

HTML 5

What is HTML5?

- HTML 5 is the latest version of HTML
- It is developed to address the latest web requirements
- In 2004 W3C was concentrating on the development of XHTML 2.0, which is the controlled version of HTML 4.1 with no backward compatibility and new additions to address latest requirements.
- In 2004 by frustrated with the approach of W3C, a group of browser manufacturers like apple, Mozilla, Opera and corporations like Microsoft and some of the web professionals formed WAHTWG (Web Hypertext Application Technology Work Group) to address latest trends
- WAHTWG started its own specification called Web Application 1.0 to continue the development of HTML with required additions.
- In 2007 W3C adapted WHATWG specification and named as HTML 5

HTML TimeLine

1995	HTML 2.0
1997	HTML 3.2
1998	Web Standard Project
1999	HTML 4.0
2000	HTML 1.0
2004	HTML 2.0
2004	WHATWG Web Application 1.0
2007	W3C renamed WAHTWG Web Application as HTML5.0
2010	Apple, Google, Firefox and IE supported and promoted usage of HTML 5

Benefits with HTML5

- We can develop web pages in more organized way
- Search engine friendly web pages can be developed
- It contains many features to facilitate the development of applications
- We no need to use adobe flash and Microsoft Silverlight that results slow page loading
- HTML5 provides native support to audio and video elements to play youtube videos and podcasts
- HTML5 retains its backward compatibility with previous versions of HTML
- The HTML specification contains explicit rules about parsing documents and error handling

HTML 4 Vs HTML 5

1. DOC type is so simple

```
<!DOCTYPE HTML>
```

2. Character coding meta tag is so simple

```
<meta charset="UTF-8"/>
```

3. New elements

Article, aside, audio, canvas, command, datalist, details, embed, figcaption, figure, footed, header, hgroup, keygen, mark, meter, nav, output, progress, rip, ruby, section, source, summery, time, video

4. Web Forms 2.0

5. New Controls

Date Picker, color Picker, Numeric Stepper

6. New Input Controls

E-mail, Search, URL

7. PUT and DELETE methods

8. Integrated API's

Video, Audio, Offline API's, Editable, Drag & Drop, History, Protocols, Geo Location, Web Messaging

A simple program in HTML4.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>lgate Solutions</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <link type="text/css" rel="stylesheet" href="mystyles.css">
    <script type="text/javascript" src="animation.js"></script>
  </head>
  <body>
    <h1>Welcome to lgate Solutions</h1>
    <p>
    </p>
  </body>
</html>
```

A simple program in HTML5

```
<!doctype html>
<html>
```

```
<head>
<title>Igate Solutions</title>
<meta charset="utf-8">
<link rel="stylesheet" href="mystyles.css">
<script src="animation.js"></script>
</head>
<body>
<h1>Welcome to Igate Solutions</h1>
<p>
  -----
  -----
</p>
</body>
</html>
```

HTML 5 Tags

<figure>

HTML5 includes a new tag for marking figures: <figure>. It is not a replacement for , rather it is a container into which we place an tag. The main advantage to using <figure> is that we can use the <figcaption> tag to associate a caption with the image. The caption will stick with the image.

Note: These tags <figure> and <figcaption> have no attributes, hence need to use styles.

Example:

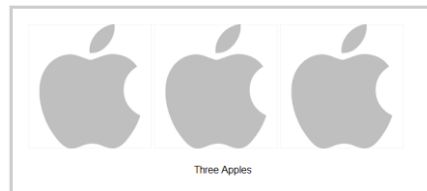
```
<!DOCTYPE html>
<html>
<head>
  <title>Igate Solutions</title>
  <style>
    figure{
      border:5px solid #ccc;
      padding:25px;
      margin:10px auto;
      width: 630px;
    }
    figcaption{
      text-align: center;
      font:12pt Arial;
      padding-top:20px;
    }
  </style>
</head>
<body>
  <figure>
    <img alt="A placeholder for an image." data-bbox="111 596 344 886"/>
    <figcaption>
      A figure with a caption.
    </figcaption>
  </figure>
</body>
</html>
```

