PHOENIX

COMPUTER EDUCATION & TRAINING

Certificate Program in Web Designing

PHOENIX COMPUTER EDUCATION & TRAINING

CONTENTS

HTML

Introduction of HTML

HTML stands for **Hyper Text Markup Language**. HTML is the computer language which is used to create and design **Web Pages**. HTML was created by Berners Lee in 1991. But in 1995 HTML 2.0 was the first standard HTML specification. HTML 4.01 was published in 1999. Currently we are using HTML 5 which was published in 2012.

Why HTML is called Markup Language?

HTML is called Markup Language because it allows us to improve the appearance of any webpage, and Hyper Text means machine readable text.

Features of HTML

1. HTML is easy to use and very simple language.
2. HTML is not case sensitive language.
3. With the help of **HTML tags** it is easy to design web pages. There are a lot of tags in HTML.
4. We can add links, images, audios and videos in the web pages to make web pages more attractive.
5. HTML is **platform independent** because it can be displayed on any platform like Linux, Windows, Mac etc.

HTML Editors

Web pages can be created and modified through **Notepad** and **Dreamweaver Application**. Follow some steps to create a web page by using Notepad and Dreamweaver.

For Notepad:

1. Open Notepad, then write HTML tags. Save the notepad with **.html** extension (eg. Example.html).

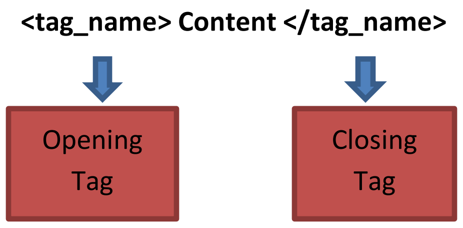
2. Open the saved document with any browser to check the design.

For Dreamweaver:

1. Open Dreamweaver, then choose HTML. An editor will open to write HTML tags. Save it.
2. Open the saved document with any browser to check the design.

HTML Tags

HTML uses various tags to format the contents. Tags contain three parts. Opening tag, text and closing tag. But some tags do not have any closing tag. Tags are enclosed within angle braces.



<html> … </html>

This <html> tag encloses the complete HTML document.

<head> … </head>

This tag performs as the document's header.

<title> … </title>

The tag is used inside the <head> tag to mention the document title.

<body> … </body>

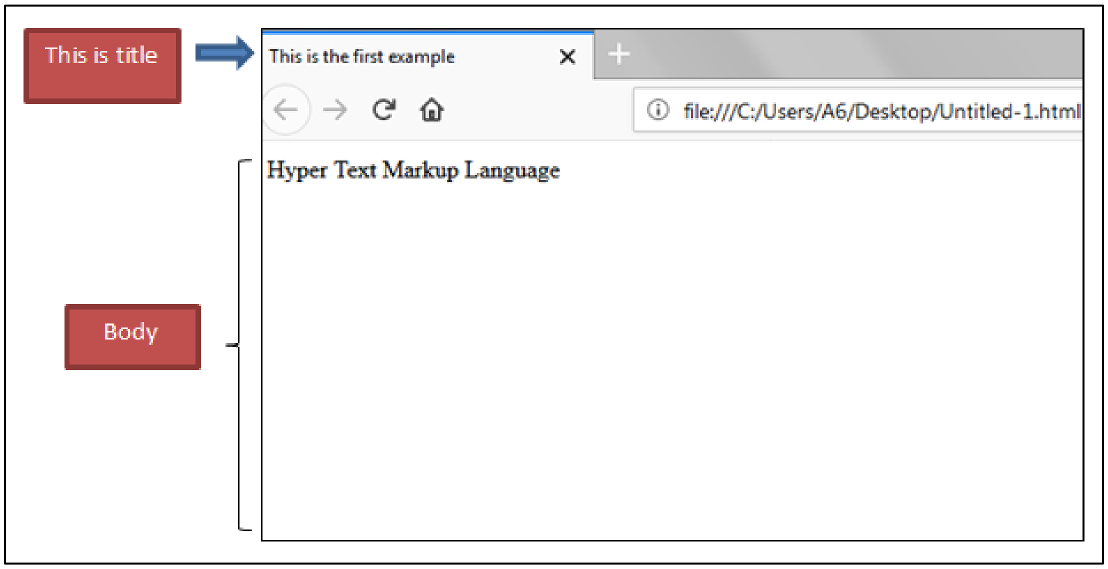
This tag performs as the document's body which keeps other HTML tags inside.

Example of HTML Tags

```
<html>
```

```
 <head>
   <title> This is the first example </title>
 </head>
 <body>
   <p> Hyper Text Markup Language </p>
 </body>
</html>
```

Output


This is title
This is the first example
file:///C:/Users/A6/Desktop/Untitled-1.html
Body

Hyper Text Markup Language

Heading Tags

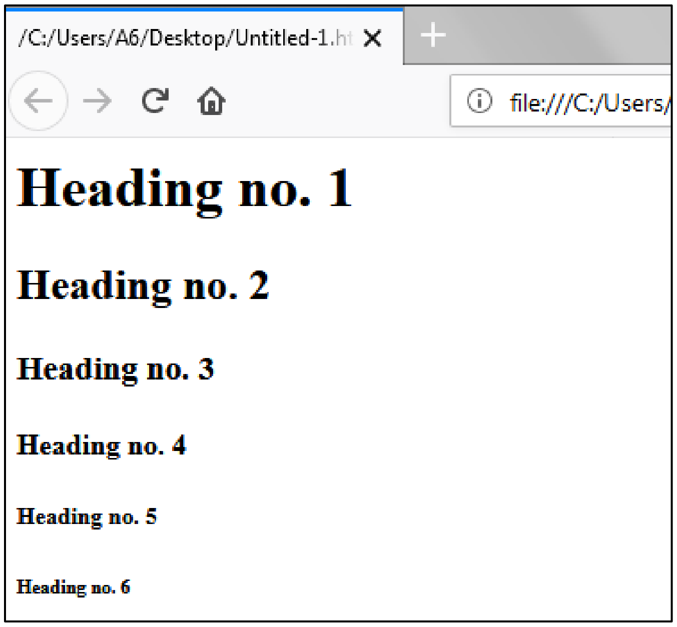
There are six different heading tags defined, <h1> to <h6> tags. Heading tags display in bold format and text size depends on the tag. <h1> is the **largest** heading tag and <h6> is the **smallest** heading tag.

Example of Heading tags

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

```
  <h1>Heading no. 1</h1>

  <h2>Heading no. 2</h2>

  <h3>Heading no. 3</h3>

  <h4>Heading no. 4</h4>

  <h5>Heading no. 5</h5>

  <h6>Heading no. 6</h6>

 </body>

</html>
```

Output



/C:/Users/A6/Desktop/Untitled-1.ht

file:///C:/Users/

Heading no. 1

Heading no. 2

Heading no. 3

Heading no. 4

Heading no. 5

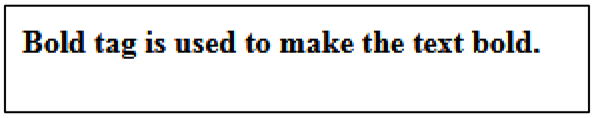
Heading no. 6

Formatting Tags

HTML Formatting tags are used to format text for better look. There are many formatting tags.

1. Bold

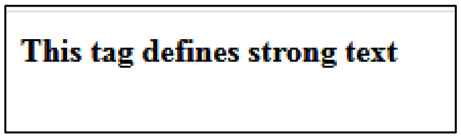
```
<b> This tag is used to make the text bold. </b>
```



Bold tag is used to make the text bold.

2. Strong

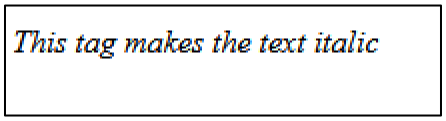
```
<strong> This tag defines strong text </strong>
```



This tag defines strong text

3. Italic

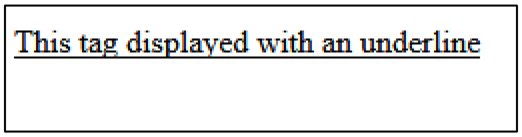
```
<i> This tag makes the text italic </i>
```



This tag makes the text italic

4. Underline

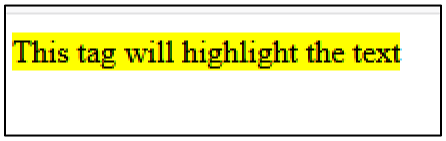
```
<u> This tag displayed with an underline </u>
```



This tag displayed with an underline

5. Mark

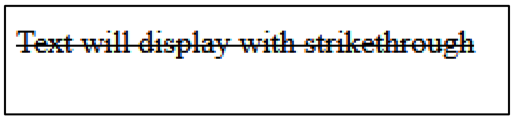
```
<mark> This tag will highlight the text </mark>
```



This tag will highlight the text

6. Strike

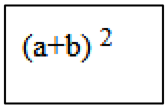
```
<strike> Text will display with strikethrough </strike>
```



~~Text will display with strikethrough~~

7. Superscript

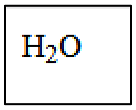
```
(a+b) <sup> 2 </sup>
```



$(a+b)^2$

8. Subscript

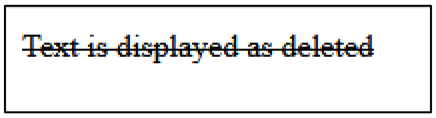
```
H <sub> 2 </sub> O
```



H_2O

9. Delete

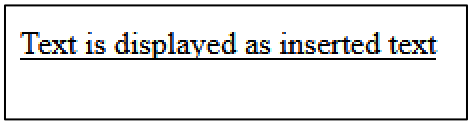
```
<del> Text is displayed as deleted <del>
```



~~Text is displayed as deleted~~

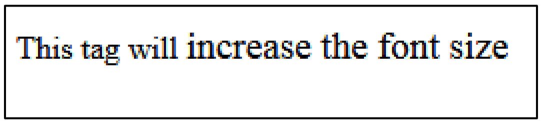
10. Insert

```
<ins> Text is displayed as inserted text <ins>
```


Text is displayed as inserted text

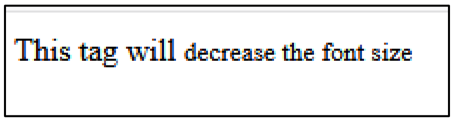
11. Big

```
This tag will <big> increase the font size </big>
```


This tag will increase the font size

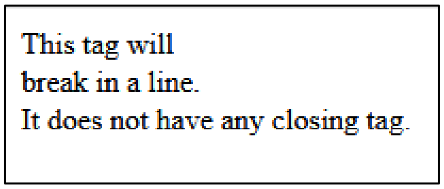
12. Small

```
This tag will <small> decrease the font size </small>
```


This tag will decrease the font size

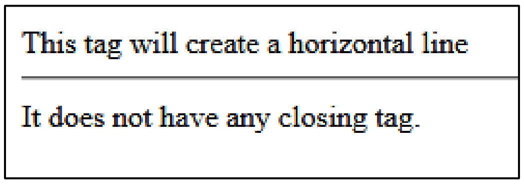
13. Break

```
  This tag will <br> break in a line. <br> It does not
have any closing tag.
```


This tag will
break in a line.
It does not have any closing tag.

14. Horizontal Line

```
This tag will create a horizontal line
  <hr>
  It does not have any closing tag.
```

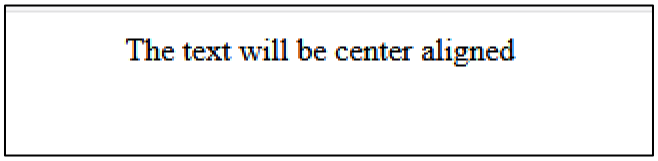


This tag will create a horizontal line

It does not have any closing tag.

13. Center

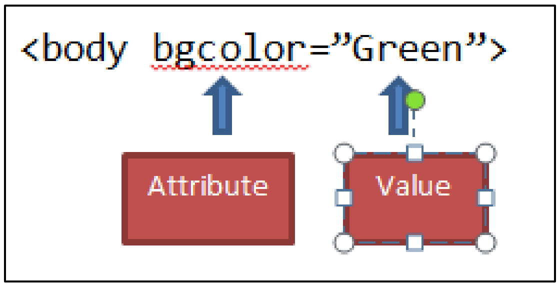
```
<center> The text will be center aligned </center>
```



The text will be center aligned

HTML Attributes

1. All HTML tags can have attributes.
2. Attributes gives additional information to HTML elements.
3. Attributes are specified in starting tag.
4. Attribute names always comes with value. Eg name=”value”.
5. Values are always in double quotation.

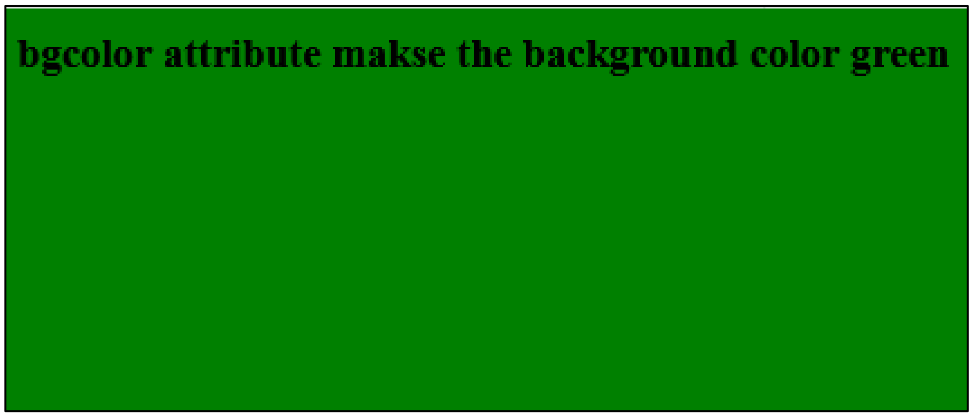


Example of HTML Attributes - 1

```
<html>

 <body bgcolor="green">
```

```
  <h3> bgcolor attribute makes the background color green

   </h3>

 </body>

</html>
```

Output

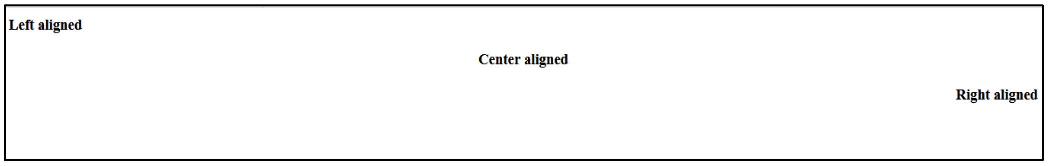


bgcolor attribute makse the background color green

Example HTML Attributes - 2

```
<html>

 <body>

  <h3 align = "left">Left aligned</h3>

  <h3 align = "center">Center aligned</h3>

   <h3 align = "right">Right aligned</h3>

 </body>

</html>
```

Output


Left aligned
Center aligned
Right aligned

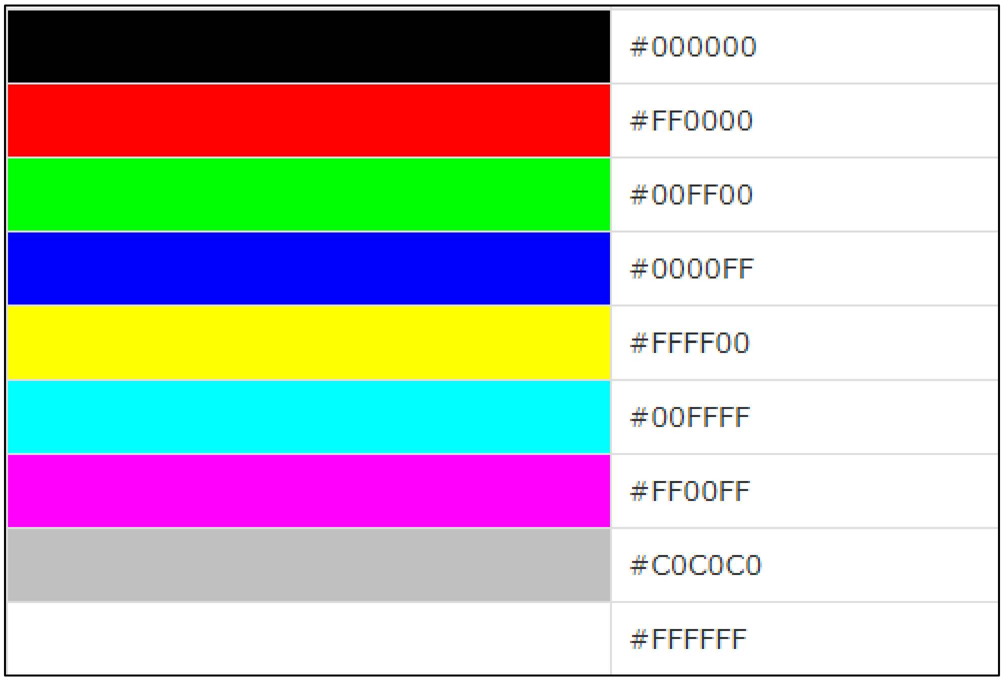
HTML Colors

Colors are very important to make any website look good. We can write color names directly as we did in bgcolor attribute and we can also use hex codes.

Hex code or hexadecimal value is a six digit number which represents red, green and blue (RGB) that makes a color. Each hex code preceded by a hash sign. Eg #FFFF00.

Here first two digits represent red color, second two digits represent green color and third two digits represent blue color.

Following is a list of some few colors using hex code –

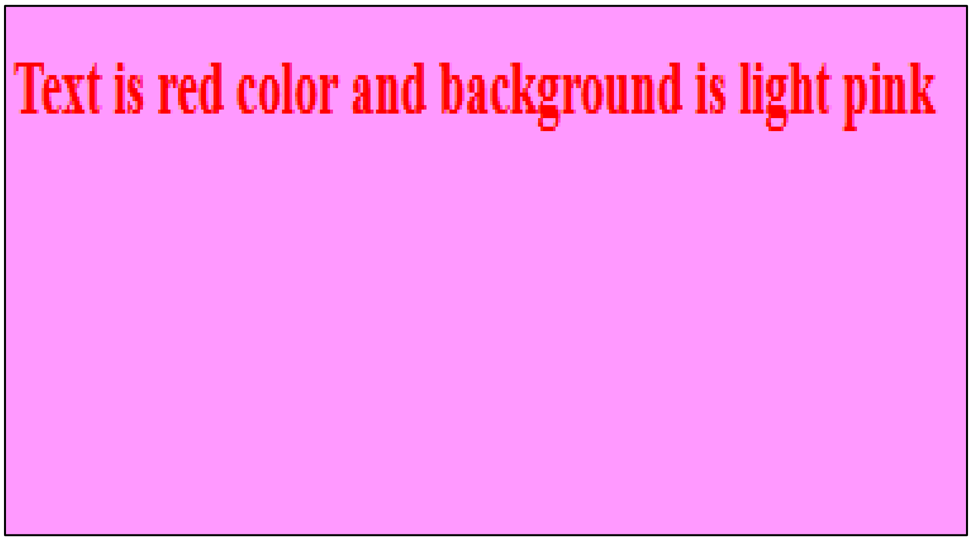


⬛	#000000
🟥	#FF0000
🟩	#00FF00
🟦	#0000FF
🟨	#FFFF00
🟦	#00FFFF
🟪	#FF00FF
▨	#C0C0C0
⬜	#FFFFFF

Example of HTML Colors

```
<html>

 <body bgcolor="#FF99FF" text="#FF0000">

  <h3> Text is red color and background is light pink </h3>

 </body>

</html>
```

Output



Text is red color and background is light pink

Font Tag

Using <font> tag we can change text color, size and style. Font tag has three attributes.

Size – This attribute will change font size.

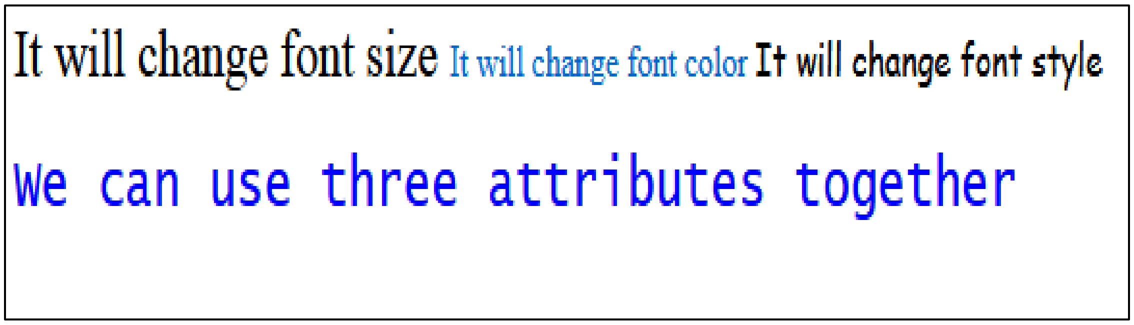
Color – This attribute will change font color.

Face – This attribute will change font style.

Example of Font Tag

```
<html>

 <body>

  <font size="5"> It will change font size </font>

  <font color="#0066CC"> It will change font color </font>

  <font face="Comic Sans MS, cursive"> It will change font style
</font>

   <br>

   <br>

  <font size="5" color="blue" face="Lucida Console, Monaco,
monospace"> We can use three attributes together </font>

 </body>

</html>
```

Output



It will change font size It will change font color It will change font style

We can use three attributes together

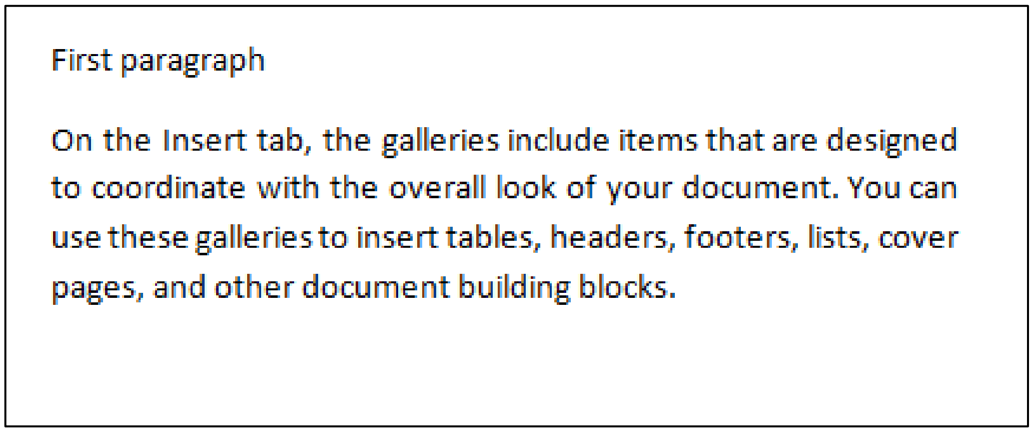
HTML Paragraph

HTML Paragraph is used to describe paragraph in a web page. <p> … </p> tag is used to define paragraph.

Example of HTML Paragraph

```
<html>

 <body>

  <p>First paragraph</p>

  <p align="justify">On the Insert tab, the galleries include items
that are designed to coordinate with the overall look of your
document. You can use these galleries to insert tables, headers,
footers, lists, cover pages, and other document building blocks.<p>

 </body>

</html>
```

Output



First paragraph

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks.

Marquee Tag

Marquee tag used to move any image or text horizontally or vertically, means up to down or down to up and left to right or right to left.

Marquee attributes

Behavior – It defines different types of scrolling. There are three values of this attribute. **Scroll, alternate and slide**.

Bgcolor – It defines background color.

Direction – It defines which direction the text or image will move. There are four values of this attribute. **Left, right, up and down**.

Height – It defines height of marquee. The value can be in pixels or in %.

Width – It defines width of marquee. The value can be in pixels or in %.

Scrollamount – It defines marquee speed. It can be define in number.

Scrolldelay – It defines scroll delay. It can be define in seconds.

Loop – It defines how many times a text or an image will move. It can be define in number.

OnMouseOver – Scrolling will stop when we put the mouse pointer on the marquee. Value will be **this.stop().**

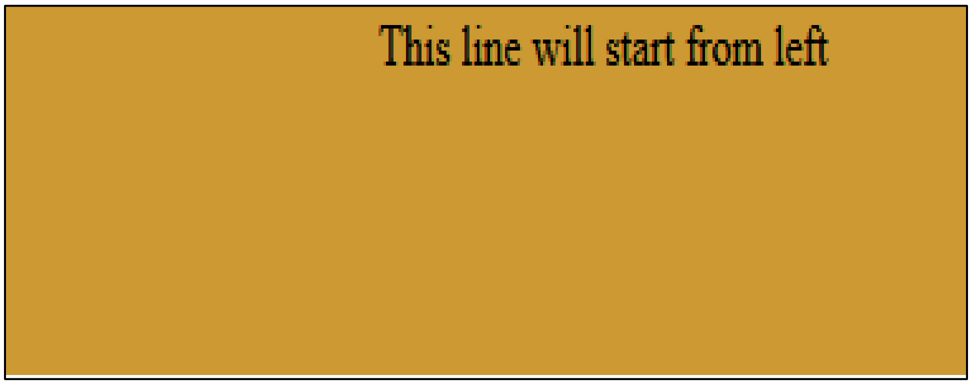
OnMouseOut – Scrolling will start again when we move away the pointer from marquee. Value will be **this.start().**

Example of Marquee Tag

```
<html>

 <body>
```

```
  <marquee behavior="scroll" bgcolor="#CC9933" direction="right"
height="100" width="1100" scrollamount="3" scrolldelay="4" loop="3"
onMouseOver="this.stop()" onMouseOut="this.start()">
This line will start from left </marquee>

 </body>

</html>
```

Output



HTML Lists

HTML Lists make lists of information. There are three types of lists.

1. Ordered List
2. Unordered List
3. Definition List

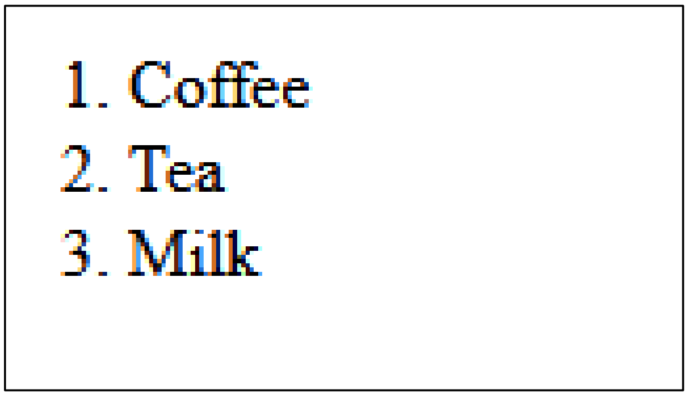
Ordered List – We can put the items in numbered list. We use **<ol>** … **</ol>** tag and we put the items in **<li>** … **</li>** tag.

Example of Ordered List

```
<html>

 <body>
```

```
  <ol>
   <li>Coffee</li>
   <li>Tea</li>
   <li>Milk</li>
  </ol>
 </body>
</html>
```

Output



```
1. Coffee
2. Tea
3. Milk
```

The Type Attribute (Ordered List)

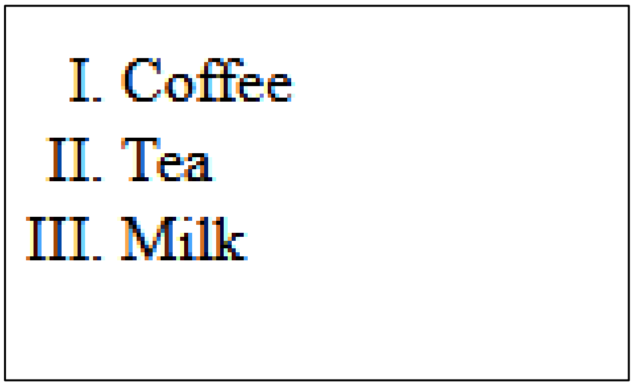
We can use **type** attribute with <ol> tag. Value can be five types. **1, A, a, i and I**.

Example of Type Attribute

```
<html>
 <body>
  <ol type="I">
```

```
    <li>Coffee</li>

    <li>Tea</li>

    <li>Milk</li>

  </ol>

 </body>

</html>
```

Output



```
  I. Coffee
 II. Tea
III. Milk
```

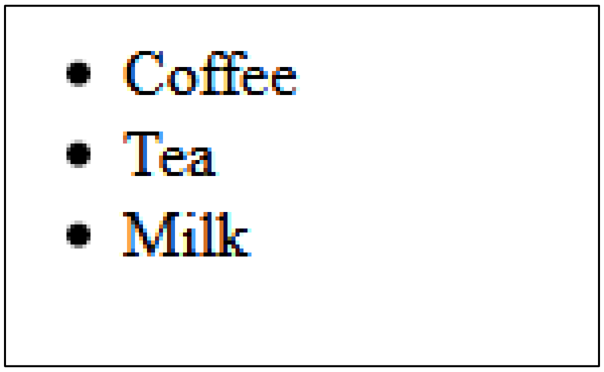
Unordered List – Here we can mark the items with bullets. We use <ul> … </ul> tag and we put the items in <li> … </li> tag.

Example of Unordered List

```
<html>

 <body>

  <ul>

    <li>Coffee</li>

    <li>Tea</li>
```

```
   <li>Milk</li>

  </ul>

 </body>

</html>
```

Output



- Coffee
- Tea
- Milk

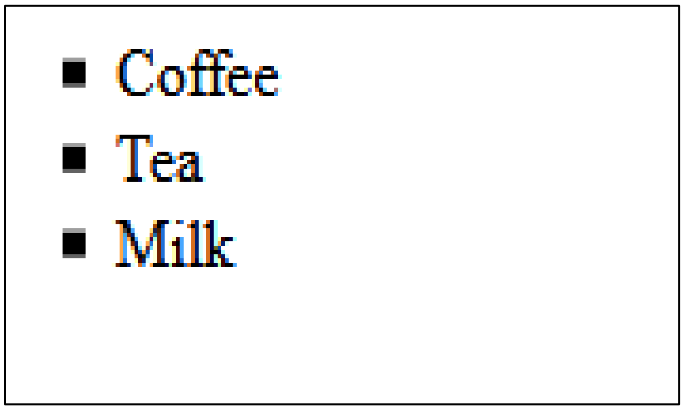
The Type Attribute (Unordered List)

We can also use **type** attribute with <ul> tag. Value can be three types. **Square, disc and circle.**

Example of Type Attribute

```
<html>

 <body>

  <ul type="square">

   <li>Coffee</li>

   <li>Tea</li>

   <li>Milk</li>
  </ul>

 </body>

</html>
```

Output



```
■ Coffee
■ Tea
■ Milk
```

Definition List - In this list texts are listed like a dictionary. Definition list contains three tags.

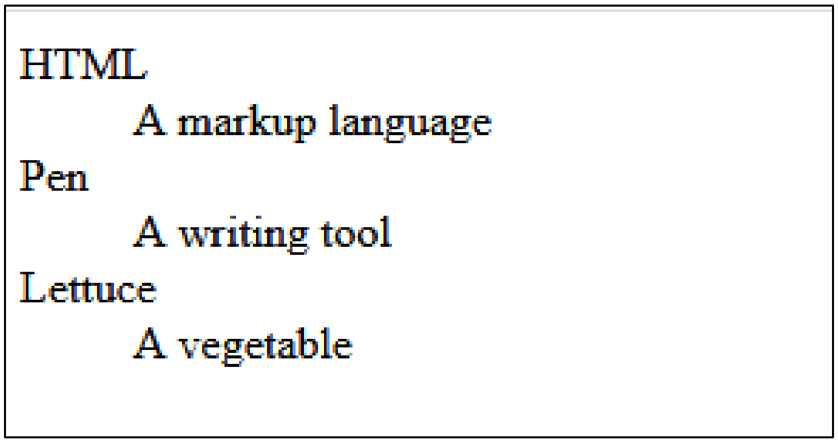
1. **<dl>** - Definition list
2. **<dt>** - Definition term
3. **<dd>** - Definition data

Example of Definition List

```
<html>

 <body>

  <dl>

   <dt>HTML</dt>

    <dd>A markup language</dd>

   <dt>Pen</dt>
```

```
    <dd>A writing tool</dd>

   <dt>Lettuce</dt>

    <dd>A vegetable</dd>

  </dl>

 </body>

</html>
```

Output



```
HTML
        A markup language
Pen
        A writing tool
Lettuce
        A vegetable
```

Nested List

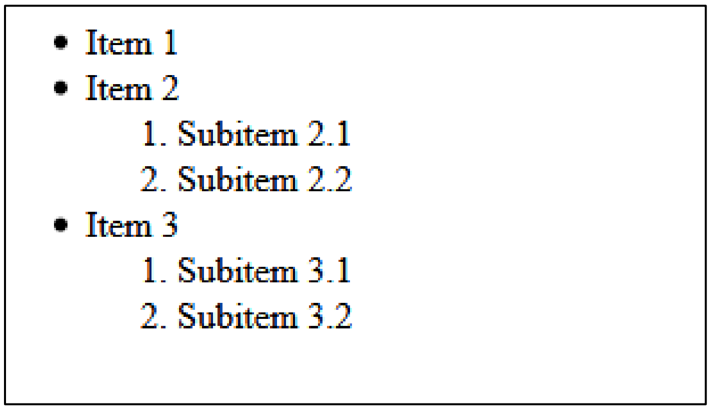
Nested list means a list within a list.

Example of Nested List

```
<html>

 <body>

  <ul>

   <li>Item 1</li>
```

```
    <li>Item 2</li>
     <ol>
      <li>Subitem 2.1</li>
      <li>Subitem 2.2</li>
     </ol>
    <li>Item 3</li>
     <ol>
      <li>Subitem 3.1</li>
      <li>Subitem 3.2</li>
     </ol>
    </ul>
  </body>
</html>
```

Output



```
• Item 1
• Item 2
    1. Subitem 2.1
    2. Subitem 2.2
• Item 3
    1. Subitem 3.1
    2. Subitem 3.2
```

HTML Table

HTML Table allows us to arrange data in a table. We can use 4 tags to create a table.

<table> - This tag defines a table.

<tr> - This tag creates a row in a table.

<th> - This tag defines heading cell in a table.

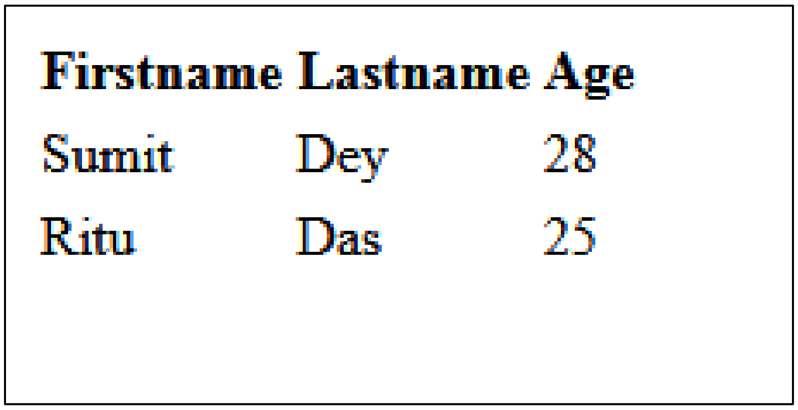
<td> - This tag defines a cell in a table.

Example of HTML Table

```
<html>
 <body>
  <table>
   <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
   </tr>
   <tr>
    <td>Sumit</td>
    <td>Dey</td>
    <td>28</td>
  </tr>
  <tr>
```

```
    <td>Ritu</td>

    <td>Das</td>

    <td>25</td>

   </tr>

  </table>

 </body>

</html>
```

Output



```
Firstname Lastname Age
Sumit     Dey      28
Ritu      Das      25
```

Table Attributes

Following attributes we can use to create a table.

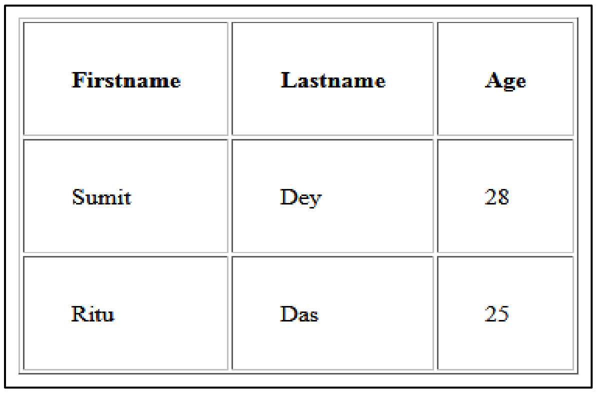
Cellpadding, Cellspacing and Border Attribute

Cell padding attribute provides the distance between cell borders and data within the table. We can use Cell spacing attribute to adjust space in table cell. Border attribute creates border.

Example of Cellpadding, Cellspacing and Border Attribute

```
<html>
 <body>
  <table cellpadding="30" cellspacing="2" border="1">
   <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
   </tr>
   <tr>
    <td>Sumit</td>
    <td>Dey</td>
    <td>28</td>
   </tr>
   <tr>
    <td>Ritu</td>
    <td>Das</td>
    <td>25</td>
   </tr>
  </table>
 </body>
</html>
```

Output



Firstname	Lastname	Age
Sumit	Dey	28
Ritu	Das	25

Bgcolor, Background and Bordercolor Attribute

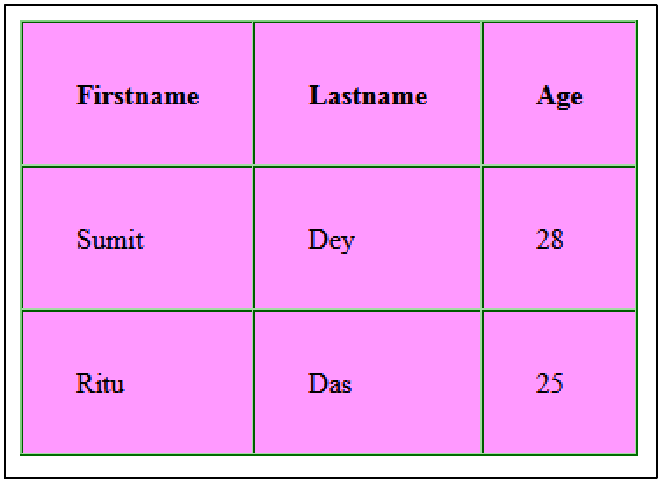
Using bgcolor attribute we can color the whole table background or just for one cell. Background attribute will set a background image for whole table or just for one cell. Bordercolor attribute will color the table borders.

Example of Bgcolor, Background and Bordercolor Attribute -1

```
<html>

 <body>

  <table cellpadding="30" cellspacing="0" border="1" bgcolor="#FF99FF" 
bordercolor="#009900">

   <tr>

    <th>Firstname</th>

    <th>Lastname</th>

    <th>Age</th>

   </tr>

   <tr>
```

```
    <td>Sumit</td>

    <td>Dey</td>

    <td>28</td>

   </tr>

   <tr>

    <td>Ritu</td>

    <td>Das</td>

    <td>25</td>

   </tr>

  </table>

 </body>

</html>
```

Output



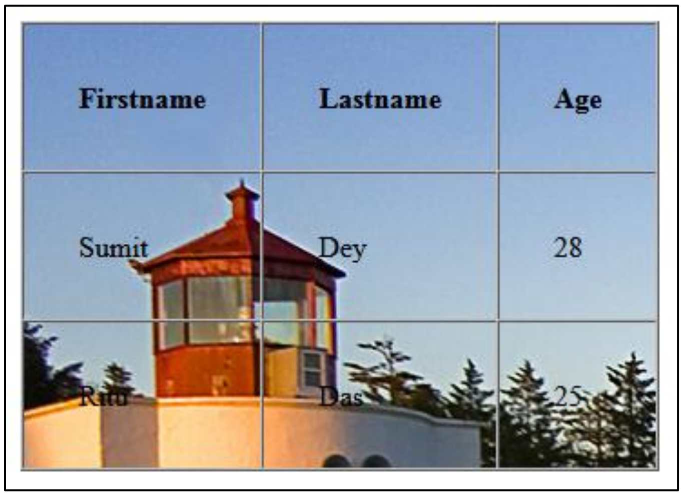
Firstname	Lastname	Age
Sumit	Dey	28
Ritu	Das	25

Example of Bgcolor, Background and Bordercolor Attribute - 2

```
<html>

 <body>

  <table cellpadding="30" cellspacing="0" border="1"
background="Lighthouse.jpg">

   <tr>

    <th>Firstname</th>

    <th>Lastname</th>

    <th>Age</th>

   </tr>

   <tr>

    <td>Sumit</td>

    <td>Dey</td>

    <td>28</td>

   </tr>

   <tr>

    <td>Ritu</td>

    <td>Das</td>

    <td>25</td>

    </tr>

  </table>

 </body>
```

```
</html>
```

Output



Firstname	Lastname	Age
Sumit	Dey	28
Ritu	Das	25

Rowspan and Colspan Attributes

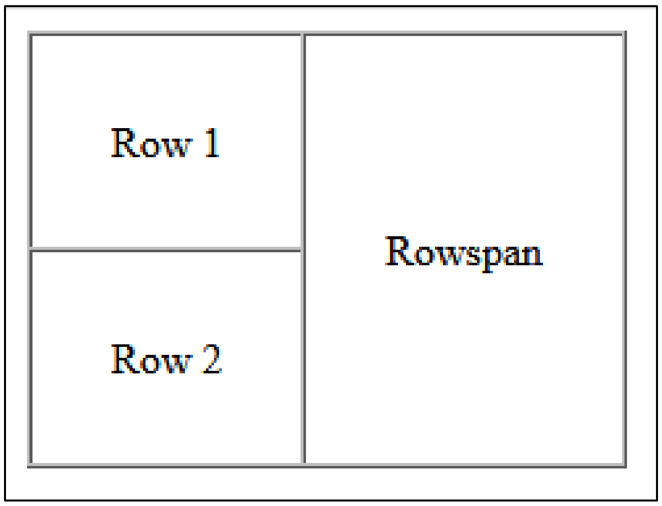
We use rowspan attribute when we want to merge two or more rows into one row and similarly we use colspan attribute when we want to merge two or more columns into one column.

Example of Rowspan and Colspan Attributes -1

```
<html>

 <body>

  <table cellpadding="30" cellspacing="0" border="1">

   <tr>

    <td>Row 1</td>

    <td rowspan="2">Rowspan</td>

   </tr>
```

```
    <tr>
     <td>Row 2</td>
    </tr>
   </table>
  </body>
</html>
```

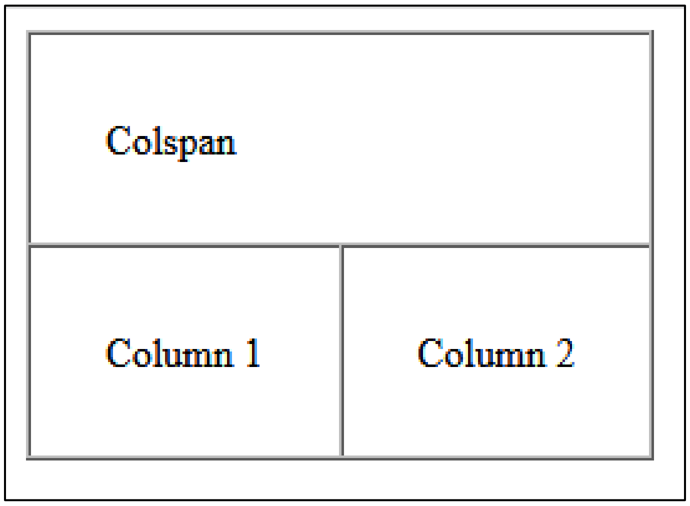
Output



Example of Rowspan and Colspan Attributes - 2

```
<html>
 <body>
  <table cellpadding="30" cellspacing="0" border="1">
   <tr>
    <td colspan="2">Colspan</td>
   </tr>

   <tr>

    <td>Column 1</td>

     <td>Column 2</td>

   </tr>

  </table>

 </body>

</html>
```

Output



Colspan	
Column 1	Column 2

HTML Links

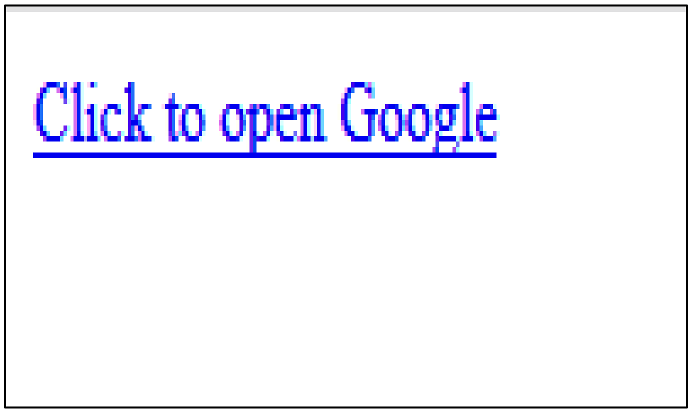
A webpage can have various links that can take us to one page to another page by clicking on it. Those links are called **Hyperlink.** We can create hyperlinks using words or images.

We use <a> … </a> tag to create hyperlink. This tag is called **Anchor** tag. The **href** attribute is the most important attribute of anchor tag. Href attribute defines the address to be linked.

Example of HTML Links

```
<html>

 <body>

  <a href="http://www.google.co.in/">Click to open Google</a>

 </body>

</html>
```

Output



Target Attribute

Target attribute used to provide the location where the linked document will open. This attribute has following values.

_blank – It will open the linked document in the new window or tab.

_parent – It will open the linked document in the parent frame.

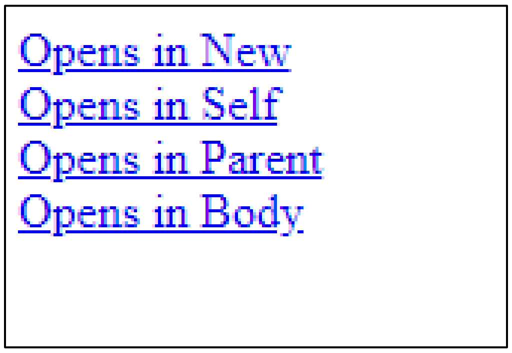
_self – It will open the linked document in the same window or tab. This is default.

_top – It will open the linked document in the full body of the window.

Example of Target Attribute

```
<html>

 <body>
  <a href = "http://www.google.co.in/" target = "_blank">Opens in
New</a><br>

  <a href = "http://www.google.co.in/" target = "_self">Opens in
Self</a><br>

  <a href = "http://www.google.co.in/" target ="_parent">Opens in
Parent</a><br>

  <a href = "http://www.google.co.in/" target = "_top">Opens in
Body</a><br>

 </body>

</html>
```

Output


Opens in New
Opens in Self
Opens in Parent
Opens in Body

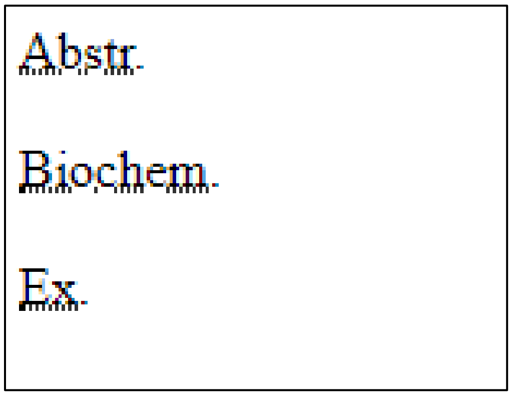
HTML Abbreviation & Acronym

HTML abbreviation and an acronym are both shortened version of any word. **<abbr>** tag is used for abbreviation and **<acronym>** tag is used for acronym. Both tag uses **title** attribute. Move mouse over an abbreviation or acronym to check the full forms.

Example of Abbreviation

```
<html>

 <body>

  <abbr title="Abstract">Abstr.</abbr>

   <br /><br />

  <abbr title="Biochemistry">Biochem.</abbr>

   <br /><br />

  <abbr title="Example">Ex.</abbr>

 </body>

</html>
```

Output



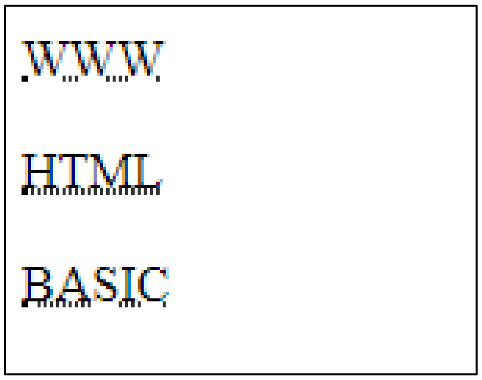
```
Abstr.

Biochem.

Ex.
```

Example of Acronym

```
<html>

 <body>

  <acronym title="World Wide Web">WWW</acronym>
```

```
   <br /><br />

  <acronym title="Hyper Text Markup Language">HTML</acronym>

   <br /><br />

  <acronym title="Beginners All Purpose
 Symbolic Instruction Code">BASIC

  </acronym>

 </body>

</html>
```

Output



```
W.W.W.

HTML

BASIC
```

HTML Image

HTML image is used to add images in webpages. We use **<img>** tag to add images. This tag does not have any closing tag.

