sites