```

    </style>
</head>
<body>
<figure>
    
    
    
    <figcaption>Three Apples</figcaption>
</figure>
</body>
</html>

```

Output:



```
<nav>
```

The `<nav>` tag is used to define a set of navigation links. All links of a document may not be inside a `<nav>` element. The `<nav>` element is intended only for major blocks of navigation links. Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

Note: This tag has no attributes in particular, but can be formatted using styles

Example:

```

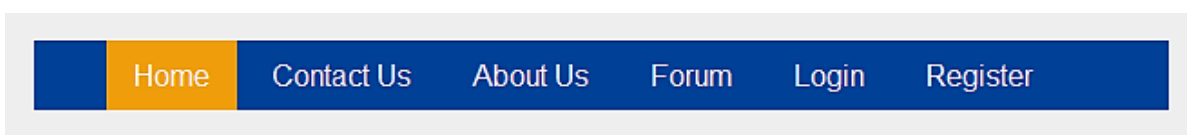
<!DOCTYPE html>
<html>
  <head>
    <title>Igate Solutions</title>
    <meta charset="utf-8">

    <style>
body{
  background: #eee;
}
#container{
  width: 80%;
  margin: 0 auto;

```

```
}
nav{
  font: 1em Arial;
  list-style: none;
  background-color: #004097;
  text-transform: capitalize;
}
nav li{
  display: inline-block;
}
nav li a{
  display: block;
  color: #fff;
  text-decoration: none;
  line-height: 1.8em; /* important*/
  padding: 5px 15px 4px 15px;
}
nav li a:hover{
  background-color:#F09D0C;
}
</style>
</head>
<body>
  <div id="container">
    <nav>
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">Contact Us</a></li>
        <li><a href="#">About Us</a></li>
        <li><a href="#">Forum</a></li>
        <li class='selected'><a href="#">Login</a></li>
        <li><a href="#">Register</a></li>
      </ul>
    </nav>
  </div>
</body>
</html>
```

Output:



HTML 5 Semantic Tags

Semantic element is an element which gives meaning by its name. Initially tables were used to design and structure a webpage. In place of <table> tag <div> is been using to structure a web page. HTML 5 has introduced some of the semantic elements to structure the page. Using of these tags give more meaning to the structure of a webpage. These tags give more meaning to the Search engines while indexing.

Note: It also helps to search engines to identify and give priority to the contents of the web page.

<header> Defines the masthead or other header information on the page. Typically the header is repeated on every page of a site, although that is not required.

<footer> Defines the text at the bottom of a page, such as the copyright or contact information. Again, it is typically repeated on every page of the site.

<article> Defines a block of text that represents a single article, story, or message. An article can be distinguished from other text in that it can logically stand alone. For example, on a news site, each news story is an article.

<aside> Defines a block of text that is different to the main discussion, such as a note, tip, or caution. An aside can be distinguished from other text in that it could be pulled out and discarded without disrupting the main document in which it appears.

<section> Defines a generic content or application section. Examples of sections would be book chapters or the numbered sections of a thesis; a site's home page could be split into sections such as Introduction, News, and Contact Information. A section begins with a heading such as <h1> followed by other content. A general rule is to use <section> if the area being defined would be included in an outline of the document or page.

Example: **Footer**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Igate Solutions</title>
    <style>
body{
  background: #eee;
}
#container{
  width: 80%;
  margin: 0 auto;
}
```