Img Tag Attributes

Following are the <img> tag attributes –

Src – This is the necessary attribute of <img> tag. It defines the image path.

Title – This attribute used to provide a title of the image. Move mouse over the image to display the image title.

Alt – This attribute defines an alternative name of the image. If the image does not display, the value of the alt attribute will describe the image.

Height – This attribute defines the height of the image.

Width – This attribute defines the width of the image.

Border – This attribute provides border around the image.

Example of Img Tag Attributes

```
<html>

 <body>

  <img src="Tulips.jpg" title="Tulip" alt="Flower" height="250"
width="300" border="2">

 </body>

</html>
```

Output

HTML Audio

HTML audio is used to add sound or music in the webpage. HTML 5 supports three file format – **mp3, ogg, wmv.** We use **<audio>** and **<source>** tags to add audio in the webpage.

Audio Attributes

Src – This attribute defines the audio path.

Controls – This attribute defines the audio control which displays play, pause and mute button.

Autoplay – This attribute defines that the audio will start automatically.

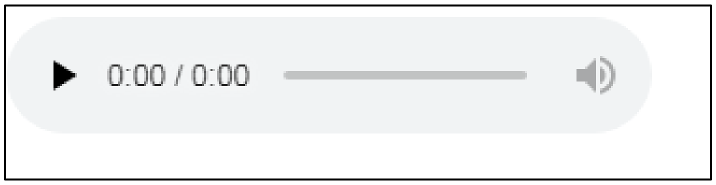
Loop – This attribute defines that the audio will start again whenever it stops.

Type – This attribute defines format of the audio.

Example of HTML Audio

```
<html>

 <body>

  <audio controls loop>

   <source src="music.mp3" type="audio/mp3">

  </audio>

 </body>

</html>
```

Output



HTML Video

HTML video is used to add videos in the webpage. HTML 5 supports three file format – **mp4, ogg, webM.** We use **<video>** and **<source>** tags to add video in the webpage.

Video Attributes

Src – This attribute defines the video path.

Controls – This attribute defines the video control which displays play, pause, full screen and mute button.

Autoplay – This attribute defines that the video will start automatically.

Loop – This attribute defines that the video will start again whenever it stops.

Height – This attribute provides the height of the video player.

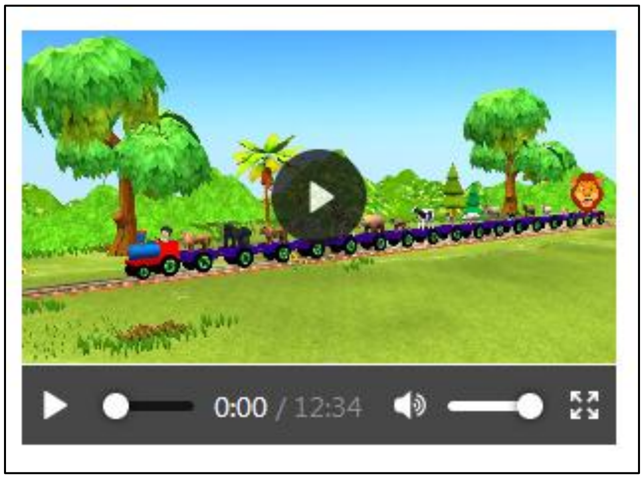
Width – This attribute provides the width of the video player.

Type – This attribute defines format of the video.

Example of HTML Video

```
<html>

 <body>

  <video height="250" width="300" controls loop>

  <source src="Animals.mp4" type="video/mp4">

  </video>

 </body>

</html>
```

Output



Useful HTML Character Entities

HTML character entities represent replacement of reserved characters in HTML. We can also replace characters that are not present on keyboard.

Following are the most used HTML character entities –

Result	Description	Entity Name
	non-breaking space	&nbsp;
<	less than	&lt;
>	greater than	&gt;
&	ampersand	&amp;
"	double quotation mark	&quot;
'	single quotation mark (apostrophe)	&apos;
¢	cent	&cent;
£	pound	&pound;
¥	yen	&yen;
€	Euro	&euro;
©	copyright	&copy;
®	registered trademark	&reg;

HTML Form

HTML form is used to collect some data from any website visitors. For example, during user registration we can collect information like name, address, email address etc. Form will take the information then send it to back end application.
<form> tag is used to create form. We can use **<pre>** tag to create a HTML form properly. <pre> tag defined as **preformatted text**.

Form Attributes

Following are the most frequently used attributes with <form> tag –

Name - This attribute provides the name of form.

Method – This attribute is used to submit form data. There are two types of methods, **Get** and **Post**.

- **Get** – When get is used, the submitted data will be shown in the address field. We should never use sensitive data while using get. It is better for non-secure data.
- **Post** – When post is used, the submitted data will not be shown in the address field. Always use post method for sensitive information.

Action – This attribute defines what action will be taken when the form is submitted. Basically when user submits the form, all the information is sent to a webpage.

Input Tag

The **<input>** tag is the most important tag in HTML form. It can be displayed in several ways. This tag does not have any closing tag. This tag uses many attributes -

Type – This attribute defines the type of the inputs user can write, like text, number or date etc.

Name – This attribute defines the name input field.

Size – This attribute defines the size of input field.

Maxlength – This attribute provides length of the data.

Title – This attribute provides title of the text box. Move mouse over the input field to display the title.

Placeholder – This attribute defines a hint which describes the expected data in the input field.

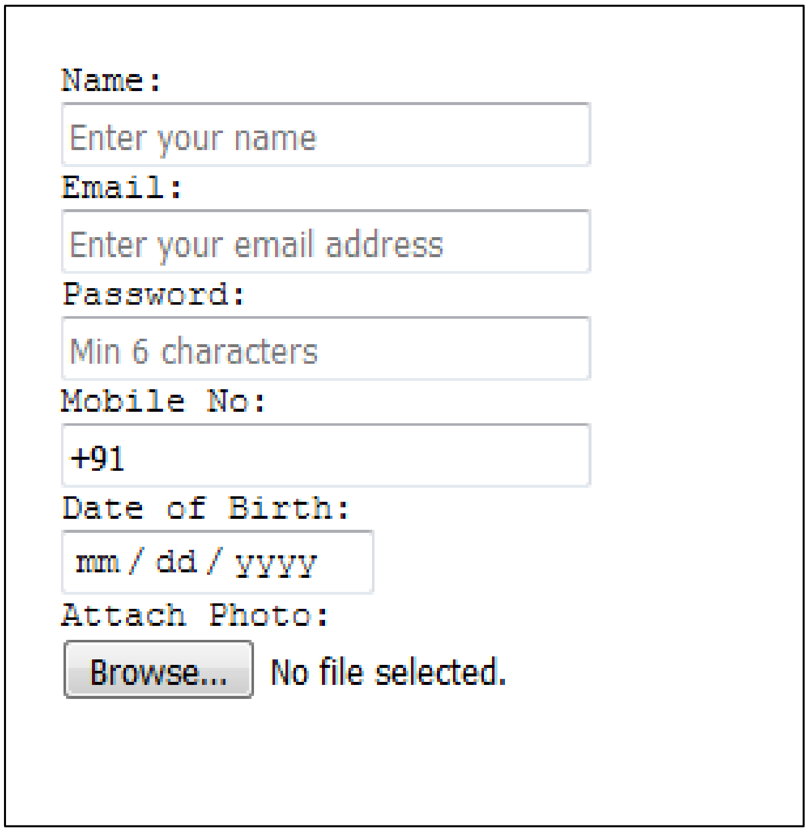
Value – This attribute defines the value of the input field.

Required – This attribute defines an input field must be filled before submitting data.

Example of Input tag

```
<html>

 <body>

  <pre>

   <form name="f1" method="post" action="/action_page.php">

    Name:
    <input type="text" name="n1" size="30" maxlenght="40" title="Enter
text only" placeholder="Enter your name" required>
    Email:
    <input type="email" name="e1" size="30" maxlenght="20"
 title="Enter email" placeholder="Enter your email address" required>

    Password:
    <input type="password" name="p1" size="30" title="Enter password"
placeholder="Min 6 characters" required>

    Mobile No:
    <input type="text" name="m1" size="30" maxlenght="10" title="Enter
phone no" value="+91" required>

Date of Birth:
    <input type="date" placeholder="dd-mm-yyyy">

Attach Photo:
    <input type="file" name="file">

   </form>
  </pre>
 </body>
</html>
```

Output



Name:
Enter your name
Email:
Enter your email address
Password:
Min 6 characters
Mobile No:
+91
Date of Birth:
mm / dd / yyyy
Attach Photo:

No file selected.

Multi Line Text Input

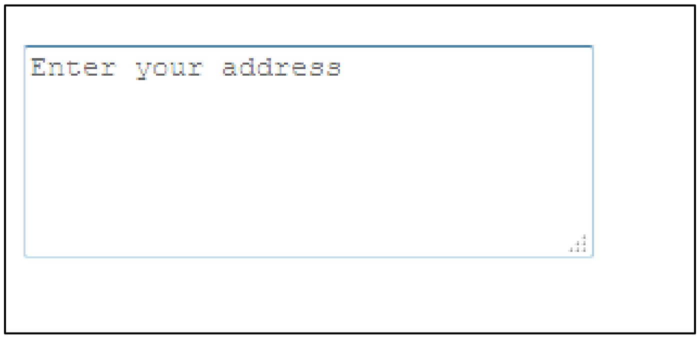
When user data is longer than single line, we use Multi Line Text Input Control. **<textarea>** tag is used to create multi line text input. This tag has two attributes – **rows and cols**.

Example of Multi Line Text Input

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

```
  <form name="f1" method="post" action="/action_page.php">

   <textarea rows="5" cols="30" placeholder="Enter your address">
    </textarea>

   </form>

 </body>

</html>
```

Output



```
Enter your address
```

HTML Buttons

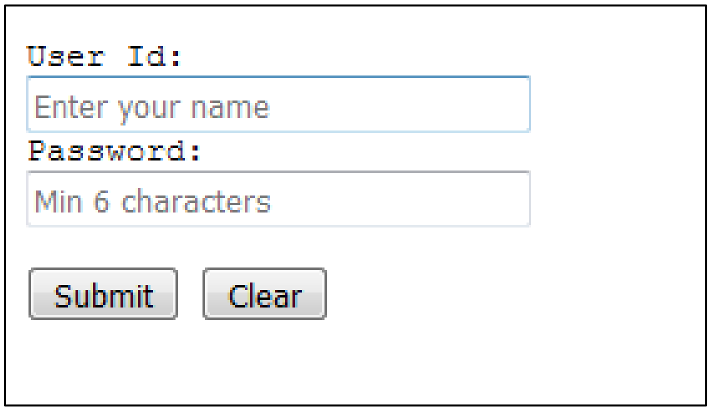
By clicking on button we can submit data to the back end application. We can create two types of button, **submit** (to submit the data) and **reset** (to reset the data).

Example of Buttons

```
<html>

 <body>

  <pre>

   <form name="f1" method="post" action="/action_page.php">

    User Id:
```

```
    <input type="text" name="n1" size="30" maxlenght="40" title="Enter
text only" placeholder="Enter your name" required>

     Password:
    <input type="password" name="p1" size="30" title="Enter password"
placeholder="Min 6 characters" required>

     <input type="submit" value="Submit"> <input type="reset"
value="Clear">

   </form>

  </pre>

 </body>

</html>
```

Output



User Id:

Enter your name

Password:

Min 6 characters

Radio Button

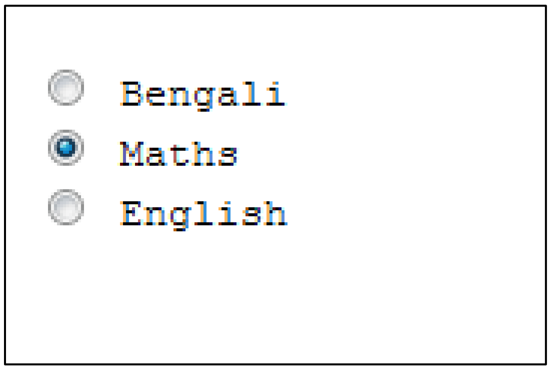
Radio buttons are used to select one option from multiple options. If we use same name for all radio button, we can select only one button at a time.

Example of Radio Button

```
<html>

 <body>
```

```
  <pre>
   <form>
     <input type="radio" name="r1"> Bengali
     <input type="radio" name="r1"> Maths
    <input type="radio" name="r1"> English
   </form>
  </pre>
 </body>
</html>
```

Output



```
○ Bengali
● Maths
○ English
```

Check Box

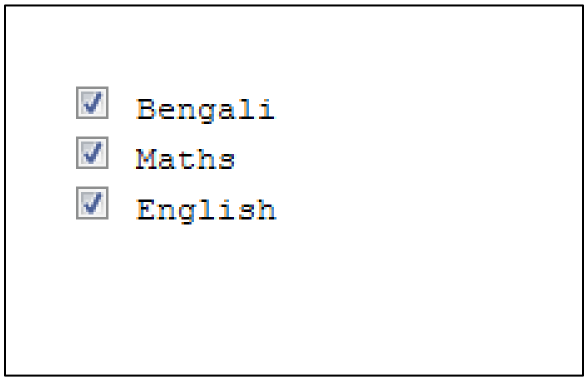
Check boxes are used to select multiple options.

Example of Check Box

```
<html>
 <body>
  <pre>
```

```
    <form>
     <input type="checkbox" name="ch1"> Bengali
     <input type="checkbox" name="ch2"> Maths
     <input type="checkbox" name="ch3"> English
    </form>
  </pre>
 </body>
</html>
```

Output



```
☑ Bengali
☑ Maths
☑ English
```

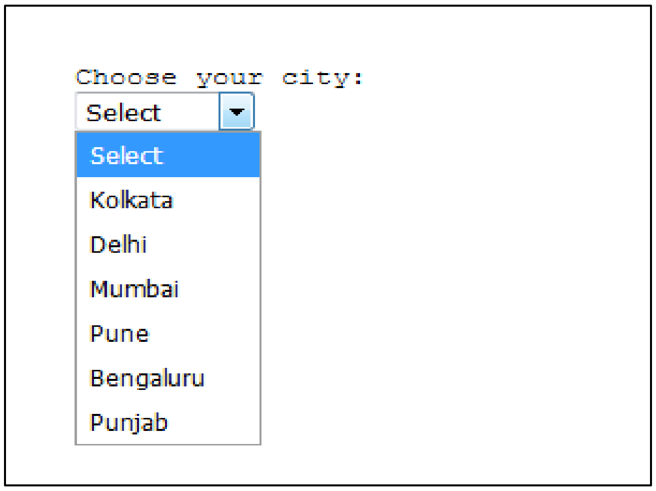
Select Tag

Select tag is used to create drop down list.

Example of Select tag

```
<html>
 <body>
  <pre>
    <form>
    Choose your city:

    <select name="city">

     <option value="0">Select</option>

     <option value="1">Kolkata</option>

     <option value="2">Delhi</option>

     <option value="3">Mumbai</option>

     <option value="4">Pune</option>

     <option value="5">Bengaluru</option>

     <option value="6">Punjab</option>

    </select>

   </form>

  </pre>

 </body>

</html>
```

Output



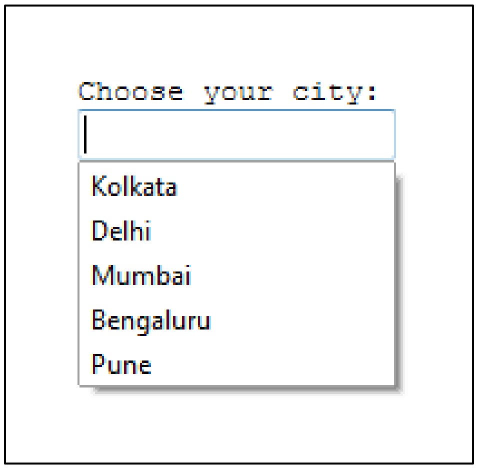
Datalist Tag

Datalist tag is also used to create drop down menu, but we can write to search any data within the list.

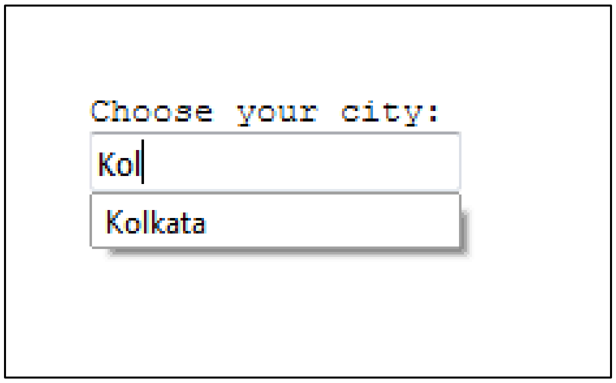
Example of Datalist tag

```
<html>

 <body>

  <pre>

   <form>

    Choose your city:

    <input list="city">

     <datalist id="city">

      <option value="Kolkata">

      <option value="Delhi">

      <option value="Mumbai">

      <option value="Bengaluru">

      <option value="Pune">

    </datalist>

   </form>

  </pre>

 </body>

</html>
```

Output



```
Choose your city:

Kolkata
Delhi
Mumbai
Bengaluru
Pune
```



```
Choose your city:
Kol
Kolkata
```

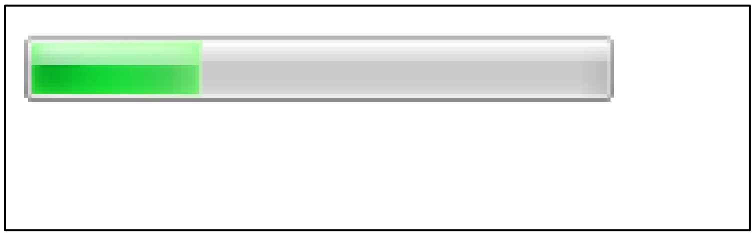
Progress Bar

Progress bar represents progress in any task.

Example of Progress Bar

```
<html>

 <body>

  <progress value="30" max="100"></progress>

 </body>

</html>
```

Output



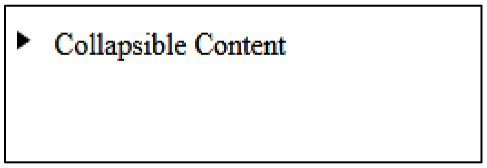
Collapsible Paragraph

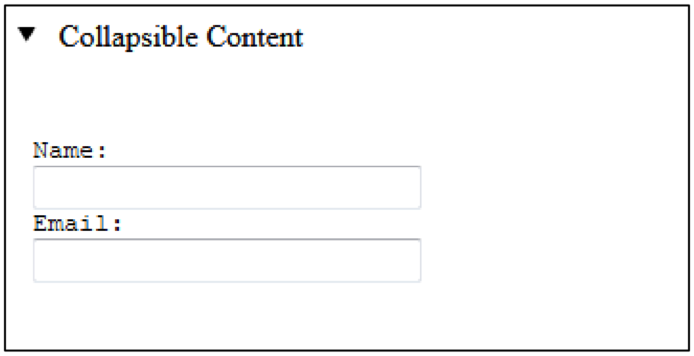
Collapsible paragraph provides us a button which shows and hides the collapsible content.

Example of Collapsible Paragraph

```
<html>

 <body>

  <details>

   <summary> Collapsible Content </summary>

    <pre>

     <form>

      Name:
      <input type="text" size="30">

      Email:
      <input type="email" size="30">

     </form>

    </pre>

   </details>

  </body>

 </html>
```

Output


▸ Collapsible Content



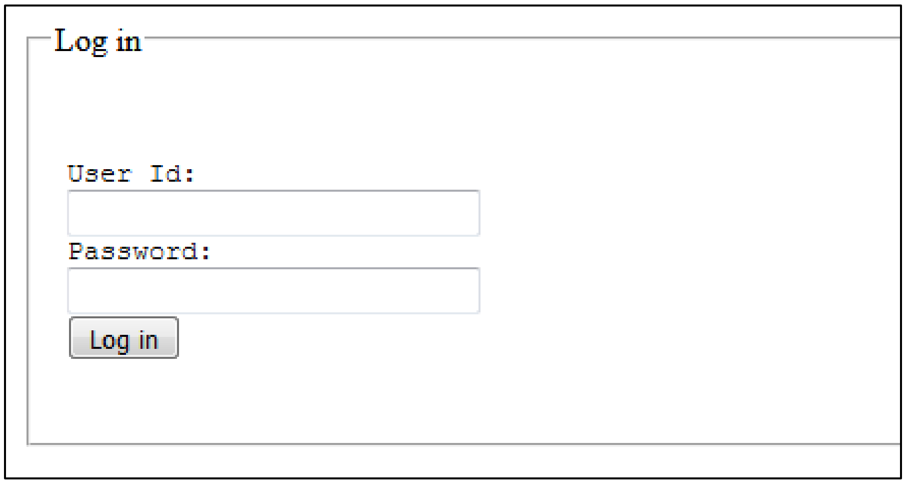
Fieldset Tag

<fieldset> tag provides grouping related forms elements. To create grouping related forms elements we use **<fieldset>** and **<legend>** tags.

Example of Fieldset tag

```
<html>

 <body>

  <fieldset>

   <legend> Log in </legend>

    <pre>

     <form>

      User Id:
      <input type="text" size="30">

      Password:
      <input type="password" size="30">
      <input type="submit" value="Log in">

     </form>

    </pre>

   </body>
  </html>
```

Output



Form Pattern Attribute

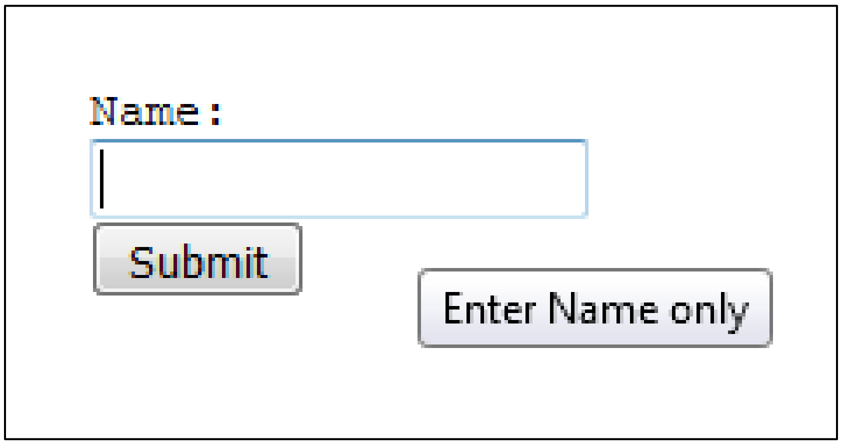
The pattern attribute defines an expression that the <input> element's value is checked against. For example, if users write numbers in the name text field, then they can't submit the form, if we restrict it using pattern attribute.

Example of Form Pattern Attribute - 1

In this example users can only write lowercase and uppercase letters in the name text field.

```
<html>

 <body>

  <pre>

   <form>

    Name:
    <input type="text" name="name" pattern="[a-zA-Z]+" title="Enter 
Name only">

    <input type="submit" value="Submit">
   </form>

  </pre>

 </body>

</html>
```

Output

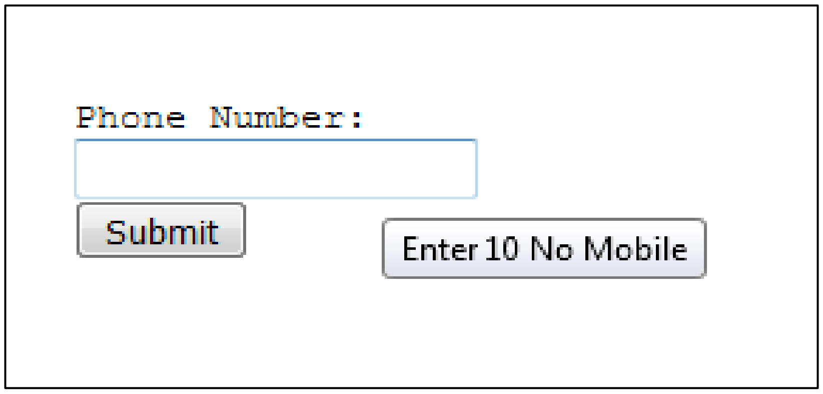


Example of Form Pattern Attribute - 2

In this example users can only write 10 numbers.

```
<html>

  <body>

   <pre>

    <form>

     Phone Number:
     <input type="text" name="mobile" pattern="[0-9]{10}" title="Enter
Mobile No Only">

     <input type="submit" value="Submit">

    </form>
   </pre>

  </body>

</html>
```

Output



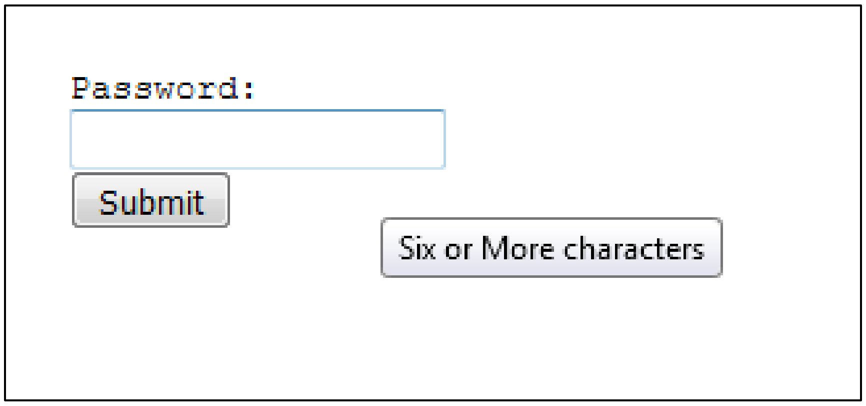
Example of Form Pattern Attribute - 3

In this example, users have to write minimum 6 or more characters in the password.

```
<html>

 <body>

  <pre>

   <form>

    Password:
    <input type="password" name="pass" pattern=".{6,}" title="Six or
more characters" >

    <input type="submit" value="Submit">

   </form>

  </pre>

 </body>
```

```
</html>
```

Output

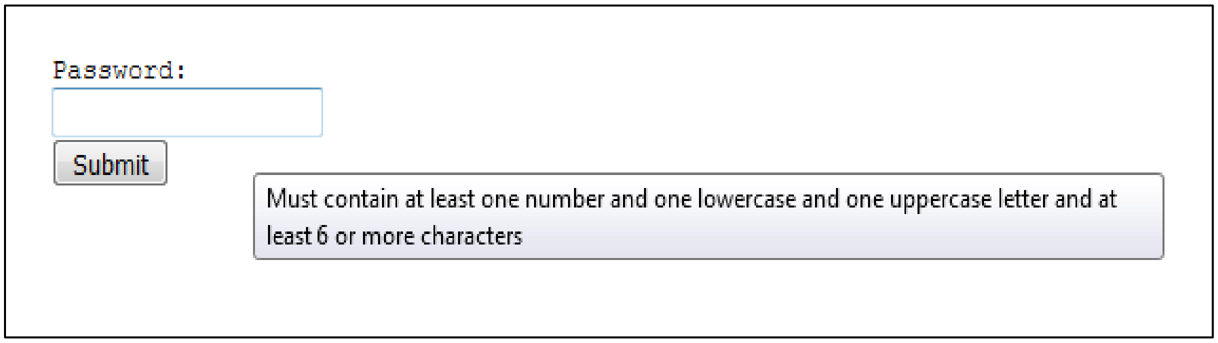


Example of Form Pattern Attribute - 4

In this example, users have to write at least one number and one lowercase and one uppercase letter and at least 6 or more characters.

```
<html>

 <body>

  <pre>

   <form>

    Password:
    <input type="password" name="pw" pattern="(?=.*\d)(?=.*[a-
z])(?=.*[A-Z]).{6,}" title="Must contain at least one number and one
lowercase and one uppercase letter and at least 6 or more characters">

    <input type="submit" value="Submit">

   </form>

  </pre>

 </body>

</html>
```

Output



Must contain at least one number and one lowercase and one uppercase letter and at least 6 or more characters

JavaScript

Introduction to Javascript

Javascript is an **Object Oriented Scripting Language** which is used to create **Client Side Dynamic pages**. It is light weight and cross platform. Javascript codes are written within the HTML document. We can make our webpage more interactive using Javascript. **Brendan Eich** developed Javascript in 1995, appeared in Netscape (Browser). Javascript was known as **Live Script**. Later it was named as Javascript.

Features of Javascript

1. Javascript controls the behavior of a webpage.
2. Javascript is case sensitive language.
3. We can validate user inputs before sending the page to the server.
4. Using Javascript we can display date and time.
5. We can also display pop-up windows and dialog boxes.
6. Javascript cannot be used in any networking application.
7. It does not have any multithreading or multiprocessor capabilities.

Javascript Syntax

Javascript codes are written within **<script> … </script>** tag. We can put <script> tag containing Javascript codes into 2 places – between <body> tag and between <head> tag. But it is recommended to write the codes within the <head> tag**.**

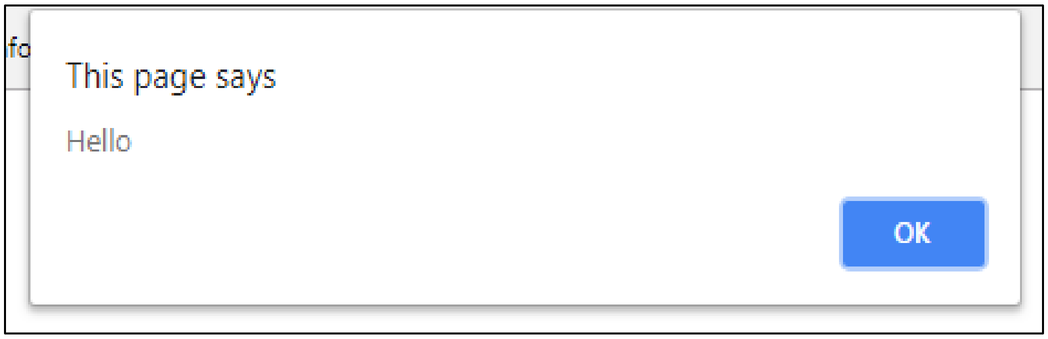
Example of Javascript Syntax

In <body> tag

```
<html>

 <body>
```

```
  <script>
   alert("Hello")
  </script>
 </body>
</html>
```

Output



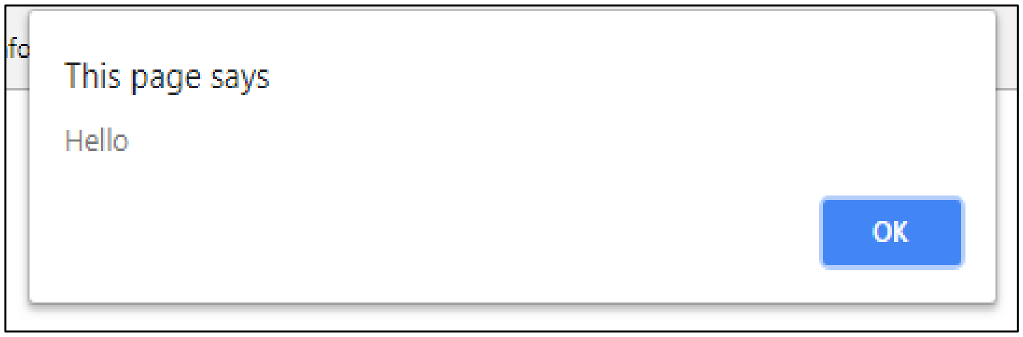
This page says

Hello

In <head> tag

```
<html>
 <head>
  <script>
   alert("Hello")
  </script>
 </head>
<body>
</body>
</html>
```

Output



```
This page says
Hello
```

Javascript Variables

Named storage location is known as **Variable**. A variable can store one data at a time. Variables are also known as **Identifiers**. We can use **var** keyword to initialize a variable in Javascript. Following are some rules while declaring a variable –

1. Variable name must start with a letter, a to z or A to Z.
2. After first letter we can use numbers (0-9). For eg, name1.
3. No special characters are allowed in Javascript variable except underscore (_).
4. Variable names are case sensitive. For eg, **A** and **a** are different variable.

Types of Variable

There are two types of variables –
Local Variable – Local variables are accessible within the function. We declare local variable inside the function.

Global Variable - Global variables are accessible anywhere in the Javascript Code. We declare global variable outside the function.

Correct Variable Declaration	Incorrect Variable Declaration
var x = 10 var name = 20	var 12 = 30 var &name = 40

Example of Variable

```
var x = 4
var y = 5
var z = x+y
```

In this example, x variable stores 4, y variable stores 5 and z variable stores 9.

Javascript Reserved Keywords

There are some reserved keywords which we cannot use as variables, functions, methods or any object names. Following are the list of reserved keywords –

abstract	else	instanceof	switch
boolean	enum	int	synchronized
break	export	interface	this
byte	extends	long	throw
case	false	native	throws
catch	final	new	transient
char	finally	null	true
class	float	package	try
const	for	private	typeof
continue	function	protected	var
debugger	goto	public	void
default	if	return	volatile
delete	implements	short	while
do	import	static	with
double	in	super	

Javascript Datatypes

Javascript provides different datatypes to hold different types of data. It defines which type of value a variable can hold. There are five types of primitive datatypes –

String – This datatype represents text. Eg, “Hello”.

Number – This datatype represents numbers. Eg, 123.

Boolean – This datatype represents boolean value, either true or false.

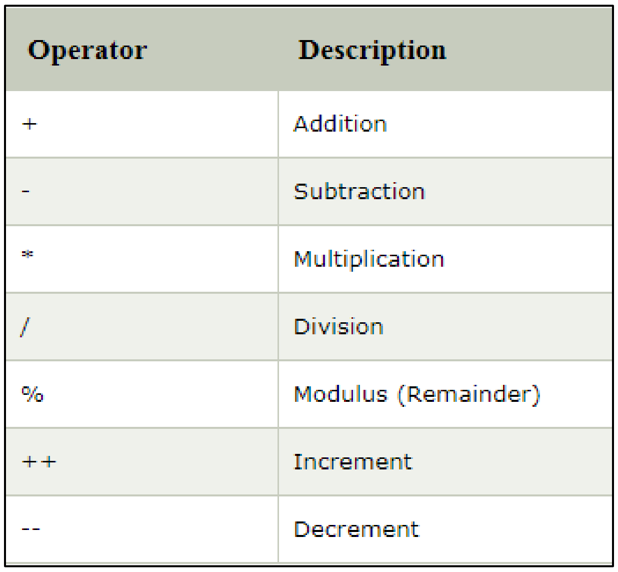
Undefined – This datatype represents undefined value.

Null – This datatype represents null value.

Javascript Operators

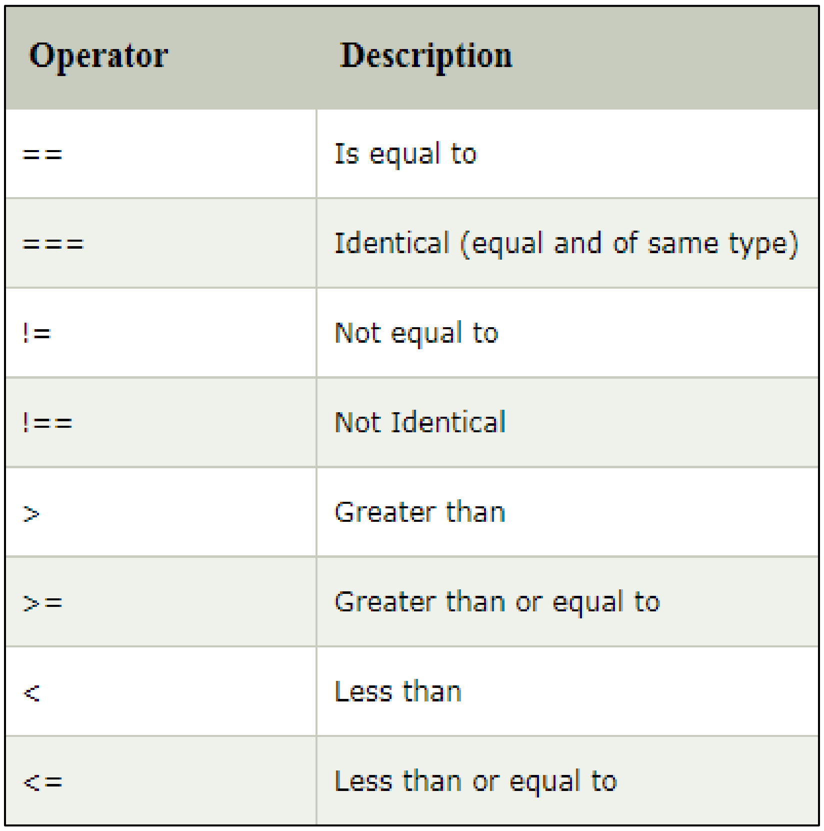
Javascript operators are the symbols which are used to perform operations on operands. For eg, 10 + 20 equals to 30. Here 10 and 20 are the operands, and ‘+’ is the operator. Following are the types of operators which javascript supports –

Arithmetic Operators



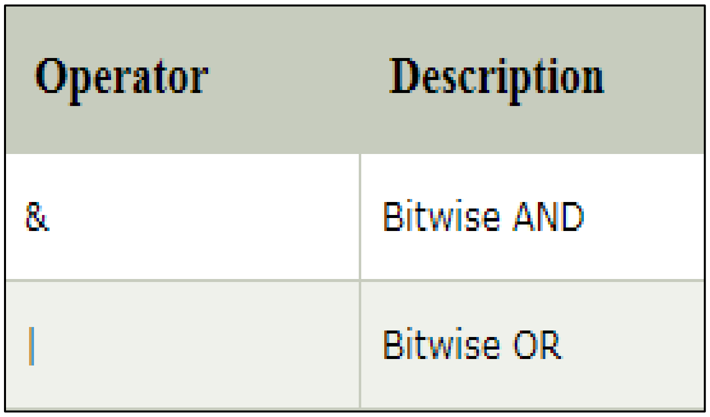
Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (Remainder)
++	Increment
--	Decrement

Comparison (Relational) Operators



Operator	Description
==	Is equal to
===	Identical (equal and of same type)
!=	Not equal to
!==	Not Identical
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

Bitwise Operators



Operator	Description
&	Bitwise AND
|	Bitwise OR

Logical Operators

Operator	Description
&&	Logical AND
||	Logical OR
!	Logical Not

Assignment Operators

Operator	Description
=	Assign
+=	Add and assign
-=	Subtract and assign
*=	Multiply and assign
/=	Divide and assign
%=	Modulus and assign

Conditional (or ternary) Operator

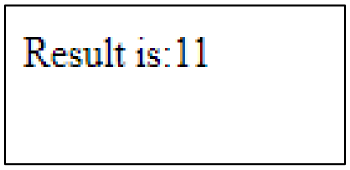
Operator	Description
(?:)	Conditional Operator returns value based on the condition. It is like if-else.

Example of Variable

```
<html>
 <head>
  <script>
   var x = 5
   var y = 6
   var z = x + y
   document.write("Result is:" + z)
  </script>
 </head>
 <body>
 </body>
</html>
```

Here, **document.write()** writes the given string in the document.

Output



```
Result is:11
```

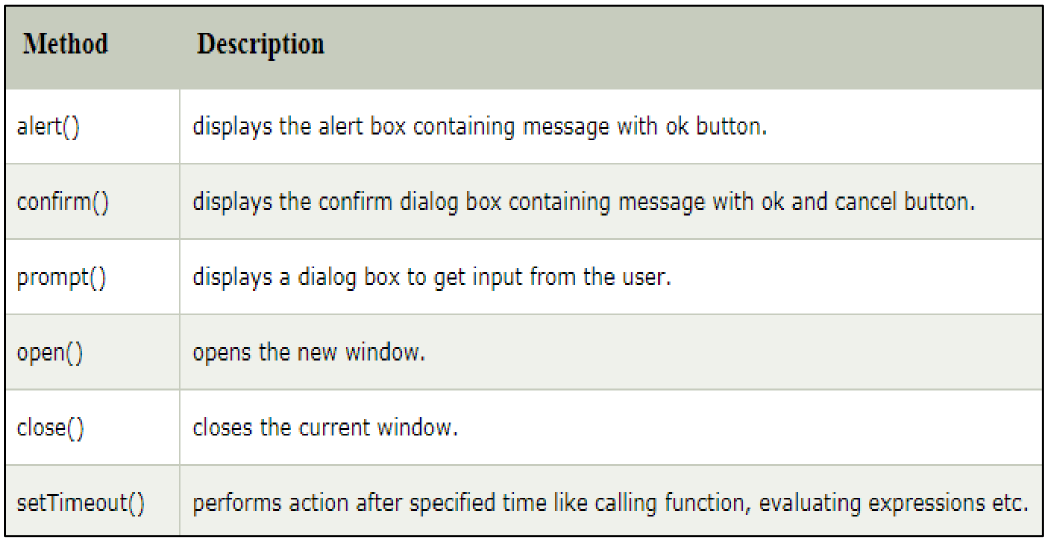
Browser Object Model (BOM)

The Browser Object Model is used to interact with the browser. The objects are –

Window Object

The window object defines the window in browser. It is automatically created by the browser.

Methods of Window Object

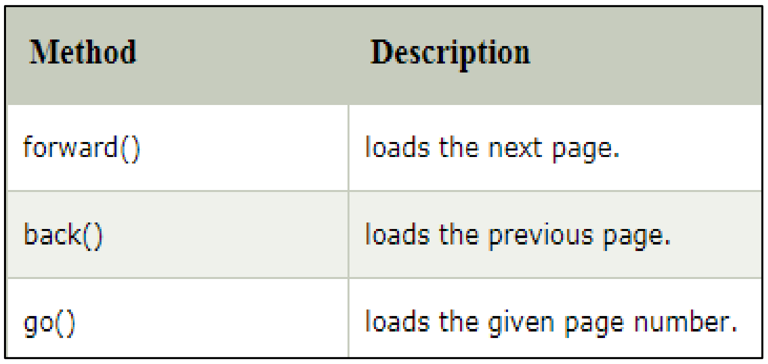


Method	Description
alert()	displays the alert box containing message with ok button.
confirm()	displays the confirm dialog box containing message with ok and cancel button.
prompt()	displays a dialog box to get input from the user.
open()	opens the new window.
close()	closes the current window.
setTimeout()	performs action after specified time like calling function, evaluating expressions etc.

History Object

The history object defines URLs visited by the user. We can load previous, forward and a particular page using this object.

Methods of History Object



Method	Description
forward()	loads the next page.
back()	loads the previous page.
go()	loads the given page number.

Screen Object

The screen object defines the information about browser screen like width, height etc.

Property of Screen Object

Property	Description
width	returns the width of the screen
height	returns the height of the screen
availWidth	returns the available width
availHeight	returns the available height
colorDepth	returns the color depth
pixelDepth	returns the pixel depth.

Document Object Model (DOM)

The Document Object Model defines whole HTML document. HTML document becomes document object when it's loaded in browser. We can change or access HTML document contents by its methods.

Methods of document object

Method	Description
write("string")	writes the given string on the doucment.
writeln("string")	writes the given string on the doucment with newline character at the end.
getElementById()	returns the element having the given id value.
getElementsByName()	returns all the elements having the given name value.

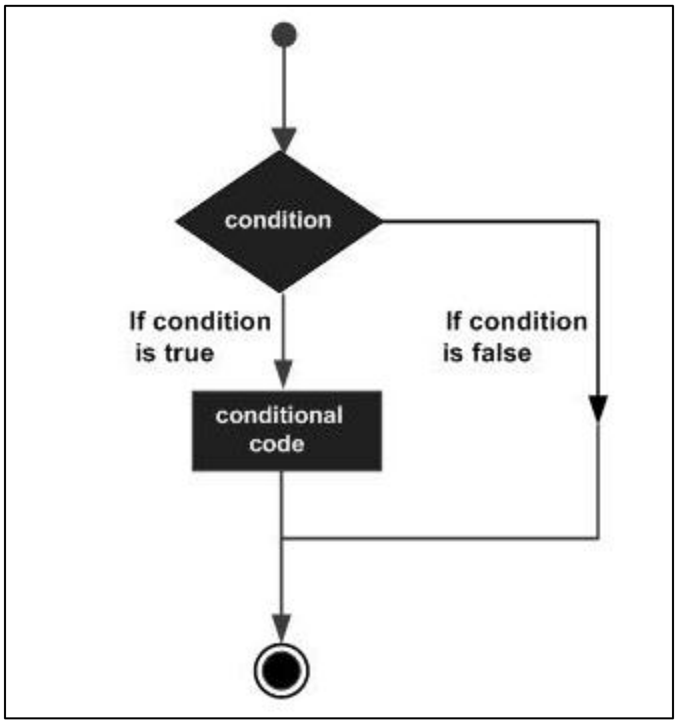
Javascript innerHTML and innerText Property

innerHTML and innerText properties are used to write dynamic HTML on the HTML document . In innerText, text will not be interpreted as HTML text. It will interpret as a normal text. Both properties are used in document object methods.

Decision Making

Decision making defines that the programmer specifies one or more conditions to be evaluated by the program along with a statement or statements to be executed if the condition is true and the other statement to be executed if the condition is false.

Flow Chart



There are three types decision making in Javascript –

If Statement

In if statement, the content will be executed only if the condition is true.

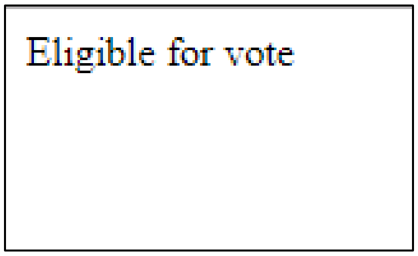
Syntax

```
if (condition)
 {
   Code block to be executed if the condition is true
 }
```

Example of if Statement

```
<html>

 <head>

  <script>

   var age = 25

    if( age >= 18 )

     {

      document.write("Eligible for vote")

     }

  </script>

 </head>

</html>
```

Output



```
Eligible for vote
```

If Else Statement

An if statement can be followed by an else statement, which executes when the condition is false.

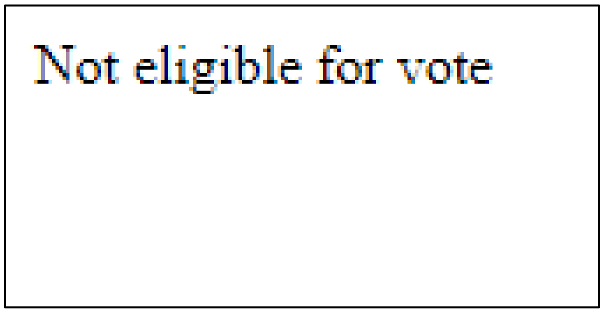
Syntax

```
if (condition)
{
    Code block to be executed if the condition is true
}
else
{
    Code block to be executed if the condition is false
}
```

Example of if else Statement

```
<html>

 <head>

  <script>

   var age = 16

    if( age >= 18 )

     {

      document.write("Eligible for vote")

     }

    else

     {

      document.write("Not eligible for vote")
     }

  </script>

 </head>

</html>
```

Output



```
Not eligible for vote
```

If Else If statement

An if statement can be followed by an else if statement to check multiple conditions.

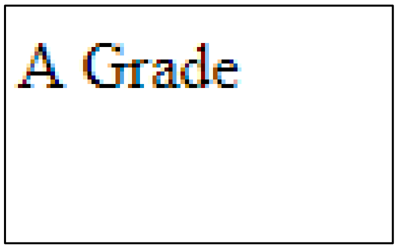
Syntax

```
if (condition1)
{
    Code block to be executed if condition1 is true
}
else if (condition2)
 {
    Code block to be executed if the condition1 is false and
condition2 is true
}
else
{
    Code block to be executed if the condition1 is false and
condition2 is false
}
```

Example of if else if Statement

```
<html>

 <head>

  <script>

    var marks = 85

     if( marks > 80 )

      {

        document.write("A Grade")

      }

     else if ( marks > 60 )

      {

        document.write("B Grade")

      }

     else if ( marks > 40 )

       {

        document.write("C Grade")

     }

     else

     {

         document.write("Fail")

     }

  </script>
 </head>

</html>
```

Output



```
A Grade
```

Javascript Switch Statement

The switch statement is just like if else if statement. Switch statement is used to execute one code from multiple expressions.

Syntax

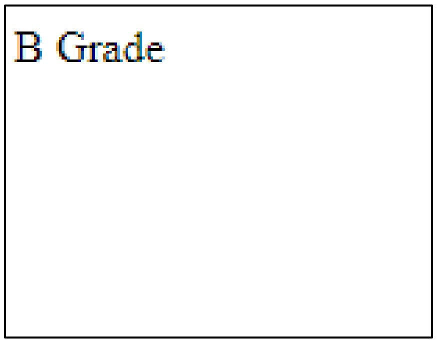
```
switch (expression)
{
 case value1:
   code block
 break

 case value2:
  code block
 break

 default:
 code block
}
```

Here, switch expression is evaluated once. Expression value is compared with each case values. If there is a match, the associated code block will be executed. **Break** statement represents the end of particular case. If nothing matches, **default** condition will be executed.

Example of Switch Statement

```
<html>

 <head>

  <script>

   var grade='B'

   var result

    switch(grade)
     {

      case 'A':
      result="A Grade"
      break

      case 'B':
      result="B Grade"
      break

      case 'C':
      result="C Grade
      break

      default:
      result="No Grade"

     }

    document.write(result)

  </script>

 </head>

</html>
```

Output



```
B Grade
```

Javascript Loops

Loop statement allows us to execute a statement or a group of statement multiple times. Following are the types of loops use in Javascript –

While Loop

While loop executes a statement or a code block repeatedly as long as the condition is true. Once the condition becomes false, loop terminates.

Syntax

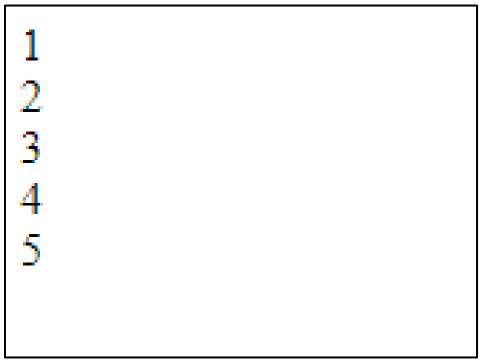
```
while (condition)
{
    Code to be executed
}
```

Example of While Loop

```
<html>

 <head>
```

```
  <script>

  var i=1

    while (i<=5)

    {

      document.write(i + "<br>")

      i++

    }

  </script>

 </head>

</html>
```

Output



```
1
2
3
4
5
```

Do While Loop

Do while loop is similar to the while loop but here the condition is checked at end of the loop. That is why the loop will be executed at least once, even if condition is false.

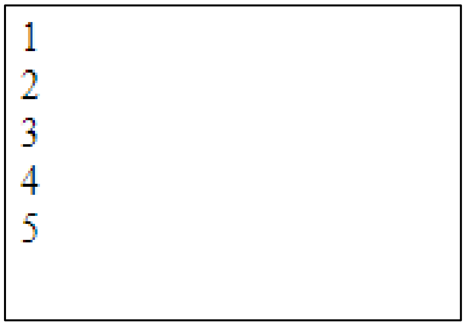
Syntax

```
Do
{
 code to be executed
}
while (condition)
```

Example of Do While Loop

```
<html>

 <head>

  <script>

   var i=1

    do

    {

     document.write(i + "<br/>")

     i++

     }

    while (i<=5)

  </script>

 </head>

</html>
```

Output



```
1
2
3
4
5
```

For Loop

For loop includes following three parts –

Initialization – Here we initialize a starting value.

Condition – Here we write the condition. If the condition is true, the statement will be executed and if the condition is false, loop terminates.

Increment/Decrement – Here we increase or decrease the value.

Syntax

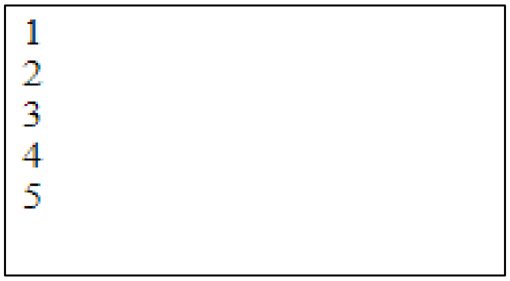
```
for (initialization; condition; increment)
{
    code to be executed
}
```

Example of For Loop

```
<html>

 <head>

  <script>
```

```
   for (i=1; i<=5; i++)

   {

     document.write(i + "<br/>")

   }
  </script>

 </head>

</html>
```

Output



```
1
2
3
4
5
```

Jump Statement

Javascript provides full control to handle loops. If we want to exit from loop any time or skip any code block of loop, we can use **Jump Statement.** There are two types of jump statement –

Break Statement

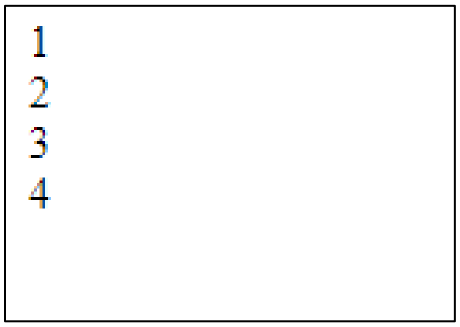
We have already used break statement in switch statement. Break statement is used to exit from the loop when given condition is satisfied.

Example of Break Statement

```
<html>

 <head>
```

```
  <script>
    var i
     for (i = 1; i < 10; i++)
     {
       if (i === 5)
        {
          break
        }
      document.write(i + "<br/>")
     }
  </script>
 </head>
</html>
```

Output



```
1
2
3
4
```

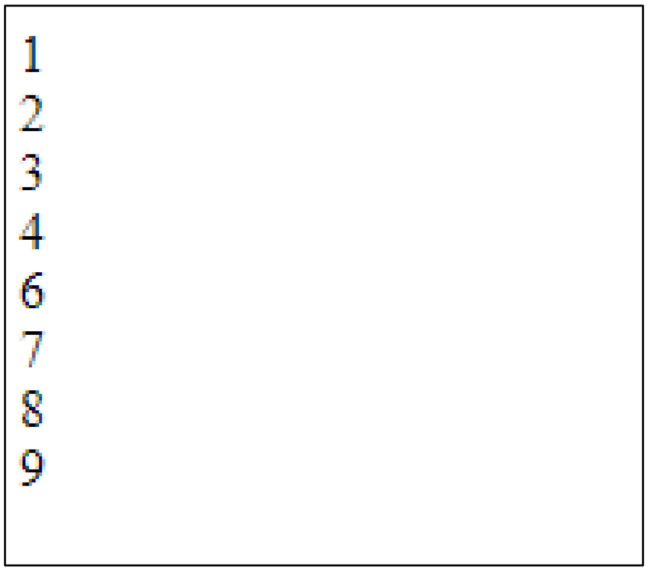
Continue Statement

Continue Statement skips one iteration in the loop, if given condition occurs and continues with the next iteration.

Example of Continue Statement

```
<html>

 <head>

  <script>

    var i

     for (i = 1; i < 10; i++)

      {

       if (i === 5)

         {

           continue

         }

      document.write(i + "<br/>")
      }

  </script>

 </head>

</html>
```

Output


```
1
2
3
4
6
7
8
9
```

Javascript Function

Functions are used to perform specific task. We can call functions many times to reuse the code. This eliminates the need of writing the same code again and again. Function divides a big program into a number of small functions.

Function Definition

We need to define a function before using it. To define a function we use **function** keyword followed by a function name, parameter list (list might be empty) and a code block surrounded by braces.

Calling a Function

To call a function we just need to mention the exact function name, later in the code block, using **onclick** attribute.

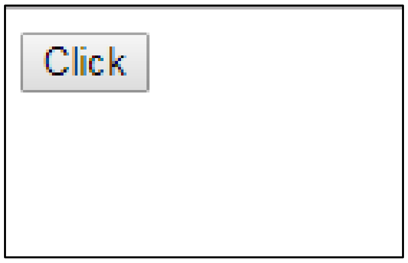
Syntax

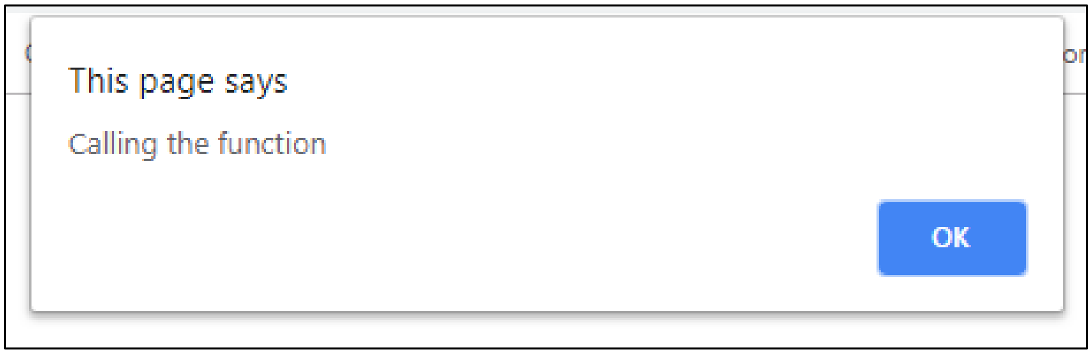
```
function functionName (parameter list)
{
 Code block
}
```

Example of Calling a Function

```
<html>

 <head>

  <script>

   function call()

   {

     alert("Calling the function")

   }

  </script>

 </head>

 <body>

  <input type="button" onclick="call()" value="Click">

 </body>

</html>
```

Output





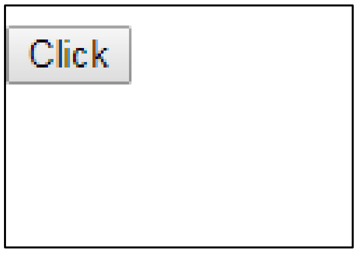
Function Parameters/Arguments

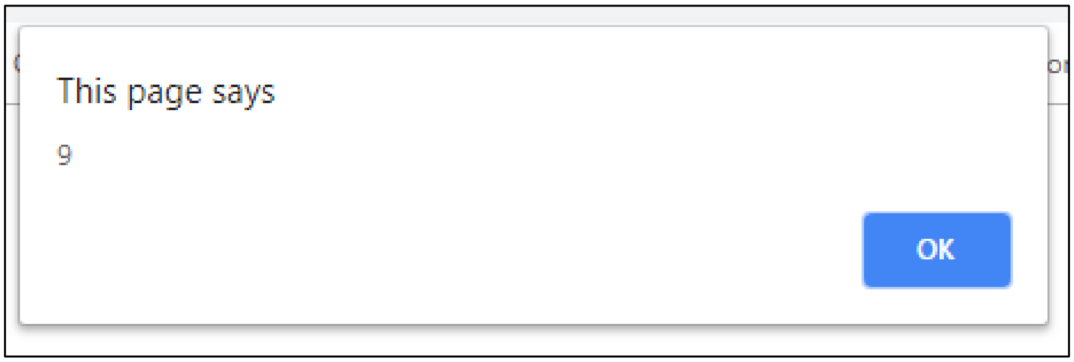
We can call a function by passing parameters or arguments.

Example of Function Parameters/Arguments

```
<html>

 <head>

  <script>

    function square(number)

    {

     alert(number*number)
    }

  </script>

 </head>
 <body>
  <input type="button" onclick="square(3)" value="Click">
 </body>
</html>
```

Output





Function with Return Value

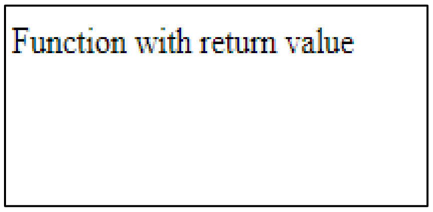
We can call a function that returns a value which we can use it in our program.

Example of Function with Return Value

```
<html>
 <head>
  <script>
   function msg()
```

```
    {

      return "Function with return value"

    }

   document.write(msg())

  </script>

 </head>

</html>
```

Output



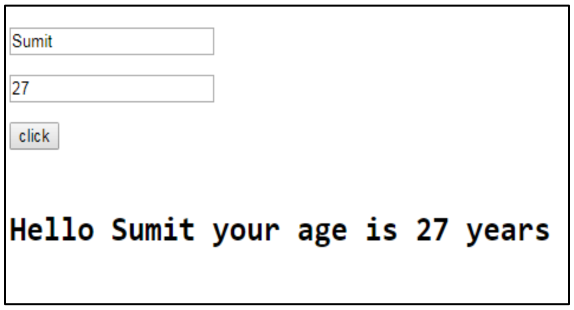
```
Function with return value
```

Example of Show Name and Age Using Function

```
<html>

 <head>

  <script>

    function Age()

     {
       var name=document.f1.t1.value

       var age=document.f1.t2.value

       document.getElementById('demo').innerHTML="hello "+name+" your
age is "+age+" years"
```

```
     }
   </script>
  </head>
 <body>
  <pre>
   <form name="f1" method="post">
    <input type="text" name="t1" placeholder="Enter name">
    <input type="text" name="t2" placeholder="Enter age">
    <input type="button" value="click" onClick="Age()">
   </form>
   <h1 id="demo"></h1>
  </pre>
 </body>
</html>
```

Output



Hello Sumit your age is 27 years

External JavaScript File

We can create external javascript file and use it in many HTML document. It provides reusability of codes. It is recommended that write all external javascript files into a single file. It increases webpage speed. Always save external javascript file as **.js** extension.

Example of External Javascript File

Suppose we create a named **script.js** and create a prompt box.

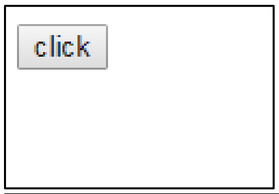
```
function msg()
 {
   var x = prompt ("Enter Name", "Full Name")
   document.write("<h2> Hello "+x+" Happy Birthday</h2>")
 }
```

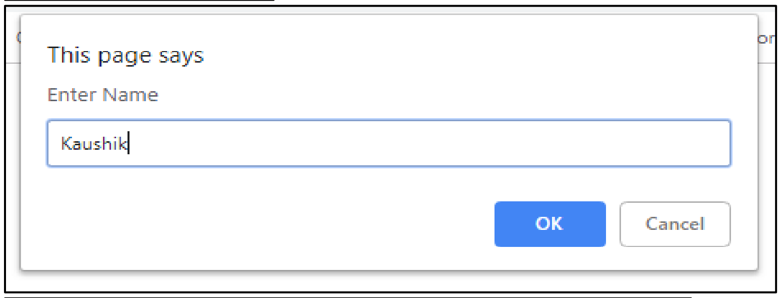
Now include this file into a HTML page.

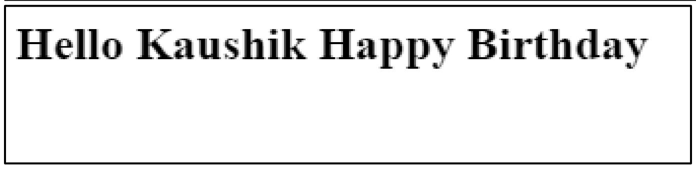
```
<html>
  <head>
   <script type="text/javascript" src="script.js">
   </script>
  </head>
 <body>
  <form>
   <input type="button" value="click" onclick="msg()">
  </form>
 </body>
```

```
</html>
```

Output







Hello Kaushik Happy Birthday

Javascript Number Methods

Javascript Number Methods represent a numeric value. It may be integer or floating point.

Number Properties

Property	Description
MAX_VALUE	Returns the largest maximum value. Eg, var x = Number.MAX_VALUE
MIN_VALUE	Returns the largest minimum value. Eg, var x = Number.MIN_VALUE
POSITIVE_INFINITY	Returns positive infinity, overflow value.

	Eg, var x = Number.POSITIVE_INFINITY
NEGATIVE_INFINITY	Returns negative infinity, overflow value. Eg, var x = Number.NEGATIVE_INFINITY
NaN	Represents "Not a Number" value. Eg, var x = Number.NaN

Number Methods

Methods	Description
parseFloat()	Converts the given string into a floating point number. Eg, parseFloat("10") //returns 10
parseInt()	Converts the given string into an integer number. Eg, parseInt("10.33") //returns 10
toExponential()	Returns the string that represents exponential notation of the given number. Eg, var x = 9.656 x.toExponential(2) //returns 9.66e+0
toFixed()	Returns the string that represents a number with exact digits after a decimal point. Eg, var x = 9.656 x.toFixed(2) //returns 9.66
toPrecision()	Returns the string representing a number of specified precision. Eg, var x = 9.656 x.toPrecision(2) //returns 9.7

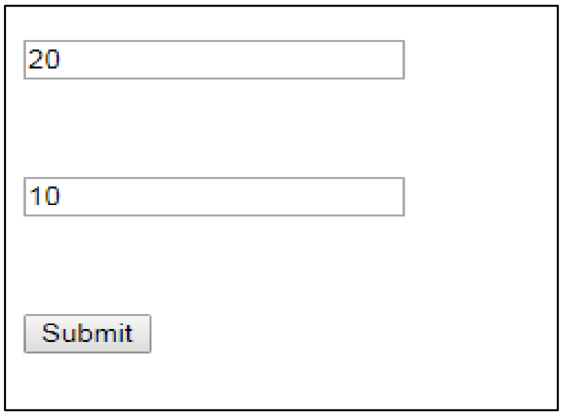
toString()	Returns the given number in the form of string. Eg, var x = 123 x.toString() //returns 123

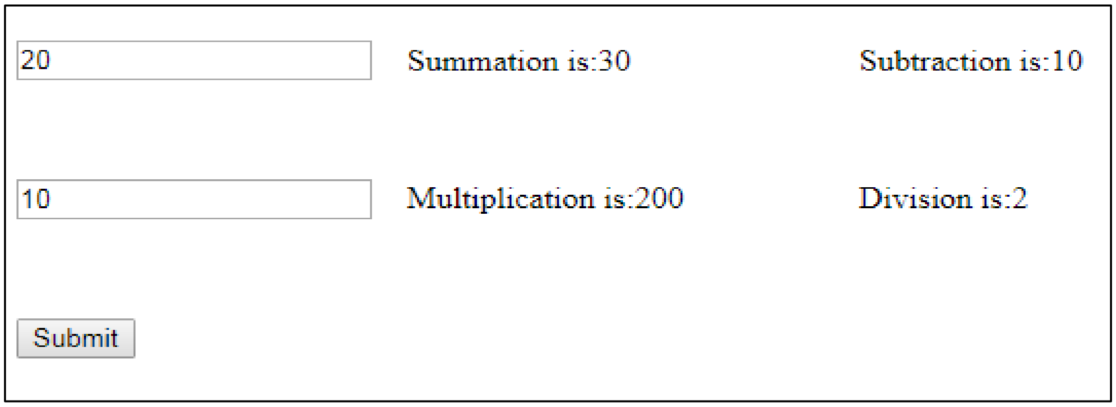
Example of Number Methods

```
<html>

 <head>

  <script>

    function Func()

     {

        var a=parseInt(document.form1.n1.value)

        var b=parseInt(document.form1.n2.value)

        var c=a+b

        var d=a-b

        var e=a*b


        var f=a/b

        document.getElementById('demo1').innerHTML="Summation is:"+c

      document.getElementById('demo2').innerHTML="Subtraction is:"+d

  document.getElementById('demo3').innerHTML="Multiplication is:"+e

       document.getElementById('demo4').innerHTML="Division is:"+f

     }

   </script>
  </head>
 <body>
  <form name="form1" method="post">
   <table width="50%" height="220" border="0">
    <tr>
     <td width="30%">
        <input type="text" id="n1" placeholder="1st no"></td>
     <td width="35%" id="demo1"></td>
     <td width="35%" id="demo2"></td>
   </tr>
   <tr>
    <td>
     <input type="text" id="n2" placeholder="2nd no"></td>
    <td id="demo3"></td>
    <td id="demo4"></td>
  </tr>
  <tr>
    <td>
    <input type="button" value="Submit" onclick="Func()">
    </td>
     <td></td>
     <td></td>
```

```
      </tr>

    </table>

  </form>

 </body>

</html>
```

Output

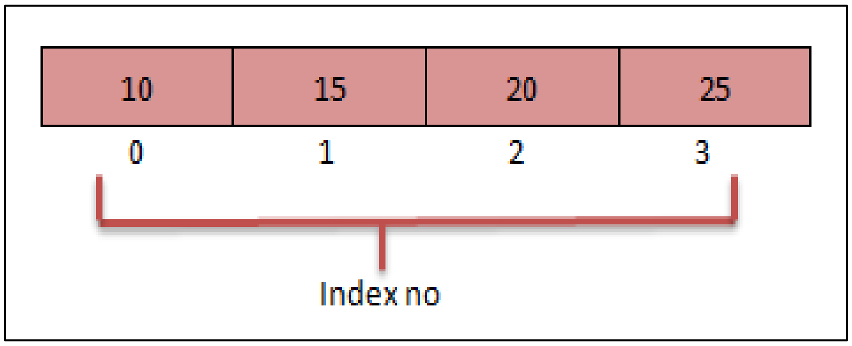




```
Summation is:30          Subtraction is:10
Multiplication is:200    Division is:2
```

Javascript Array

Array is a collection of similar elements, stored in a contiguous memory location. We can store multiple values in a single variable. Array values can be identified by **Index No**. Index no always starts with '0'.



There are three ways we can create an array in Javascript –

Array Literal

Syntax

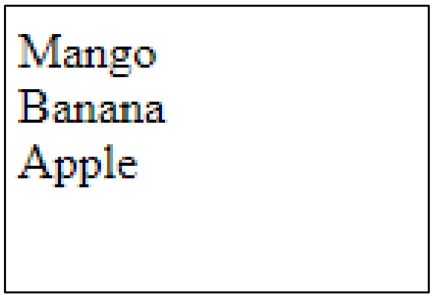
```
var arrayname=[value1,value2.....valueN]
```

Example of Array Literal

```
<html>

 <head>

  <script>

   var i

   var fruit=["Mango","Banana","Apple"]

    for (i=0;i<3;i++)
```

```
     {
       document.write(fruit[i] + "<br/>")

     }

  </script>

 </head>

</html>
```

Output



```
Mango
Banana
Apple
```

Array Directly (Using new Keyword)

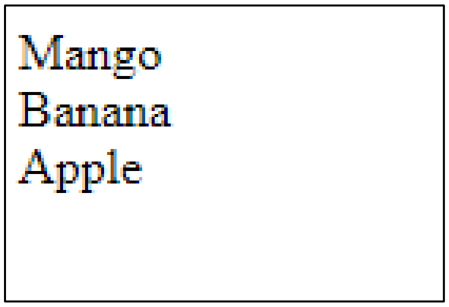
Syntax

```
var arrayname=new Array()
```

Example of Array Directly (Using new Keyword)

```
<html>

 <head>

  <script>

    var i
    var fruit = new Array()

     fruit[0] = "Mango"

     fruit[1] = "Banana"

     fruit[2] = "Apple"

      for (i=0;i<3;i++)

       {

         document.write(fruit[i] + "<br>")

       }

  </script>

 </head>

</html>
```

Output



```
Mango
Banana
Apple
```

Array Constructor (Using new Keyword)

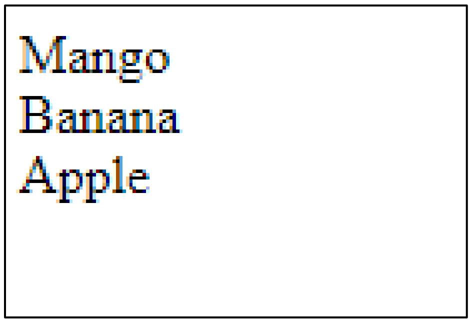
Syntax

```
var arrayname=new Array(value1,value2...valueN)
```

Example of Array Constructor - 1

```
<html>

 <head>

  <script>

   var i

   var fruit=new Array("Mango","Banana","Apple")

    for (i=0;i<3;i++)

     {

       document.write(fruit[i] + "<br>")

     }

  </script>

 </head>

</html>
```

Output



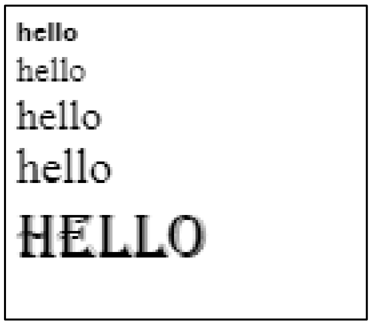
```
Mango
Banana
Apple
```

Example of Array Constructor - 2

```
<html>
```

```
 <head>

  <script>

   function func()
    {

     var a,x

     x=new Array('aharoni','ariel','times new roman','bauhaus
93','algerian')

      for(a=0;a<5;a++)

       document.write("<font size="+(a+1)+";
face="+x[a]+">hello<br></font>")

    }

  </script>

 </head>

 <body>

  <input type="button" value="Click" onClick="func()">

 </body>

</html>
```

Output



```
hello
hello
hello
hello
HELLO
```

Javascript String

String represents series of characters.

There are two ways we can create strings in Javascript –

String Literal

String literals are created using double quotes.

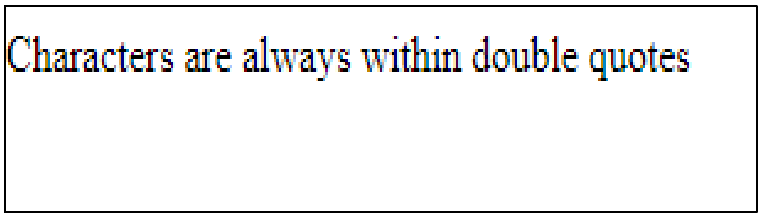
Syntax

```
var stringname="string value"
```

Example of String Literal

```
<html>

 <head>

  <script>

    var str="Characters are always within double quotes"

      document.write(str)

  </script>

 </head>

</html>
```

Output



```
Characters are always within double quotes
```

String Object (Using new Keyword)

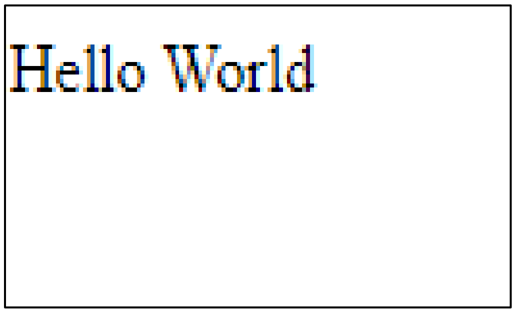
Syntax

```
var stringname=new String("stringname")
```

Example of String Object

```
<html>
 <head>
  <script>
    var str=new String("Hello World")
     document.write(str)
  </script>
 </head>
</html>
```

Output



```
Hello World
```

Javascript Math Object

Math Object allows us to perform mathematical task on numbers.

Math Methods

Methods	Description
.round()	Returns closest integer value of the given number. Eg, Math.round(4.7) // returns 5
.pow()	Returns value of base to the power of exponent. Eg, Math.pow(8, 2) // returns 64
.sqrt()	Returns the square root of the given number Eg, Math.sqrt(64) // returns 8
.abs()	Returns the absolute value of the given number. Eg, Math.abs(-4.7) // returns 4.7
.ceil()	Returns a smallest integer value, greater than or equal to the given number. Eg, Math.ceil(4.4) // returns 5
.floor()	Returns largest integer value, lower than or equal to the given number. Eg, Math.floor(4.7) // returns 4
.min()	Returns minimum value of the given numbers. Eg, Math.min(0, 150, 30, 20, -8, -200) // returns -200
.max()	Returns maximum value of the given numbers. Eg, Math.max(0, 150, 30, 20, -8, -200) // returns 150
.random()	Returns random number between 0 (inclusive) and 1 (exclusive). Eg, Math.random() // returns a random number

Example of Changing Background Colors Using Math Methods

```
<html>

 <head>

  <script>

    function bgcolor()

    {
      window.setTimeout("bgcolor();","5000")

      x=new
Array('red','green','blue','indigo','skyblue','violet','orange','yello
w')


document.body.style.background=x[Math.floor(Math.random()*x.length)]

    }

  </script>

 </head>

 <body onload="bgcolor()">

 </body>

</html>
```

Javascript Date Object

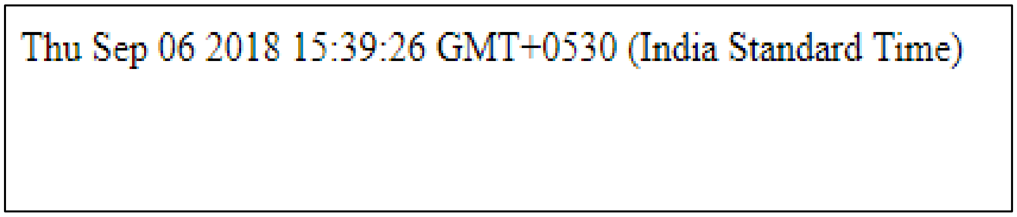
Date Object can be used to display time, year, month and day.

Example of Date Object

```
<html>

 <head>
```

```
  <script>
    var time=new Date()
    document.write(time)
  </script>
 </head>
</html>
```

Output



```
Thu Sep 06 2018 15:39:26 GMT+0530 (India Standard Time)
```

Date Methods

Methods	Description
getDate()	Returns the integer value between 1 and 31 that represents the day for the specified date. Eg, var d = new Date() document.write(d.getDate())
getHours()	Returns the integer value between 0 and 23 that represents the hours. Eg, var d = new Date() document.write(d.getHours())
getMinutes()	Returns the integer value between 0 and 59 that represents the minutes.

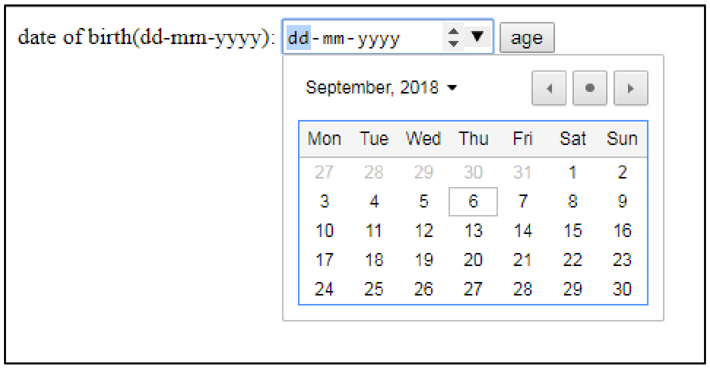
	Eg, var d = new Date() document.write(d.getMinutes())
getSeconds()	Returns the integer value between 0 and 60 that represents the seconds. Eg, var d = new Date() document.write(d.getSeconds())
getMilliseconds()	Returns the integer value between 0 and 999 that represents the milliseconds. Eg, var d = new Date() document.write(d.getMilliseconds())
getDay()	Returns the integer value between 0 and 6 that represents the day of the week. Eg, var d = new Date() document.write(d.getDay())
getFullYear()	Returns the integer value that represents the year. Eg, var d = new Date() document.write(d.getFullYear())

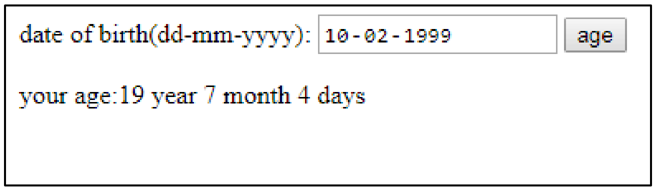
Example of Date Methods

```
<html>

 <head>

  <script>

   function ageCount()

    {

      var date1=new Date()

      var dob=document.getElementById('dob').value
```

```
       var date2=new Date(dob)

       if(dob)

        {

          var y1=date1.getFullYear()

          var m1=date1.getMonth()

          var d1=date1.getDate()

          var y2=date2.getFullYear()

          var m2=date2.getMonth()

          var d2=date2.getDate()

          var year=y1-y2

          var month=Math.abs(m1-m2)

          var days=Math.abs(d1-d2)

       if(date1<=date2)

        {

         alert("invalid date")

         return false

        }

      document.getElementById('output').innerHTML="your age:"+year+"
year "+month+" month "+days+" days"

    return true

     }

    else
    {
     alert("invalid date format")
     return false
    }
   }
  </script>
 </head>
<body>
 date of birth(dd-mm-yyyy):
  <input type="date" id="dob" placeholder="dd-mm-yyyy">
  <input type="submit" value="age" onClick="ageCount()">
  <p id="output"></p>
 </body>
</html>
```

Output





```
date of birth(dd-mm-yyyy): 10-02-1999  age

your age:19 year 7 month 4 days
```

Form Validation

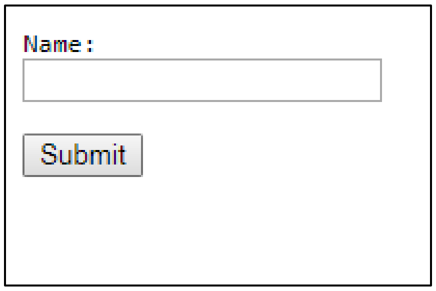
Form Validation provides a way to enter all the necessary data before submitting the form. If the form has inappropriate values or field is empty then, server send all the data back to the client and request to resubmit the form again with the correct information.

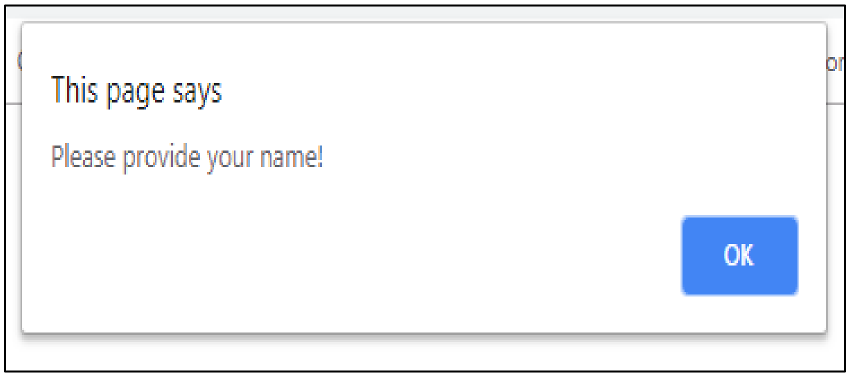
Example Form Validation

```
<html>

 <head>

  <script>

    function validate()

     {

      if( document.myForm.Name.value == "" )

       {

        alert( "Please provide your name!" )

        document.myForm.Name.focus()

        return false

       }

     }

   </script>

  </head>

 <body>

  <pre>

   <form name="myForm">

    Name:
```

```
    <input type="text" name="Name">

    <input type="submit" value="Submit" onClick="validate()">

   </form>

  </pre>

 </body>

</html>
```

Output




This page says

Please provide your name!

Number Validation

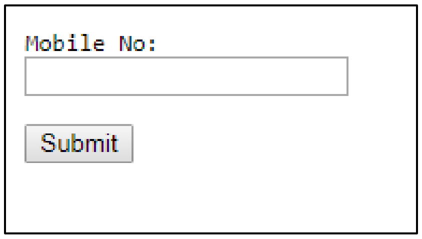
We can validate any text field for numeric value only using isNaN() function.

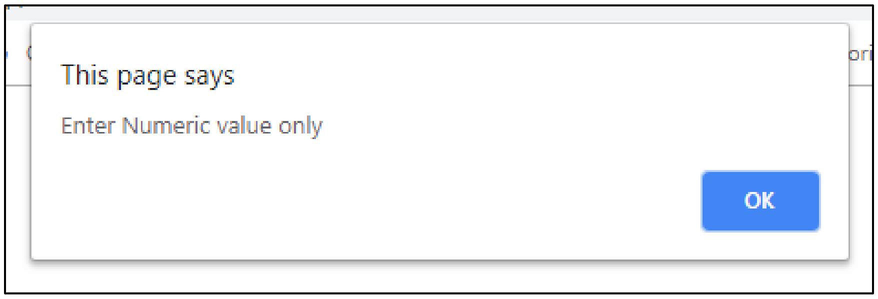
Example of Number Validation

```
<html>

 <head>

  <script>

    function validate()

     {
       var num=document.myform.num.value

       if (isNaN(num))

        {

          alert("Enter Numeric value only")

          return false

        }

      else

       {

         return true

       }

     }

    </script>

   </head>

  <body>

   <pre>

    <form name="myform">

     Mobile No:
     <input type="text" name="num">

     <input type="submit" value="Submit" onClick="validate()">

   </form>

  </pre>

 </body>
```

```
</html>
```

Output


Mobile No:


This page says

Enter Numeric value only

Password Validation

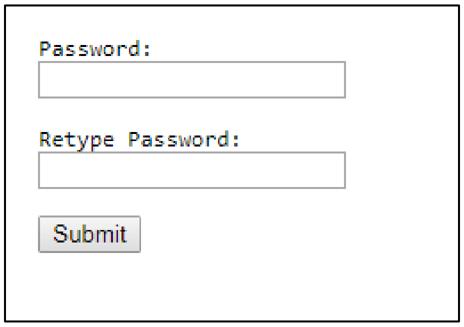
We can also validate password.

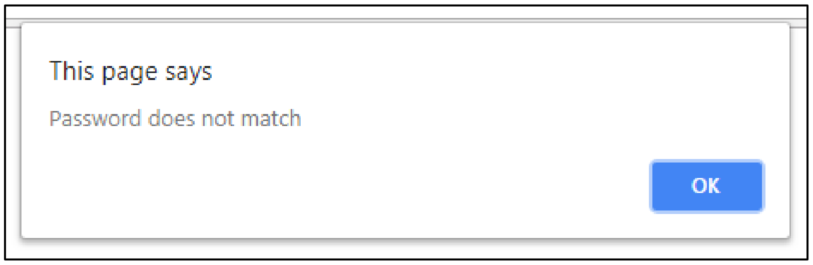
Example of Password Validation

```
<html>

 <head>

  <script>

    function validate()

     {

       if(document.myForm.p1.value == ""
||document.myForm.p1.value.length != 6 )

        {

          alert("Password must be fill with 6 characters" )

          document.myForm.p1.focus()

          return false

        }
```

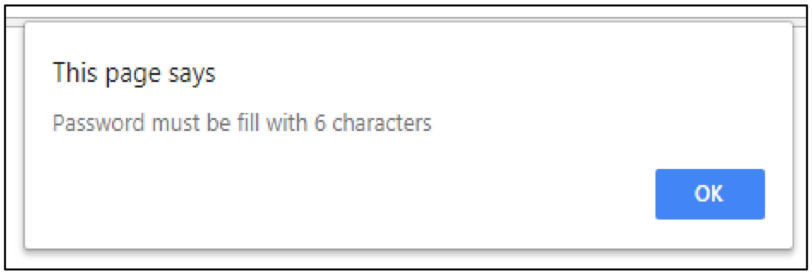
```
      var pass=document.myForm.p1.value

      var pass1=document.myForm.p2.value

        if(pass!=pass1)

         {

           alert( "Password does not match" )

           document.myForm.p2.focus()

           return false

         }

      }

    </script>

   </head>

  <body>

   <pre>

    <form name="myForm">
     Password:
     <input type="password" name="p1">

     Retype Password:
     <input type="password" name="p2">

    <input type="submit" value="Submit" onClick="validate()">

   </form>

  </pre>

 </body>

</html>
```

Output




This page says

Password does not match


This page says

Password must be fill with 6 characters

Email Validation

There are some criteria we need to follow while validate email address –

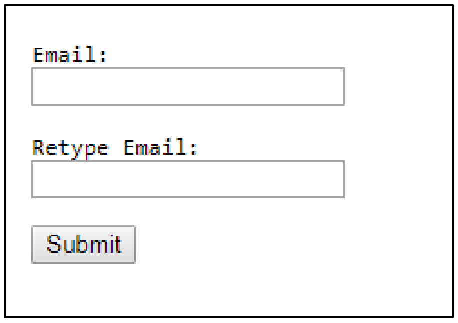
- An email id must contain '@ 'and '.' character.
- There must be at least one character before and after '@'.
- There must be at least two characters after '.'

Example of Email Validation

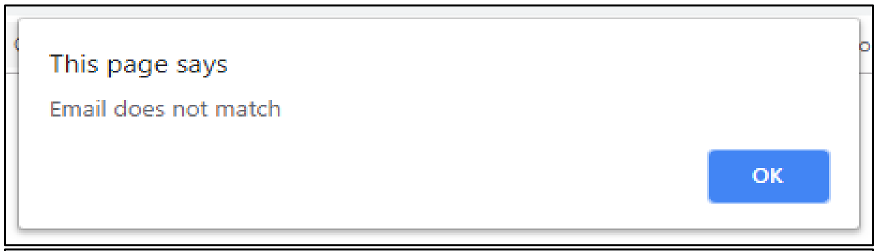
```
<html>

 <head>

  <script>

   function validate()

    {

     var emailID = document.myForm.e1.value
    atpos = emailID.indexOf("@")

    dotpos = emailID.lastIndexOf(".")

      if(atpos < 1 || ( dotpos - atpos < 2 ))

       {

         alert( "Please provide @ and . in your Email!" )

         document.myForm.e1.focus()

         return false

       }
    var em1=document.myForm.e1.value

    var em2=document.myForm.e2.value

      if(em2!=em1)

       {

         alert("Email does not match")

         document.myForm.em2.focus()

         return false

       }

    }

   </script>

  </head>

 <body>

  <pre>

   <form name="myForm">
```

```
      Email:
       <input type="email" name="e1">

      Retype Email:
       <input type="email" name="e2">

       <input type="submit" value="Submit" onClick="validate()">

    </form>

   </pre>

  </body>

 </html>
```

Output

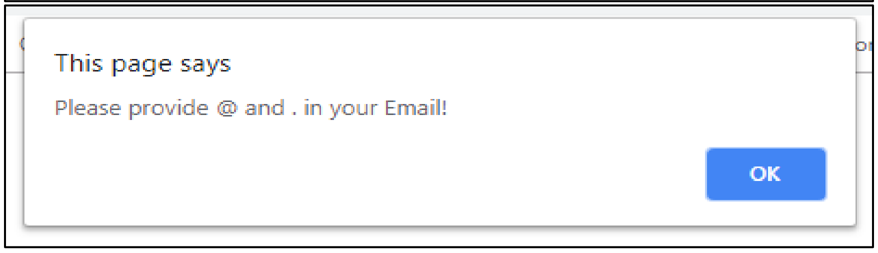


```
Email:

Retype Email:
```



This page says

Email does not match



This page says

Please provide @ and . in your Email!

CSS

Introduction of CSS

Cascading Style Sheet (CSS) simplify the process of making a web page presentable. CSS mostly handles the look of the web page. Håkon Wium Lie invented CSS in 1994. The latest version of CSS is CSS 3.

Features of CSS

1. CSS is case sensitive.
2. Semicolon is mandatory after each CSS declaration.
3. CSS is used to design HTML tags.
4. We can write CSS once and then reuse same sheet in multiple HTML pages.
5. CSS has a much wider array of attributes than HTML, so we can give a better look to HTML pages.
6. To make a change, simply change the style, and all elements in the web pages will be updated automatically.
7. CSS is platform independent.

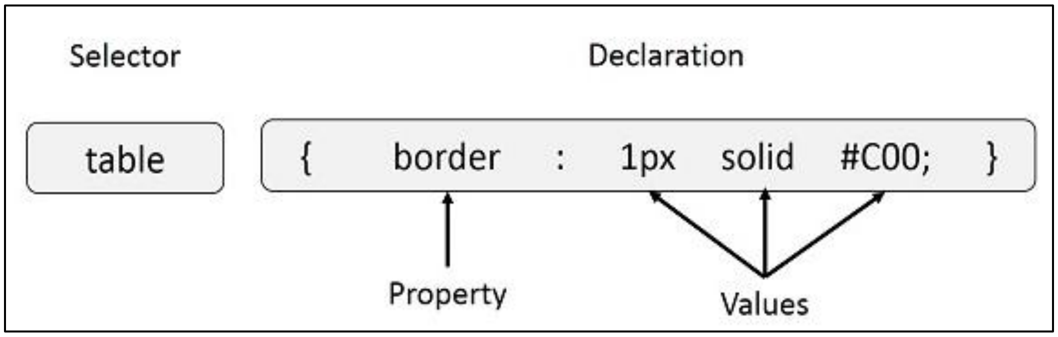
CSS Syntax

We write CSS in **<style>** … **</style>** tag. We write <style> tag inside <head> tag. A CSS rule set contains a selector and a declaration block.

Selector – A selector is an HTML tag at which a style will be applied. This could be any tag.

Property - A property is a type of attribute of HTML tag. All the HTML attributes are converted into CSS properties.

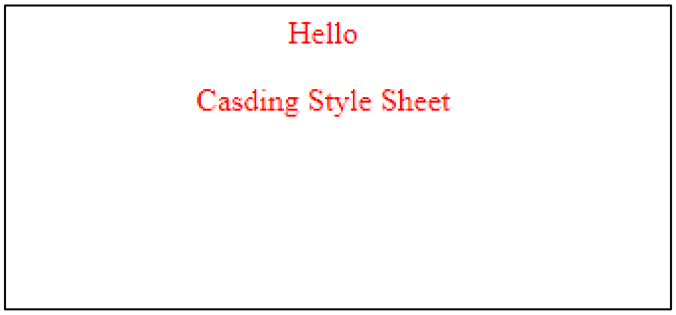
Value – Values are assigned to properties.



Example of CSS Syntax

```
<html>
 <head>
  <style>
    p
    {
      color: red;
      text-align: center;
    }
  </style>
 </head>
 <body>
  <p>Hello</p>
  <p>Casding Style Sheet</p>
 </body>
</html>
```

Output



```
Hello

Casding Style Sheet
```

CSS Selectors

CSS selectors are used to select HTML elements based on their element name, id, class, attribute etc.

The Element Selector

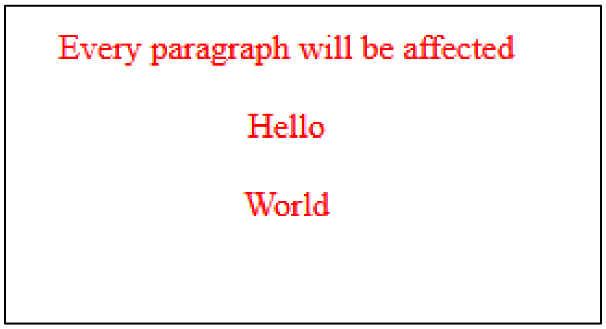
The element selector selects elements based on the element name.

We can select all <p> elements on a page.

Example of CSS Selectors

```
<html>

 <head>

  <style>

   p

   {

     text-align: center;

     color: red;

   }
   </style>
 </head>
 <body>
  <p>Every paragraph will be affected</p>
  <p>Hello</p>
  <p>World</p>
 </body>
</html>
```

Output



Every paragraph will be affected

Hello

World

The Id Selector

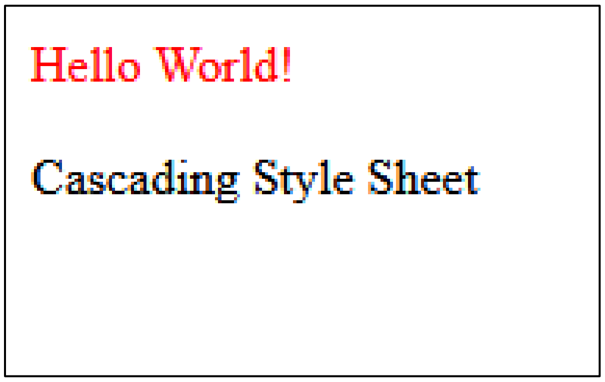
The id selector uses the id attribute of an HTML element to select a specific element. The id of an element should be unique. Use hash (#) to select an element with a specific id, followed by the id of the element.

Example of Id Selector

```
<html>
 <head>
  <style>
```

```
    #para1

    {

      text-align: left;


      color: red;
    }

   </style>

  </head>

  <body>

   <p id="para1">Hello World!</p>

   <p>Cascading Style Sheet</p>

  </body>

</html>
```

Output



```
Hello World!

Cascading Style Sheet
```

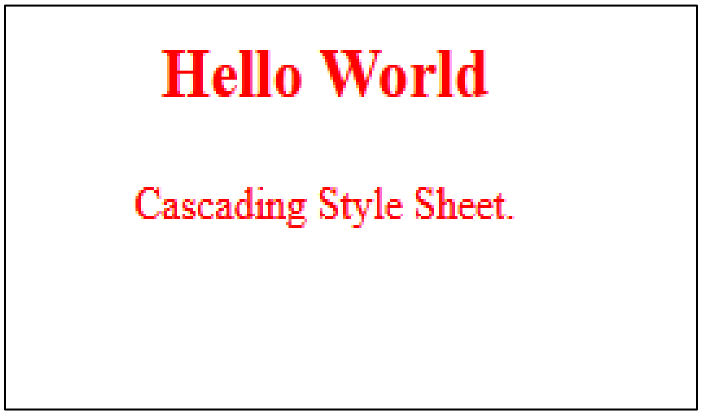
The Class Selector

The class selector selects elements with a specific class attribute. Use dot (.) to select elements with a specific class, followed by the class name.