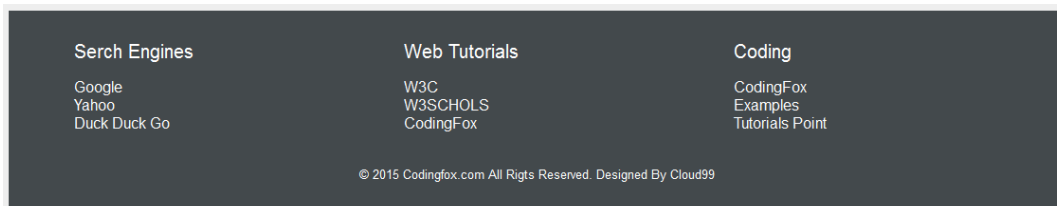
```
footer{
  background-color: #43494C;
  font:12pt Arial;
  color: white;
  padding:20px;
}
footer section{
  position: relative;
  float: left;
  padding:10px 49px;
  min-width: 250px;
}
footer section h1{
  font:15pt Arial;
  margin: 0px;
}
footer section ul{
  list-style: none;
  line-height:1.2;
}
}
footer section ul li{
  margin-left: -40px;
}
.clear{
  clear: both;
}
</style>
</head>
<body>
  <div id="container">
    <footer>
      <section>
        <h1>Serch Engines</h1>
        <ul>
          <li>Google</li>
          <li>Yahoo</li>
          <li>Duck Duck Go</li>
        </ul>
      </section>
      <section>
        <h1>Web Tutorials</h1>
        <ul>
          <li>W3C</li>
```

```

        <li>W3SCHOLS</li>
        <li>CodingFox</li>
    </ul>
</section>
<section style="border-right: none;">
    <h1>Coding</h1>
    <ul>
        <li>CodingFox</li>
        <li>Examples</li>
        <li>Tutorials Point</li>
    </ul>
</section>
<div class="clear"></div>
<section style="font:10pt Arial;text-align: center;margin: 0px auto;float: none;">
    &copy; 2015 Codingfox.com All Rigts Reserved.
    Designed By Cloud99
</section>
<div class="clear"></div>
</footer>
</div>
</body>
</html>

```

Output:



Example: **Page Layout**

```

<!DOCTYPE html>
<html>
  <head>
    <title>Igate Solutions</title>
    <style>
body{
  font: 0.8em Arial,sans-serif;
}
h1, ul{
  margin: 0;
  padding: 0;

```



```
}

.clear{
  clear: both;
}

#wrap{

  width: 940px;
  padding: 10px;
  margin: 0 auto;
}
head{
  margin-bottom: 10px;
}
nav{
  background-color: #004097;
}
nav ul{
  list-style: none;
  margin-bottom: 0px;
  float: left;
}
nav ul li{
  float: left;
}
nav ul li a{
  text-decoration: none;
  color: #fff;
  display: block;
  padding: 5px 30px;
  line-height: 1.8;
}
nav ul li a:hover{
  background: #F09D0C;
  color: #fff;
}
article section{
  width: 680px;
  float: left;
  padding: 10px;
}
article h1{
  font-size: 1.6em;
```

```
}
article,aside{
  min-height: 200px;
}
aside{
  width: 219px;
  float: right;
  padding: 10px;
  border-left: 1px solid #aaa;
}
footer{
  background: #004097;
  color: #fff;
  padding: 10px 10px;
}
</style>
</head>
<body>
  <div id="wrap">
    <head>
      <h2>igate Solutions</h2>
    </head>
    <nav>
      <ul>
        <li> <a href="#">Home</a></li>
        <li><a href="#">About Us</a></li>
        <li><a href="#">Contact Us</a></li>
        <li><a href="#">Refer a friend</a></li>
      </ul>
      <div class="clear"></div>
    </nav>
    <article>
      <section>
        <h1>This is our site</h1>
      </section>
      <aside> This is a side bar.</aside>
    </article>
    <div class="clear"/>
    <footer>
      This is footer
    </footer>
  </div>
</body>
</html>
```

Output:

Igate Solutions



Adding video to the web page

Before HTML 5, <embed> was the tag used to embed a video clip in a web page. Here the video clip would be loaded and played by a third party player like adobe flash player or silver light. It requires installation of a third party plugin.