Example of Class Selector

```
<html>

 <head>

  <style>

   .center

   {

     text-align: center;

     color: red;

   }

  </style>

 </head>

 <body>

  <h2 class="center">Hello World</h2>

  <p class="center">Cascading Style Sheet.</p>

 </body>

</html>
```

Output



Hello World

Cascading Style Sheet.

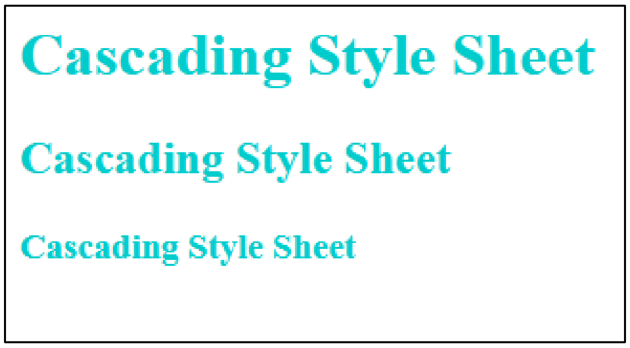
Grouping Selectors

We can apply a style to many selectors. Separate the selectors with a comma.

Example of Grouping Selector

```
<html>

 <head>

  <style>

   h1, h2, h3
   {

     color:#0CC;

   }


  </style>

 </head>

 <body>

  <h1> Cascading Style Sheet </h1>
  <h2> Cascading Style Sheet </h2>

  <h3> Cascading Style Sheet </h3>

 </body>

</html>
```

Output



```
Cascading Style Sheet

Cascading Style Sheet

Cascading Style Sheet
```

Types of CSS

There are three ways we can apply CSS to HTML document –

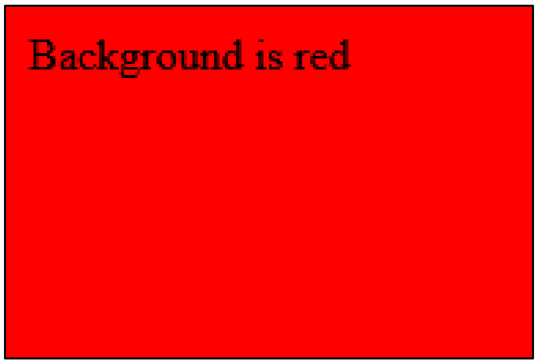
Inline CSS

One way to apply CSS to HTML is by using the **style** attribute.

Example of Inline CSS

```
<html>

 <body style="background-color: #FF0000;">

  <p>Background is red</p>

 </body>

</html>
```

Output


Background is red

Internal CSS

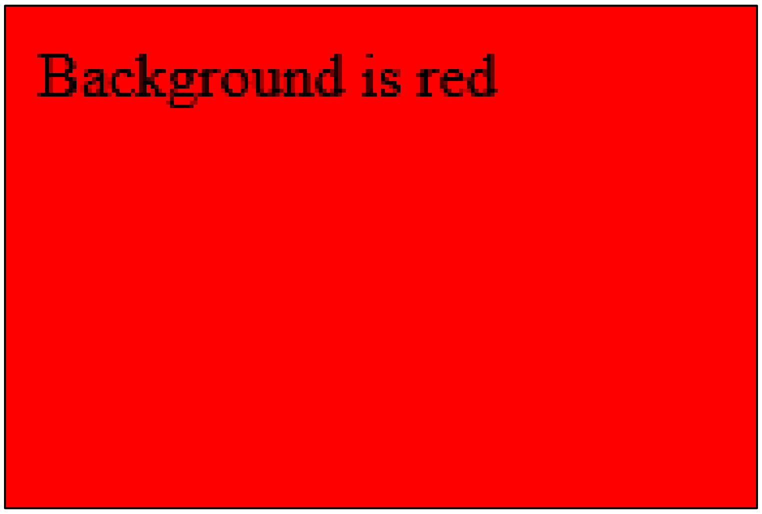
Another way is to include the CSS codes, using the HTML tag **<style>.**

Example Internal CSS

```
<html>

 <head>

  <style>

   body

   {

     background-color: #FF0000;

   }

  </style>

 </head>

 <body>
```

```
   <p>Background is red</p>

  </body>

</html>
```

Output



Background is red

External CSS

In this type of CSS, we link an external css file to HTML document. An external style sheet is a text file with the extension **.css**. We create the link between external CSS and HTML document using **href** attribute. We can reuse this external CSS file in many HTML document.

Example of External CSS

Save this file as **css.html**.

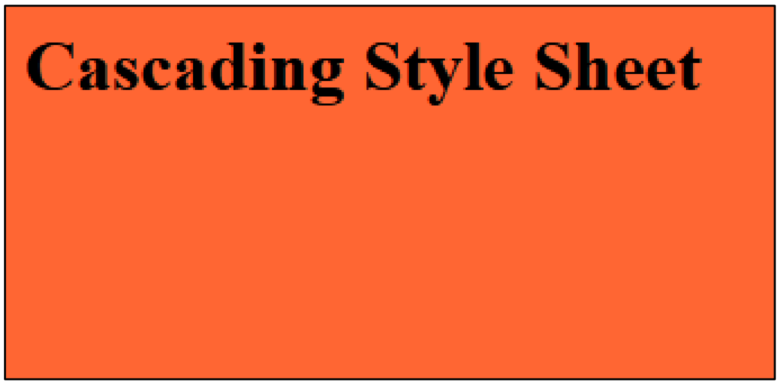
```
<html>

  <head>

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

```
 </head>

 <body>

  <h1>Cascading Style Sheet</h1>

 </body>

</html>
```

Now write the following code and save this file as **style.css**.

```
body

{
  background-color:#F63

}
```

Output



CSS Background

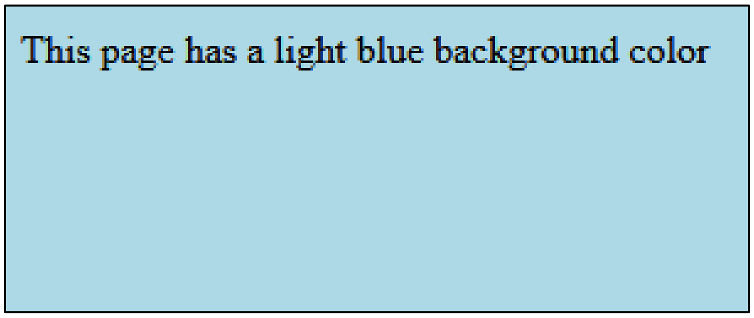
CSS background property is used to define the background effects on any web page. CSS background properties are –

background-color - This attribute demonstrates how to set the background color for an element.

Example of CSS Background

```
<html>
 <head>
  <style>
   body
    {
      background-color: lightblue;
    }
  </style>
 </head>
 <body>
  <p>This page has a light blue background color</p>
 </body>
</html>
```

Output


This page has a light blue background color

background-image - We can set the image as background.

background-repeat - This property used to control the repetition of an image in the background. Some images are repeated horizontally or vertically. The background looks better if the image repeated horizontally. To repeat the image horizontally set the **background-image:repeat-x**. To repeat the image vertically set the **background-image:repeat-y.**

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

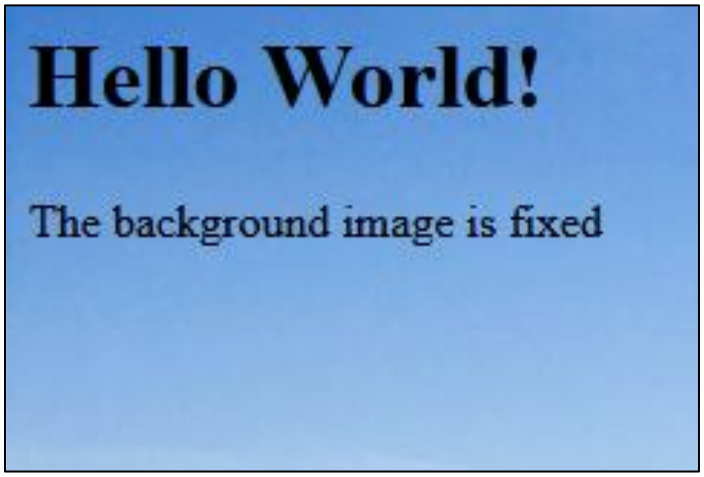
background-position - This property is used to control the image position in the background.

Example of background image

```
<html>

 <head>

  <style>

    body

    {

      background-image:url(Desert.jpg);

      background-repeat: no-repeat;

      background-position: left top;

      background-attachment: fixed;

    }

  </style>

 </head>

 <body>

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

```
   <p>The background image is fixed</p>
  </body>
</html>
```

Output


Hello World!

The background image is fixed

CSS Border

The border properties allow us to specify how the border of the box representing an element should look. CSS border properties are –

border-style - Border style property is used to specify the border type to display on the web page.

border-width - The width can be set as a specific size in px, pt, cm, em, etc or by using pre-defined values - thin, medium, and thick.

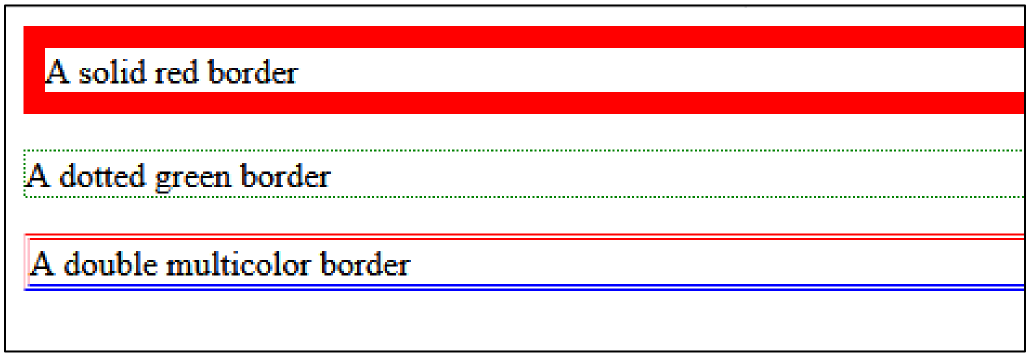
border-color - This property defines the border color.

border-radius - This property is used to add rounded borders to an element.

Example of CSS Border

```
<html>

 <head>

  <style>

    .one

     {

       border-style: solid;

       border-color: red;

        border-width: 10px;


     }

   .two

    {

      border-style: dotted;

      border-color: green;

       border-width: thin;

    }

  .three

   {
      border-style:double;

      border-color: red green blue pink;

   }
 </style>
</head>
<body>
  <p class="one">A solid red border</p>
  <p class="two">A dotted green border</p>
  <p class="three">A double multicolor border</p>
 </body>
</html>
```

Output



```
A solid red border

A dotted green border

A double multicolor border
```

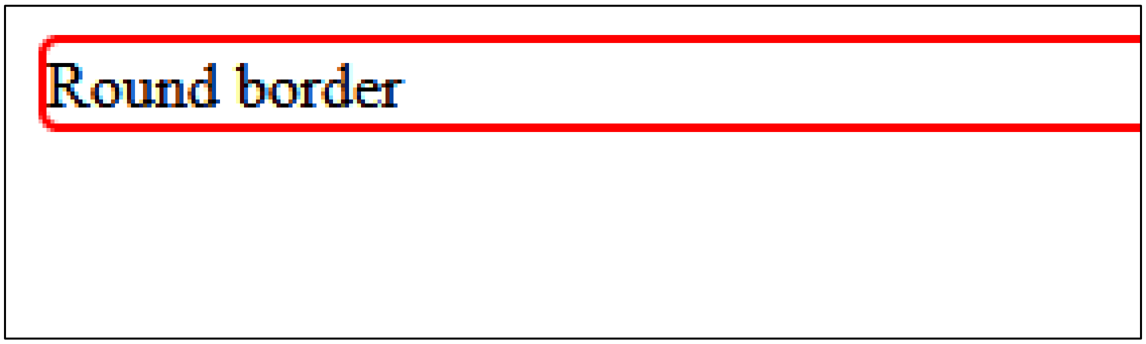
Border Shorthand Property

The border shorthand property is the following individual border properties – width, style and color.

Example of Border Shorthand Property

```
<html>
 <head>
  <style>
  .round
   {
     border: 2px solid red;
     border-radius: 5px;
   }
 </style>
</head>
<body>
  <p class="round">Round border</p>
 </body>
</html>
```

Output


Round border

CSS Margin

The margin property defines the space around an HTML element. The properties are –

margin-top - defines the top margin of an element.

margin-right - defines the right margin of an element.

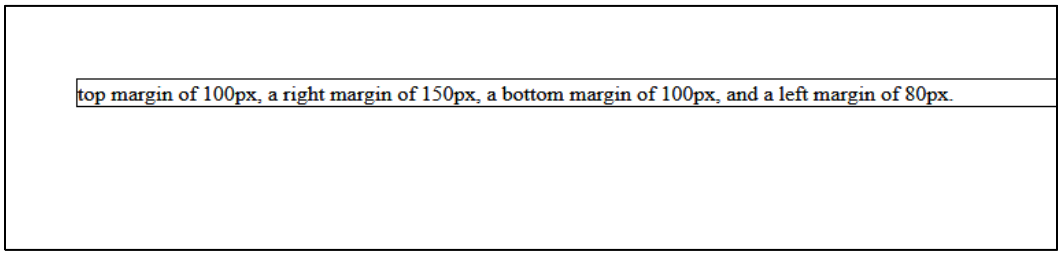
margin-bottom - defines the bottom margin of an element.

margin-left - defines the left margin of an element.

Example of CSS Margin

```
<html>
 <head>
   <style>
     div
     {
       border: 1px solid black;
       margin-top: 100px;
       margin-bottom: 100px;
       margin-right: 150px;
       margin-left: 80px;
     }
  </style>
 </head>
 <body>
   <div>top margin of 100px, a right margin of 150px, a bottom margin
of 100px, and a left margin of 80px.
   </div>
 </body>
</html>
```

Output


top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

Margin - Shorthand Property

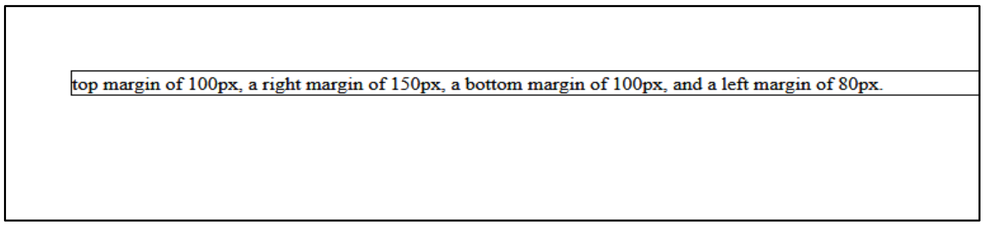
The margin shorthand property is the following individual margin properties – top, right, bottom and left.

Example of Margin - Shorthand Property

```
<html>
 <head>
  <style>
    div
     {
       border: 1px solid black;
       margin: 100px 150px 100px 80px;
     }
  </style>
 </head>
 <body>
```

```
   <div>top margin of 100px, a right margin of 150px, a bottom margin
of 100px, and a left margin of 80px.

  </div>

 </body>

</html>
```

Output



top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

CSS Padding

The CSS padding properties define space around content. The padding clears the area around the content inside the border. The properties are –

padding-top – Defines top padding of an element.

padding-right - Defines right padding of an element.

padding-bottom - Defines bottom padding of an element.

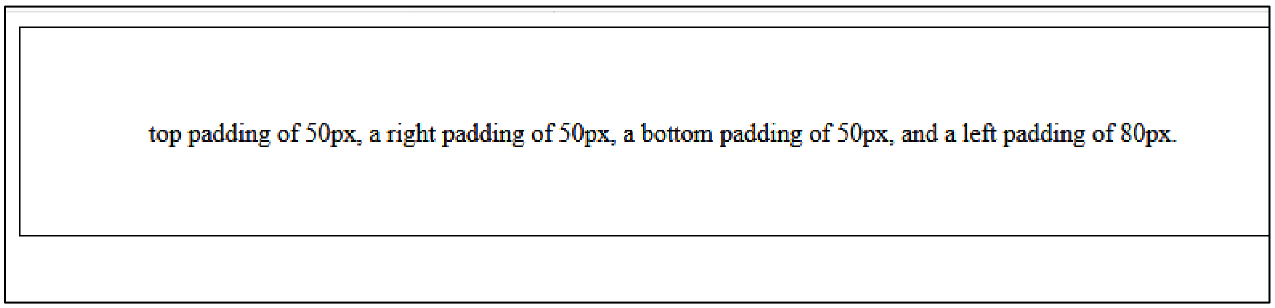
padding-left - Defines left padding of an element.

Example of CSS Padding

```
<html>

 <head>

  <style>

    div
```

```
     {
        border: 1px solid black;
        padding-top: 50px;
        padding-right: 30px;
        padding-bottom: 50px;
        padding-left: 80px;
     }
  </style>
 </head>
 <body>
   <div>top padding of 50px, a right padding of 50px, a bottom padding
of 50px, and a left padding of 80px.
   </div>
 </body>
</html>
```

Output



```
top padding of 50px, a right padding of 50px, a bottom padding of 50px, and a left padding of 80px.
```

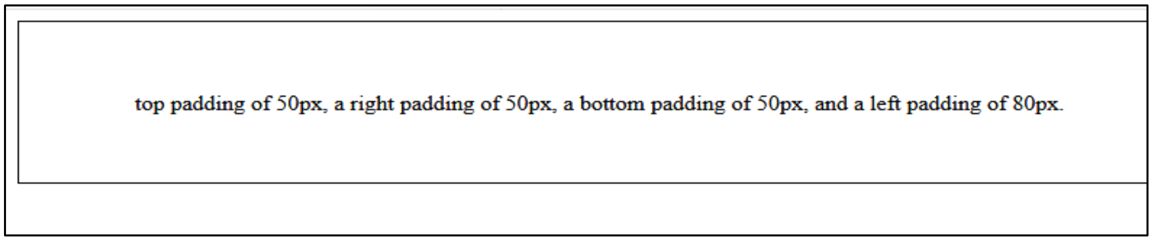
Padding - Shorthand Property

The padding shorthand property is the following individual padding properties – top, right, bottom and left.

Example of Padding - Shorthand Property

```
<html>

 <head>

  <style>

    div

     {

        border: 1px solid black;

        padding: 50px 30px 50px 80px;

     }

  </style>

 </head>

 <body>

   <div>top padding of 50px, a right padding of 30px, a bottom padding
of 50px, and a left padding of 80px.

   </div>

 </body>

</html>
```

Output



```
top padding of 50px, a right padding of 50px, a bottom padding of 50px, and a left padding of 80px.
```

CSS Height and Width

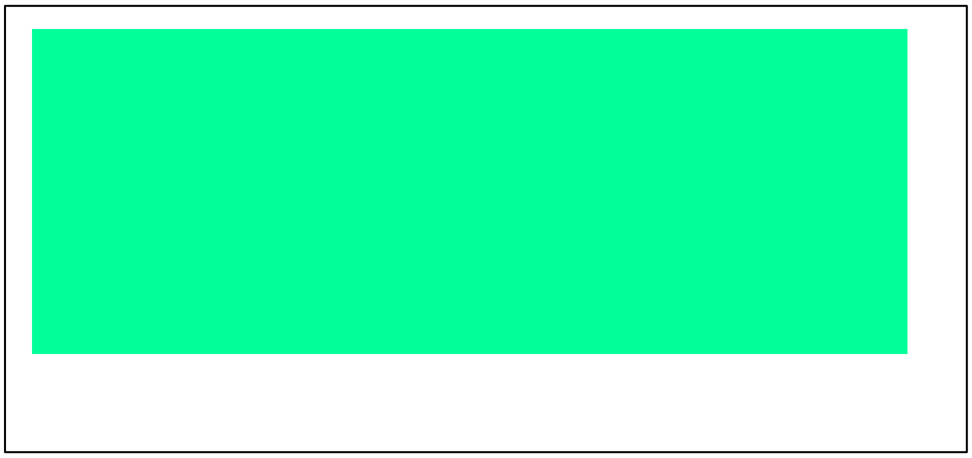
The CSS height and width properties are used to set the height and width of the content. The height and width can be set to auto (this is default) or can be specified in px, cm, percent (%).

Example of CSS Height and Width

```
<html>

 <head>

  <style>

    div

     {

        height: 100px;

        width: 20%;

        background-color:#0F9;

    }

  </style>

 </head>

 <body>
```

```
   <div></div>

 </body>

</html>
```

Output



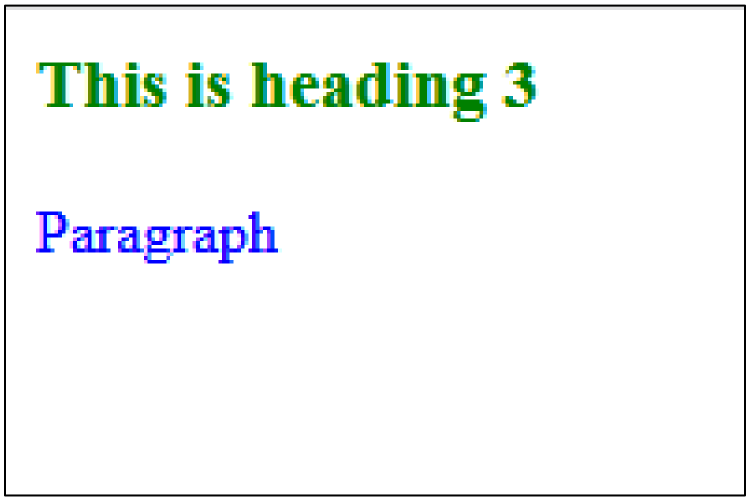
CSS Color

The color property is used to set the text color.

Example of CSS Color

```
<html>

 <head>

  <style>

    body

     {

      color: blue;

    }

    h3
    {
      color: green;
    }
  </style>
 </head>
 <body>
   <h3>This is heading 3</h3>
   <p>Paragraph</p>
 </body>
</html>
```

Output



```
This is heading 3

Paragraph
```

CSS Font

CSS Font property represents the look of texts. The font properties are –

font-family – This property is used to change the font face.

font-style – This property is used to make a font normal, italic or oblique.

font-size – This property is used to increase or decrease the font size in px or em.

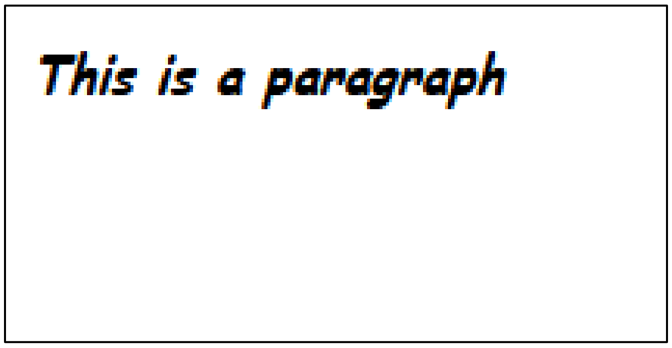
font-weight – This property is used to increase or decrease how bold or light a font appears.

Example of CSS font

```
<html>
 <head>
  <style>
   .f1
     {
        font-family:"Comic Sans MS", cursive;
        font-style:italic;
        font-size:16px;
        font-weight:bold;
     }
  </style>
 </head>
 <body>
```

```
   <p class="f1">This is a paragraph</p>

 </body>

</html>
```

Output



This is a paragraph

CSS Position

The CSS position property is used to set position for HTML element. There are four position values –

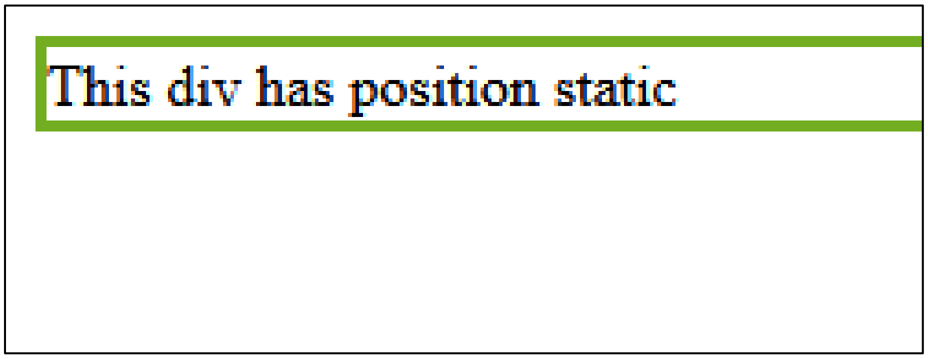
Position: static - HTML elements are positioned static by default. HTML elements are not affected by the top, bottom, left and right properties.

Example Static Position

```
<html>

 <head>

  <style>

   .static
     {

       position: static;

       border: 3px solid #73AD21;
```

```
    }
  </style>
 </head>
 <body>
   <div class="static">
This div element has position static
   </div>
 </body>
</html>
```

Output



```
This div has position static
```

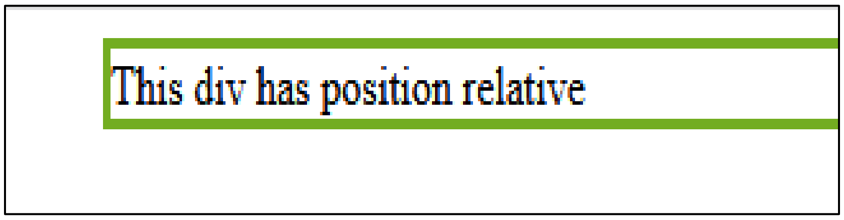
Position: relative – Relative position is positioned relative to its normal position. Top, right, bottom and left properties of a relatively positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap.

Example of Relative Position

```
<html>
 <head>
  <style>
```

```
   .relative
     {
        position: relative;
        left: 30px;
        border: 3px solid #73AD21;
     }
  </style>
 </head>
 <body>
   <div class="relative">
This div has position relative
   </div>
 </body>
</html>
```

Output



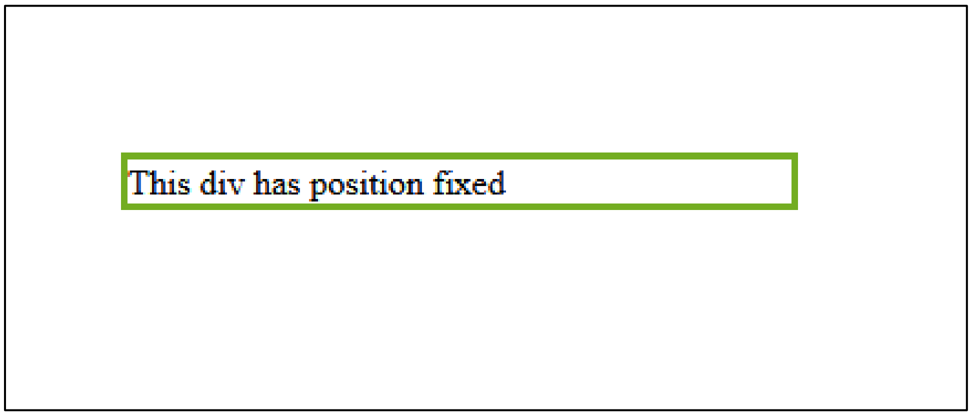
```
This div has position relative
```

Position: fixed – Fixed position is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

Example of Fixed Position

```
<html>

 <head>

  <style>

   .fixed

     {

        position: fixed;

        bottom: 100px;

        right: 100px;

        width: 300px;

        border: 3px solid #73AD21;

     }

  </style>

 </head>

 <body>

   <div class="fixed">

    This div has position fixed

   </div>

 </body>

</html>
```

Output



This div has position fixed

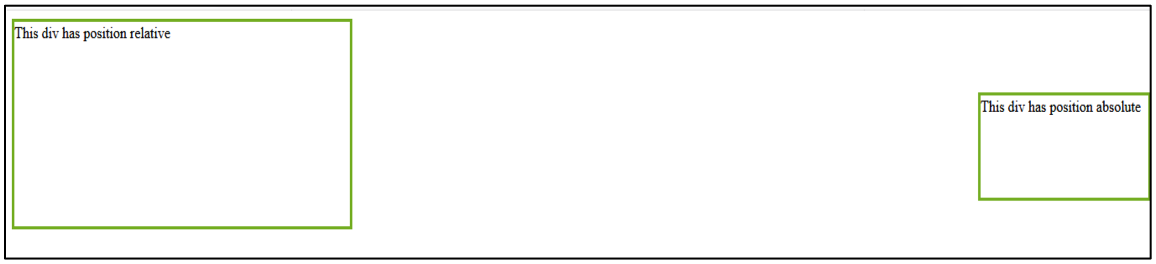
Position: absolute – Position absolute is positioned relative to the nearest positioned ancestor. If an absolute positioned element has no ancestors, it uses the document body and moves along with page scrolling.

Example of Absolute Position

```
<html>

 <head>

  <style>

   .relative

     {


        position: relative;

        width: 400px;

        height: 200px;

        border: 3px solid #73AD21;

     }

   .absolute
```

```
    {
      position: absolute;
      top: 80px;
      right: 0;
      width: 200px;
      height: 100px;
      border: 3px solid #73AD21;
    }
  </style>
 </head>
 <body>
   <div class="relative"> This div has position relative </div>
   <div class="absolute"> This div has position absolute </div>
  </div>
 </body>
</html>
```

Output



This div has position relative

This div has position absolute

Overlapping Elements

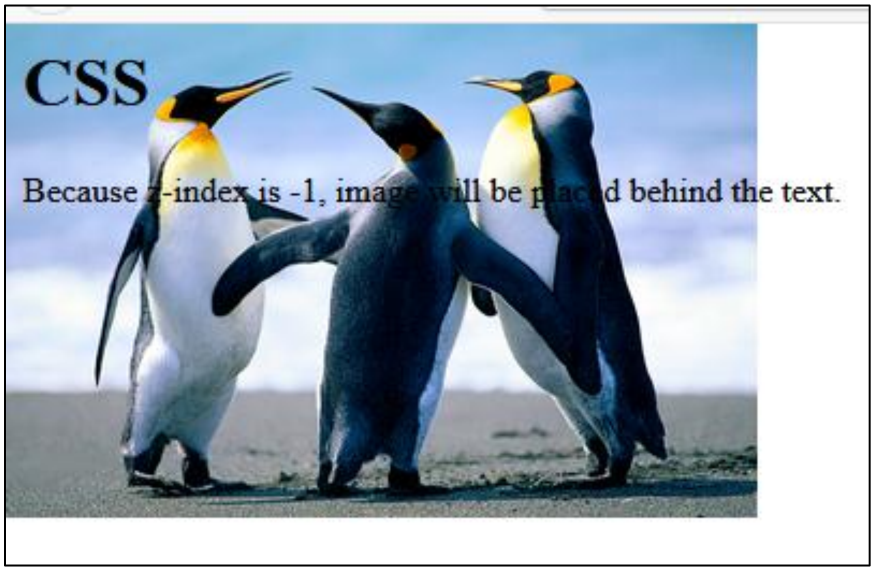
When elements are positioned, they can overlap with other elements. The z-index property specifies which element should be placed in front of or behind the others.

Example of Overlapping Elements

```
<html>
 <head>
  <style>
    img
     {
        position: absolute;
        left: 0px;
        top: 0px;
        z-index: -1;
     }
  </style>
 </head>
 <body>
   <h1>CSS</h1>
   <img src="Penguins.jpg" width="350" height="230">
   <p>Because z-index is -1, image will be placed behind the text.</p>
 </body>
```

```
</html>
```

Output



CSS

Because z-index is -1, image will be placed behind the text.

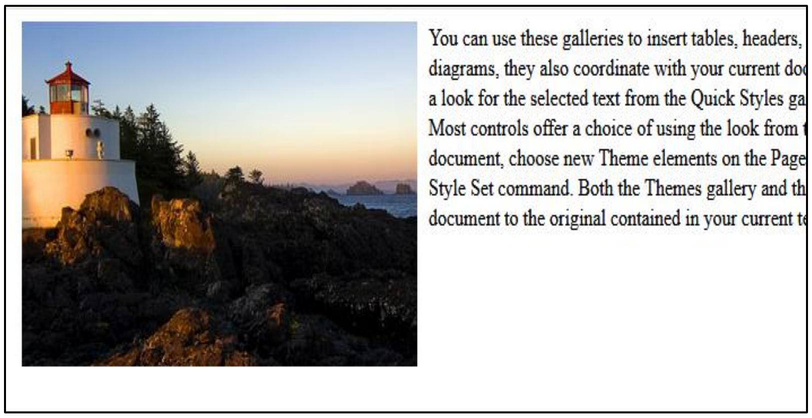
CSS Float

The CSS Float property is used to wrap text around images.

Example of CSS Float

```
<html>

 <head>

  <style>

    img

    {

       float: left;


       margin: 0 10px 10px 10px;
    }

  </style>

 </head>

 <body>

   <p>   <img   src="Lighthouse.jpg"   alt="W3Schools.com"   width="350"
height="230">You  can  use  these  galleries  to  insert  tables,  headers,
footers, lists, cover pages, and other document building blocks.</p>

 </body>

</html>
```

Output



```
You can use these galleries to insert tables, headers,
diagrams, they also coordinate with your current do
a look for the selected text from the Quick Styles ga
Most controls offer a choice of using the look from
document, choose new Theme elements on the Page
Style Set command. Both the Themes gallery and th
document to the original contained in your current t
```

CSS Overflow

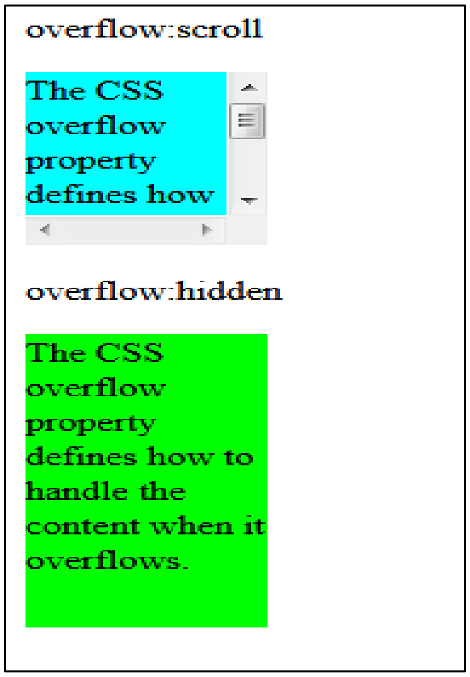
The CSS overflow property defines how to handle the content when it overflows. If we don't set the height of the box, it will grow as large as the content. But if we set a specific height or width of the box and the content inside cannot fit then the CSS overflow property will solve this problem.

Example of CSS Overflow

```
<html>
 <head>
   <style>
    .scroll
      {
         background-color: #00ffff;
         width: 100px;
         height: 100px;
         overflow: scroll;
      }
   .hidden
     {

        background-color: #00FF00;
        width: 100px;
        height: 170px;
        overflow: hidden;
     }
   </style>
 </head>
 <body>
```

```
  <p>overflow:scroll</p>

  <div class="scroll">The CSS overflow property defines how to handle
the content when it overflows.</div>

  <p>overflow:hidden</p>

  <div class="hidden">The CSS overflow property defines how to handle
the content when it overflows.</div>

 </body>

</html>
```

Output



```
overflow:scroll

The CSS
overflow
property
defines how

overflow:hidden

The CSS
overflow
property
defines how to
handle the
content when it
overflows.
```

CSS Display

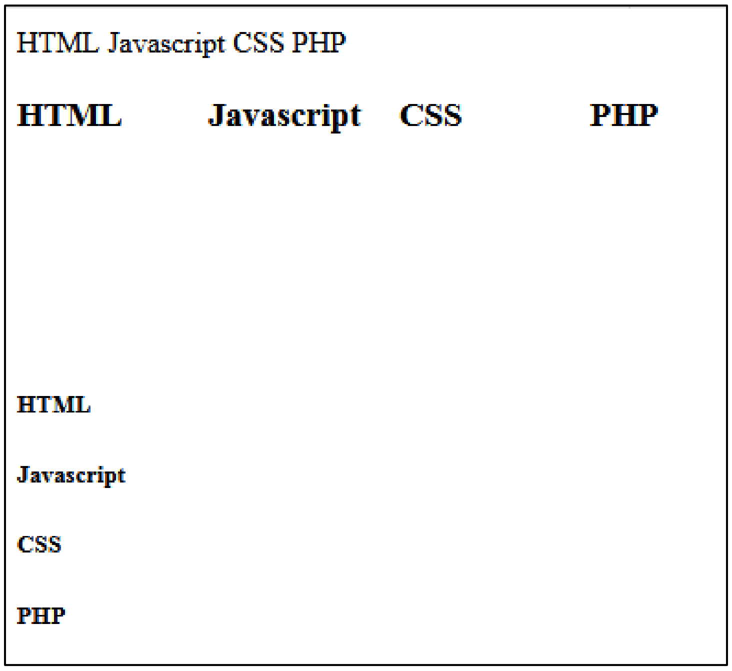
CSS display property is used to control the layout of the element.

Example of CSS Display

```
<html>

 <head>

  <style>

    .line

     {
       display: inline;

     }

    .block

     {

        display: inline-block;

        height:100;

        width:100;

     }

    .n1

     {

        display:none;
     }

    .n2

     {

       display:block;

     }

  </style>
```

```
</head>
<body>
 <p class=line>HTML</p>
 <p class=line>Javascript</p>
  <p class=line>CSS</p>
 <p class=line>PHP</p>
 <br>
 <h3 class=block>HTML</h3>
 <h3 class=block>Javascript</h3>
 <h3 class=block>CSS</h3>
 <h3 class=block>PHP</h3>
 <br>
 <h4 class=n1>HTML</h4>
 <h4 class=n1>Javascript</h4>
 <h4 class=n1>CSS</h4>
 <h4 class=n1>PHP</h4>
 <br>
 <h5 class=n2>HTML</h5>
 <h5 class=n2>Javascript</h5>
 <h5 class=n2>CSS</h5>
 <h5 class=n2>PHP</h5>
 </body>

</html>
```

Output



```
HTML Javascript CSS PHP

HTML        Javascript    CSS          PHP






HTML

Javascript

CSS

PHP
```

CSS Pseudo Classes

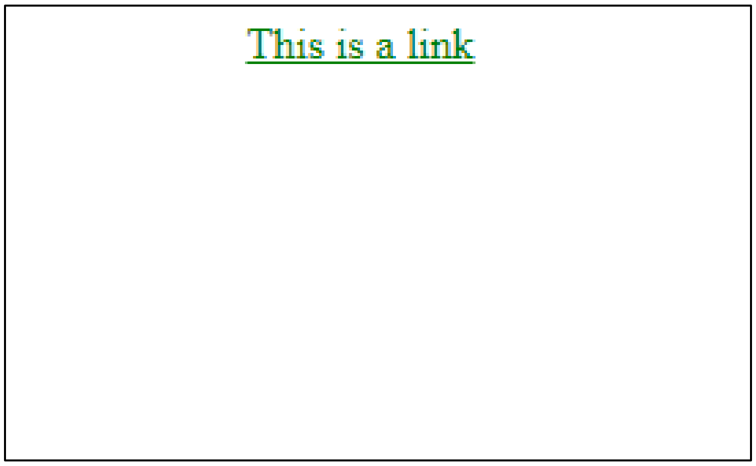
CSS Pseudo classes are used to add special effects to some selectors. Following are the most used pseudo classes –

1	**:link** Use this class to add special style to an unvisited link.
2	**:visited** Use this class to add special style to a visited link.
3	**:hover** Use this class to add special style to an element when you mouse over it.
4	**:active** Use this class to add special style to an active element.

Example of CSS Pseudo Classes

```
<html>
  <head>
    <style>
       a:link
        {
          color: red;
        }
       a:visited
        {
          color: green;
       }
       a:hover
       {
         color: hotpink;
       }
     a:active
      {
         color: blue;
       }
    </style>
  </head>
   <body>
     <a href="https://www.google.co.in" target="_blank">This is a
link</a>
   </body>
</html>
```

Output


This is a link

CSS Opacity

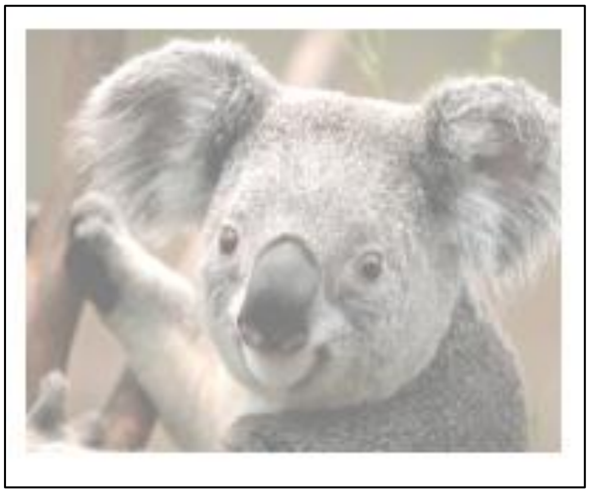
CSS opacity is used to specify transparency of any element. We can use opacity value from 0.0 to 1.0. Keep the opacity value lower to increase the opacity of the element.

Transparent Image

Example of CSS Opacity

```
<html>
  <head>
    <style>
       img
        {
          opacity: 0.5;
        }
    </style>
  </head>
  <body>
    <img src="koala.jpg" alt="Forest" width="190" height="150">
  </body>
</html>
```

Output



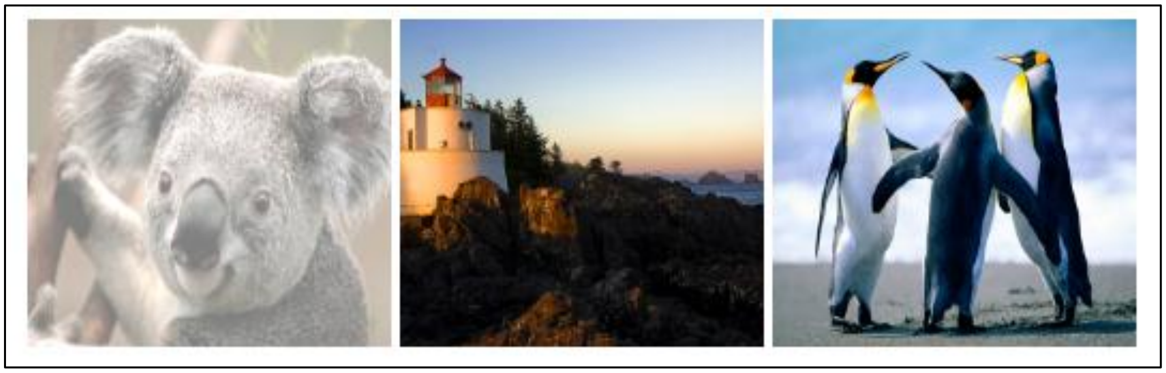
Transparent Hover Effect

Example of Transparent Hover Effect

```
<html>
  <head>
    <style>
      img:hover
        {
          opacity: 0.5;
        }
     </style>
    </head>
  <body>
    <img src="koala.jpg" alt="Forest" width="170" height="150">
    <img src="lighthouse.jpg" alt="Mountains" width="170"
height="150">
    <img src="penguins.jpg" alt="Italy" width="170" height="150">
  </body>
</html>
```

Output

This will change the opacity of these pictures on mouse over.

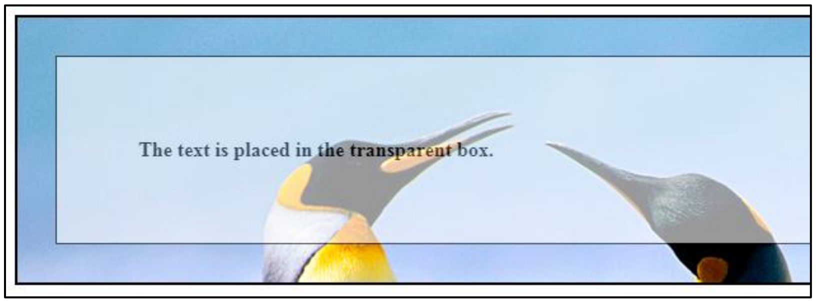


Text in Transparent Box

Example of Text in Transparent Box

```
<html>
  <head>
    <style>

    .background
       {
         background: url(penguins.jpg);
         border: 2px solid black;
       }

    .transbox
       {
         margin: 30px;
         background-color: #ffffff;
         border: 1px solid black;
         opacity: 0.6;
       }

     .text
       {
         margin: 5%;
         font-weight: bold;
         color: #000000;
       }
    </style>
 </head>
 <body>
    <div class="background">
      <div class="transbox">
       <p class="text">The text is placed in the transparent box.</p>
    </div>
  </div>
 </body>
</html>
```

Output



CSS Navigation Bar

With the help of CSS navigation bar we can make HTML menus into good looking navigation bars.

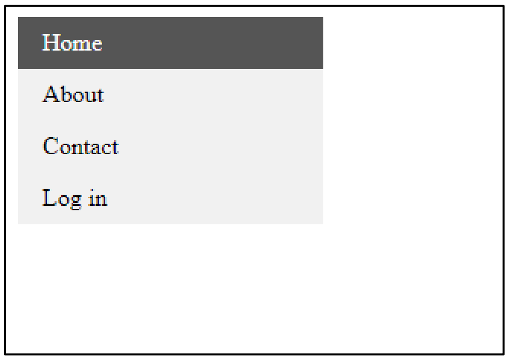
Example of Vertical Navigation Bar

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

```
            width: 200px;
            background-color: #f1f1f1;
          }
        a
         {
            display: block;
            color: #000;
            padding: 8px 16px;
            text-decoration: none;
         }
       a:hover
         {
           background-color: #555;
           color: white;
         }
      </style>
   </head>
   <body>
     <ul>
       <li><a href="#Home">Home</a></li>
       <li><a href="#About">About</a></li>
       <li><a href="#Contact">Contact</a></li>
       <li><a href="#Log in">Log in</a></li>
     </ul>
   </body>
</html>
```

We use href="#" for test links. In a web site this would be URLs.

Output

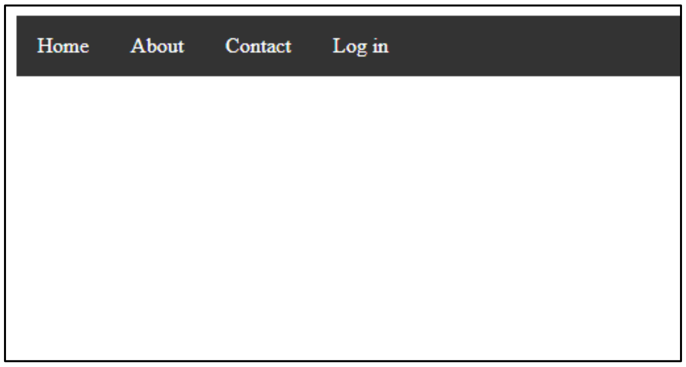


Example of Horizontal Navigation Bar

```
<html>
  <head>
    <style>
        ul
         {
            list-style-type: none;
            margin: 0;
            padding: 0;
            overflow: hidden;
            background-color: #333;
         }
        li
         {
            float: left;
         }
         a
          {
            display: block;
            color: white;
```

```
              text-align: center;
              padding: 14px 16px;
              text-decoration: none;
            }
         a:hover
            {
              background-color: #111;

            }
        </style>
 </head>
 <body>
   <ul>
     <li><a href="#home">Home</a></li>
     <li><a href="#news">About</a></li>
     <li><a href="#contact">Contact</a></li>
     <li><a href="#about">Log in</a></li>
   </ul>
 </body>
</html>
```

Output



CSS Gradients

CSS gradient is used to display smooth transition between two or more colors. There are two types of CSS gradient –

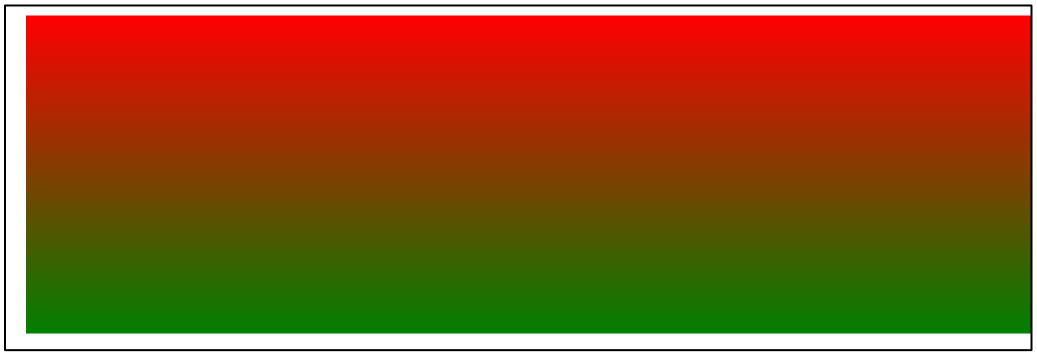
Linear Gradients

The CSS linear gradient goes up/down/left/right and diagonally. We have to define two or more colors to create linear gradient. Starting point and direction can also be added along with the gradient effect.

Example of Top to Bottom Linear Gradient

```
<html>
  <head>
    <style>
      #grad1
        {
          height: 200px;
          background: linear-gradient(red, green);
        }
    </style>
  </head>
  <body>
    <div id="grad1"></div>
  </body>
</html>
```

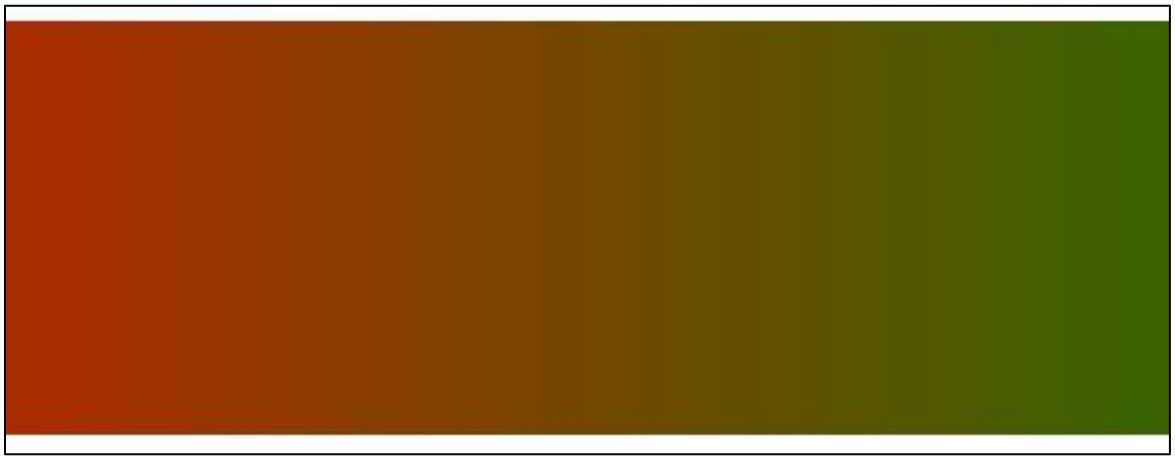
Output



Example of Left to Right Linear Gradient

```
<html>
  <head>
    <style>
      #grad1
        {
          height: 200px;
          background: linear-gradient(to right, red , green);
        }
     </style>
  </head>
  <body>
    <div id="grad1"></div>
  </body>
</html>
```

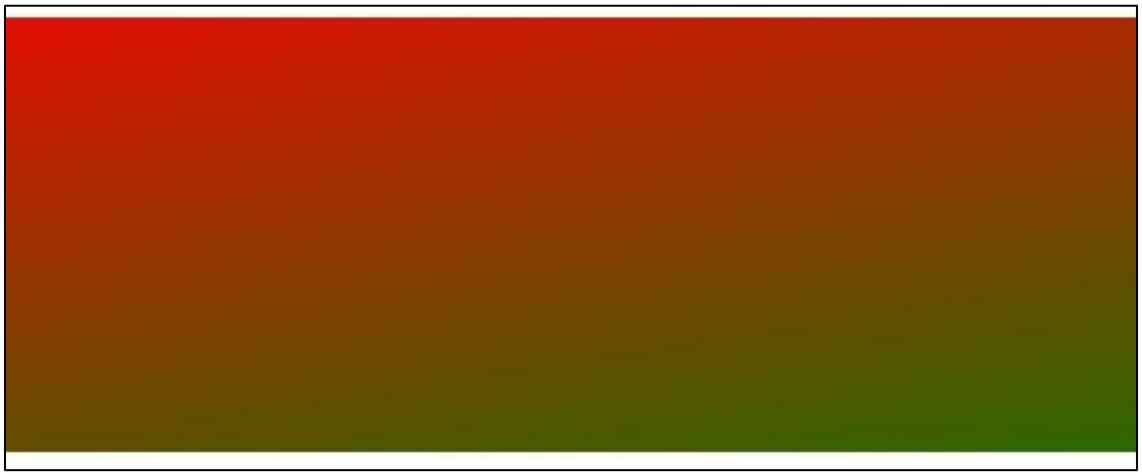
Output



Example of Diagonal Linear Gradient

```
<html>
  <head>
    <style>
      #grad1
        {
          height: 200px;
          background: linear-gradient(to bottom right, red , green);
        }
    </style>
  </head>
  <body>
   <div id="grad1"></div>
  </body>
</html>
```

Output



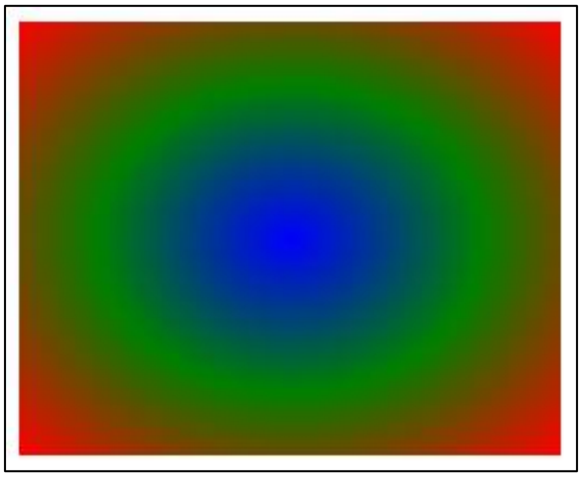
Radial Gradients

We have to define two or more colors to create radial gradients. It is defined by its corner.

Example of Evenly Spaced Colors

```
<html>
  <head>
    <style>
      #grad1
        {
          height: 200px;
          width: 250px;
          background: radial-gradient(blue, green, red);
        }
    </style>
  </head>
  <body>
    <div id="grad1"></div>
  </body>
</html>
```

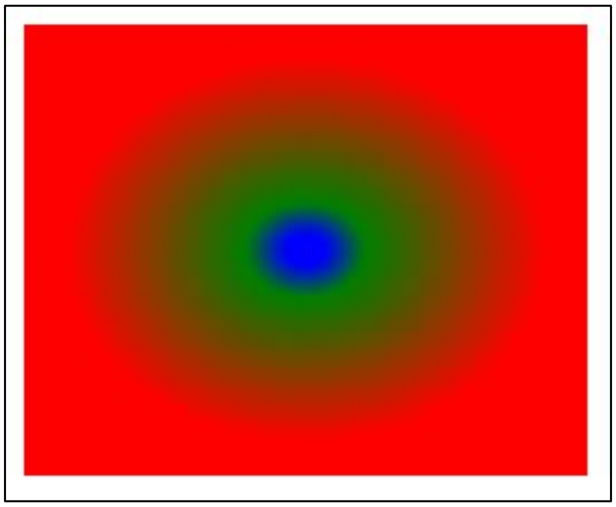
Output



Example of Differently Spaced Color

```
<html>
  <head>
    <style>
      #grad1
        {
          height: 200px;
          width: 250px;
          background: radial-gradient(blue 5%, green 15%, red 60%);
}
    </style>
  </head>
  <body>
    <div id="grad1"></div>
  </body>
</html>
```

Output



CSS Shadow

With the help of CSS shadow we can add shadows to text and to elements. There are two types of shadow properties –

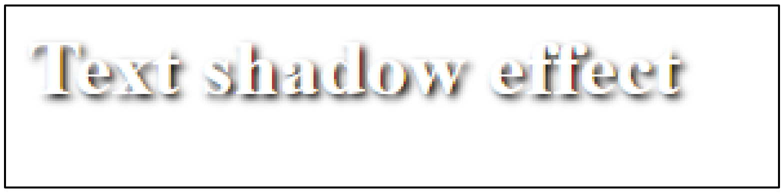
Text Shadow Property

Text shadow property applies shadow to text.

Example of Text Shadow Property

```
<html>
  <head>
    <style>
      h1
       {
         color: white;
         text-shadow: 2px 2px 4px #000000; /* x-offset, y-offset,
blur, shadow-color*/
       }
    </style>
  </head>
  <body>
    <h1>Text shadow effect</h1>
  </body>
</html>
```

Output



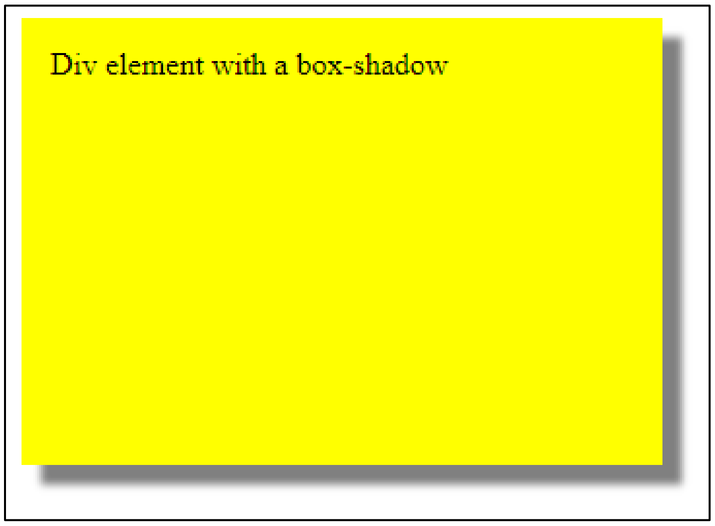
Box Shadow Property

Box shadow property applies shadow to elements.

Example of Box Shadow Property

```
<html>
  <head>
    <style>
      div
       {
          width: 300px;
          height: 200px;
          padding: 15px;
          background-color: yellow;
          box-shadow: 10px 10px 5px grey; /* x-offset, y-offset, blur,
shadow-color*/
        }
    </style>
  </head>
  <body>
    <div>Div element with a box-shadow</div>
  </body>
</html>
```

Output


Div element with a box-shadow

CSS Animation

CSS Animation property is used to create animations on the webpage. An animation makes an element change from one style to another style. We can add as many as. We can specify the changes in percentage. 0% specifies the start of the animation and 100% specify its completion.

@keyframes Rule

The animation is created in the **@keyframe** rule. It is used to control the intermediate steps in a CSS animation sequence. To get an animation to work, we must bind the animation to an element.

Animation Properties

Property	Description
@keyframes	It is used to specify the animation.
animation	This is a shorthand property, used for setting all the properties, except the animation-play-state and the animation-fill- mode property.
animation-delay	It specifies when the animation will start.
animation-direction	It specifies if or not the animation should play in reserve on alternate cycle.
animation-duration	It specifies the time duration taken by the animation to complete one cycle.
animation-fill-mode	it specifies the static style of the element. (when the animation is not playing)
animation-iteration-count	It specifies the number of times the animation should be played.
animation-play-state	It specifies if the animation is running or paused.
animation-name	It specifies the name of @keyframes animation.
animation-timing-function	It specifies the speed curve of the animation.

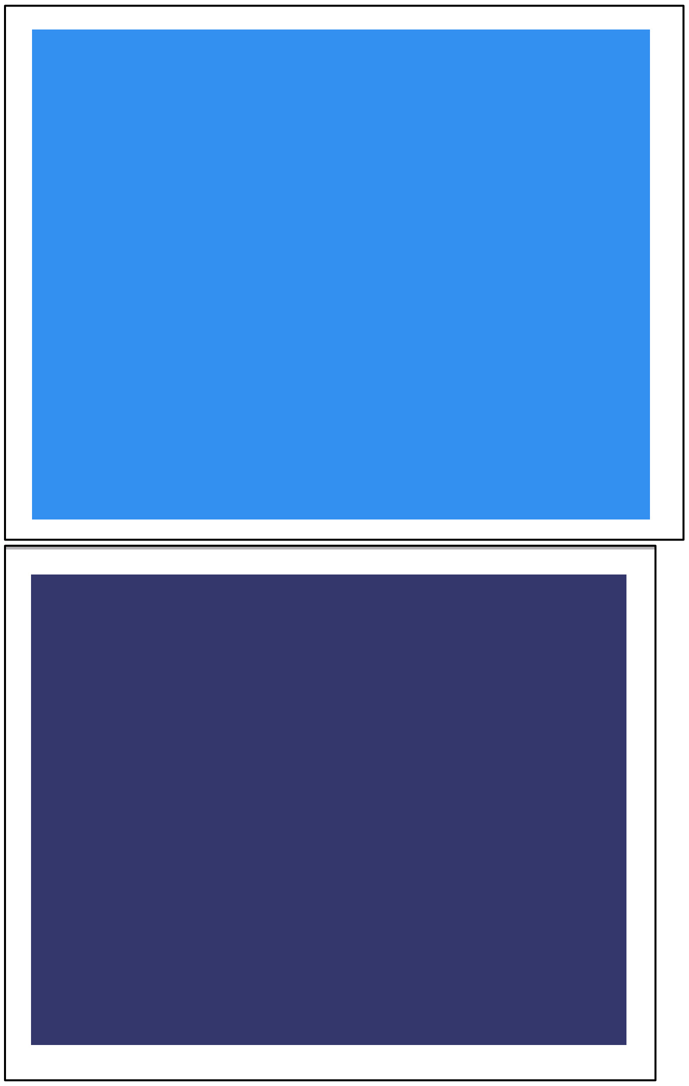
Example of Changing Background Color

```
<html>
```

```
  <head>
    <style>
      .box1
        {
           width:190px;
           height:150px;
           background-color:#336;
           animation-name:anim;
          -webkit-animation-name:anim; /*for old version chrome &
safari*/
           animation-duration:4s;
           animation-iteration-count:infinite;
           animation-delay:1s
        }
     @keyframes anim  /*animation name as per my choice*/
        {
           from{background-color:#336}
           to{background-color:#39F}
        }
    </style>
  </head>
  <body>
   <div class="box1"></div>
  </body>
```

```
</html>
```

Output

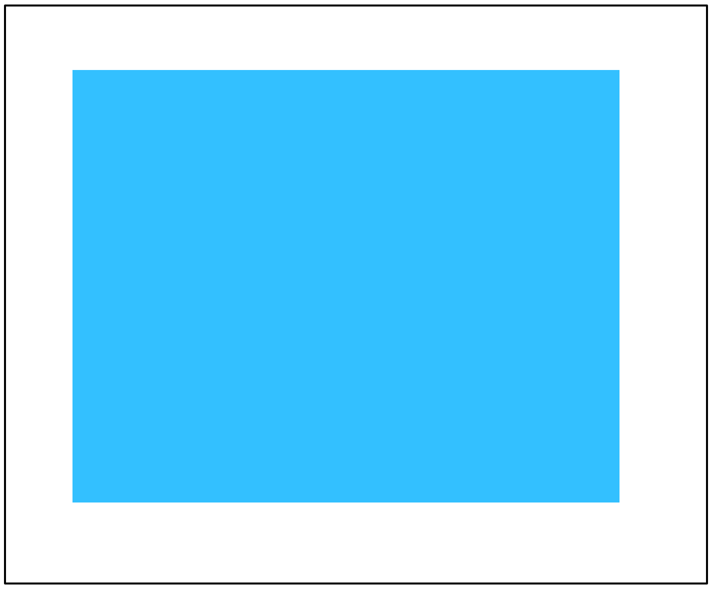


Example of Moving box

```
<html>
  <head>
    <style>
      .box1
```

```
        {
            position:absolute;
            width:190px;
            height:150px;
            background-color:#30F;
            animation-name:anim;
           -webkit-animation-name:anim;
            animation-duration:8s;
            animation-iteration-count:infinite;
         }
      @keyframes anim
         {
            0%{position:absolute; top:150px; left:0px; background-
color:#30F}
            25%{position:absolute; top:150px; left:300px; background-
color:#36F}
            50%{position:absolute; top:250px; left:300px; background-
color:#3CF}
            75%{position:absolute; top:250px; left:0px; background-
color:#3F6}
            100%{position:absolute; top:150px; left:0px; background-
color:#30F}
}
     </style>
  </head>
  <body>
    <div class="box1"></div>
  </body>
</html>
```

Output



CSS Transitions

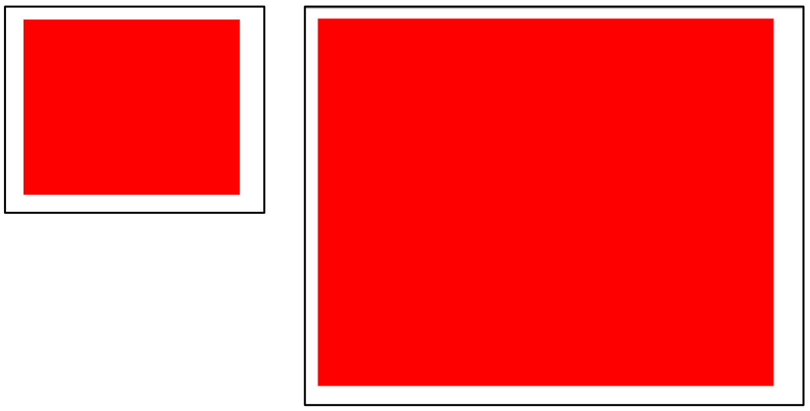
CSS transitions allow us to change property values from one value to another. To create a transition effect, we must define two things - the CSS property we want to add an effect to and the duration of the effect. If the duration is not specified, the transition will have no effect because the default value is 0.

Example of Transition effect for width and height property

```
<html>
  <head>
    <style>
```

```
      div
       {
          width: 100px;
          height: 100px;
          background: red;
          transition: width 2s, height 4s;
      }
     div:hover
      {
         width: 300px;
         height: 300px;
      }
   </style>
 </head>
 <body>
   <div></div>
 </body>
</html>
```

Output



Transition-timing-function Property

transition-timing-function property defines the transition effect's speed curve. This property can have following values –

ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)

linear - specifies a transition effect with the same speed from start to end.

ease-in - specifies a transition effect with a slow start.

ease-out - specifies a transition effect with a slow end.

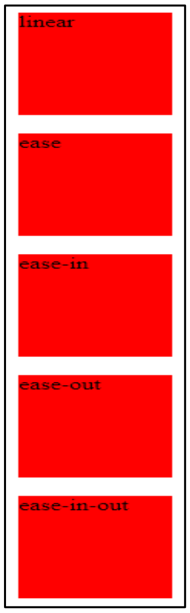
ease-in-out - specifies a transition effect with a slow start and end.

Example of Transition-timing-function Property

```
<html>
  <head>
    <style>
      div
       {
          width: 100px;
          height: 100px;
          background: red;
          transition: width 2s;
       }
     #div1 {transition-timing-function: linear;}
     #div2 {transition-timing-function: ease;}
     #div3 {transition-timing-function: ease-in;}
     #div4 {transition-timing-function: ease-out;}
     #div5 {transition-timing-function: ease-in-out;}
```

```
        div:hover
         {
            width: 300px;
         }
      </style>
    </head>
    <body>
      <div id="div1">linear</div><br>
      <div id="div2">ease</div><br>
      <div id="div3">ease-in</div><br>
      <div id="div4">ease-out</div><br>
      <div id="div5">ease-in-out</div><br>
    </body>
  </html>
```

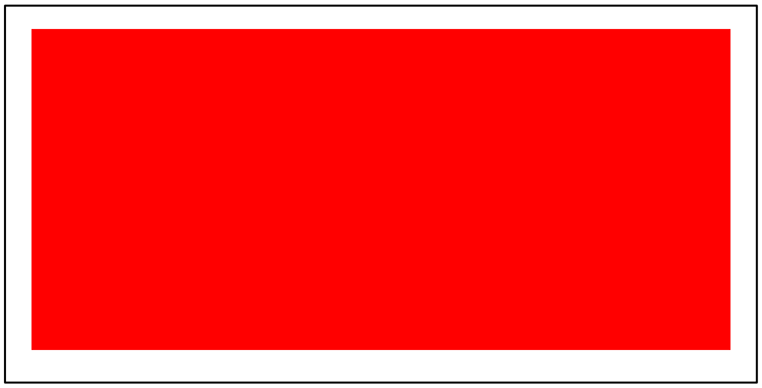
Output



Example of Using all the properties

```
<html>
  <head>
    <style>
      div
       {
          width: 100px;
          height: 100px;
          background: red;
          transition-duration: 2s;
          transition-timing-function: linear;
          transition-delay: 1s;
      }
     div:hover
      {
          width: 300px;
      }
   </style>
 </head>
  <body>
    <div></div>
  </body>
</html>
```

Output



CSS Transforms

This transform property allows us to translate, rotate, scale, and skews elements. Transformation is an effect that changes shape, size and position. There are two types of transformation –

2D Transforms

The CSS 2D transforms are used to re change the structure of the element. Following is a list of 2D transforms methods –

translate(x,y) - It is used to transform the element along X-axis and Y-axis.

translateX(n) - It is used to transform the element along X-axis.

translateY(n) - It is used to transform the element along Y-axis.

rotate() - It is used to rotate the element on the basis of an angle.

scale(x,y) - It is used to change the width and height of an element.

scaleX(n) - It is used to change the width of an element.

scaleY(n) - It is used to change the height of an element.

skew() - it is used to skew an element along with X-axis and Y.

skewX() - It defines the skew transforms along with X-axis.

skewY() - It defines the skew transforms along with Y-axis.

matrix() - It defines matrix transforms.

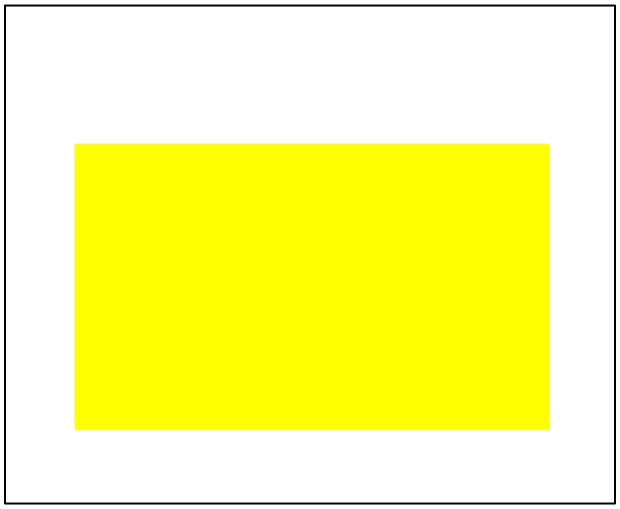
translate() Method

The CSS translate() method is used to move an element from its current position according to the given parameters i.e. X-axis and Y-axis.

Example of translate() Method

```
<html>
  <head>
    <style>
      div
       {
          width: 250px;
          height: 150px;
          background-color: yellow;
          transition-duration:2s;
       }
      div:hover
       {
          transform: translate(50px,100px);
       }
    </style>
  </head>
   <body>
      <div>
     </div>
  </body>
</html>
```

Output



rotate() Method

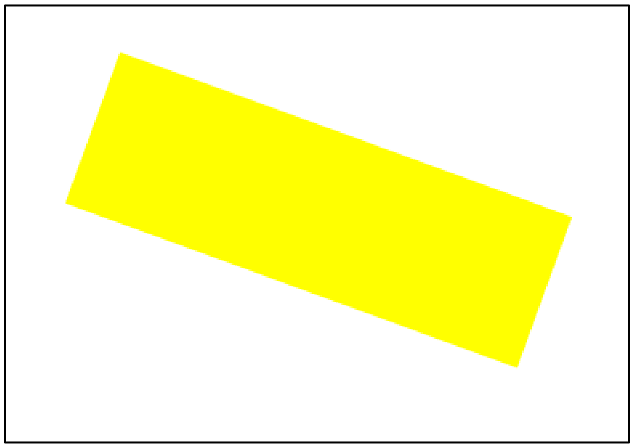
The CSS rotate() method is used to rotate an element clockwise or anti-clockwise according to the given degree.

Example of rotate() Method

```
<html>
  <head>
    <style>
      div
```

```
        {
           width: 300px;
           height: 100px;
           background-color: yellow;
           transition-duration:2s;
        }
      div:hover
       {
          transform: rotate(20deg);
       }
    </style>
  </head>
  <body>
    <div>
    </div>
  </body>
</html>
```

Output



scale() Method

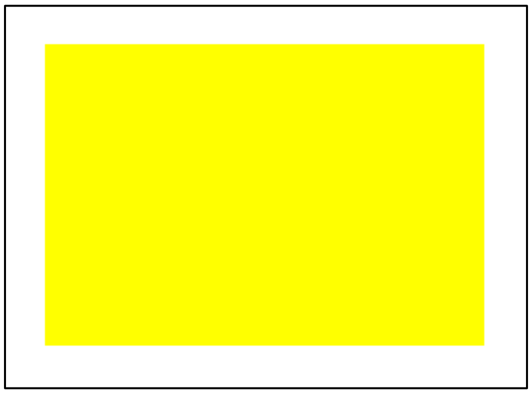
The CSS scale() method is used to increase or decrease the size of an element according to the given width and height.

Example of scale() Method

```
<html>
  <head>
    <style>
      div
       {
          margin: 150px;
         width: 200px;
         height: 100px;
         background-color: yellow;
         transition-duration:2s;
      }
    div:hover
     {
        transform: scale(2,3);
     }
   </style>
  </head>
  <body>
    <div>
    </div>
  </body>
```

```
</html>
```

Output



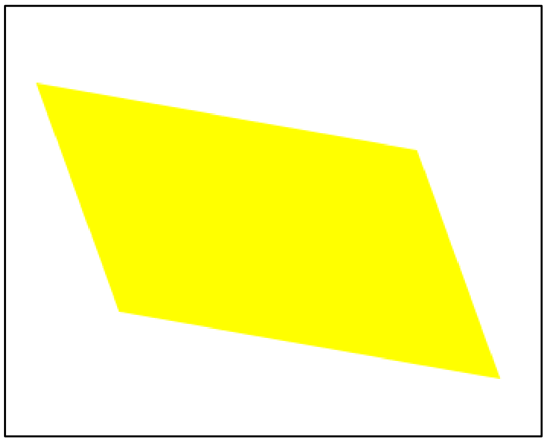
skew() Method

The skew() method skews an element along the X and Y-axis by the given angles.

Example of skew() Method

```
<html>
  <head>
    <style>
      div
       {
           width: 250px;
           height: 150px;
           background-color: yellow;
           transition-duration:2s;
      }
    div:hover
     {
            transform: skew(20deg,10deg);
         }
      </style>
   </head>
   <body>
      <div>
      </div>
   </body>
</html>
```

Output



matrix() Method

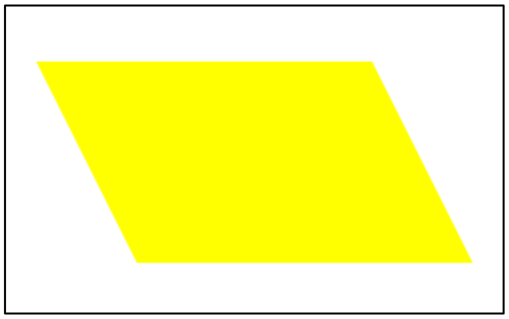
The matrix() method combines all the 2D transform methods into one. The matrix() method take six parameters, containing mathematic functions, which allows you to rotate, scale, translate, and skew elements. The parameters are matrix (scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY()).

Example of matrix() Method

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

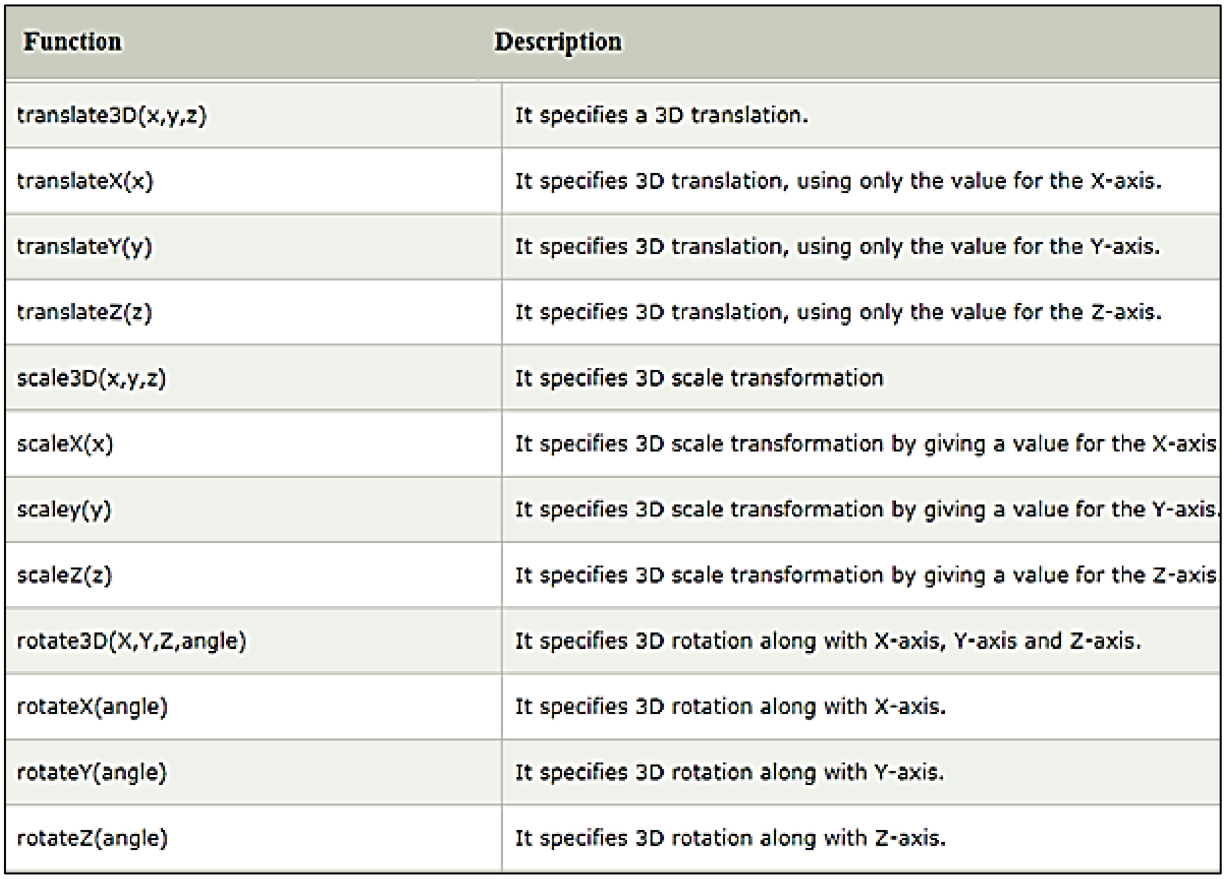
```
    <style>
      div
       {
          width: 250px;
         height: 150px;
         background-color: yellow;
         transition-duration:2s;
    }
   div:hover
    {
        transform: matrix(1, 0, 0.5, 1, 150, 0);
    }
  </style>
 </head>
 <body>
   <div>
   </div>
 </body>
</html>
```

Output



3D Transforms

The CSS 3D transforms allows us to move an element to X-axis, Y-axis and Z-axis. Following is a list of 3D transforms methods –



Function	Description
translate3D(x,y,z)	It specifies a 3D translation.
translateX(x)	It specifies 3D translation, using only the value for the X-axis.
translateY(y)	It specifies 3D translation, using only the value for the Y-axis.
translateZ(z)	It specifies 3D translation, using only the value for the Z-axis.
scale3D(x,y,z)	It specifies 3D scale transformation
scaleX(x)	It specifies 3D scale transformation by giving a value for the X-axis
scaley(y)	It specifies 3D scale transformation by giving a value for the Y-axis.
scaleZ(z)	It specifies 3D scale transformation by giving a value for the Z-axis
rotate3D(X,Y,Z,angle)	It specifies 3D rotation along with X-axis, Y-axis and Z-axis.
rotateX(angle)	It specifies 3D rotation along with X-axis.
rotateY(angle)	It specifies 3D rotation along with Y-axis.
rotateZ(angle)	It specifies 3D rotation along with Z-axis.

rotateX() Method

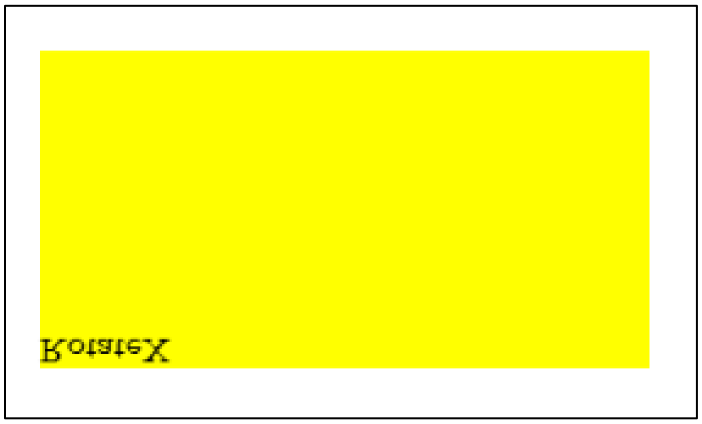
The rotateX() method rotates an element around its X-axis according to given degree.

Example of rotateX() Method

```
<html>
  <head>
    <style>
      div
```

```
        {
           width: 250px;
           height: 150px;
           background-color: yellow;
           transition-duration:2s;
        }
      div:hover
       {
          transform: rotateX(150deg);
       }
    </style>
  </head>
  <body>
    <div>RotateX</div>
  </body>
</html>
```

Output



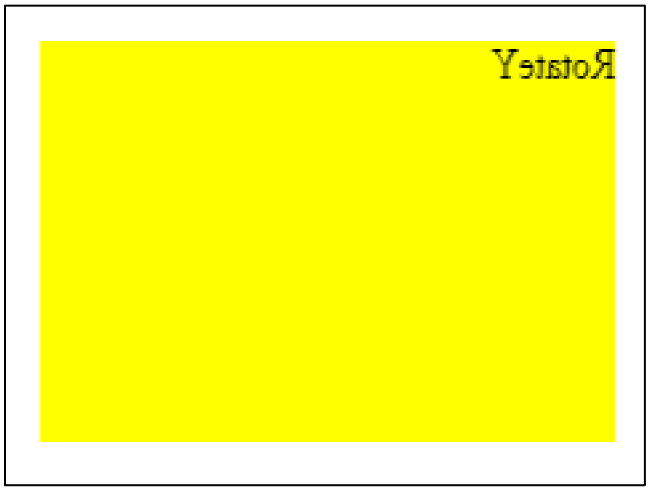
rotateY() Method

The rotateY() method rotates an element around its X-axis according to given degree.

Example of rotateY() Method

```
<html>
  <head>
    <style>
      div
       {
          width: 250px;
          height: 150px;
          background-color: yellow;
          transition-duration:2s;
       }
     div:hover
      {
         transform: rotateY(150deg);
      }
</style>
</head>
<body>
  <div>RotateY</div>
</body>
</html>
```

Output



rotateZ() Method

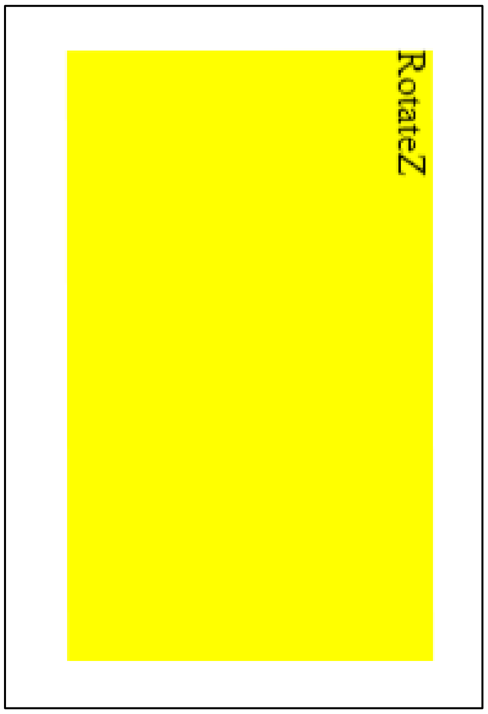
The rotateZ() method rotates an element around its X-axis according to given degree.

Example

```
<html>
  <head>
    <style>
      div
       {
           width: 250px;
           height: 150px;
           background-color: yellow;
           transition-duration:2s;


       }
     div:hover
```

```
        {
           transform: rotateZ (90deg);
        }
     </style>
   </head>
   <body>
     <div>RotateZ</div>
   </body>
</html>
```

Output

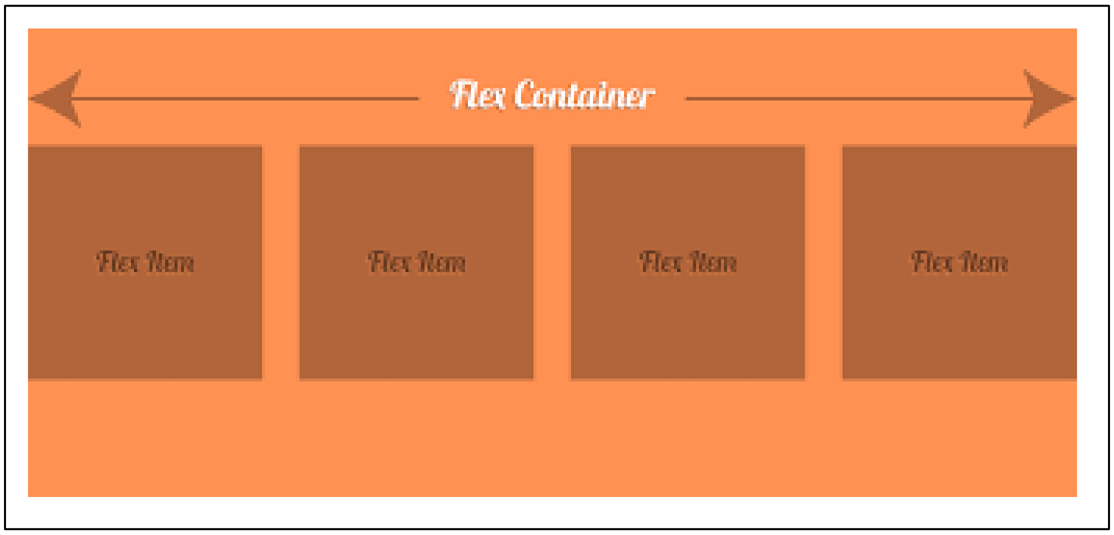


CSS Flexbox

CSS Flexible boxes also known as CSS Flexbox. It is a new layout mode in CSS. The CSS flexbox is used to make the elements behave predictably when they are used with different screen sizes and different display devices. It is mainly used to make CSS capable to change its items width and height to fit for all available spaces. It is preferred over block model. The CSS flexbox contains flex containers and flex items.

Flex container - It specifies the properties of the parent. It is declared by setting the display property of an element to either flex or inline-flex.

Flex items - It specifies properties of the children. There may be one or more flex items inside a flex container.



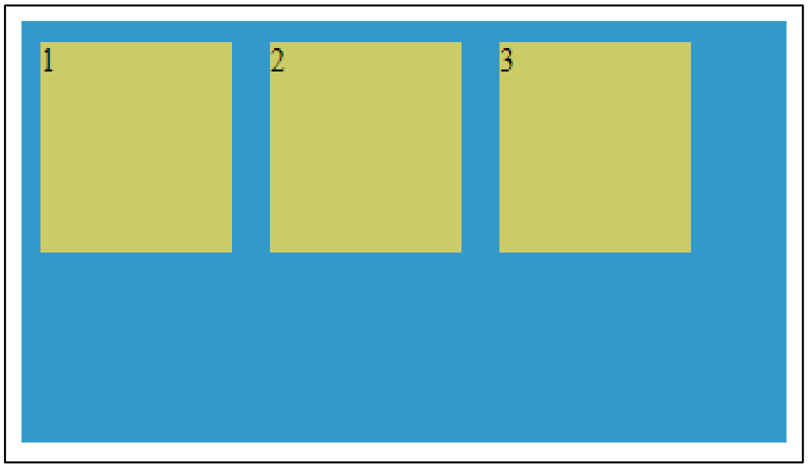
Flexbox Properties

property	description
display	it is used to specify the type of box used for an html element.
flex-direction	it is used to specify the direction of the flexible items inside a flex container.
justify-content	it is used to align the flex items horizontally when the items do not use all available space on the main-axis.
align-items	it is used to align the flex items vertically when the items do not use all available space on the cross-axis.
flex-wrap	it specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line.
align-content	it is used to modify the behavior of the flex-wrap property. it is similar to align-items, but instead of aligning flex items, it aligns flex lines.
flex-flow	it specifies a shorthand property for flex-direction and flex-wrap.
order	it specifies the order of a flexible item relative to the rest of the flex items inside the same container.
align-self	it is used on flex items. it overrides the container's align-items property.
flex	it specifies the length of a flex item, relative to the rest of the flex items inside the same container.

Example of Flexbox Properties

```
<html>
  <head>
    <style>
      .flex-container
        {
           display: flex;
           width: 400px;
           height: 200px;
           background-color:#39C;
       }
    .flex-item
      {
          background-color:#CC6;
          width: 100px;
          height: 100px;
          margin: 10px;
      }
   </style>
 </head>
 <body>
   <div class="flex-container">
   <div class="flex-item">1</div>
    <div class="flex-item">2</div>
    <div class="flex-item">3</div>
   </div>
 </body>
</html>
```

Output

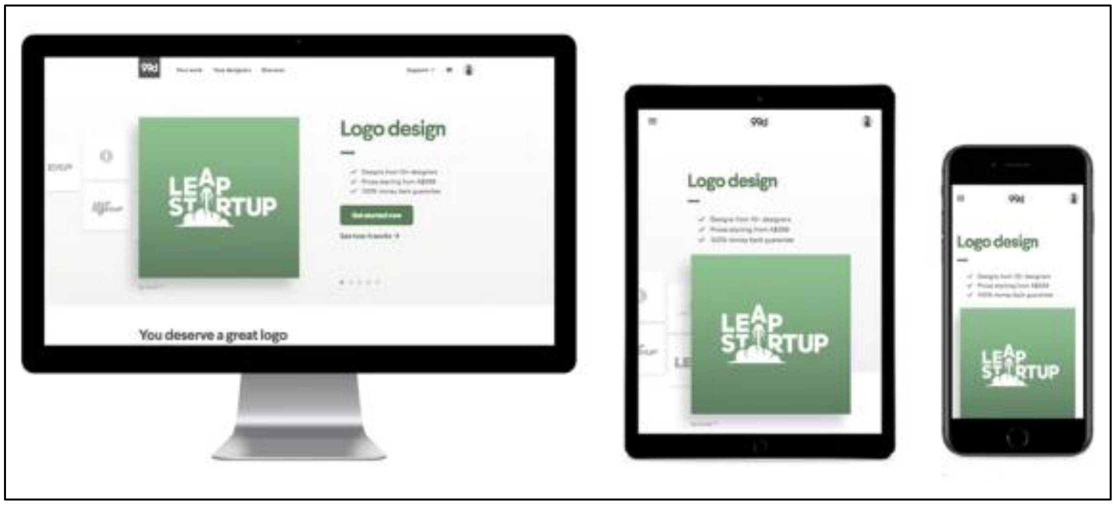


CSS Responsive

What is Responsive CSS

Responsive web design makes website rearrange whole elements according to the device screen size.

Web pages can be viewed in different devices - desktops, tablets, and phones without resizing the entire pager scale down.



Viewport

The viewport is the user's visible area on the digital device. The viewport may varies with different type of device like a smaller on a mobile phone to a large computer screen. Earlier days web pages were designed only for computer screens, but now we are using different type of devices and their screen sizes are different. Then we started surfing the internet using tablets and mobile phones, but the fixed size web pages were too large to fit the viewport. So the modern browsers on those devices scaled down the entire web page to fit the screen.

Setting the Viewport

HTML5 introduced a method set the web element according to the viewport size using **<meta>** tag.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

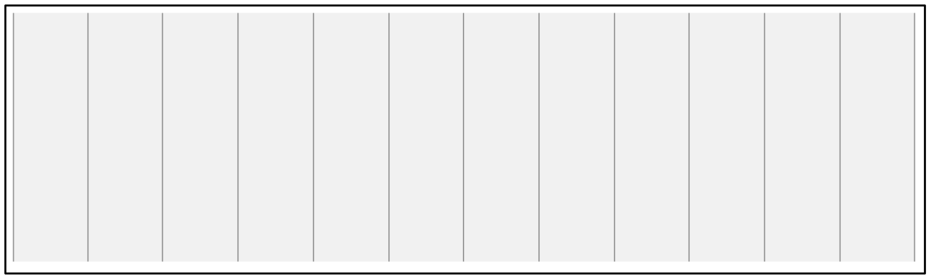
A <meta> element gives the browser instructions to control the page's width, height and scaling.

The `width=device-width` part is sets the page width to follow the screen-width current device.

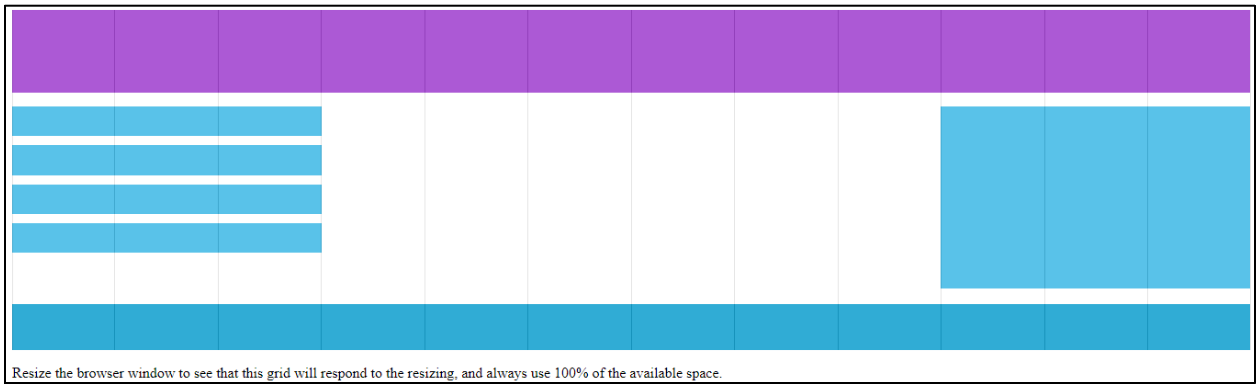
The `initial-scale=1.0` part sets the initial magnification level when the page in different size of viewports.

Grid-View

Based on a grid-view, CSS divided into 12 equal columns -



It makes it easier to place elements on the web page.

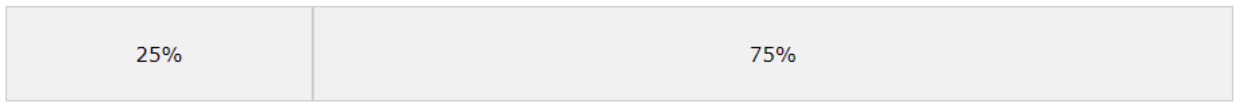


Resize the browser window to see that this grid will respond to the resizing, and always use 100% of the available space.

Building a Responsive Grid-View

Ensure that all HTML elements have the css box-sizing property and it is set with border-box property. This makes the padding and border is included in the width and height of the elements.

Example of responsive web page, with two columns -



```
.menu {
    width: 25%;
    float: left;
}
.main {
    width: 75%;
    float: left;
}
```

Media Query

Media query is a CSS property introduced in CSS3.