Now HTML 5 provides a tag called <video> by using which we can directly place and play a video in a browser without using the flash player. However old versions of browsers don't support <video> tag. Hence code must be written to support backward compatibility.

Browsers that support different formats

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES from Firefox 21 from Firefox 3.0 for Linux	YES	YES
Safari	YES	NO	NO
Opera	YES From Opera 2.5	YES	YES

- MP4 = MPEG 4 files with H264 video codec and AAC audio codec
- WebM = WebM files with VP8 video codec and Vorbis audio codec
- Ogg = Ogg files with Theora video codec and Vorbis audio codec

MIME Types for video formats

Format	MIME-type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

Attributes:

Attribute	Value	Description
autoplay	autoplay	Specifies that the video will start playing as soon as it is ready
controls	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
height	pixels	Sets the height of the video player
loop	loop	Specifies that the video will start over again, every time it is finished
muted	muted	Specifies that the audio output of the video should be muted
poster	URL	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
src	URL	Specifies the URL of the video file
width	pixels	Sets the width of the video player

Example:

```
<video width="320" height="240" controls poster="../images/pic.png">
<source src="myvideo.mp4" type="video/mp4">
<source src="myvideo.ogg" type="video/ogg">
<source src="myvideo.webm" type="video/webm">
Your Browser Could not Play the Video
</video>
```

Here "mp4" format is played if the page is loaded into Internet Explorer, Chrome and Safari, "ogg/ogv" format is played if the page is loaded into Firefox and Opera.

"Your Browser Could not Play the Video" is displayed if browser fails to load the video by any reason.

Even include <embed> tag to load video even by the old browsers.

```
<video width="320" height="240" controls poster="../images/pic.png">
<source src="myvideo.mp4" type="video/mp4">
```

```
<source src="myvideo.ogg" type="video/ogg">
<source src="myvideo.webm" type="video/webm">
<embed src="myvideo.mp4" type="application/x-shockwave-flash" width="320" height="240"
allowscripaccess="always"
allowfullscreen="true">
Your Browser Could not Play the Video
</video>
```

Embedding a YouTube Video with in a page:

We can embed a YouTube video within a web page by <iframe>.

```
<iframe width="854" height="480" src="https://www.youtube.com/embed/a-605ZGfeec"
frameborder="0" allowfullscreen></iframe>
```

Adding an Audio file

Inserting an audio file is similar to inserting video file in a web page except using of <audio> tag in place of <video> tag.

Here **mp3** and **ogg** formats are commonly supported by most of the browsers.

Browser	MP3	Wav	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

MIME types of audio types

Format	MIME-type
MP3	audio/mpeg
Ogg	audio/ogg
Wav	audio/wav

Example:

```
<audio autoplay loop>
<source src="myaudio.mp3" type="audio/mpeg">
<source src="myaudio.ogg" type="audio/ogg">
<embed src="myaudio.mp3">
```

<progress>

It is used to display the total progress of a task using progress bar

Attributes:

Attribute	Value	Description
max	<i>number</i>	Specifies how much work the task requires in total
value	<i>number</i>	Specifies how much of the task has been completed

<meter>

The <meter> tag is used to define a scalar measurement within a known range, or a fractional value. This is also known as a gauge. It is generally used to show the status of usage, the relevance of a query result, etc.

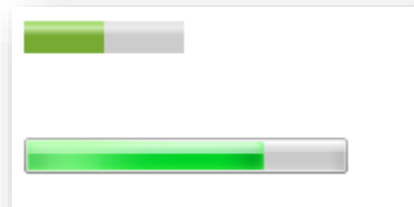
Attributes:

Attribute	Value	Description
max	<i>number</i>	Specifies how much work the task requires in total
value	<i>number</i>	Specifies how much of the task has been completed

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Meter and Progress Elements</title>
  </head>
  <body>
    <meter min="0" max="100" value="50">
      Your browser does not support the meter element!
    </meter>
    <br>
    <progress value="75" max="100">
      Your browser does not support the progress element!
    </progress>
  </body>
</html>
```

Output:



<details> (Collapsible Text)

It is introduced from HTML 5. It is used to display collapsible text. By default summary would be displayed, would be expanded on click on the summary.

Note: It is only supported in Chrome, Safari and Opera

Attribute:

Attribute	Value	Description
open	open	Specifies that the details should be visible (open) to the user

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Collapasible Text</title>
    <style>
      details summary{
        font: 12pt Arial;
        color: #aaa;
        margin:10px 100px;
      }
      details p{
        font: 12pt Arial;
        margin: 10px 100px;
        text-indent: 100px;
        width: 500px;
      }
    </style>
```

```
</head>
<body>
  <details>
    <summary>lgate Solutions</summary>
    <p>Some Text Here..Some Text Here....</p>
  </details>
</body>
</html>
```

Output:

▼ Igate Solutions

[illegible]

```
<iframe>
```

`<iframe>` is used to embed a website in another website. It is also used to embed a YouTube video and google maps etc.

Property	Purpose
width	It specifies the width of iframe
height	It specifies the height of iframe
src	Source of web page, YouTube video, Google map
name	To get used with target attribute of an anchor tag
align	left, right, center, top, bottom, middle

Embedding a website:

```
<div class="container">
  <iframe src="http://www.codingfox.com" scrolling="yes"></iframe>
</div>
```

Embedding a YouTube Video

Select the share button under any YouTube Video

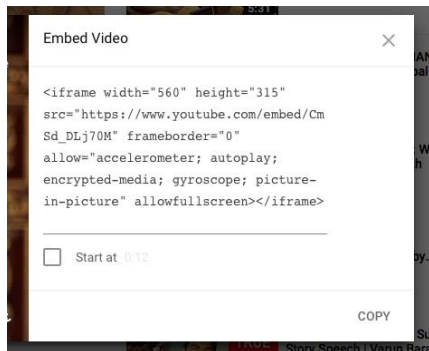


Select embed button



Embed

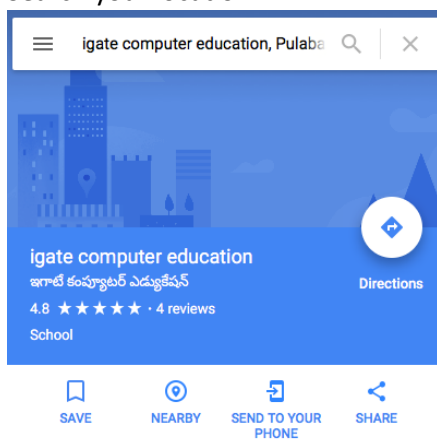
Copy the <iframe> code generated by the YouTube