`@media` rule to include a block of CSS properties and work only if a certain condition is true.

```
@media only screen and (max-width: 500px)
{
    body {
        background-color: lightblue;
    }
}
```

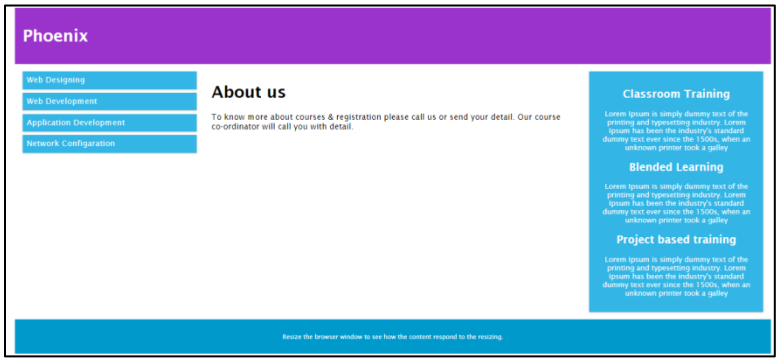
Example of Media Query

```
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
   * <style>
      {
         box-sizing: border-box;
      }
    .row::after
      {
         content: "";
         clear: both;
         display: table;
      }
   [class*="col-"]
     {
        float: left;
        padding: 15px;
     }
    html
    {
       font-family: "Lucida Sans", sans-serif;
    }
  .header
    {
       background-color: #9933cc;
       color: #ffffff;
       padding: 15px;
   }
 .menu ul
   {
       list-style-type: none;
       margin: 0;
      padding: 0;
  }
```

```
.menu li
  {
     padding: 8px;
     margin-bottom: 7px;
     background-color: #33b5e5;
     color: #ffffff;
     box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px
rgba(0,0,0,0.24);
  }
.menu li:hover
  {
     background-color: #0099cc;
  }
.aside
  {
     background-color: #33b5e5;
     padding: 15px;
     color: #ffffff;
     text-align: center;
     font-size: 14px;
     box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px
rgba(0,0,0,0.24);
  }
 .footer
   {
      background-color: #0099cc;
      color: #ffffff;
      text-align: center;
      font-size: 12px;
      padding: 15px;
  }
/* For mobile phones: */
 [class*="col-"]
   {
      width: 100%;
   }
@media only screen and (min-width: 768px)
  {
    /* For desktop: */
     .col-1 {width: 8.33%;}
     .col-2 {width: 16.66%;}
     .col-3 {width: 25%;}
     .col-4 {width: 33.33%;}
     .col-5 {width: 41.66%;}
     .col-6 {width: 50%;}
     .col-7 {width: 58.33%;}
     .col-8 {width: 66.66%;}
     .col-9 {width: 75%;}
     .col-10 {width: 83.33%;}
     .col-11 {width: 91.66%;}
     .col-12 {width: 100%;}
  }
 </style>
</head>
<body>
  <div class="header">
    <h1>Phoenix</h1>
  </div>
 <div class="row">
   <div class="col-3 menu">
     <ul>
       <li>Web Designing</li>
       <li>Web Development</li>
       <li>Application Development</li>
       <li>Network Configaration</li>
    </ul>
  </div>
  <div class="col-6">
    <h1>About us</h1>
     <p>To know more about courses & registration please call us or
send your detail. Our course co-ordinator will call you with
detail.</p>
  </div>
  <div class="col-3 right">
  <div class="aside">
    <h2>Classroom Training</h2>
      <p>Lorem Ipsum is simply dummy text of the printing and
typesetting industry. Lorem Ipsum has been the industry's standard
dummy text ever since the 1500s, when an unknown printer took a galley
</p>
    <h2>Blended Learning</h2>
      <p>Lorem Ipsum is simply dummy text of the printing and
typesetting industry. Lorem Ipsum has been the industry's standard
dummy text ever since the 1500s, when an unknown printer took a galley
</p>
    <h2>Project based training</h2>
```

```
      <p>Lorem Ipsum is simply dummy text of the printing and
typesetting industry. Lorem Ipsum has been the industry's standard
dummy text ever since the 1500s, when an unknown printer took a galley
</p>
  </div>
 </div>
</div>
<div class="footer">
  <p>Resize the browser window to see how the content respond to the
resizing.</p>
 </div>
</body>
</html>
```

Output (Desktop View)



Phoenix

Web Designing
Web Development
Application Development
Network Configaration

About us

To know more about courses & registration please call us or send your detail. Our course co-ordinator will call you with detail.

Classroom Training

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley

Blended Learning

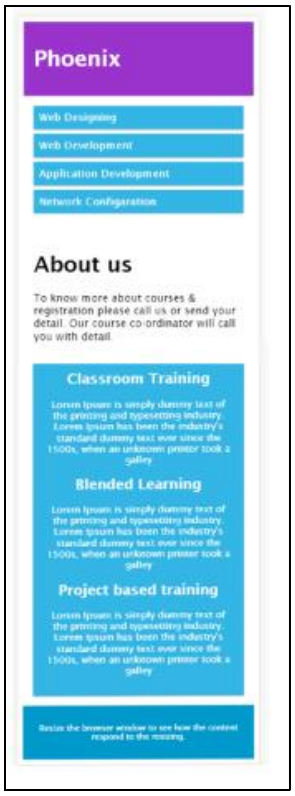
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley

Project based training

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley

Resize the browser window to see how the content respond to the resizing.

Mobile View



Phoenix

Web Designing
Web Development
Application Development
Network Configaration

About us

To know more about courses & registration please call us or send your detail. Our course co-ordinator will call you with detail.

Classroom Training

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley

Blended Learning

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley

Project based training

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley

Resize the browser window to see how the content respond to the resizing.

Bootstrap

Introduction of Bootstrap

Bootstrap is an open source library of CSS, and JavaScript .Developing with HTML pages with minimal timing .Bootstrap also has responsive features , mobile-first approach on the web with the world's most popular front-end CSS & JavaScript component library design and developed by Twitter group by Mark Otto and Jacob Thornton at Twitter, and released as an open source product in August 2011 on GitHub.

Features of Bootstrap

The main feature of bootstrap is its mobile fast approach and easy to use. Apart from this bootstrap is also simple and responsive component library. Few important features of bootstrap is given below.

1. Easy to use

   Creating various web page by just basic knowledge of HTML and CSS.

2. Responsive features

   Bootstrap's responsive CSS adjusts to any viewport likes phones, tablets, and desktops etc.

3. Mobile-first approach

   From Bootstrap 3, mobile-first styles are part of the core framework for easy access from anywhere. Mobile-first styles are part of the core framework. add the following `<meta>` tag inside the `<head>` element.

   ```
   <meta name="viewport" content="width=device-width, initial-scale=1">
   ```

   The `width=device-width` the width of the page to follow the screen-width of the display which will depending on the different devices.

4. Browser compatibility

   Bootstrap is supports all modern browsers (**Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera**)

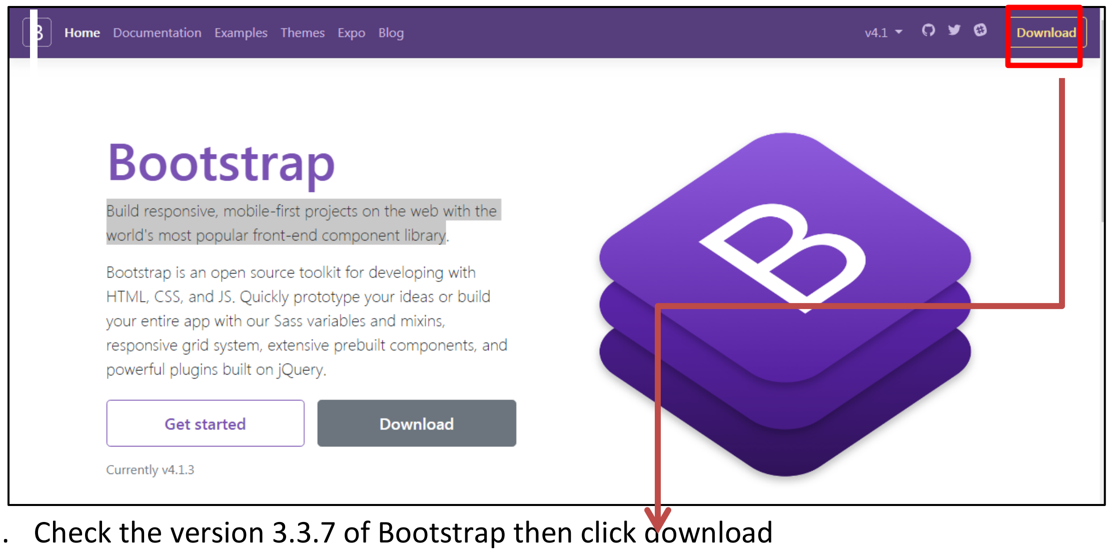
Downloads and Install

There are two ways to start using Bootstrap on your own web site.

- **Download Bootstrap from getbootstrap.com**
- **Include Bootstrap from a CDN**

Download Bootstrap from getbootstrap.com

1. Go to http://getbootstrap.com



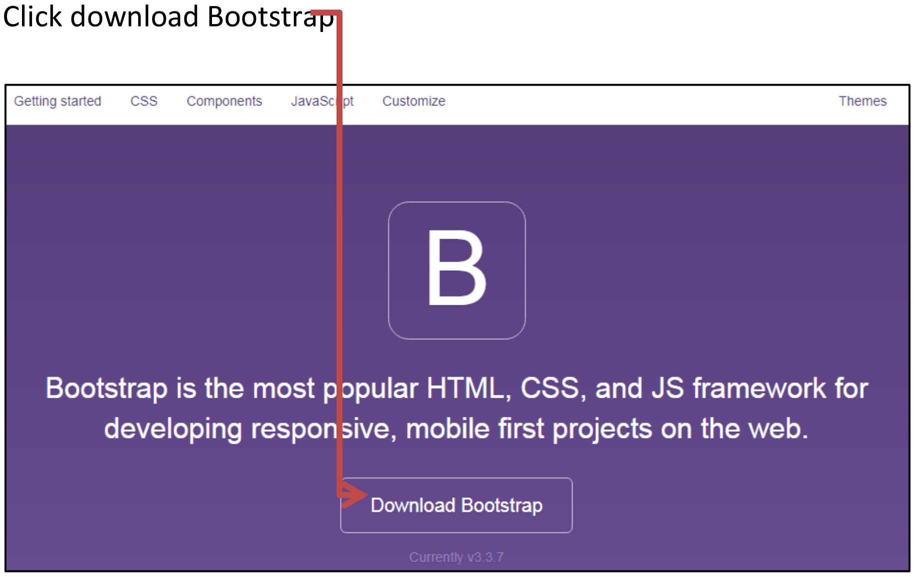
Bootstrap

Build responsive, mobile-first projects on the web with the world's most popular front-end component library.

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

Currently v4.1.3

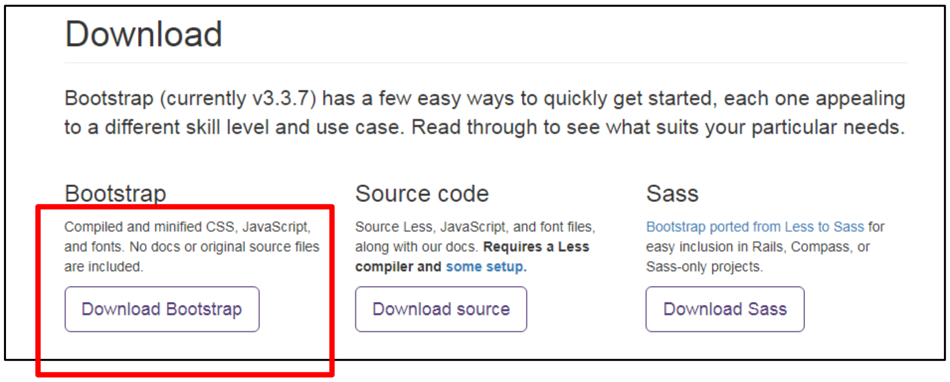
2. Check the version 3.3.7 of Bootstrap then click download
3. Click download Bootstrap



Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

Currently v3.3.7

4. Download from download bootstrap button



Download

Bootstrap (currently v3.3.7) has a few easy ways to quickly get started, each one appealing to a different skill level and use case. Read through to see what suits your particular needs.

Bootstrap

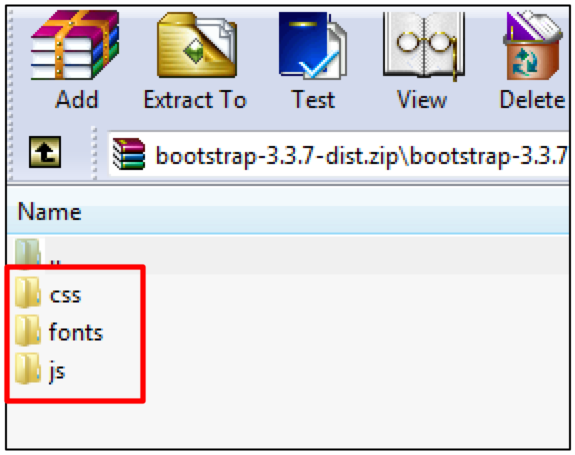
Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.

Source code

Source Less, JavaScript, and font files, along with our docs. **Requires a Less compiler and** some setup.

Sass

Bootstrap ported from Less to Sass for easy inclusion in Rails, Compass, or Sass-only projects.



5. From download folder extract the Zip file and copy **css, fonts, js** folder to your project folder.
6. Now link those file (bootstrap.css, bootstrap-theme.css, bootstrap.js) with **html** or **php** files.
7. Add jquery-mini-3.3.1.js file in **js** folder and link with **html** or **php** pages.

```
<link href="css/bootstrap.css" rel="stylesheet">
<link href="css/bootstrap-theme.css" rel="stylesheet">
<script src="js/bootstrap.js"></script>
<script src="js/jquery-3.3.1.min.js"></script>
```

Include Bootstrap from a CDN

Don't want to download and host Bootstrap in offline?? ,Optionally you can include it from a **CDN (Content Delivery Network).**

MaxCDN provides CDN online support for Bootstrap's CSS and JavaScript and jQuery .

```
<!-- Latest compiled and minified CSS from getbootstrap -->


<link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/
3.3.7/css/bootstrap.min.css">

<!-- jQuery library from jquery.com -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery
.min.js"></script>

<!-- Latest compiled JavaScript from getbootstrap -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootst
rap.min.js"></script>
```

Responsive Design

Responsive web design fits our web content according to the device screen size and looks good on all devices. Responsive web design is created by only uses of HTML and CSS. Bootstrap components have responsive fetchers to reduce the effort.

Web pages can be viewed using many different devices: desktops, tablets, and phones are given below:

Bootstrap Typography

Bootstrap's default font-size is 14px, with a line-height of 1.428 for to all the content inside `<body>` element and paragraphs (`<p>`).

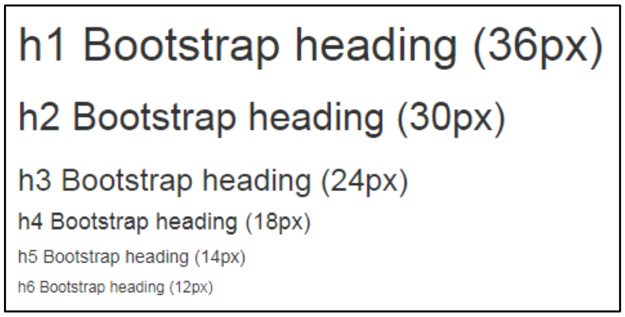
Heading text <h1> to <h6>

By default, Bootstrap will style the HTML headings (`<h1>` to `<h6>`) example given below:

Example of Heading Text

```
<div class="container">

  <h1>h1 Bootstrap heading (36px)</h1>
  <h2>h2 Bootstrap heading (30px)</h2>
  <h3>h3 Bootstrap heading (24px)</h3>
  <h4>h4 Bootstrap heading (18px)</h4>
  <h5>h5 Bootstrap heading (14px)</h5>
  <h6>h6 Bootstrap heading (12px)</h6>

</div>
```

Output



```
h1 Bootstrap heading (36px)
h2 Bootstrap heading (30px)
h3 Bootstrap heading (24px)
h4 Bootstrap heading (18px)
h5 Bootstrap heading (14px)
h6 Bootstrap heading (12px)
```

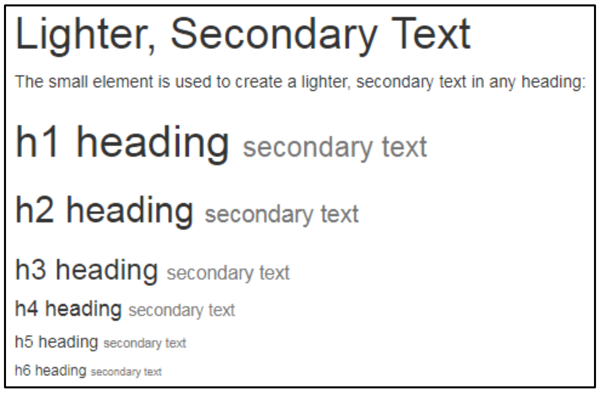
<small>

The HTML `<small>` element is used to create a lighter, secondary text in any heading

Example of <small>

```
<div class="container">
  <h1>Lighter, Secondary Text</h1>
  <p>The small element is used to create a lighter, secondary text in
any heading:</p>
  <h1>h1 heading <small>secondary text</small></h1>
  <h2>h2 heading <small>secondary text</small></h2>
  <h3>h3 heading <small>secondary text</small></h3>
  <h4>h4 heading <small>secondary text</small></h4>
  <h5>h5 heading <small>secondary text</small></h5>
  <h6>h6 heading <small>secondary text</small></h6>
</div>
```

Output



```
Lighter, Secondary Text
The small element is used to create a lighter, secondary text in any heading:
h1 heading secondary text
h2 heading secondary text
h3 heading secondary text
h4 heading secondary text
h5 heading secondary text
h6 heading secondary text
```

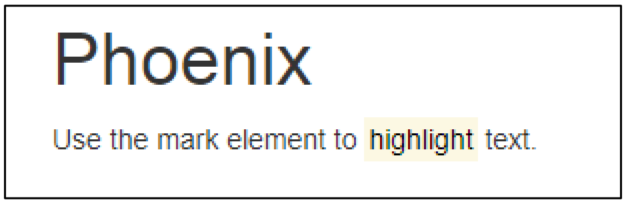
<mark>

Bootstrap will style the HTML <mark> element, example is given below:

Example of <mark>

```
<div class="container">
  <h1>Phoenix </h1>
  <p>Use the mark element to <mark>highlight</mark> text.</p>
</div>
```

Output :



Phoenix

Use the mark element to highlight text.

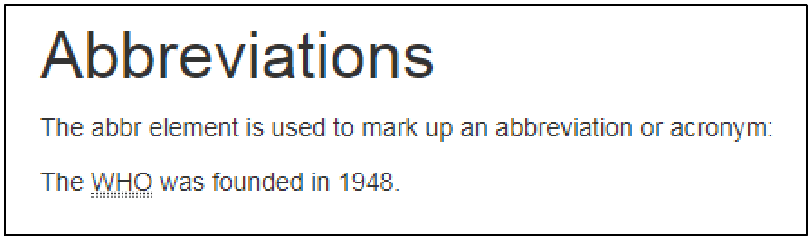
<abbr>

Bootstrap will style for abbreviation in the HTML `<abbr>` element, example is given below -

Example of <abbr>

```
<div class="container">
  <h1>Abbreviations</h1>
  <p>The abbr element is used to mark up an abbreviation or
acronym:</p>
  <p>The <abbr title="World Health Organization">WHO</abbr> was
founded in 1948.</p>
</div>
```

Output :



Abbreviations

The abbr element is used to mark up an abbreviation or acronym:

The WHO was founded in 1948.

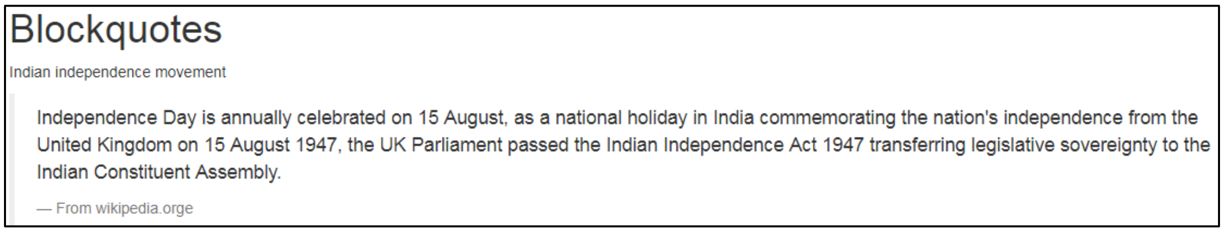
<blockquote>

Bootstrap will style the HTML `<blockquote>` element. Example is given below:

Example of <blockquote>

```
<div class="container">
  <h1>Blockquotes</h1>
  <p>Indian independence movement</p>
  <blockquote>
    <p>Independence Day is annually celebrated on 15 August, as a
national holiday in India commemorating the nation's independence from
the United Kingdom on 15 August 1947, the UK Parliament passed the
Indian Independence Act 1947 transferring legislative sovereignty to
the Indian Constituent Assembly.</p>
    <footer>From wikipedia.orge</footer>
  </blockquote>
</div>
```

Output :



Blockquotes

Indian independence movement

Independence Day is annually celebrated on 15 August, as a national holiday in India commemorating the nation's independence from the United Kingdom on 15 August 1947, the UK Parliament passed the Indian Independence Act 1947 transferring legislative sovereignty to the Indian Constituent Assembly.

— From wikipedia.orge

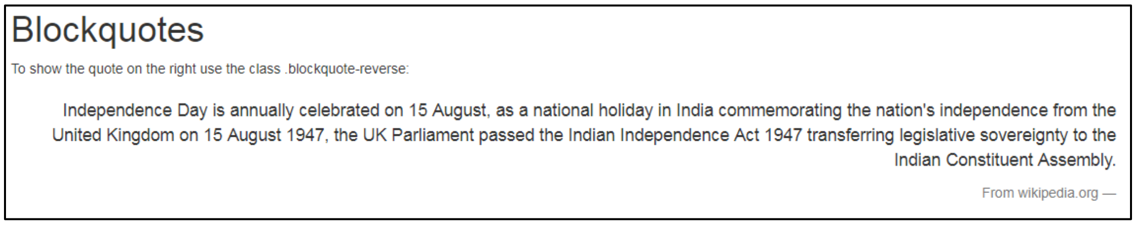
The quote on the right, use the `.blockquote-reverse` class:

Example of blockquote-reverse

```
<div class="container">
  <h1>Blockquotes</h1>
  <p>To show the quote on the right use the class .blockquote-
reverse:</p>
  <blockquote class="blockquote-reverse">
<p>Independence Day is annually celebrated on 15 August, as a national
holiday in India commemorating the nation's independence from the
United Kingdom on 15 August 1947, the UK Parliament passed the Indian
```

```
Independence Act 1947 transferring legislative sovereignty to the
Indian Constituent Assembly.</p>
    <footer>From wikipedia.orge</footer>
  </blockquote>
</div>
```

Output:



Blockquotes

To show the quote on the right use the class .blockquote-reverse:

Independence Day is annually celebrated on 15 August, as a national holiday in India commemorating the nation's independence from the United Kingdom on 15 August 1947, the UK Parliament passed the Indian Independence Act 1947 transferring legislative sovereignty to the Indian Constituent Assembly.

From wikipedia.org —

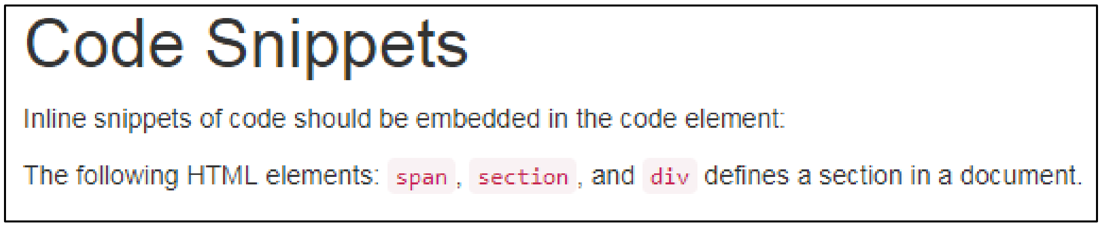
<code>

Bootstrap will style the HTML <code> element. Example is given below:

Example of <code>

```
<div class="container">
  <h1>Code Snippets</h1>
  <p>Inline snippets of code should be embedded in the code
element:</p>
  <p>The following HTML elements: <code>span</code>,
<code>section</code>, and <code>div</code> defines a section in a
document.</p>
</div>
```

Output:



Code Snippets

Inline snippets of code should be embedded in the code element:

The following HTML elements: `span`, `section`, and `div` defines a section in a document.

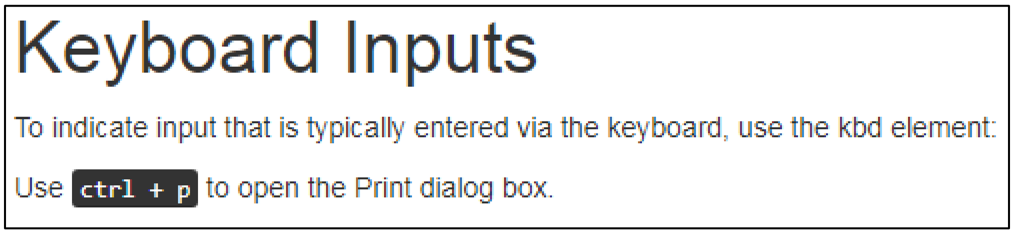
<kbd>

Bootstrap will style the HTML `<kbd>` element. Example is given below:

Example of <kbd>

```
<div class="container">
  <h1>Keyboard Inputs</h1>
  <p>To indicate input that is typically entered via the keyboard, use
the kbd element:</p>
  <p>Use <kbd>ctrl + p</kbd> to open the Print dialog box.</p>
</div>
```

Output:



Keyboard Inputs

To indicate input that is typically entered via the keyboard, use the kbd element:

Use `ctrl + p` to open the Print dialog box.

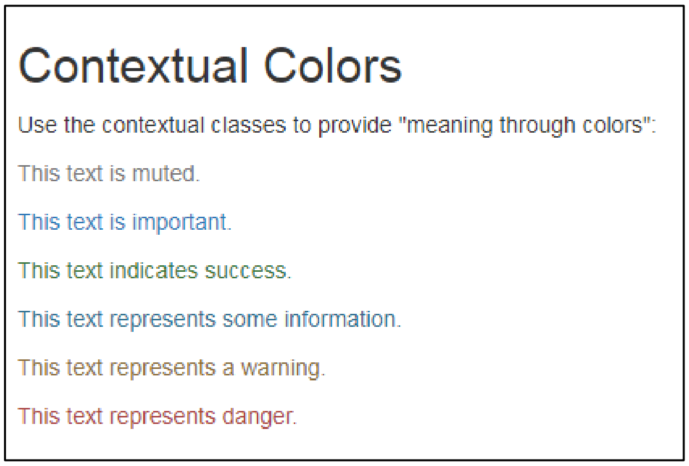
Contextual Colors and Backgrounds

Contextual classes mean - "meaning through colors".
The classes for text colors are: `.text-muted`, `.text-primary`, `.text-success`, `.text-info`, `.text-warning`, and `.text-danger`:

Example of Contextual Colors and Backgrounds

```
<div class="container">
  <h2>Contextual Colors</h2>
  <p>Use the contextual classes to provide "meaning through
colors":</p>
  <p class="text-muted">This text is muted.</p>
  <p class="text-primary">This text is important.</p>
  <p class="text-success">This text indicates success.</p>
  <p class="text-info">This text represents some information.</p>
  <p class="text-warning">This text represents a warning.</p>
  <p class="text-danger">This text represents danger.</p>
</div>
```

Output:



Contextual Colors

Use the contextual classes to provide "meaning through colors":

This text is muted.

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

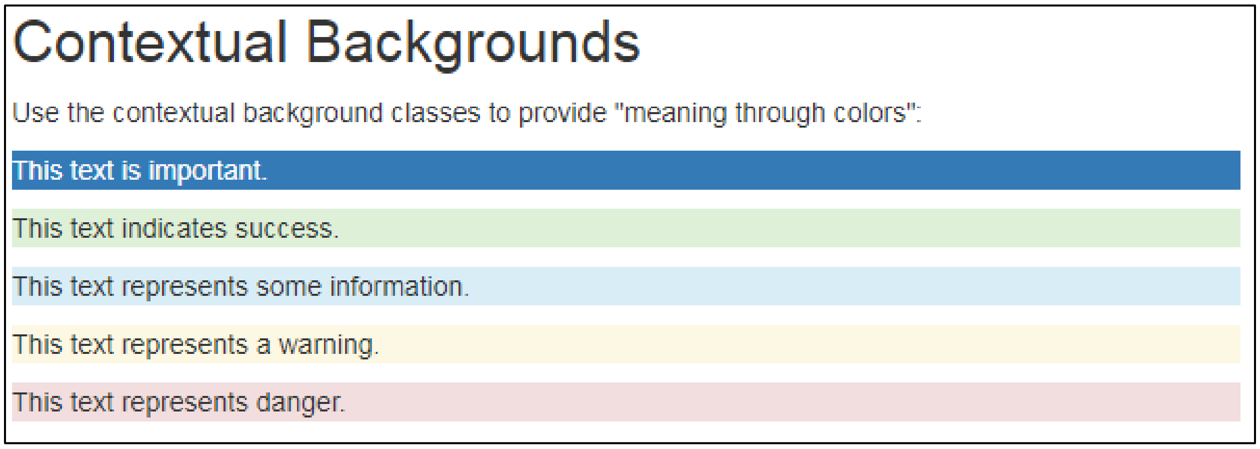
This text represents danger.

The classes for background colors are: `.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, and `.bg-danger`:

Example of background colors

```
<div class="container">
  <h2>Contextual Backgrounds</h2>
  <p>Use the contextual background classes to provide "meaning through
colors":</p>
  <p class="bg-primary">This text is important.</p>
  <p class="bg-success">This text indicates success.</p>
  <p class="bg-info">This text represents some information.</p>
  <p class="bg-warning">This text represents a warning.</p>
  <p class="bg-danger">This text represents danger.</p>
</div>
```

Output:



Contextual Backgrounds

Use the contextual background classes to provide "meaning through colors":

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

More Typography Classes

More typography classes are given below :

Class	Description
.lead Makes	To create a paragraph stand out.
.small	To create smaller text (set to 85% of the size of the parent).
.text-left	To create left-aligned text.
.text-center	To create center-aligned text.
.text-right	To create right-aligned text.
.text-justify	To create justified text.
.text-nowrap	To create no wrap text.
.text-lowercase	To create lowercased text.
.text-uppercase	To create uppercased text.
.text-capitalize	To create capitalized text.

`.initialism`	To create displays the text inside an <abbr> element in a smaller font size.
`.list-inline`	Places all <li> list items on a single line.
`.pre-scrollable`	Makes all <pre> element contents are scrollable.

Bootstrap Containers

a containing element to wrap site contents and making more adjustable in responsive design .

There are two container classes to choose from

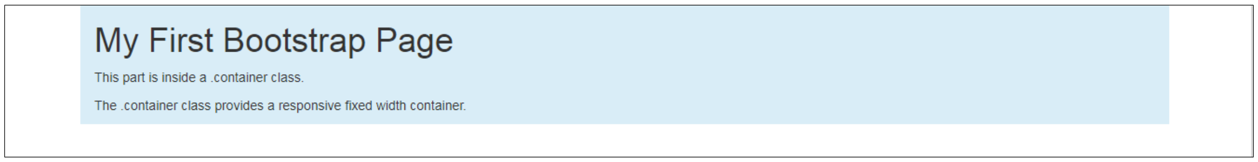
1. The `.container` class create a responsive **fixed width container**
2. The `.container-fluid` class create **a full width container**, display the entire 100% width of the viewport

Example of Fixed Width Container

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>

<div class="container">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>
```

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

Output:



My First Bootstrap Page

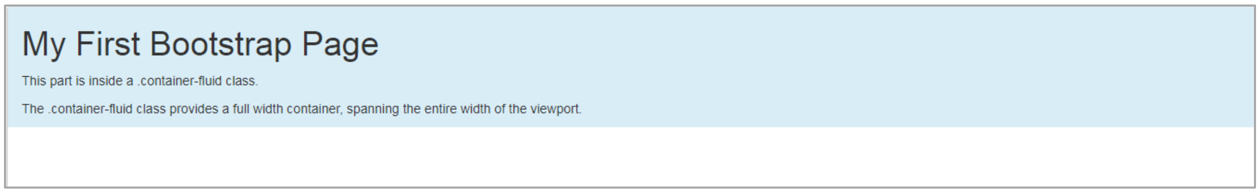
This part is inside a .container class.

The .container class provides a responsive fixed width container.

Example of Full Width Container

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>

<div class="container-fluid">
  <h1>My First Bootstrap Page</h1>
  <p>This is some text.</p>
</div>

</body>
</html>
```

Output



My First Bootstrap Page

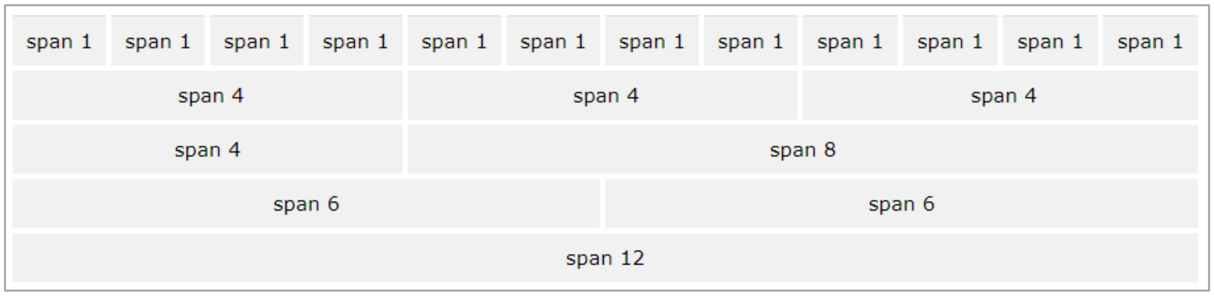
This part is inside a .container-fluid class.

The .container-fluid class provides a full width container, spanning the entire width of the viewport.

Bootstrap Grid System

Bootstrap's grid system divides body into 12 columns across the page. All girds are treated as a column .From 1 to 12 all columns are re-arranged automatically depending on the screen size.

Grid system has four classes according to device screen size -



- `xs` (Extra small for phones - screens < 768px wide)
- `sm` (Small for tablets – screens >= 768px wide)
- `md` (Mid-range for small laptops and PC - screens >= 992px wide)
- `lg` (Large for laptops and desktops - screens >= 1200px wide)

Bootstrap Grid Basic Structure

```
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
<div class="row">
  ...
</div>
```

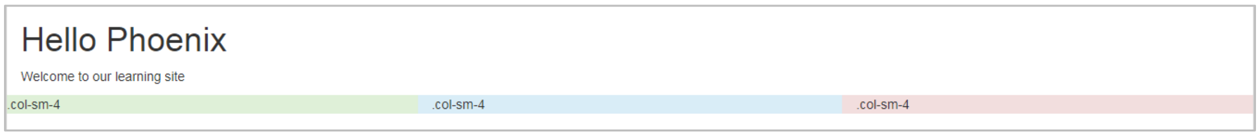
Here `div class="col-*-*"` mean col-md-6 (col-class-number).

Before use column class first put (`<div class="row">`) to create row.

Example of 3 Equal Column

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>

<div class="container-fluid">
  <h1>Hello Phoenix</h1>
  <p>Welcome to our learning site</p>
</div>
<div class="row">
  <div class="col-sm-4 bg-success">.col-sm-4</div>
  <div class="col-sm-4 bg-info">.col-sm-4</div>
  <div class="col-sm-4 bg-danger">.col-sm-4</div>
</div>
</body>
</html>
```

Output:



```
Hello Phoenix
Welcome to our learning site
.col-sm-4    .col-sm-4    .col-sm-4
```

Also you can use pictures, videos, forms elements inside of column instance of text.

Bootstrap Images

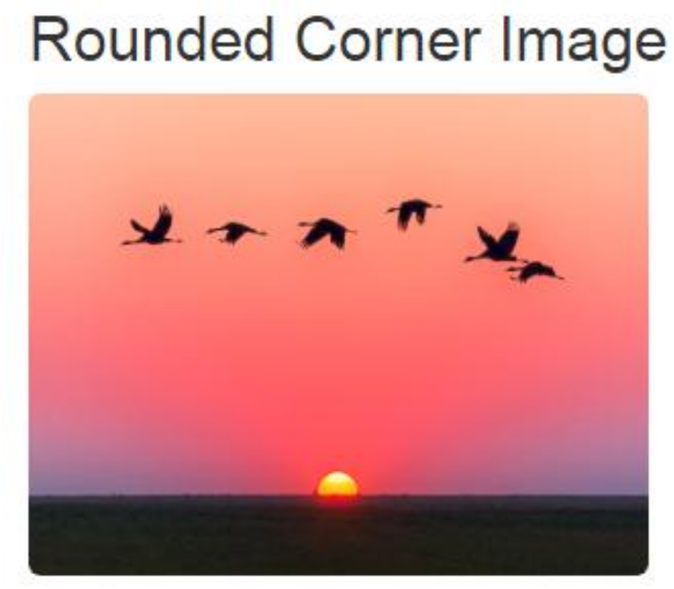
Rounded Corners

The `.img-rounded` class adds rounded corners to an image (Use updated browser. For Microsoft browser use **EI9** or above):

Example of Rounded Corners

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>

<div class="container-fluid">
  <h1>Rounded Corner Image</h1>
  <p>Welcome to our learning site</p>
</div>
<div class="row">
 <img src="animals-avian-backlit-1126384.jpg" class="img-rounded" alt="Cinque
Terre" width="304" height="236">

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

Output:



Circle

The `.img-circle` class shapes the image to a circle image (Use updated browser. For Microsoft browser use **EI9** or above):

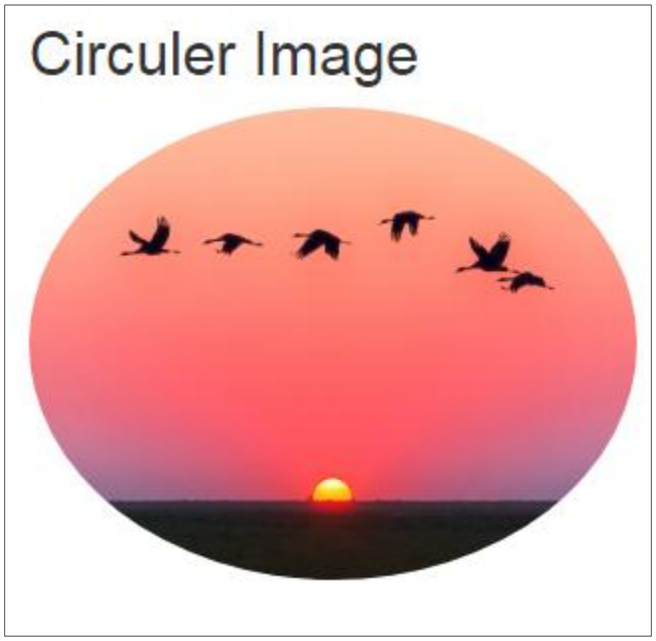
For proper circle use equal width and height

Example of Circle

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>
```

```
<div class="container-fluid">
  <h1>Rounded Corner Image</h1>
  <p>Welcome to our learning site</p>
</div>
<div class="row">
 <img src="animals-avian-backlit-1126384.jpg" class="img-circle" alt="Cinque
Terre" width="304" height="236">

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

Output:



Thumbnail

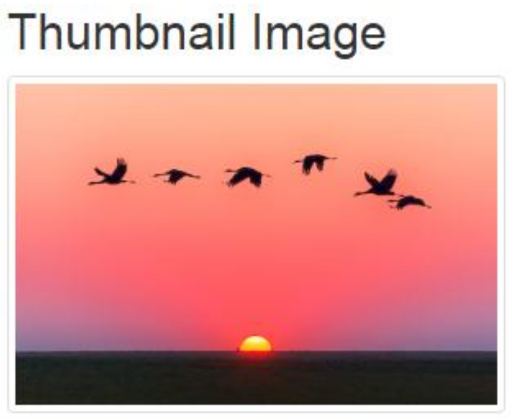
The `.img-thumbnail` class shapes the image to a thumbnail (Use updated browser. For Microsoft browser use **EI9** or above):

Example of Thumbnail

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>
<div class="container-fluid">
  <h1>Rounded Corner Image</h1>
  <p>Welcome to our learning site</p>
</div>
<div class="row">
 <img src="animals-avian-backlit-1126384.jpg" class="img-circle" alt="Cinque
Terre" width="304" height="236">

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

Output:


Thumbnail Image

Responsive Images

Images come in actual size . Responsive images fetchers automatically adjust image size to fit on the Viewport.

Create responsive images by adding an class `.img-responsive` to the `<img>` attribute.

Insite the `.img-responsive` class use `display: block;` and `max-width: 100%;` and `height: auto;` to the image:

Example of Responsive Images

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>

<div class="container">

  <h2>Image Gallery</h2>

 <p>Click on the images to enlarge them.</p>
<div class="row">

  <div class="col-md-4">
    <div class="thumbnail">
      <a href="animals-avian-backlit-1126384.jpg">
        <img src="animals-avian-backlit-
1126384.jpg" alt="Animal" style="width:100%">
        <div class="caption">
          <p>Lorem ipsum donec id elit non mi porta gravida at eget
metus.</p>

        </div>
      </a>
    </div>
  <div class="col-md-4">
    <div class="thumbnail">
      <a href="Chrysanthemum.jpg">
        <img src="Chrysanthemum.jpg" alt="Chrysanthemum" style="width:100%">
        <div class="caption">
          <p>Lorem ipsum donec id elit non mi porta gravida at eget
metus.</p>
        </div>
      </a>
    </div>
  </div>
  <div class="col-md-4">
    <div class="thumbnail">
      <a href="Desert.jpg">
        <img src="Desert.jpg" alt="Desert" style="width:100%">
        <div class="caption">
          <p>Lorem ipsum donec id elit non mi porta gravida at eget
metus.</p>
        </div>
      </a>
    </div>
  </div>

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

Note: The `.caption` class create proper padding and a dark grey color to text. You can use this class as a contextual text color inside thumbnails.

Output(Large screen)

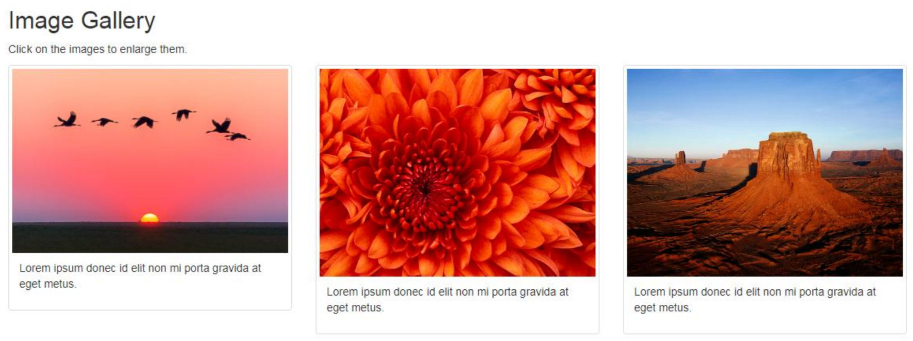


Image Gallery

Click on the images to enlarge them.

Lorem ipsum donec id elit non mi porta gravida at eget metus.

Lorem ipsum donec id elit non mi porta gravida at eget metus.

Lorem ipsum donec id elit non mi porta gravida at eget metus.

Output(Slandered cellphone screen)

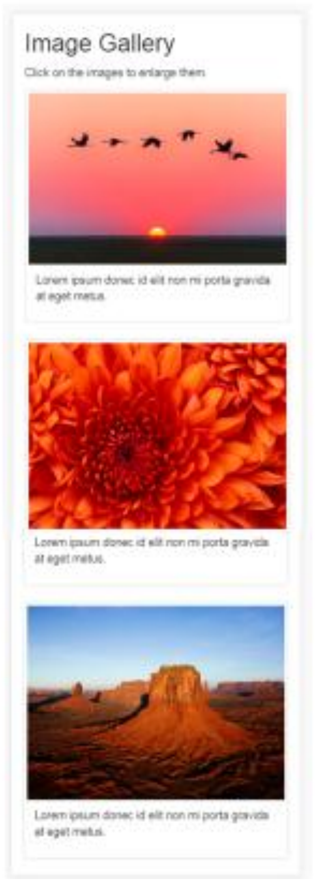


Image Gallery

Click on the images to enlarge them

Lorem ipsum donec id elit non mi porta gravida at eget metus.

Lorem ipsum donec id elit non mi porta gravida at eget metus.

Lorem ipsum donec id elit non mi porta gravida at eget metus.

Bootstrap Table

Bootstrap creates a clean layout for tables. The table elements supported by Bootstrap are given below -

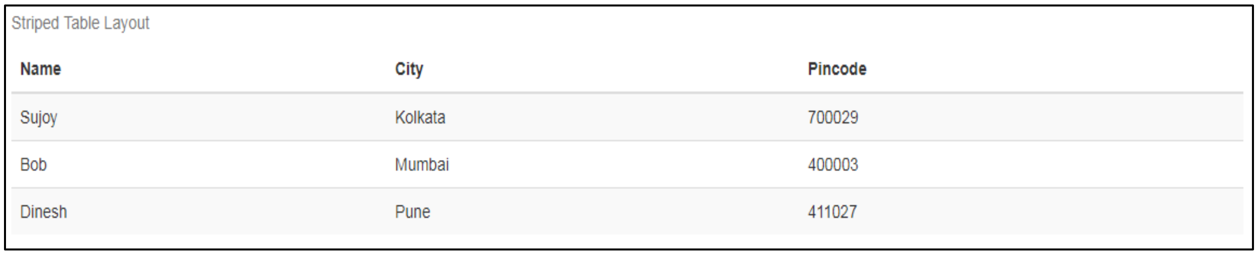
<table>	Displaying data in a table format
<thead>	Table header rows (<tr>) to label table columns.
<tbody>	Table rows (<tr>) in the body of the table.
<tr>	table cells (<td> or <th>) that appears on a single row.
<td>	Default table cell use to display data (images, text, videos).
<th>	table cell for column to create heading of contents categories .

<caption> Summary about the table .

Example of Table

```
<table class = "table table-striped">
   <caption>Striped Table Layout</caption>
   <thead>
      <tr>
         <th>Name</th>
         <th>City</th>
         <th>Pincode</th>
      </tr>
   </thead>
   <tbody>
      <tr>
         <td>Sujoy</td>
         <td>Kolkata</td>
         <td>700029</td>
      </tr>
      <tr>
         <td>Bob </td>
         <td>Mumbai</td>
         <td>400003</td>
      </tr>
      <tr>
         <td>Dinesh</td>
         <td>Pune</td>
         <td>411027</td>

     </tr>

   </tbody>

</table>
```

Output



Striped Table Layout

Name	City	Pincode
Sujoy	Kolkata	700029
Bob	Mumbai	400003
Dinesh	Pune	411027

Bordered Table

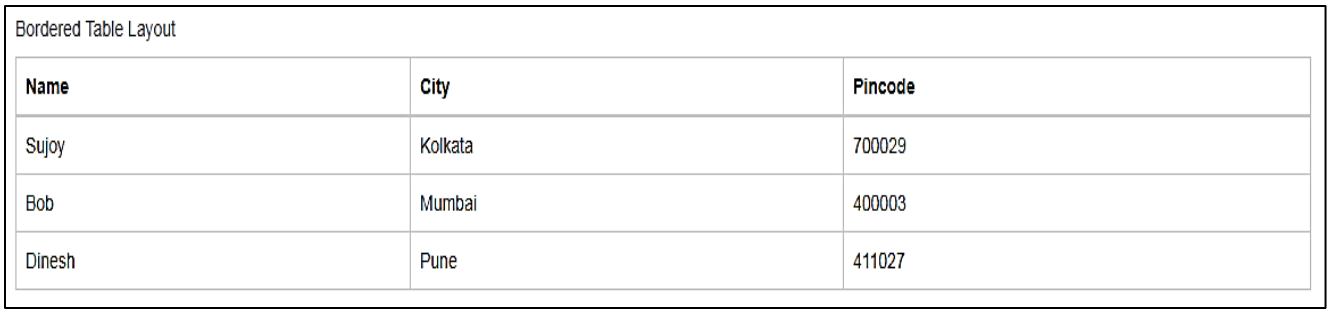
The `.table-bordered` class, create borders surrounding every row & columns of the entire table.

Example of Bordered Table

```
<table class = "table table-bordered">
```

Just add this class inside the table.

Output



Bordered Table Layout

Name	City	Pincode
Sujoy	Kolkata	700029
Bob	Mumbai	400003
Dinesh	Pune	411027

Hover Table

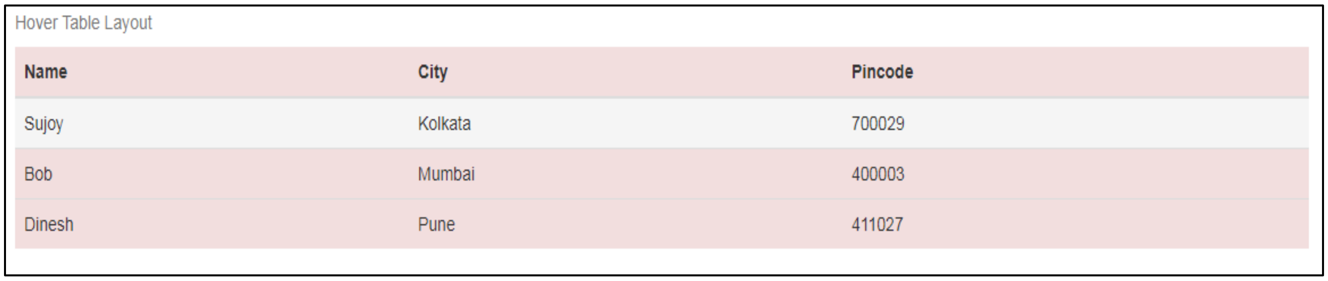
The `.table-hover` class, a light gray background will be added to rows while the cursor hovers over. Before creating Hover add contextual background color to the table.

Example of Hover Table

```
<table class = "table table-hover bg-danger">
```

Just add this class inside the table.

Output



Hover Table Layout

Name	City	Pincode
Sujoy	Kolkata	700029
Bob	Mumbai	400003
Dinesh	Pune	411027

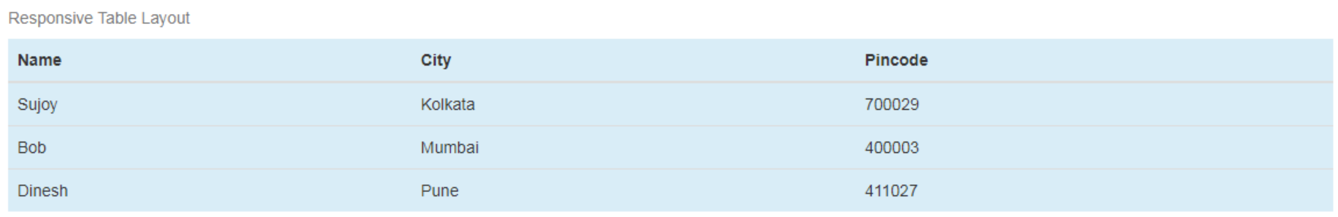
Responsive Tables

The `.table` in `.table-responsive` class, convert the table scroll horizontally up to small devices (under 768px). When viewing on anything larger device than 768px wide screen, you will not see any difference.

Example of Responsive Tables (768px wide screen)

```
<div class = "table table-responsive bg-info">
```

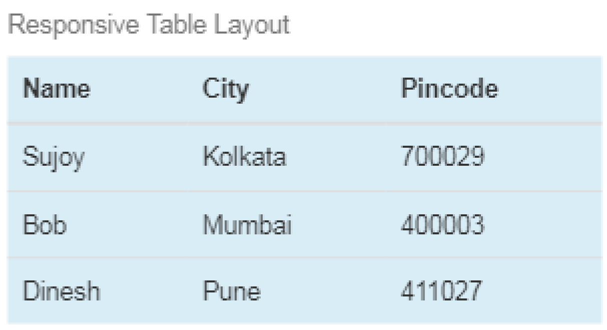
Output (768px wide screen)



Responsive Table Layout

Name	City	Pincode
Sujoy	Kolkata	700029
Bob	Mumbai	400003
Dinesh	Pune	411027

Output (over 768px wide screen)



Responsive Table Layout

Name	City	Pincode
Sujoy	Kolkata	700029
Bob	Mumbai	400003
Dinesh	Pune	411027

Contextual Classes

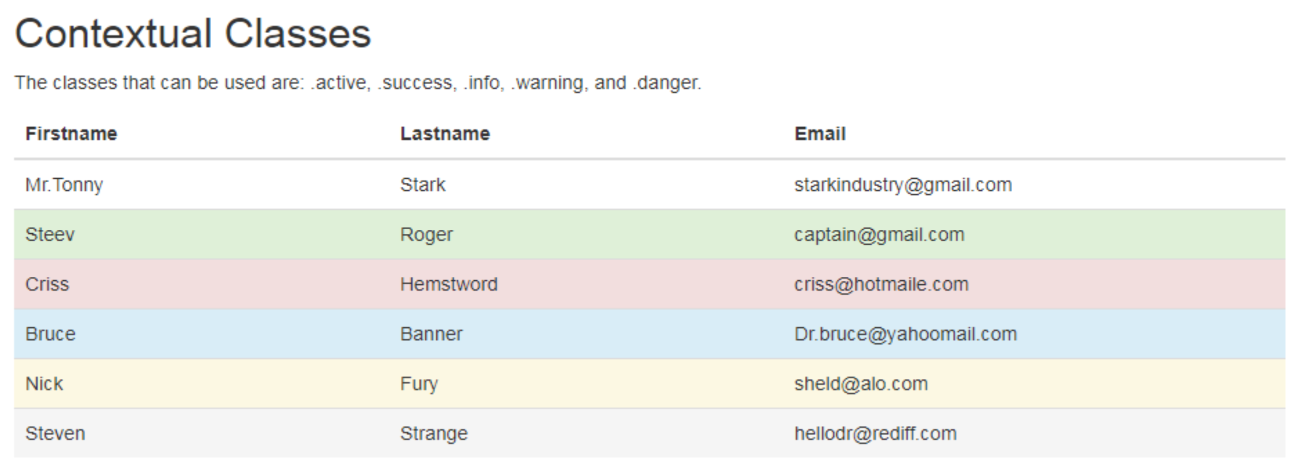
Contextual classes can be used to color table rows (`<tr>`) or table cells (`<td>`) or entire table at a time.

Example of Contextual Classes

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.mi
n.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.
js"></script>
</head>
<body>

<div class="container">
  <h2>Contextual Classes</h2>
  <p>The classes that can be used are: .active, .success, .info,
.warning, and .danger.</p>
  <table class="table">
    <thead>
      <tr>
        <th>Firstname</th>
        <th>Lastname</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
```

```
        <td>Mr.Tonny</td>
        <td>Stark</td>
        <td>starkindustry@gmail.com</td>
      </tr>
      <tr class="success">
        <td>Steev</td>
        <td>Roger</td>
        <td>captain@gmail.com</td>
      </tr>
      <tr class="danger">
        <td>Criss</td>
        <td>Hemstword</td>
        <td>criss@hotmaile.com</td>
      </tr>
      <tr class="info">
        <td>Bruce</td>
        <td>Banner</td>
        <td>Dr.bruce@yahoomail.com</td>
      </tr>
      <tr class="warning">
        <td>Nick</td>
        <td>Fury</td>
        <td>sheld@alo.com</td>
      </tr>
      <tr class="active">
        <td>Steven</td>
        <td>Strange</td>
        <td>hellodr@rediff.com</td>
      </tr>
    </tbody>
  </table>
</div>
</div>
</body>
</html>
```

Output


Contextual Classes

The classes that can be used are: .active, .success, .info, .warning, and .danger.