Embedding google map

Open the google maps using <https://maps.google.com>

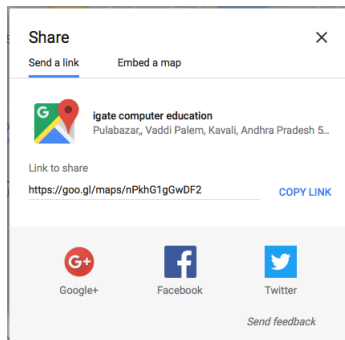
Search your location



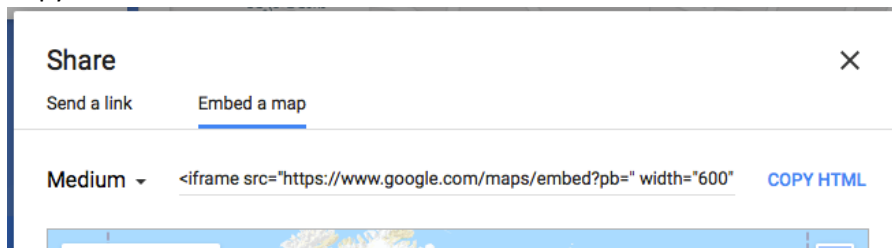
Select share button



Select Embed



Copy the html code



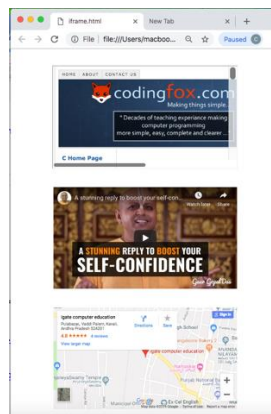
Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    .container{
      text-align: center;
      margin: 50px;
    }
    iframe{
      width:80%;
      height: 300px;
    }
  </style>
</head>
<body>
  <div class="container">
    <iframe src="http://www.codingfox.com" scrolling="yes"></iframe>
  </div>
  <div class="container">
    <iframe width="560" height="315" src="https://www.youtube.com/embed/CmSd_DLj70M"
    frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture"
    allowfullscreen></iframe>
```

```

</div>
<div class="container">
  <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3855.315023167853!2d79.98781
691483688!3d14.919532389599906!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x3a4b7b
edbae62187%3A0x2d41f3ff43847b63!2sigate+computer+education!5e0!3m2!1sen!2sin!4v1548647411
468" width="600" height="450" frameborder="0" style="border:0" allowfullscreen></iframe>
  </div>
</body>
</html>

```



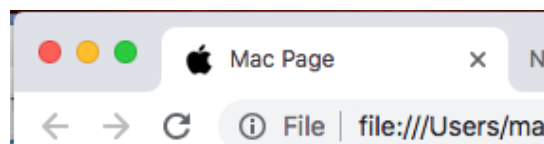
Adding favicon to a web page

Favicon is an icon that appears on the top of every tab of html page. The allowed images for favicon are .png/.ico. The minimum size of favicon is generally 16x16px. It can be added by using a link tag in the head section of the web page

```

<head>
<link rel="shortcut icon" type="image/png" href="/images/mac.png"/>
<title>Mac Page</title>
</head>

```



HTML Forms

HTML forms are very powerful tools to interact with the user. These are used to access the data from the user and send to the server-side program. The server-side program may be a PHP, ASP, JSP, a servlet or a python program.

HTML has the following user interface controls

1) Form elements

- a) Form
- b) Input
- c) Textarea
- d) Label
- e) Fieldset
- f) Legend
- g) Select
- h) Optgroup
- i) Option
- j) Button
- k) Datalist
- l) Output

2) Input Elements

- a) Text
- b) Password
- c) Radio
- d) Checkbox
- e) Submit
- f) Reset
- g) Button
- h) color
- i) date

- j) datetime-local
- k) email
- l) month
- m) number
- n) range
- o) search
- p) tel
- q) time
- r) url
- s) week

Form element

HTML form is an interface among the user and a server-side program. The form and its user controls take the data from the user and send to the server-side program specified by the **post** attribute of form.

An HTML Form is made of one or more widgets. Those widgets can be text fields (single line or multiline), select boxes, buttons, checkboxes, or radio buttons. Most of the time those widgets are paired with a label that describes their purpose.

The main difference between a HTML form and a regular HTML document is that most of the time, the data collected by the form is sent to a web server. In that case, we need to set up a web server to receive and a server-side program to process the received data.

Attribute	Purpose
action	Specifies the server-side program to which data must be submitted
method	Specifies how the data must be send to the server-side program
autocomplete	Specifies whether form controls must be auto completed (on/off)
novalidate	Specifies should not perform validations

Action: Specifies the URL of server-server program to which form must be submitted to.

Method: There are several methods using which we can send the data to the server, that are GET, POST, HEAD, TRACE, OPTION, PUT, DELETE and CONNECT. Out of these GET and POST methods are more popular and used frequently.

GET:

- Appends the data to the http request header and sends to the server
- The length of a URL is limited to 3000 characters
- Can't use this method to send sensitive data as it is visible and book marked
- GET is better for non-secure data, like query strings used for google search

POST

- Any amount of data can be send using POST method

Example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Form</title>
  <style>
    .container{
      padding: 10px;
      width: 50%;
      margin: 10px auto;
    }
    label{
      width: 100px;
      display:inline-block;
      text-align: right;
      padding: 10px;
    }
    [type=submit],[type=reset]{
      width: 100px;
      padding: 10px;
      margin: 10px;
    }
  </style>
</head>
<body>
  <div class="container">
    <form action="emp.php" method="GET">
      <fieldset>
        <legend>Emp Details</legend>
        <label for="a">Empno</label>
        <input type="text" id="a"> <br>
        <label for="b">Ename</label>
        <input type="text" id="b"> <br>
        <label for="c">Salary</label>
        <input type="text" id="c"> <br>
        <div style="text-align: center">
          <input type="submit"><input type="reset">
        </div>
      </fieldset>
    </form>
  </div>
```

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

Here **<fieldset>** is used to visually group elements

Here **<legend>** is used to display legend heading to the field set

Label

It is an element used to give description to another form/input element. It has **for** property which should be match to the **id** property of an element which is been describing by the label

One more advantage with the label is that, when we click on the label the described element would get focus

```
<label for="a">Empno</label>
<input type="text" id="a"> <br>
```

TextArea

It is an element used to display multiline text box. It has name, cols and rows properties

```
<label for="txt1" style="vertical-align: top">Address</label>
<textarea name="ads" id="txt1" cols="30" rows="10"></textarea> <br>
```

Select

It is an element used to create a dropdown list box where only one item can be selected out of a list of items and a list box where multiple items can be selected by holding ctrl/command button

In case of dropdown box, the size must be 1 and in case of list box size must be more than 1 and **multiple** property must be used in **<select>** tag


By default, first option is displayed in case of dropdown list. We can set any option as the default option using **selected** property

```
<label for="select1">Courses</label>
```

```

<select name="course" size="1" id="select1">
  <option value="java">Java</option>
  <option value="py" selected>Python</option>
  <option value="js">JavaScript</option>
  <option value="angl">Angular</option>
  <option value="react">React</option>
</select>

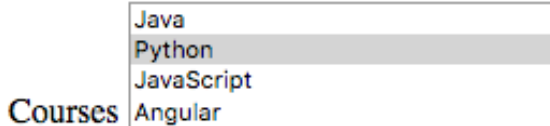
```



```

<label for="select2" style="vertical-align:middle">Courses</label>
<select name="courses" size="4" id="select2" multiple>
  <option value="java">Java</option>
  <option value="py" selected>Python</option>
  <option value="js">JavaScript</option>
  <option value="angl">Angular</option>
  <option value="react">React</option>
</select>

```



Datalist

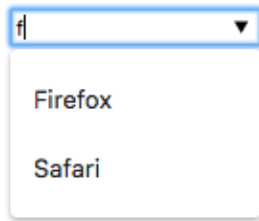
By default, datalist alone represents nothing, it is used to define a list of predefined options to another `<input>` element.

While attaching datalist with `<input>` make sure that the **list** property of `<input>` must be equal to the **id** property of datalist

```

<input list="browsers">
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist> <br>

```

<input> elements

Password:

It is an input element generally used to accept password with mask character *. We can also create our own password element using text box and text-security CSS property.

```
<label for="p1">Password</label>
<input type="password" id="p1"><br><br>
<label for="p2">Text Password</label>
<input type="text" id="p2" style="-webkit-text-security:square"><br>
```

Password

Text Password

Submit

It is an input element used to submit the form data to the server-side program that is specified in the action attribute of form.

Each form object has only one submit button. The default value of submit button is "Submit Query"

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

Reset

It is the input element that sets the values of UI controls in the form to their default values. Each form object has only one reset button

```
<input type="reset">
```

Radio:

It is an input type used select one out of number of choices. Independent input elements need to be used for independent options. Make sure that all the options must have the same name to make all the options as a group.