Firstname	Lastname	Email
Mr.Tonny	Stark	starkindustry@gmail.com
Steev	Roger	captain@gmail.com
Criss	Hemstword	criss@hotmaile.com
Bruce	Banner	Dr.bruce@yahoomail.com
Nick	Fury	sheld@alo.com
Steven	Strange	hellodr@rediff.com

Note: You can use all table classes at a time as per your need

Bootstrap Alert

Bootstrap provides an easy way to create alert messages box.

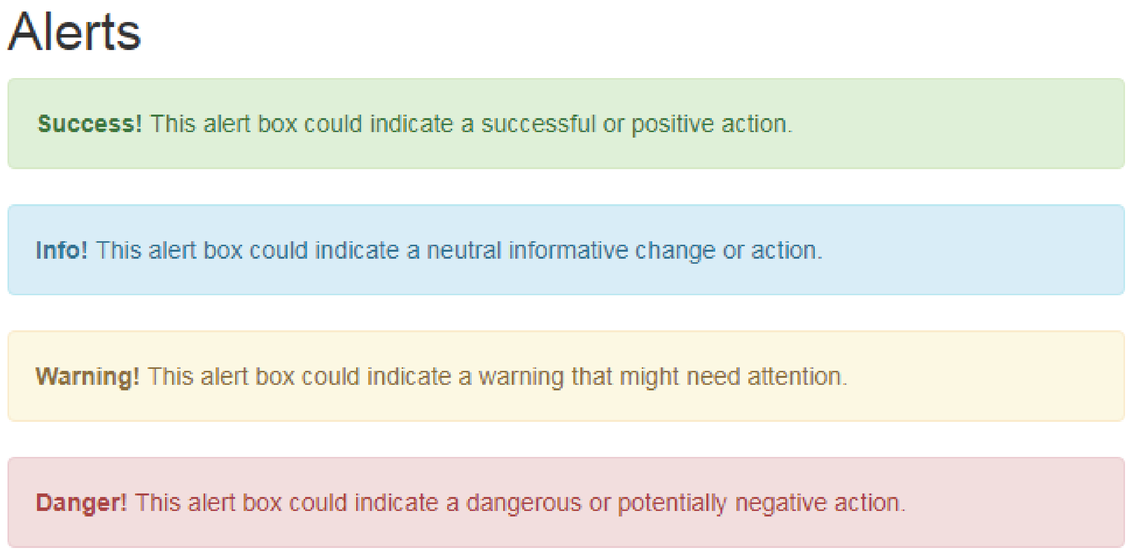
the `.alert` class, followed by one of the four contextual classes `.alert-success`, `.alert-info`, `.alert-warning` or `.alert-danger.`

Example of Bootstrap Alert

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>
```

```
<div class="container">

<div class="alert alert-success">
  <strong>Success!</strong> Indicates a successful or positive action.
</div>

<div class="alert alert-info">
  <strong>Info!</strong> Indicates a neutral informative change or action.
</div>

<div class="alert alert-warning">
  <strong>Warning!</strong> Indicates a warning that might need attention.
</div>

<div class="alert alert-danger">
  <strong>Danger!</strong> Indicates a dangerous or potentially negative
action.
</div>

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

Output



Alerts

Success! This alert box could indicate a successful or positive action.

Info! This alert box could indicate a neutral informative change or action.

Warning! This alert box could indicate a warning that might need attention.

Danger! This alert box could indicate a dangerous or potentially negative action.

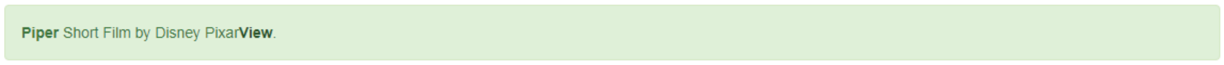
Alert Links

The `.alert-link` class creates hyperlink inside the alert message box.

Example of Alert Links

```
<div class="alert alert-success">
  <strong>Piper</strong> Short Film by Disney
Pixar<a href="https://www.youtube.com/results?search_query=piper+oscar
+winning+short+film" class="alert-link">&nbsp;View</a>.
</div>
```

Output



Piper Short Film by Disney PixarView.

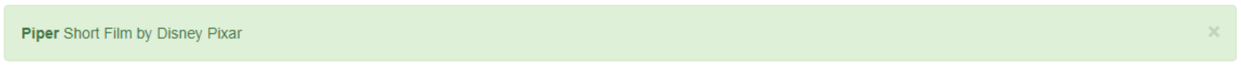
Closing Alerts

To close the alert message box, add a `.alert-dismissible` class inside of the alert container. Then add `class="close"` and `data-dismiss="alert"` to a link or a button element. For more effects use `.fade in class.`

Example of Closing Alerts

```
<div class="alert alert-success alert-dismissible fade in "><a href="
#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
<strong>Piper</strong> Short Film by Disney Pixar </div>
```

Output



Piper Short Film by Disney Pixar

Bootstrap List Group

Basic List Groups

The purpose of list group classes is to create customized content in lists. To get a basic list group using –

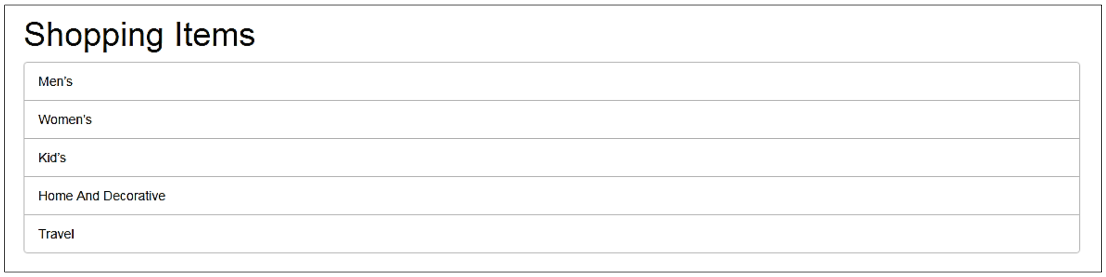
- Add the class `.list-group` to element <ul>.
- Add class `.list-group-item` to <li>.

Example of Basic List Groups

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>
<div class="container">
<h1>Shopping Items</h1>
<ul class = "list-group">
   <li class = "list-group-item">Men’s</li>
   <li class = "list-group-item">Women’s</li>
   <li class = "list-group-item">Kid’s</li>
   <li class = "list-group-item">Home And Decorative</li>
   <li class = "list-group-item">Travel</li>
</ul>
```

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

Output



Shopping Items

```
Men's
Women's
Kid's
Home And Decorative
Travel
```

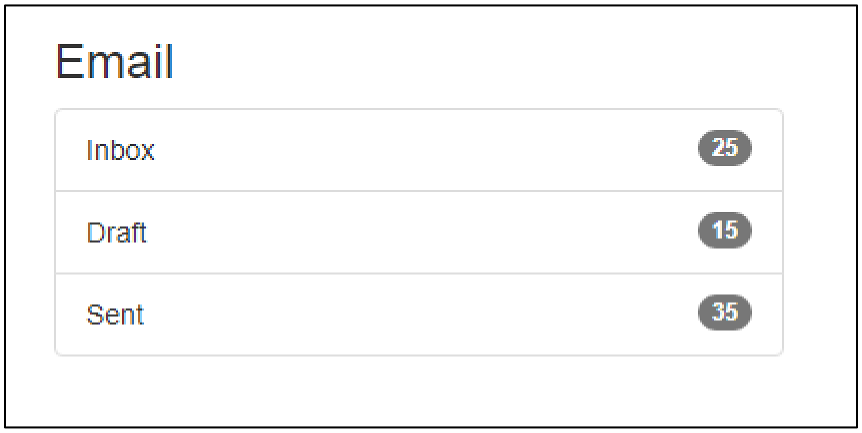
List Group with Badges

Badges to a list group will automatically be positioned on the right side as a notification symbol: To create a badge, create a `<span>` element with class `.badge` inside the list item.

Example of List Group with Badges

```
<h3>Email </h3>

<ul class="list-group col-md-4">
  <li class="list-group-item">Inbox<span class="badge">25</span></li>
  <li class="list-group-item">Draft<span class="badge">15</span></li>
  <li class="list-group-item">Sent <span class="badge">35</span></li>
</ul>
```

Output



Email

```
Inbox    25
Draft    15
Sent     35
```

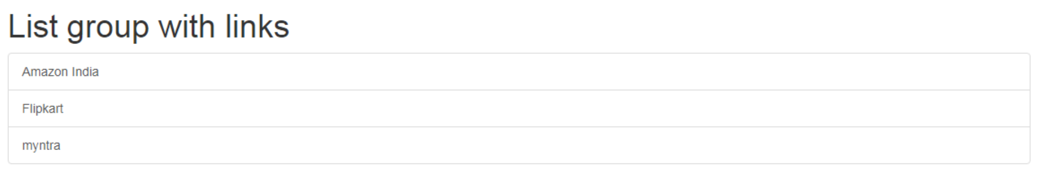
List Group with Linked Items

The items in a list group can also create hyperlinks. Add inside the `.list-group` class.

Example of List Group with Linked Items

```
<div class="list-group">
  <a href="http://www.amazon.in" class="list-group-item">Amazon India</a>
  <a href="http://www.flipkart.com" class="list-group-item">Flipkart</a>
  <a href="http://www.myntra.com" class="list-group-item">myntra</a>
</div>
```

Output


List group with links

```
Amazon India
Flipkart
myntra
```

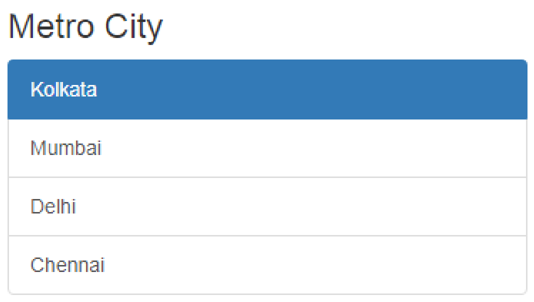
Active State

Use `.active` class to highlight the current item as on screen.

Example of Active State

```
<div class="list-group">
      <a href="#" class="list-group-item active">Kolkata</a>
      <a href="#" class="list-group-item">Mumbai</a>
      <a href="#" class="list-group-item">Delhi</a>
      <a href="#" class="list-group-item">Chennai</a>
</div>
```

Output


Metro City

```
Kolkata
Mumbai
Delhi
Chennai
```

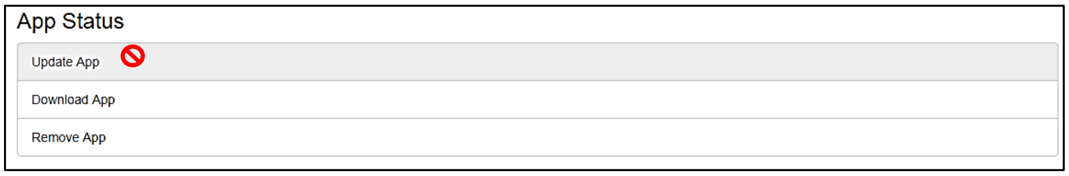
Disabled Item

You can disable any item by using add the `.disabled` class.

Example of Disabled Item

```
<div class="list-group">
  <a href="#" class="list-group-item disabled">Update App</a>
  <a href="#" class="list-group-item">Download App</a>
  <a href="#" class="list-group-item">Remove App</a>
</div>
```

Output



```
App Status
Update App
Download App
Remove App
```

Contextual Classes

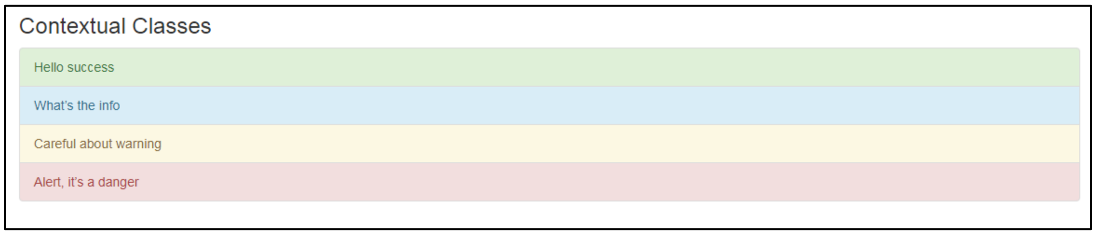
List items are also having Contextual classes.

The classes for coloring list-items are: `.list-group-item-success`, `list-group-item-info`, `list-group-item-warning`, and `.list-group-item-danger`.

Example of Contextual Classes

```
<ul class="list-group">
  <li class="list-group-item list-group-item-success">Hello success </li>
  <li class="list-group-item list-group-item-info">What’s the info </li>
  <li class="list-group-item list-group-item-warning">Careful about warning
</li>
  <li class="list-group-item list-group-item-danger">Alert, it’s a danger
</li>
</ul>
```

Output


Contextual Classes

Hello success

What's the info

Careful about warning

Alert, it's a danger

Bootstrap Dropdown

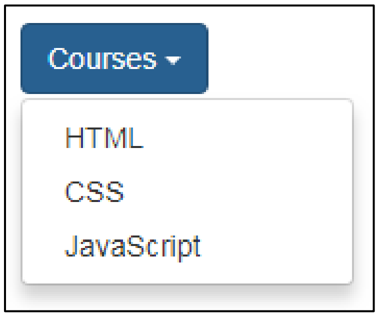
Basic Dropdown

A dropdown menu is a toggleable menu that allows the user to choose any one menu or value under main menu or main criteria for example choose Men's Women or Kid's menu from Cloth menu from a shopping site or choose EMI numbers from EMI options.

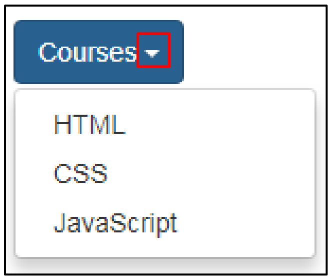
Example of Basic Dropdown

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>
<div class="container">
<div class="dropdown">
  <button class="btn btn-primary dropdown-toggle" type="button" data-
toggle="dropdown">Courses<span class="caret"></span></button>
  <ul class="dropdown-menu">
    <li><a href="#">HTML</a></li>
    <li><a href="#">CSS</a></li>
    <li><a href="#">JavaScript</a></li>
  </ul>
</div>
</body>
</html>
```

Output



Note : `<span class="caret"></span>` create dropdown arrow icon.



The `.dropdown` class creates a dropdown menu group.

To open the dropdown menu, use a `<button>` or `<a>` link or use `<li>` element with a class of `.dropdown-toggle` and the `data-toggle="dropdown"` attribute. Add the `.dropdown-menu` class to a `<ul>` element to actually build the dropdown menu items and their links.

Dropdown Divider

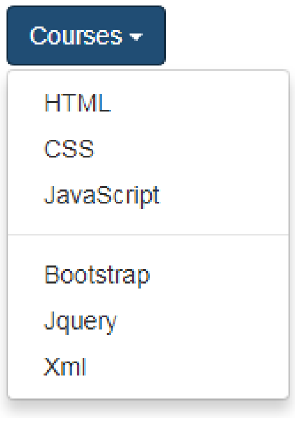
The `.divider` class is used to separate links with a thin horizontal border.

```
<li class="divider"></li>
```

Example of Dropdown Divider

```
<div class="dropdown">
  <button class="btn  btn-primary  dropdown-toggle" type="button" data-
toggle="dropdown">Courses <span class="caret"></span></button>
  <ul class="dropdown-menu">
      <li><a href="#">HTML</a></li>
      <li><a href="#">CSS</a></li>
      <li><a href="#">JavaScript</a></li>
      <li class="divider"></li>
      <li><a href="#">Bootstrap</a></li>
      <li><a href="#">jQuery</a></li>
      <li><a href="#">Xml</a></li>
  </ul>
</div>
```

Output



Dropdown Header

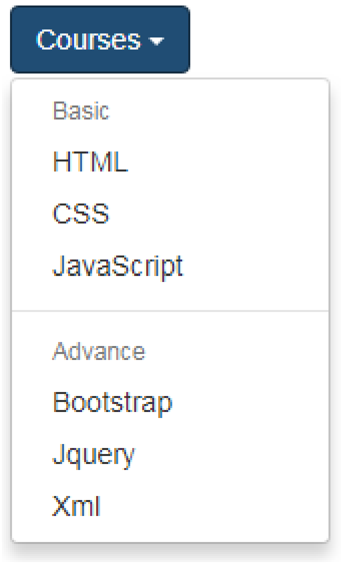
The `.dropdown-header` class is used to create headers inside the dropdown menu to indicate categories.

Example of Dropdown Header

```
<div class="dropdown">
  <button class="btn btn-primary dropdown-toggle" type="button" data-
```

```
toggle="dropdown">Courses <span class="caret"></span></button>
  <ul class="dropdown-menu">
      <li class="dropdown-header">Basic</li>
      <li><a href="#">HTML</a></li>
      <li><a href="#">CSS</a></li>
      <li><a href="#">JavaScript</a></li>
      <li class="divider"></li>
      <li class="dropdown-header">Advance</li>
      <li><a href="#">Bootstrap</a></li>
      <li><a href="#">jQuery</a></li>
      <li><a href="#">Xml</a></li>
  </ul>
</div>
```

Output



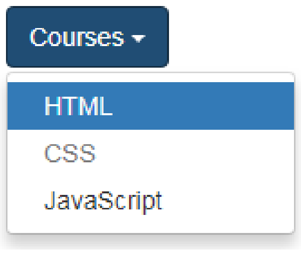
Disable and Active items

This creates a class file with the same name; this is the bytecode form of Java program. Indicate a specific dropdown item from dropdown menu with the `.active` class. Disable any item inside the dropdown menu, use the .disabled class.

Example of Disable and Active items

```
<div class="dropdown">
  <button class="btn btn-primary dropdown-toggle" type="button" data-
toggle="dropdown">Courses
  <span class="caret"></span></button>
  <ul class="dropdown-menu">
    <li class="active"><a href="#">HTML</a></li>
    <li class="disabled"><a href="#">CSS</a></li>
    <li><a href="#">JavaScript</a></li>
  </ul>
</div>
```

Output

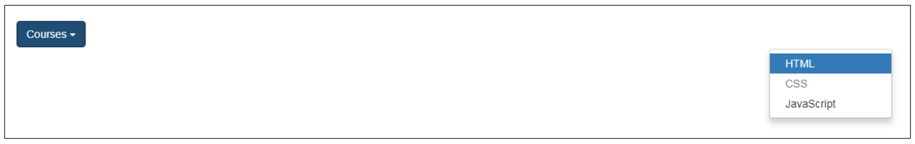


Dropdown Position

To right-align the dropdown, add the `.dropdown-menu-right` class to the element with .dropdown-menu:

Use `<ul class="dropdown-menu dropdown-menu-right">` to create right alignment dropdown from main menu.

Output



Dropup

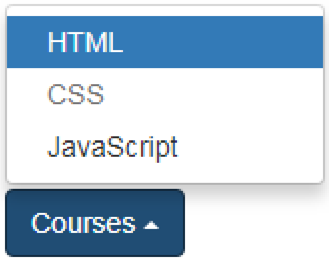
The dropdown menu to slide upwards instead of downwards, change the <div> element with class="`dropdown`" to `"dropup"`.

```
<div class="dropup">
```

Example of Dropdown Header

```
<div class="dropup">
  <button class="btn btn-primary dropdown-toggle" type="button" data-toggle="dropdown">Courses
  <span class="caret"></span></button>
  <ul class="dropdown-menu">
    <li class="active"><a href="#">HTML</a></li>
    <li class="disabled"><a href="#">CSS</a></li>
    <li><a href="#">JavaScript</a></li>
  </ul>
</div>
```

Output



Bootstrap Navbar

Navigation Bars

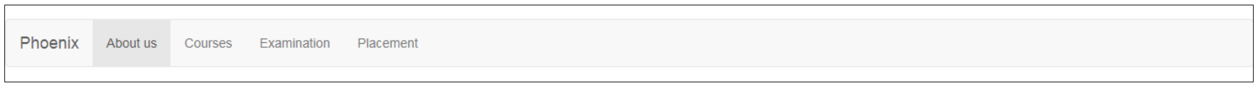
A navigation bar is a navigation menu which is placed at the top of the page. A navigation bar is responsive feature, it can extend or collapse, depending on the screen size.

Example of Navigation Bars

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>
<nav class="navbar navbar-default">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">Phoenix</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">About us</a></li>
      <li><a href="#">Courses</a></li>
      <li><a href="#">Examination</a></li>
      <li><a href="#">Placement</a></li>
    </ul>
  </div>
</nav>
```

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

Output



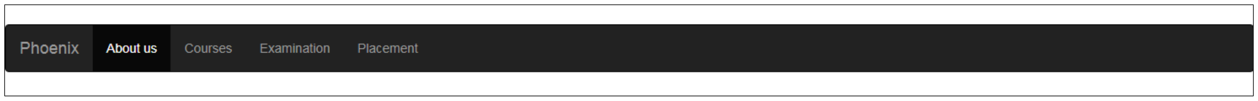
Inverted Navigation Bar

The style of the default navigation bar is bright and light gray, Bootstrap provides an alternative, black navbar with inverse color using `.navbar-inverse`.

Example of Inverted Navigation Bar

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">Phoenix</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">About us</a></li>
      <li><a href="#">Courses</a></li>
      <li><a href="#">Examination</a></li>
      <li><a href="#">Placement</a></li>
    </ul>
  </div>
</nav>

</body>
</html>
```

Output



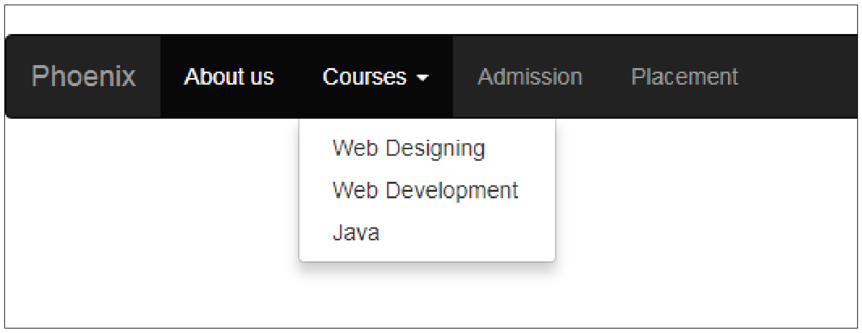
Navigation Bar with Dropdown

Inside of navigation bars you can create a dropdown menu.

Example of Navigation Bar with Dropdown

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">Phoenix</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">About us</a></li>
      <li class="dropdown">
        <a class="dropdown-toggle" data-toggle="dropdown" href="#">Courses
        <span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="#">Web Designing</a></li>
          <li><a href="#">Web Development</a></li>
          <li><a href="#">Java</a></li>
        </ul>
      </li>
      <li><a href="#">Admission</a></li>
      <li><a href="#">Placement</a></li>
    </ul>
  </div>
</nav>
</body>
</html>
```

Output



Note : Bootstrap predefined 260 glyphicons from the Glyphicons Halflings set.
Glyphicons can be used to create text, buttons, toolbars, navigation, forms, and many more.

Envelope glyphicon: ✉

Print glyphicon: 🖶

Search glyphicon: 🔍

Download glyphicon: ⊕

Glyphicon Syntax

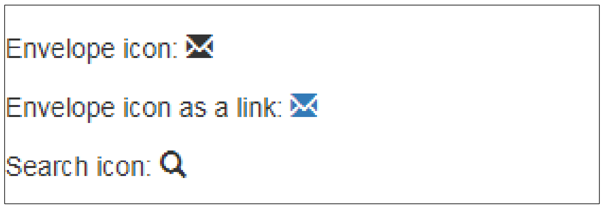
The following is a syntax of glyphicon:

```
<span class="glyphicon glyphicon-name"></span>
```

Example of Glyphicon

```
<p>Envelope icon: <span class="glyphicon glyphicon-envelope"></span></p>
<p>Envelope icon as a link:
  <a href="#"><span class="glyphicon glyphicon-envelope"></span></a>
</p>
<p>Search icon: <span class="glyphicon glyphicon-search"></span></p>
```

Output



```
Envelope icon: ✉

Envelope icon as a link: ✉

Search icon: 🔍
```

Bootstrap Navbar Collapsible

This navigation bar will automatically fit them according to the screen size and collapsed menu items inside of it. It takes up too much space on a small screen.

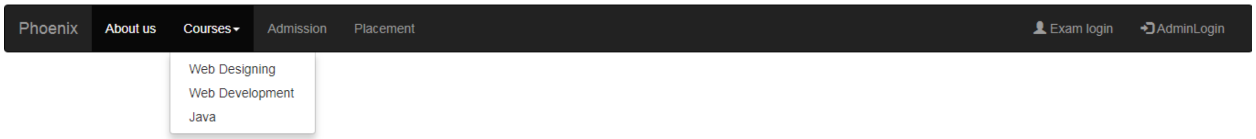
Example of Navbar Collapsible

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
```

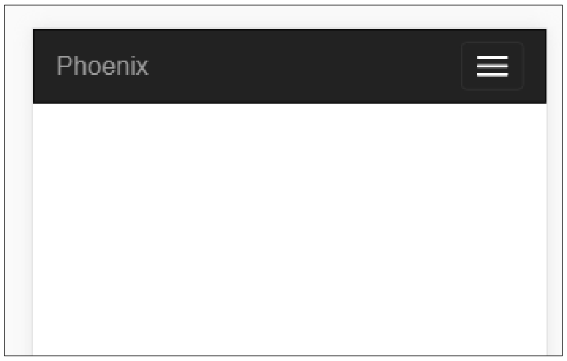
```
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target="#myNavbar">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Phoenix</a>
    </div>
    <div class="collapse navbar-collapse" id="myNavbar">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">About us</a></li>
         <li class="dropdown">
        <a class="dropdown-toggle" data-
toggle="dropdown" href="#">Courses<span class="caret"></span></a>
        <ul class="dropdown-menu">
          <li><a href="#">Web Designing</a></li>
          <li><a href="#">Web Development</a></li>
          <li><a href="#">Java</a></li>
        </ul>
      </li>

        <li><a href="#">Admission</a></li>
        <li><a href="#">Placement</a></li>
      </ul>
      <ul class="nav navbar-nav navbar-right">
        <li><a href="#"><span class="glyphicon glyphicon-
user"></span> Exam login</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-log-
in"></span> AdminLogin</a></li>
      </ul>
    </div>
  </div>
</nav>
```

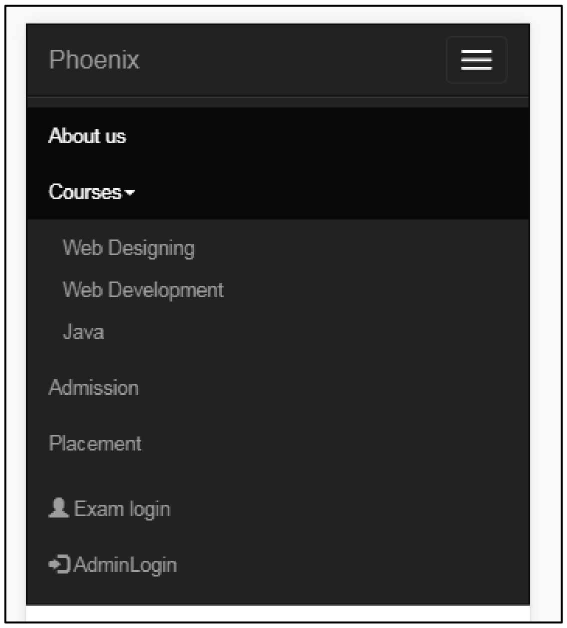
```
</body>
</html>
```

Output (Desktop)



Output (Mobile or Small Screen)





Bootstrap Form

Form controls automatically receive some global styling with Bootstrap.

All textual `<input>`, `<textarea>`, and `<select>` elements with class `.form-control` have a width of 100%.

Bootstrap Form Layouts

Three types of form layouts:

- Vertical form (this is default)
- Horizontal form
- Inline form

Rules for all three form layouts:

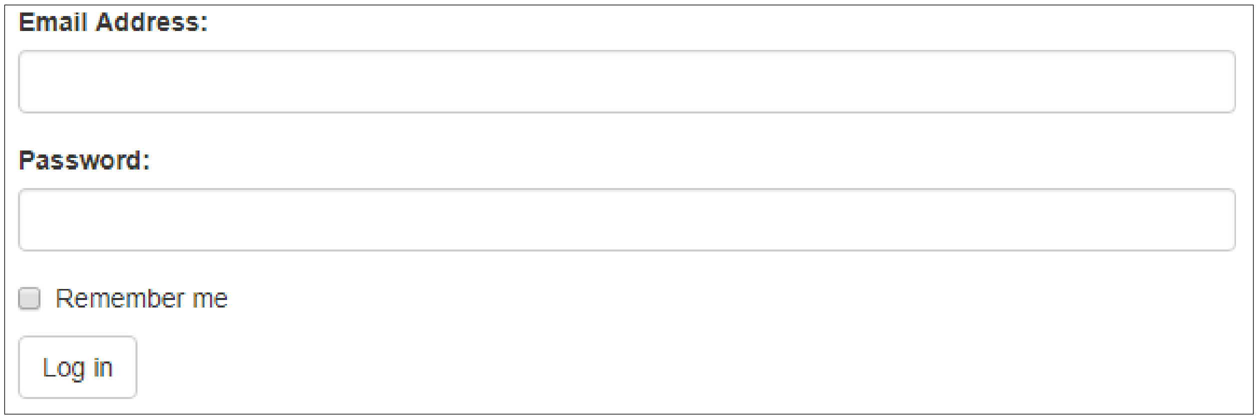
- Wrap labels and form controls in `<div class="form-group">` needed for spacing
- Add class `.form-control` to all textual `<input>`, `<textarea>`, and `<select>` elements

Example of Vertical Form (Default)

Vertical form with two input fields, one checkbox, and a submit button -

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>
<form class="col-md-6">
  <div class="form-group">
    <label for="email">Email Address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <div class="checkbox">
    <label><input type="checkbox"> Remember me</label>
  </div>
  <button type="submit" class="btn btn-default">Log in</button>
</form>
</body>
</html>
```

Output



Inline Form

Inline form, all html elements are inline, left-aligned, and the labels are alongside.
Rule for an inline form.

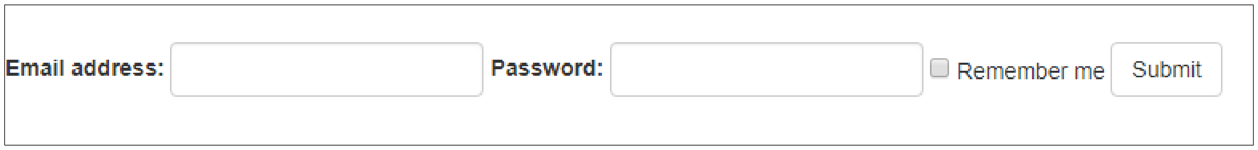
Add class `.form-inline` to the `<form>` element.

Example of Inline Form

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
```

```
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>
<form class="form-inline" action="/action_page.php">
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <div class="checkbox">
    <label><input type="checkbox"> Remember me</label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
</body>
</html>
```

Output



Bootstrap Horizontal Form

A horizontal form means that the labels are aligned beside the input field on large and medium screens. On small screens (767px and below), it will transform to a vertical form (labels are placed on top of each input).

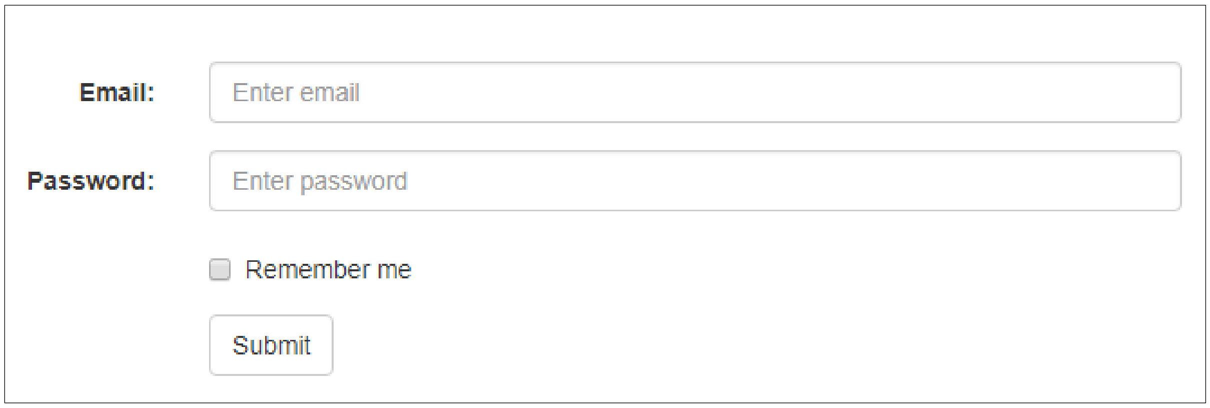
Rules for a horizontal form:

- Add class `.form-horizontal` into the `<form>` element
- Add class `.control-label` into all `<label>` elements

Example of Bootstrap Horizontal Form

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>

<form class="form-horizontal" action="/action_page.php">
  <div class="form-group">
    <label class="control-label col-sm-2" for="email">Email:</label>
    <div class="col-sm-10">
      <input type="email" class="form-
control" id="email" placeholder="Enter email">
    </div>
  </div>
  <div class="form-group">
    <label class="control-label col-sm-2" for="pwd">Password:</label>
    <div class="col-sm-10">
      <input type="password" class="form-
control" id="pwd" placeholder="Enter password">
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
      <div class="checkbox">
        <label><input type="checkbox"> Remember me</label>
      </div>
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
      <button type="submit" class="btn btn-default">Submit</button>
    </div>
  </div>
</form>
</body>
</html>
```

Output



Bootstrap Modal

The Modal Plugin

The Modal plugin is a dialog box or a popup window which is displayed on top of the current page just like JavaScript alert box.

Plugins can be included individually or all at once (using "bootstrap.js" or "bootstrap.min.js").

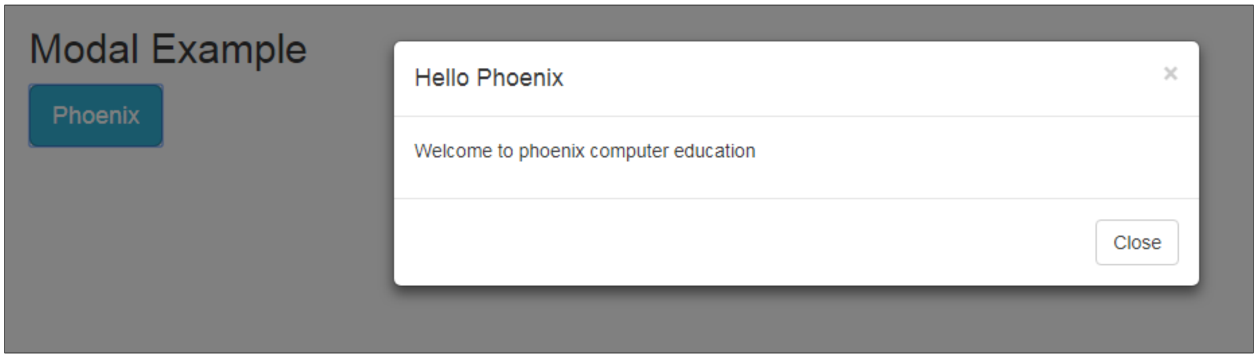
Modal window, you need to use a `<button>` or `<a>` link.

Example of Basic Modal

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>
<div class="container">
  <h2>Modal Example</h2>
<!-- Trigger the modal with a button -->
<button type="button" class="btn btn-info btn-lg" data-
toggle="modal" data-target="#myModal">Phoenix</button>

<!-- Modal -->
<div id="myModal" class="modal fade" role="dialog">
  <div class="modal-dialog">

    <!-- Modal content-->
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-
dismiss="modal">&times;</button>
        <h4 class="modal-title">Hello Phoenix</h4>
      </div>
      <div class="modal-body">
        <p>Welcome to phoenix computer education<p>
      </div>
      <div class="modal-footer">
```

```
        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
      </div>
    </div>
  </div>
 </div>
</div>
</body>
</html>
```

Output



Modal Size

Change of the modal by `.modal-sm` class for small modals and `.modal-lg` class for large modals.

Add the size class to the `<div>` element with class `.modal-dialog`:

Small Modal

```
<div class="modal-dialog modal-sm">
```

Large Modal

```
<div class="modal-dialog modal-lg">
```

Try this by yourself inside of the current modal.

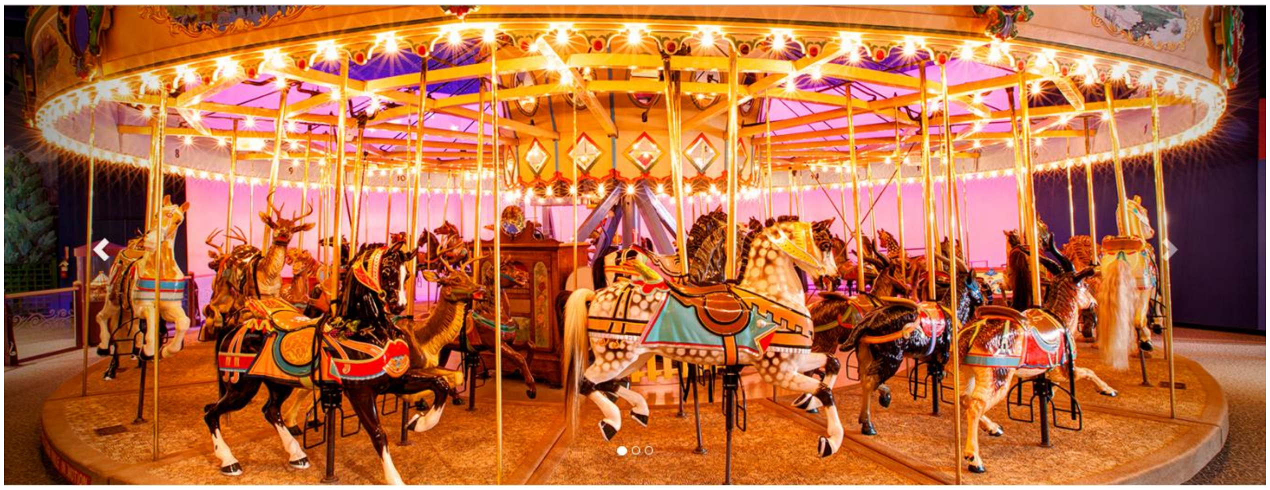
Bootstrap Carousel

The Carousel is a plugin to create component for image or text slidshow.

Example of Carousel

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstra
p/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jque
ry.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/boot
strap.min.js"></script>
</head>
<body>

<div id="myCarousel" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
  </ol>

  <!-- Wrapper for slides -->
  <div class="carousel-inner">
    <div class="item active">
      <img src="carousel.jpg" alt="carousel" class="img-responsive">
    </div>

    <div class="item">
      <img src="Carousel-Night-Hero-Image.jpg" alt="Carousel-Night"
class="img-responsive">
    </div>
    <div class="item">
      <img src="ExhibitsIconsPlaces_Carousel_Large_003.jpg"
alt="ExhibitsIconsPlaces" class="img-responsive">
    </div>
  </div>

  <!-- Left and right controls -->
  <a class="left carousel-control" href="#myCarousel" data-slide="prev">
    <span class="glyphicon glyphicon-chevron-left"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="right carousel-control" href="#myCarousel" data-slide="next">
    <span class="glyphicon glyphicon-chevron-right"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
</body>
</html>
```

Output



jQuery

Introduction to jQuery

jQuery is a fast, small, easy to access and feature-rich browser compatible JavaScript library. jQuery makes things like HTML,CSS document traversal and manipulation, event handling, animation, and Ajax much simpler with an API that works across a multiple browsers. In one sentence jQuery means do more write less.

Pre-requisite for jQuery

- HTML
- CSS
- JavaScript

jQuery Features

jQuery is to make it much easier to use JavaScript. jQuery simplify the JavaScript, like AJAX calls and DOM manipulation.

The jQuery features are given below

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

jQuery Library

There are 2 types jQuery library you can use -

- Online library from jQuery site
- Download jQuery from jQuery.com

Online library

Copy the jQuery CDN file from Google of Microsoft .(Content Delivery Network).

Both Google and Microsoft provides jQuery.

Example of jQuery from Google or Microsoft, based CDN -

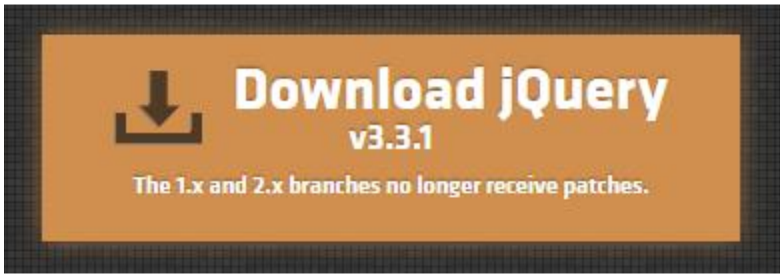
```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js
"></script><!--- Google CDN--->
```

Or

```
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-
3.3.1.min.js"></script><!--- Microsoft CDN--->
</head>
```

Download jQuery

1. Go to jQuery.com site
2. Click Download jQuery (v3.X.X)



Inside of the page there are two type versions of jQuery available:

- Production - it has been minified and compressed for faster loading.
- Development version - this is for testing or development

3. After download copy the file inside the project folder.
4. Link them with html file .

```
<head>
<script src="jquery-3.3.1.min.js"></script>
</head>
```

jQuery Syntax

The jQuery syntax is made for **selecting** HTML elements and creating some **action** on the element(s) to create more interaction with minimal script typing.

jQuery basic syntax: **$(*selector*).*action*()**

- A $ sign to access jQuery
- A (*selector*) to find HTML elements (elements, Id Selector, Class selector)
- A jQuery *action*() to be performed on the element line show and hide, click on button animation etc.

Example of Syntax writing

```
$(document).ready(function(){

   // jQuery methods go here...

});
```

Where jquery **`$(document).ready(function()`** statement means your document is ready to execute javascript/jquery code from jquery library .

jQuery Selectors

jQuery selectors allow us to select and manipulate HTML elements like ID selector, class selector or direct through tags.

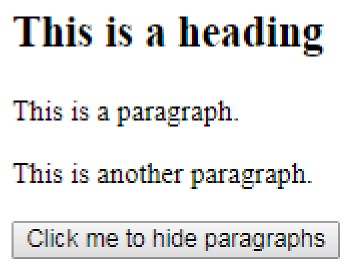
All selectors start with the dollar($) sign and parentheses: $().

Example of Selector

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>jQuery Example</title>
  <meta charset="utf-8">
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js
"></script>
<script>
```

```
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
});
</script>
</head>
<body>
<h2>This is a heading</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<button>Click me to hide paragraphs</button>
</body>
</html>
```

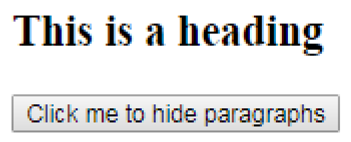
Output


This is a heading

This is a paragraph.

This is another paragraph.

After Click on the button:


This is a heading

The id Selector

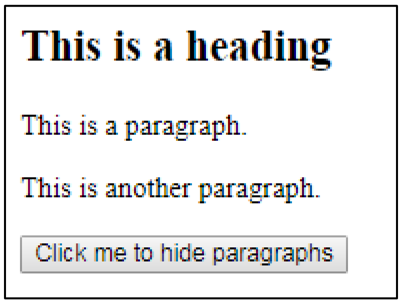
An id is an unique selector within the page, so we should use the id selector by #(hash).

Example of id Selector

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
  <title>jQuery Example</title>
  <meta charset="utf-8">
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js
"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#test").hide();
    });
});</script>
</head>
<body>
<h2>This is a heading</h2>

<p id=”test”>This is a paragraph.</p>
<p id=”test”>This is another paragraph.</p>

<button>Click me to hide paragraphs</button>
</body>
</html>
```

Output

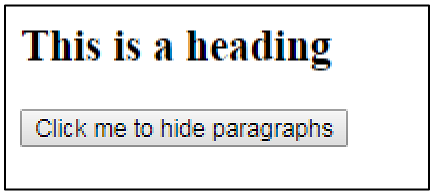


This is a heading

This is a paragraph.

This is another paragraph.

After Click on the button:



This is a heading

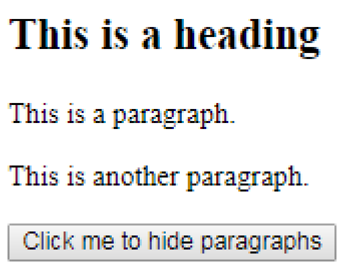
The class Selector

Find the html elements with a specific class and write a period character. Class must be defined by dot (.)

Example of class Selector

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>jQuery Example</title>
  <meta charset="utf-8">
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js
"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $(".test").hide();
    });
});</script>
</head>
<body>
<h2>This is a heading</h2>

<p class="test">This is a paragraph.</p>
<p class="test">This is another paragraph.</p>

<button>Click me to hide paragraphs</button>
</body>
</html>
```

Output

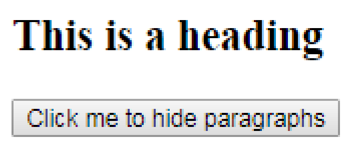


This is a heading

This is a paragraph.

This is another paragraph.

After Click on the button:



jQuery Events

What are Events?

An event represents the action moment when something happens or how an jQuery program will be execute.

Types of Event Example

- Moving a mouse over element.
- Select a radio button or click a radio button.
- Click on an element or mouse over or mobile gesture.

Some Common DOM Events

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

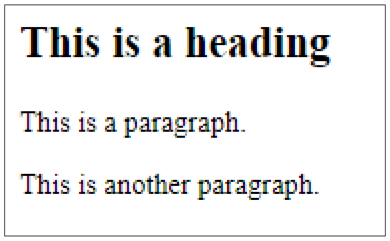
Event Methods

Most DOM events have a significance jQuery method for html elements.

Example of Event Methods

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>jQuery Example</title>
  <meta charset="utf-8">
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery
.min.js"></script>
< script >
$(document).ready(function(){

     $("p").click(function(){
         $(this).hide();
     });
});
< /script >

</head>
<body>
<h2>This is a heading</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

Output



This is a heading

This is a paragraph.

This is another paragraph.

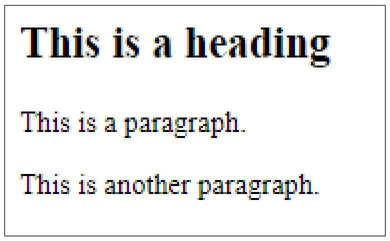
dblclick()

The dblclick() method is to run an event handler function to an HTML element by double click.

Example of dblclick()

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>jQuery Example</title>
  <meta charset="utf-8">
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery
.min.js"></script>
< script >
$(document).ready(function(){
      $("p").dblclick(function(){
          $(this).hide();
      });
});
< /script >

</head>
<body>
<h2>This is a heading</h2>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

</body>
</html>
```

Output



```
This is a heading

This is a paragraph.

This is another paragraph.
```

mouseenter()

The mouseenter() an event handler function to an HTML element for execute the function when the mouse pointer enters the HTML element:

Example of mouseenter ()

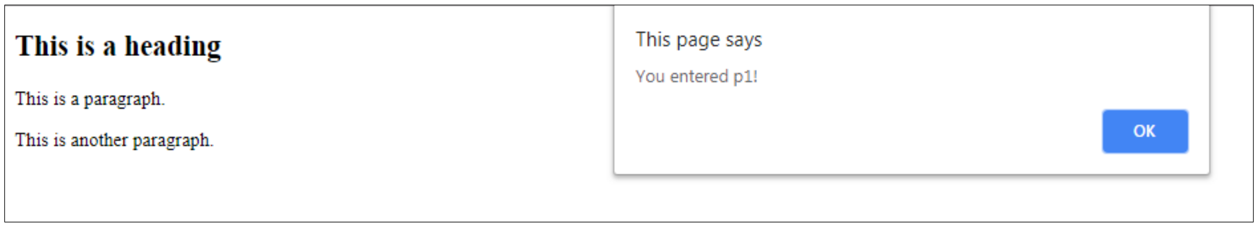
// script

```
< script >
$(document).ready(function(){
      $("#p1").mouseenter(function(){
          alert("You entered p1!");
      });
});
< /script >
```

// body

```
<p id=”p1”>You enter p1.</p>
```

Output


This is a heading

This is a paragraph.

This is another paragraph.

This page says

You entered p1!

mouseleave()

Example of mouseleave()

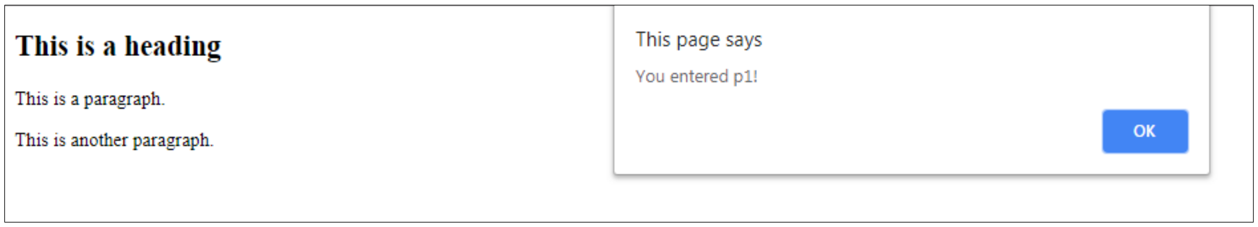
// script

```
< script >
$(document).ready(function(){
      $("#p1").mouseleave(function(){
          alert("You entered p1!");
      });
});
< /script >
```

// body

```
<p id=”p1”>You enter p1.</p>
```

Output



This is a heading

This is a paragraph.

This is another paragraph.

This page says

You entered p1!

mouseup()

Example of mouseup()

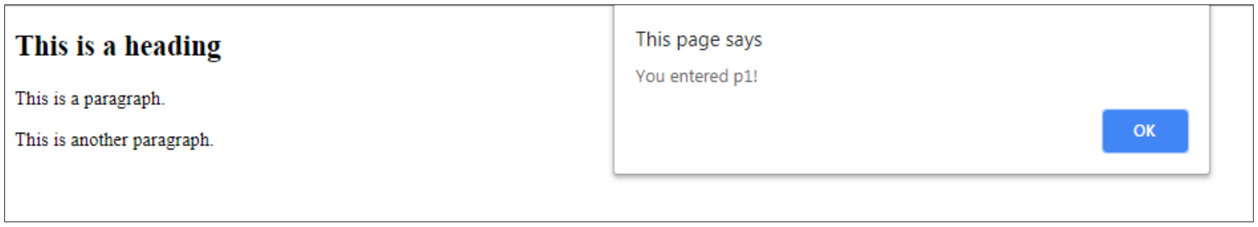
// script

```
< script >

$(document).ready(function(){
      $("#p1").mouseup(function(){
          alert("You entered p1!");
      });
});
< /script >
```

// body

```
<p id=”p1”>You enter p1.</p>
```

Output



This is a heading

This is a paragraph.

This is another paragraph.

This page says

You entered p1!

hover()

Example of hover()

// script

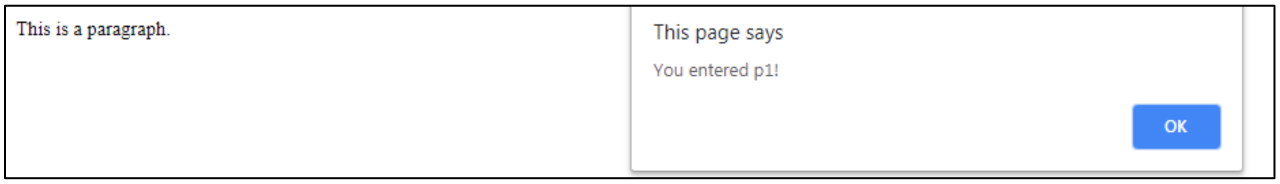
```
< script >
$(document).ready(function(){
      $("#p1").hover(function(){
          alert("You entered p1!");
      },
      function(){
          alert("Bye! You now leave p1!");
      });});
< /script >
```

// body

```
<p id=”p1”>You enter p1.</p>
```

Output

After hover

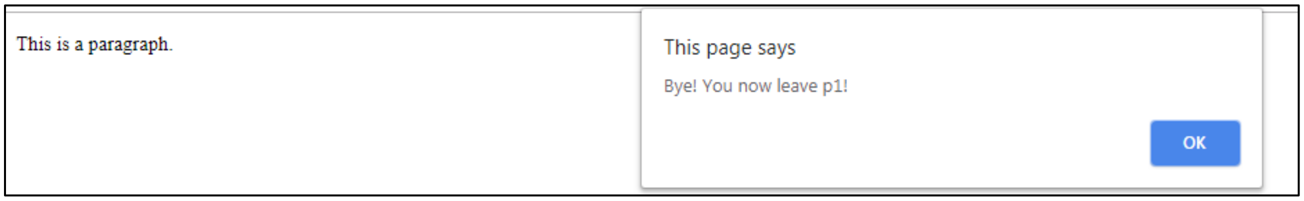


This is a paragraph.

This page says

You entered p1!

After click ok



This is a paragraph.

This page says

Bye! You now leave p1!

focus()

The `focus()`an event handler function to an HTML form field is executed when the form field gets focus.

Example of focus()

// Script

```
< script >

$(document).ready(function(){
      $("#p1").hover(function(){
          alert("You entered p1!");
      },
      function(){
          alert("Bye! You now leave p1!");
      });});
< /script >
```

// body

```
<p id=”p1”>You enter p1.</p>
```

blur()

The `blur()` method works when the form field loses focus.

Example of blur()

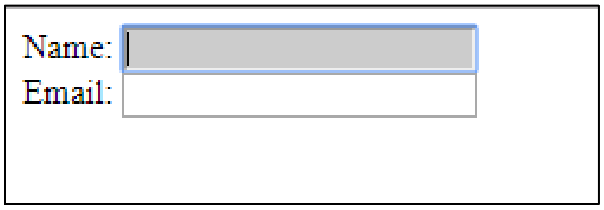
// Script

```
<script>

$(document).ready(function(){

   $("input").focus(function(){

         $(this).css("background-color", "#cccccc");

     });

   $("input").blur(function(){

         $(this).css("background-color", "#ffffff");

     });

});

</script>
```

// body

```
Name: <input type="text" name="fullname"><br>

Email: <input type="text" name="email">
```

Output:



on() Method

The on( ) method combine one or more event handlers at a time for the selected elements and execute .

Example of on() Method

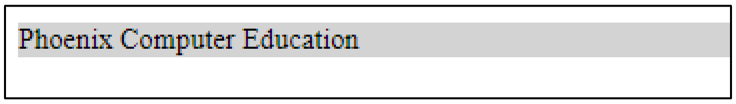
// Script

```
<script>
$(document).ready(function(){
$("p").on({
mouseenter: function(){
$(this).css("background-color", "lightgray");
},
mouseleave: function(){
$(this).css("background-color", "lightblue");
},
click: function(){
$(this).css("background-color", "yellow");
}
});
});
</script>
```

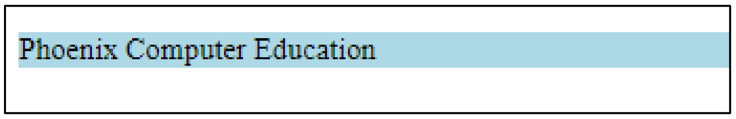
// Body

```
<p>Phoenix Computer Education</p>
```

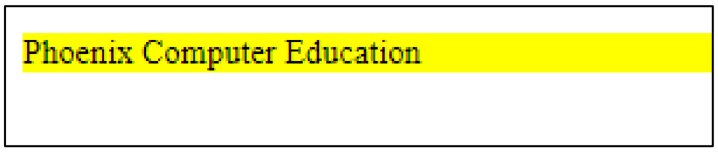
Output:


Phoenix Computer Education

Mouse Enter


Phoenix Computer Education

Mouse leave


Phoenix Computer Education

Mouse click

jQuery Show Hide

Hide and show any HTML elements with the `hide()` and `show()` methods individually or create toggle effects of show and hide.

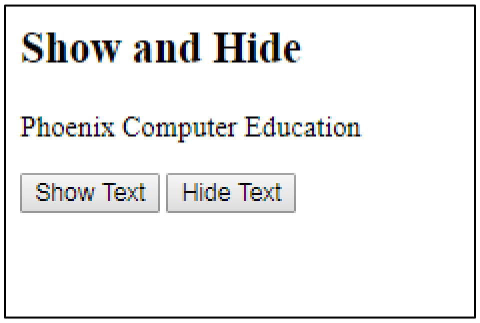
Example of Show Hide

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>jQuery Example</title>
  <meta charset="utf-8">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery
.min.js">

</script>

<script>

$(document). ready(function(e) {

$("#hide").click(function(){
    $("p").hide();
});

$("#show").click(function(){
    $("p").show();
});

});
```

```
</script>
</head>
<body>
<h2>Show and Hide </h2>
<p> Phoenix Computer Education</p>
<button id=”show” >Show Text</button>
<button id=”hide” >Hide Text</button>
</body>
</html>
```

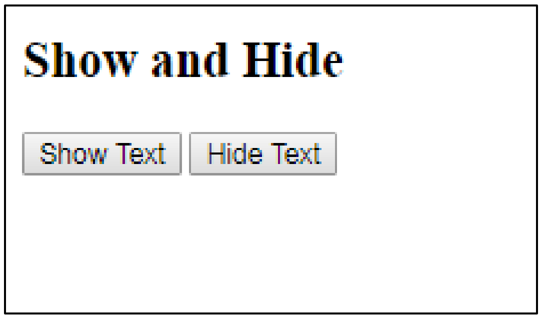
Output

Click on hide button to hide the text


Show and Hide

Phoenix Computer Education

Click on Show button to show text


Show and Hide

Show and Hide with Speed and Duration

The optional speed property to specify the speed of the hiding/showing, by using the values: "slow", "fast", or milliseconds.

Example of Show Hide with Speed and Duration

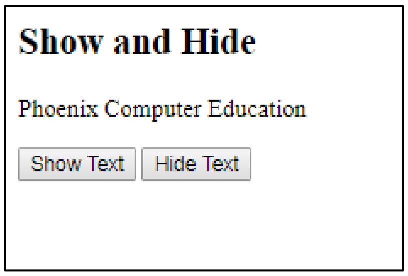
```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>jQuery Example</title>
  <meta charset="utf-8">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery
.min.js">

</script>

<script>

$(document). ready(function(e) {

$("#hide").click(function(){
    $("p").hide(“fast”);
});

$("#show").click(function(){
    $("p").show(“slow”);
});

});

</script>

</head>

<body>

<h2>Show and Hide </h2>

<p> Phoenix Computer Education</p>

<button id=”show” >Show Text</button>

<button id=”hide” >Hide Text</button>
</body>

</html>
```

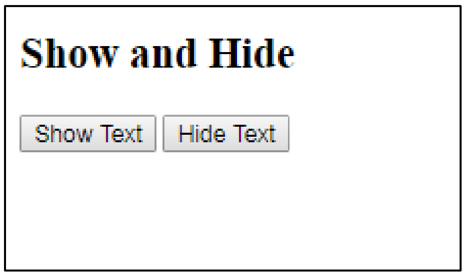
Or you can replace the property value with

```
$("#hide").click(function(){
    $("p").hide(1000);// millisecond
});
```

Output

Click on hide button to hide the text





Click on Show button to show text

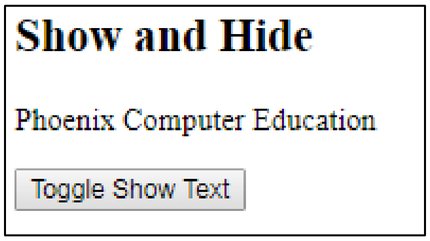
Show and Hide toggle

Toggle between the hide() and show() methods with a single toggle() method in a button.

Example of Show and Hide toggle

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>jQuery Example</title>
  <meta charset="utf-8">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery
.min.js">

</script>

<script>

$(document). ready(function(e) {

$("#tog").click(function(){
    $("p"). toggle();
});
});

</script>

</head>

<body>

<h2>Show and Hide </h2>

<p> Phoenix Computer Education</p>

<button id="tog" >Toggle Show Text</button>
</body>

</html>
```

Output


Show and Hide

Phoenix Computer Education

jQuery Slide

Using jQuery you can create a sliding effect on html elements.

jQuery has three slide methods -

- `slideDown()`
- `slideUp()`
- `slideToggle()`

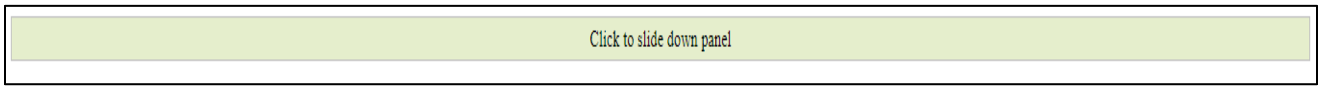
slideDown() Method

The jQuery `slideDown()` method is used to create slide down effects fro html elements just like dropdown menu.

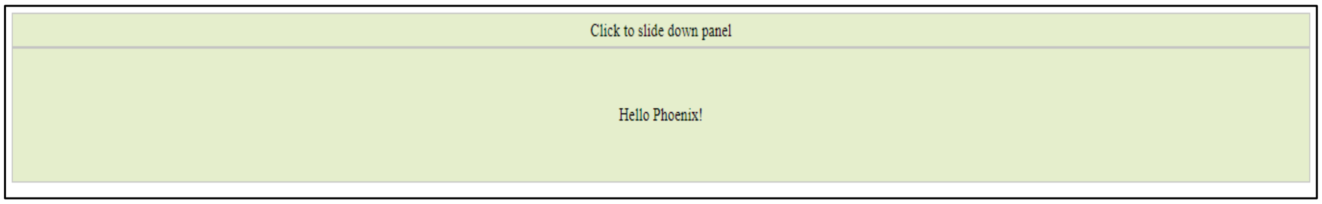
Example of slideDown() Method

```
<!DOCTYPE html>

<html>

<head>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
></script>

<script>

$(document).ready(function(){

    $("#flip").click(function(){

        $("#panel").slideDown("slow");

    });
});
</script>

<style>
#panel, #flip {
    padding: 5px;
    text-align: center;
    background-color: #e5eecc;
    border: solid 1px #c3c3c3;
}

#panel {
    padding: 50px;
    display: none;
}
</style>
</head>
<body>
<div id="flip">Click to slide down panel</div>
<div id="panel">Hello Phoenix</div>
</body>
</html>
```

Output



Afterclick



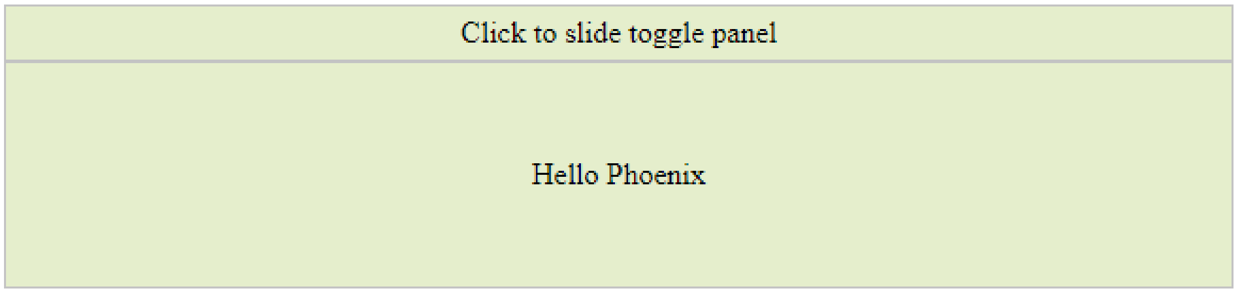
slideToggle() Method

slideToggle() method toggles between the slideDown() and slideUp(). When the elements have been slid down, slideToggle() will slide them up.When the elements have been slid up, slideToggle() will slide them down.

Example of slideToggle() Method

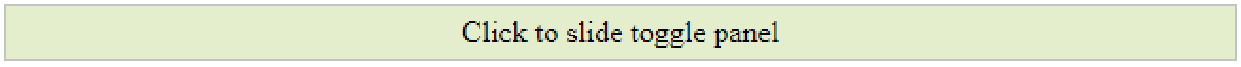
```
<html>

<head>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
></script>

<script>

$(document).ready(function(){

    $("#flip").click(function(){

        $("#panel").slideToggle("slow");

    });

});

</script>


<style>

#panel, #flip {

    padding: 5px;

    text-align: center;
    background-color: #e5eecc;
    border: solid 1px #c3c3c3;
}

#panel {
    padding: 50px;
    display: none;
}
</style>
</head>
<body>
<div id="flip">Click to slide toggle panel</div>
<div id="panel">Hello Phoenix</div>
</body>
</html>
```

Output



```
Click to slide toggle panel

Hello Phoenix
```

Then click again



```
Click to slide toggle panel
```

jQuery Animation

jQuery animate() method lets you create multiple animations and also you can customize them.

animate() Method

jQuery animate() method is used to create custom animations and it is need CSS for animation . Before use this animation you must learn about CSS.

Syntax:

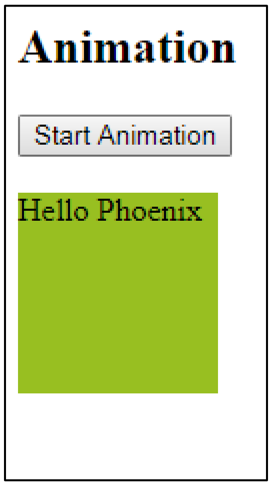
```
$(selector).animate({params},speed,callback);
```

Example of animate() Method

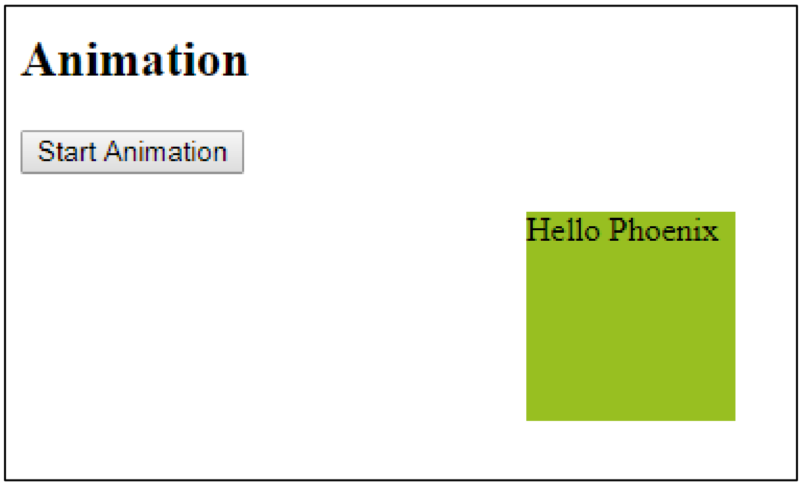
```
<html>

<head>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
></script>

<script>

$(document).ready(function(){

           $("button").click(function(){

           $("div").animate({left: '250px'});    });

});

</script>

</head>

<body>

<h2>Animation </h2>

<button>Start Animation</button>

<div
style="background:#98bf21;height:100px;width:100px;position:absolute;"
>Hello Phoenix</div>

</body>
```

```
</html>
```

Output



After click



Manipulate Multiple Properties

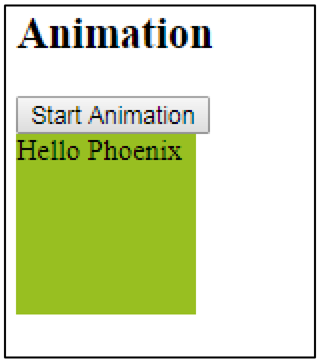
multiple properties can be animated at the same time inside of the same element

Example of Manipulate Multiple Properties

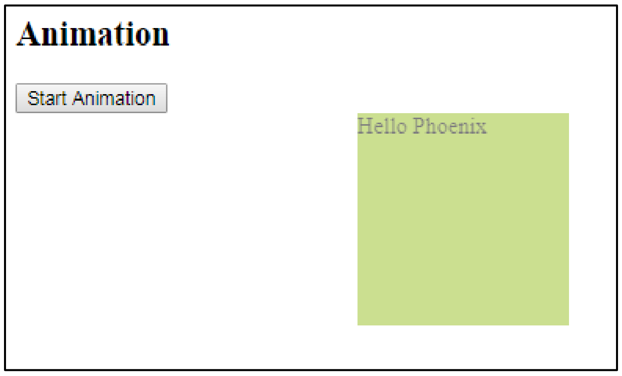
```
<html>

<head>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
></script>

<script>

$(document).ready(function(){

$("button").click(function(){
    $("div").animate({
        left: '250px',
        opacity: '0.5',
        height: '150px',
        width: '150px'
      });
```

```
   });
 });
</script>
</head>
<body>
<h2>Animation </h2>
<button>Start Animation</button>
<div
style="background:#98bf21;height:100px;width:100px;position:absolute;"
>Hello Phoenix</div>
</body>
</html>
```

Output



After click



jQuery Stop

Stop Animations

jQuery stop() method will stop any animation or effect before it is finished in a particular duration.

Syntax:

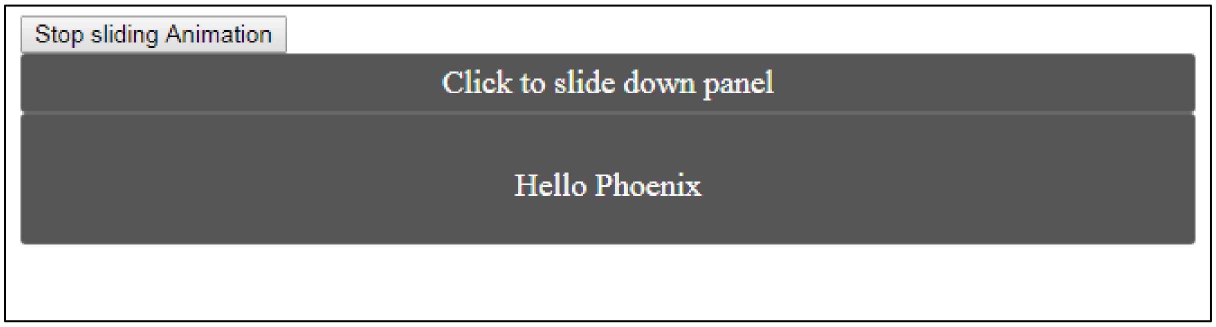
```
$(selector).stop(stopAll,goToEnd);
```

Example of Stop Animations

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
></script>
<script>
$(document).ready(function(){
    $("#flip").click(function(){
        $("#panel").slideDown(5000);
    });
    $("#stop").click(function(){
        $("#panel").stop();
    });
});
</script>

<style>
#panel, #flip {
    padding: 5px;
    font-size: 18px;
    text-align: center;
    background-color: #555;
    color: white;
    border: solid 1px #666;
    border-radius: 3px;
}

#panel {
    padding: 50px;
    display: none;
}
</style>
</head>
<body>

<button id="stop">Stop sliding Animation</button>
```

```
<div id="flip">Click to slide down panel</div>
<div id="panel">Hello Phoenix </div>
</body>
</html>
```

Output



jQuery Fade

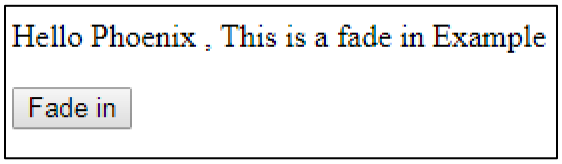
fadeIn() Method

fadeIn() method generally changes the alpha value, for selected elements, from transparent to visible.

Example of fadeIn() Method

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
></script>
<script>
$(document).ready(function(){
$("p").hide()
    $(".btn2").click(function(){
        $("p").fadeIn(1000);
    });
});
</script>
</head>
<body>
<p>Hello Phoenix , This is a fade in Example</p>
<button class="btn2">Fade in</button>
</body>
</html>
```

Output



Hello Phoenix , This is a fade in Example

After clicking fade in button

fadeOut() Method

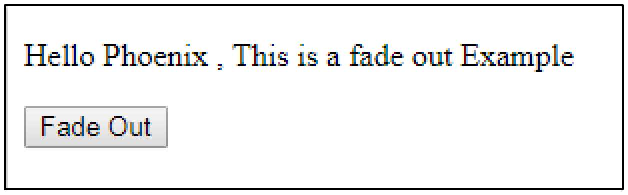
fadeIn() method generally changes the alpha value, for selected elements, from visible to transparent.

Example of fadeOut() Method

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
></script>
```

```
<script>
$(document).ready(function(){
    $(".btn2").click(function(){
        $("p").fadeOut(1000);
    });
});
</script>
</head>
<body>
<p>Hello Phoenix , This is a fade out Example</p>
<button class="btn2">Fade Out</button>
</body>
</html>
```

Output


Hello Phoenix , This is a fade out Example

Method Chaining

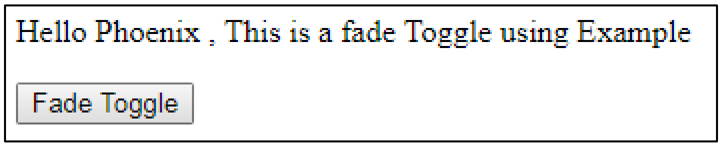
It allows us to run multiple jQuery methods on the same element within a single statement.

Example of Method Chaining

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
></script>
<script>
```

```
$(document).ready(function(){
    $(".btn2").click(function(){
        $("p").fadeTo('slow', 0.0).fadeTo('slow', 1.0);
    });
});
</script>
</head>
<body>
<p>Hello Phoenix , This is a fade Toggle using  Example</p>
<button class="btn2">Fade Toggle</button>
</body>
</html>
```

Output


Hello Phoenix , This is a fade Toggle using Example

jQuery hasClass

hasClass() Method

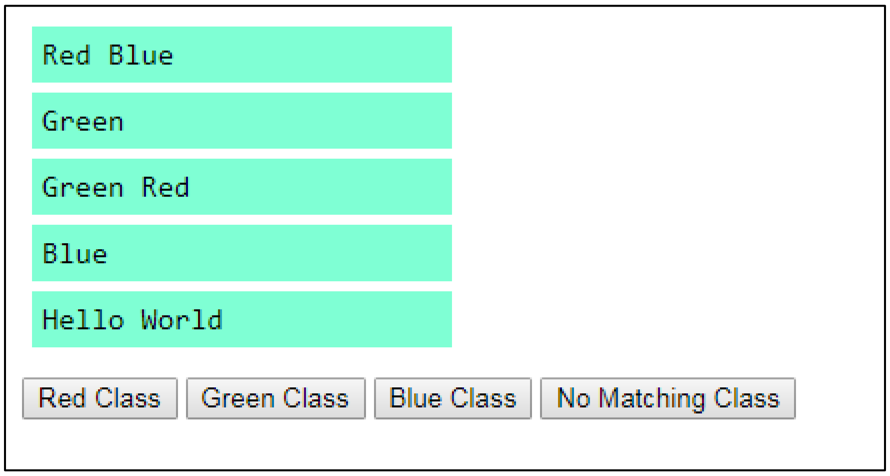
The hasClass() method checks the selected elements have a particluer class name.

Example of hasClass() Method

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
<script src="jquery-3.3.1.min.js"></script>
<script>
     $(document).ready(function() {
           $(".btn").click(function(){
           var className = $(this).attr("id");
           $("ul li").each(function() {
           if ($(this).hasClass(className)) {
           $(this).fadeTo('slow', 0.0).fadeTo('slow', 1.0);
           }
```

```
                });
            });
        });
</script>
<style>
ul{
font-family: monospace;
font-size: 15px;
font-style: normal;
font-size-adjust: none;
width:200px;
padding:0px;
}
ul li{
background-color:#7fffd4;
margin:5px;
padding:5px;
list-style-type:none;
width:200px;
}
</style>
</head>
<body>
<ul>
<li class="red blue">Red Blue</li>
<li class="green">Green</li>
<li class="green red">Green Red</li>
<li class="blue">Blue</li>
<li class="noclass">Hello World</li>
</ul>
<input type="button" class="btn" value="Red Class" id="red">
<input type="button" class="btn" value="Green Class" id="green">
<input type="button" class="btn" value="Blue Class" id="blue">
<input type="button" class="btn" value="No Matching Class"
id="noclass">
</body>
</html>
```

Output



```
Red Blue
Green
Green Red
Blue
Hello World
```

Jquery UI

jQuery UI create multiple combination of interaction, effects, widgets, utilities, and themes designed to work well together.

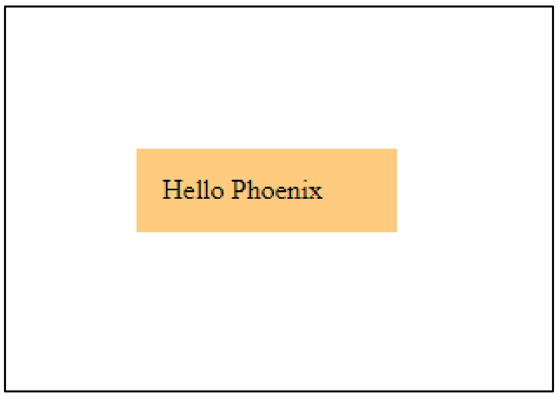
Draggable

Allow elements to be moved using the mouse.

Example of Draggable

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Draggable - Default functionality</title>
  <link rel="stylesheet"
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <style>
  #draggable { width: 150px; height: 150px; padding: 0.5em; }
  </style>
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-
ui.js"></script>
  <script>
  $( function() {
    $( "#draggable" ).draggable();
  } );
  </script>
  <style>
  p{
        background:rgba(255,153,0,.5);
        padding:15px
  }
  </style>
</head>
<body>

<div id="draggable" class="ui-widget-content">
  <p>Hello Phoenix</p>
</div>
</body>
</html>
```

Output



```
Hello Phoenix
```

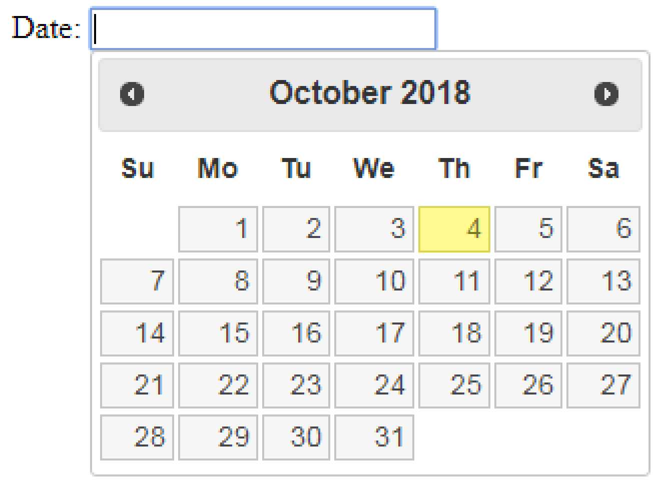
Datepicker

Select a date from a popup or inline calendar

Example of Datepicker

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Datepicker - Default functionality</title>
  <link rel="stylesheet"
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  <link rel="stylesheet" href="/resources/demos/style.css">
  <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
  <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
  <script>
  $( function() {
    $( "#datepicker" ).datepicker();
  } );
  </script>
</head>
<body>
<p>Date: <input type="text" id="datepicker"></p>
</body>
</html>
```

Output



Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

UI Animation

Using jQuery UI create animation by less typing code.

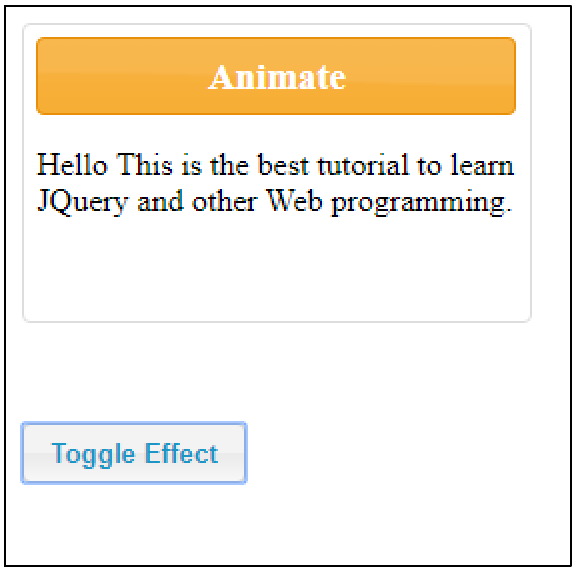
Example of UI Animation

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
<link href="http://code.jquery.com/ui/1.10.4/themes/ui-
lightness/jquery-ui.css" rel="stylesheet">
<script src="http://code.jquery.com/jquery-1.10.2.js"></script>
<script src="http://code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
<script>
$(function() {
var state = true;
$( "#button" ).click(function() {
if ( state ) {
$( "#effect" ).animate({
```

```
backgroundColor: "#aa0000",
color: "#fff",
width: 500
}, 1000 );
} else {
$( "#effect" ).animate({
backgroundColor: "#fff",
color: "#000",
width: 240
}, 1000 );
}
state = !state;
});
});
</script>
<style>
.toggler { width: 500px; height: 200px; position: relative; }
#button { padding: .5em 1em; text-decoration: none; }
#effect { width: 240px; height: 135px; padding: 0.4em; position:
relative; background: #fff; }
#effect h3 { margin: 0; padding: 0.4em; text-align: center; }
</style>
</head>
<body>
<div class="toggler">
<div id="effect" class="ui-widget-content ui-corner-all">
<h3 class="ui-widget-header ui-corner-all ">Animate</h3>
<p>Hello This is the best tutorial to learn JQuery and other Web
programming.</p>
</div>
</div>
```

```
<button id="button" class="ui-state-default ui-corner-all">Toggle
Effect</button>
</body>
</html>
```

Output



Hello This is the best tutorial to learn JQuery and other Web programming.

Click on toggle effect button to view the animation

XML

Introduction of XML

XML stands for **Extensible Markup Language**. XML is a text-based markup language derived from Standard Generalized Markup Language (SGML). XML tags are used to store and organize the data, rather than specifying how to display it like HTML tags.

Features of XML

1. XML allows us to create our own self-descriptive tags or language.
2. XML allows us to store the data irrespective of how it will be presented.
3. XML was developed by an organization called the World Wide Web Consortium (W3C) and is available as an open standard.
4. XML uses a DTD (Document Type Definition) to formally describe the data.
5. All XML elements must have a closing tag.
6. XML tags are case sensitive.
7. All XML elements must be properly nested.

Difference between XML and HTML

1. XML was designed to carry data, with focus on what data is.
2. HTML was designed to display data, with focus on how data looks.
3. XML tags are not predefined. HTML tags are predefined.
4. XML tags are case sensitive whereas HTML tags are not case sensitive.

XML Syntax

All elements can have sub elements (child elements).

```
<root>
  <child>
```

```
     <subchild>.....</subchild>
  </child>
</root>
```

Example of XML Syntax

```
<?xml version="1.0" encoding="UTF-8"?>
<person>
   <name>
      <fname>Navin</fname>
      <lname>Shah</lname>
   </name>
   <address>
      <company>Phoenix Enterprise</company>
      <city>Kolkata</city>
      <pin>700029</pin>
   </address>
   <job>Professional</job>
</person>
```

Description of Above Code

The first line in the document defines the XML version of the document. In this case the document conforms to the 1.0 specification of XML -

```
<?xml version="1.0" encoding="UTF-8"?>
```

To avoid errors, we should specify the encoding used, or save our XML files as UTF-8. UTF-8 is the default character encoding for XML documents. It is not a part of the XML document.

The next line is the root element of the document -

```
<person>
```

The next line starts a `<name>` element. The `<name>` elements have 2 child elements:

```
<fname>Navin</fname>
<lname>Shah</lname>
```

The `<address>` elements have 3 child elements -

```
<company>Phoenix Enterprise</company>
<city>Kolkata</city>
<pin>700029</pin>
```

The last line ends the `</person>` element.

Note: use elements rather than using attributes.

Using attributes -

```
<date>18/03/2018</date>
```

Using elements -

```
<note>
  <date>
    <day>18</day>
    <month>03</month>
    <year>2018</year>
  </date>
  <to>Sumit</to>
  <from>Kaushik</from>
  <heading>Invitation</heading>
  <body>Please come in the party</body>
</note>
```

Well Formed XML

XML documents that suite the syntax rules above are said to be Well Formed XML documents.

Internal DTD

A DTD is referred to as an internal DTD if elements are declared within the XML files. To refer it as internal DTD, **standalone** attribute in XML declaration must be set to yes. This means, the declaration works independent of external source.

Syntax

```
<!DOCTYPE root-element [element-declarations]>
```

root-element is the name of root element and element-declarations is where we declare the elements.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE address [
  <!ELEMENT address (name,company,email,mob)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT company (#PCDATA)>
  <!ELEMENT email (#PCDATA)>
  <!ELEMENT mob (#PCDATA)>
]>
<address>
  <name>Dinesh Agarwal</name>
  <company>Phoenix</company>
  <mob>(033) 2123-4567</mob>
</address>
```

Description of Above Code

```
<!DOCTYPE address [
```

The DOCTYPE declaration has an exclamation mark (!) at the start of the element name. The DOCTYPE informs the parser that a DTD is associated with this XML document.

DTD Body - The DOCTYPE declaration is followed by body of the DTD, where we declare elements, attributes, entities and notations –

```
<!DOCTYPE address [
  <!ELEMENT address (name,company,email,mob)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT company (#PCDATA)>
  <!ELEMENT email (#PCDATA)>
  <!ELEMENT mob (#PCDATA)>
]>
```

<!ELEMENT name (#PCDATA)> defines the element name to be of type "#PCDATA". Here #PCDATA means **parse-able text data**.

The declaration section of the DTD is closed using a closing bracket and a closing angle bracket ( ]>). This effectively ends the definition and thereafter, the XML document follows immediately.

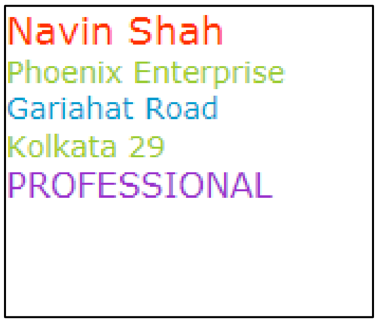
Example of Internal DTD

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<?xml-stylesheet type="text/css" href="ex.css" ?>
<!DOCTYPE person[
   <!ELEMENT person (name,address,job)>
   <!ELEMENT name (fname,lname)>
   <!ELEMENT fname (#PCDATA)>
   <!ELEMENT lname (#PCDATA)>
   <!ELEMENT address (add1+,add2+)>
   <!ELEMENT add1 (#PCDATA)>
   <!ELEMENT add2 (#PCDATA)>
   <!ELEMENT job (#PCDATA)>
]>
<person>
  <name>
    <fname>Navin</fname>
    <lname>Shah</lname>
  </name>
  <address>
    <add1>Phoenix Enterprise</add1>
    <add2>Gariahat Road</add2>
    <add1>Kolkata 29</add1>
  </address>
  <job>PROFESSIONAL</job>
</person>
```

Save this above file as **Test.xml**. Now we write the css code to provide the style to xml document. Save the css file as **ex.css**. To link the css file with the xml document, write the css file name in `href="ex.css"` as mentioned above.

```
name
 {
    text-align:left;
    font-family:Verdana, Geneva, sans-serif;
    font-size:18px;
    color:#F30;
    display:block;
 }
add1
 {
    text-align:left;
    font-family:Verdana, Geneva, sans-serif;
    font-size:14px;
    color:#9C3;
    border:thin;
    display:block;
 }
add2
 {
    text-align:left;
    font-family:Verdana, Geneva, sans-serif;
    font-size:14px;
     color:#09C;
    border:thin;
 }
job
 {
    text-align:left;
    font-family:Verdana, Geneva, sans-serif;
    font-size:16px;
    color:#93C;
    display:block;
 }
```

Output



```
Navin Shah
Phoenix Enterprise
Gariahat Road
Kolkata 29
PROFESSIONAL
```

External DTD

In external DTD elements are declared outside the XML file. They are accessed by specifying the system attributes which may be either the legal **.dtd file** or a valid URL. To refer it as external DTD, standalone attribute in the XML declaration must be set as no. This means, declaration includes information from the external source.

Syntax

```
<!DOCTYPE root-element SYSTEM "file-name">
```

where file-name is the file with .dtd extension.

Declaring Attributes

An attribute declaration has the following syntax –

```
<!ATTLIST element-name attribute-name attribute-type attribute-value>
```

Example of External DTD

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE product[
<!ELEMENT product (item) >
<!ELEMENT item (name,desc,price,qty) >
<!ELEMENT name (#PCDATA) >
<!ELEMENT desc (#PCDATA) >
<!ELEMENT price(#PCDATA) >
<!ELEMENT qty (#PCDATA) >
<!ATTLIST product pid category (BOOK)>
]>
```

Save this above file as Test.dtd.

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<?xml-stylesheet type="text/css" href="ex.css" ?>
```

```
<!DOCTYPE product SYSTEM "Test.dtd" >
<product pid="P001" category="BOOK" >
  <name>Let Us C</name>
  <desc>C programming</desc>
  <price>350</price>
  <qty>20</qty>
</product>
```

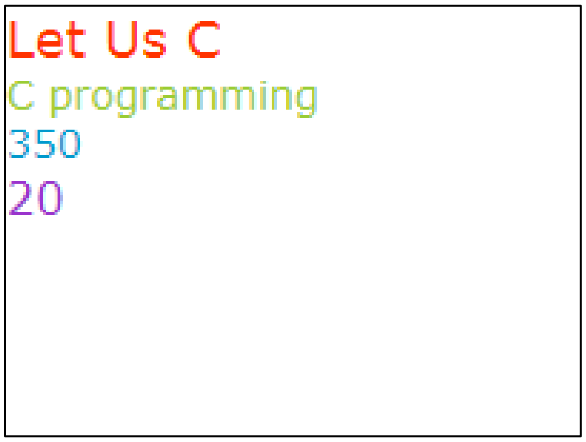
Save this above file as Test.xml.

```
name
 {
    text-align:left;
    font-family:Verdana, Geneva, sans-serif;
    font-size:18px;
    color:#F30;
    display:block;
 }
desc
 {
     text-align:left;
    font-family:Verdana, Geneva, sans-serif;
    font-size:14px;
    color:#9C3;
    border:thin;

    display:block;
 }
price
```

```
 {
    text-align:left;
    font-family:Verdana, Geneva, sans-serif;
    font-size:14px;
    color:#09C;
    border:thin;
 }
qty
 {
    text-align:left;
    font-family:Verdana, Geneva, sans-serif;
    font-size:16px;
    color:#93C;
    display:block;
 }
```

Save this above file as ex.css.

Output



```
Let Us C
C programming
350
20
```