Checked property is used to select an element by default

```
<input type="radio" name="gender" value="Male" checked>Male<br>
<input type="radio" name="gender" value="Female" selected>Female<br>
<input type="radio" name="gender" value="Other" selected>Other<br>
```

☒ Male
☐ Female
☐ Other

Checkbox:

It is an input element used to select or unselect an option.

```
<input type="checkbox" value="yes" checked> Active
```

☒ Active

Button:

It is a user interface control used to create a normal button. It is generally used to start a process to perform some action within the page using client side script.

It can be created either by using <input> or <button>

Even using <button> we can create a normal, submit or reset buttons

```
input[type=button],button{
    width: 100px;
    padding: 5px;
}
<input type="button" value="Click Me"><br><br>
<button type="button">Click Me</button><br><br>
<button type="submit">Submit</button><br><br>
<button type="reset">Reset</button><br><br>
```

Click Me

Click Me

Submit

Reset

HTML 5 elements

HTML provides number of input elements like color, date, time, week, number, range, file, tel, email, progress, meter search, url, image and hidden

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    .container{
      padding:50px;
    }
  </style>
</head>
<body>
  <div class="container">
    <form name="form1">
      <label for="age">Age</label>
      <input type="number" name="age" id="age" min="1" max="10" step="2"><br><br>

      <label for="beans">How many Beans</label>
      <input type="range" name="beans" id="beans" min="0" max="500" step="10"><br><br>

      <label for="clr">Color</label>
      <input type="color" name="mycolor" id="clr"><br><br>

      <label for="doj">Date of joining</label>
      <input type="date" name="doj" min="2013-06-01" max="2013-08-31" id="doj"><br><br>

      <label for="tm">Time</label>
      <input type="time" name="time" id="tm"><br><br>

      <label for="wk">Week</label>
      <input type="week" name="week" id="wk"><br><br>

      <label for="em">E-mail</label>
      <input type="email" id="em" name="email" multiple><br><br>

      <label for="sh">Search</label>
      <input type="search" id="sh" name="search"><br><br>

      <label for="te">Telephone</label>
      <input type="tel" id="te" name="tel"><br><br>

      <label for="url">Website Address</label>
      <input type="url" id="url" name="url"><br><br>
```

```
<input type="file" name="file" id="file" accept="image/*" multiple><br><br>
<input type="hidden" id="timestamp" name="timestamp" value="1286705410"><br><br>
<input type="image" alt="Click me!" src="/images/mac.png" width="80" height="" /><br><br>
<progress max="100" value="75">75/100</progress><br><br>
<meter min="0" max="100" value="70" low="33" high="66" optimum="50">75</meter><br><br>
<input type="submit" value="submit">
</form>
</div>
</body>
</html>
```

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5500/form2.html?age". The page contains the following form elements:

- Age**: A text input field.
- How many Beans**: A range slider.
- Color**: A color picker showing black.
- Date of joining**: A date input field showing "dd/mm/2013".
- Time**: A time input field showing "--:-- --".
- Week**: A week input field showing "Week --, ----".
- E-mail**: An email input field.
- Week**: Another week input field.
- Telephone**: A telephone input field.
- Website Address**: A text input field.
- File Upload**: A "Choose Files" button and "No file chosen" text.
- Apple Logo**: A large black Apple logo.
- Progress Bars**: Two horizontal progress bars, one blue and one yellow.
- submit**: A submit button.

Important properties of Form elements

Form properties

Property	Purpose
autocomplete	on/off
novalidate	Enable or disables form control validation capabilities

spellcheck	true/false
------------	------------

Other properties

placeholder	To show the descriptive text
required	Compulsory to fill field
autocomplete	Completes depends on the history
autofocus	To get the focus on loading the page
pattern	To validate using regular expressions
disabled	To disable the control
readonly	Cannot select the content
length, minlength, maxlength	allowed length of input text