HTML FORMS

HTML Forms are required when you want to collect some data from the site visitor. For example during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, text area fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML <form> tag is used to create an HTML form and it has following syntax:

```
<form action="Script URL" method="GET|POST">

   form elements like input, text area etc.

</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes:

Attribute	Description
Action	Backend script ready to process your passed data.
method	Method to be used to upload data. The most frequently used are GET and POST methods.
Target	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
Enctype	You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are: ● **application/x-www-form-urlencoded** - This is the standard method most forms use in simple scenarios. ● **multipart/form-data** - This is used when you want to upload binary data in the form of files like image, word file etc.

Note: You can refer to Perl & CGI for a detail on how form data upload works.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form:

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms:

- **Single-line text input controls** - This control is used for items that require only one line of user input, such as search boxes or names. They are created using the HTML <input> tag.
- **Password input controls** - This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.
- **Multi-line text input controls** - This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using the HTML <textarea> tag.

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using the HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name:

```
<!DOCTYPE html>

<html>

<head>

<title>Text Input Control</title>

</head>

<body>

<form > First name:  <input type="text" name="first_name" />
```


 Last name: <input type="text" name="last_name" />

</form>

</body>

</html>

This will produce following result:

First name:

Last name:

Attributes

Following is the list of attributes for <input> tag for creating text field.

Attribute	Description
Type	Indicates the type of input control and for text input control it will be set to text.
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Value	This can be used to provide an initial value inside the control.
Size	Allows to specify the width of the text-input control in terms of characters.
maxlength	Allows to specify the maximum number of characters a user can enter into the text box.

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using the HTML <input> tag but type attribute is set to **password.**

Example

Here is a basic example of a single-line password input used to take user password:

<!DOCTYPE html>

<html>

<head>

<title>Password Input Control</title>

</head>

<body>

<form > User ID : <input type="text" name="user_id" />

 Password: <input type="password" name="password" />

</form>

</body>

</html>

This will produce following result:

User ID:

Password:

Attributes

Following is the list of attributes for <input> tag for creating a password field.

Attribute	Description
Type	Indicates the type of input control and for password input control it will be set to password.
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Value	This can be used to provide an initial value inside the control.
Size	Allows to specify the width of the text-input control in terms of characters.
Max-length	Allows to specify the maximum number of characters a user can enter into the text box.

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using the HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description:

<!DOCTYPE html>

<html>

<head>

<title>Multiple-Line Input Control</title>

</head>

<body>

<form> Description :

<textarea rows="5" cols="50" name="description"> Enter description here... </textarea> </form>

</body>

</html>

This will produce following result:

Description:

Enter description here...

Attributes

Following is the list of attributes for <textarea> tag.

Attribute	Description
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Rows	Indicates the number of rows of text area box.
cols	Indicates the number of columns of text area box

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using the HTML <input> tag but the type attribute is set to the checkbox. Example Here is an example HTML code for a form with two **checkboxes:**

<!DOCTYPE html>

<html>

<head>

<title>Checkbox Control</title>

</head>

<body>

<form>

<input type="checkbox" name="maths" value="on"> Maths

<input type="checkbox" name="physics" value="on"> Physics

</form>

```
</body>

</html>
```

This will produce following result:

Maths Physics

Attributes

Following is the list of attributes for <checkbox> tag.

Attribute	Description
Type	Indicates the type of input control and for checkbox input control it will be set to checkbox.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the checkbox is selected.
checked	Set to checked if you want to select it by default.

Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using the HTML <input> tag but type attribute is set to radio.

Example

Here is example HTML code for a form with two radio buttons:

```
<!DOCTYPE html>

<html>

<head>

<title>Radio Box Control</title>

</head>

<body>

<form>

<input type="radio" name="subject" value="maths"> Maths

<input type="radio" name="subject" value="physics"> Physics

</form>

</body>

</html>
```

This will produce following result:

Maths Physics

Attributes

Following is the list of attributes for the radio button.

Attribute	Description
type	Indicates the type of input control and for checkbox input control it will be set to checkbox.
name	Used to give a name to the control which is sent to the server to be recognized and get the value.
value	The value that will be used if the checkbox is selected.
checked	Set to check if you want to select it by default.

Select Box Control

A select box, also called drop down box which provides an option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

Here is example HTML code for a form with one drop down box

```
<!DOCTYPE html>

<html>

<head>

<title>Select Box Control</title>

</head>

<body>

<form>

<select name="dropdown">

<option value="Maths" selected>Maths</option>

 <option value="Physics">Physics</option>

</select>

</form>

</body>

</html>
```

This will produce following result:

Maths

Attributes

Following is the list of important attributes of <select> tag:

Attribute	Description
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Size	This can be used to present a scrolling list box.
Multiple	If set to "multiple" then allows a user to select multiple items from the menu.
Following is the list of important attributes of <option> tag:	
Attribute	**Description**
value	The value that will be used if an option in the select box.box is selected.
Selected	Specifies that this option should be the initially selected value when the page loads.
Label	An alternative way of labeling options

File Upload Box

If you want to allow a user to upload a file to your website, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to file.

Example

Here is example HTML code for a form with one file upload box:

```
<!DOCTYPE html>

<html>

<head>

<title>File Upload Box</title>

</head>

<body>

<form>

<input type="file" name="fileupload" accept="image/*" />

</form>
```

</body>

</html>

This will produce following result:

Attributes

Following is the list of important attributes of file upload box:

Attribute	Description
Name	Used to give a name to the control which is sent to the server to be recognized and get the value.
Accept	Specifies the types of files that the server accepts.

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using the <input> tag by setting its type attribute to the button. The type attribute can take the following values:

Type	Description
Submit	This creates a button that automatically submits a form.
Reset	This creates a button that automatically resets form controls to their initial values.
Button	This creates a button that is used to trigger a client-side script when the user clicks that button.
Image	This creates a clickable button but we can use an image as the background of the button.

Example

Here is example HTML code for a form with three types of buttons:

<!DOCTYPE html>

<html>

<head>

<title>File Upload Box</title>

</head>

<body>

<form>

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

<input type="reset" name="reset" value="Reset" />

```
<input type="button" name="ok" value="OK" />

<input type="image" name="image button" src="/html/images/logo.png" />

</form>

</body>

</html>
```

This will produce following result:

Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, the following hidden form is being used to keep current page number. When a user clicks the next page then the value of hidden control will be sent to the web server and there it will decide which page has been displayed next based on the passed current page.

Example

Here is example HTML code to show the usage of hidden control:

```
<!DOCTYPE html>

<html>

<head>

<title>File Upload Box</title>

</head>

<body>

<form>

<p>This is page 10</p>

<input type="hidden" name="pagename" value="10" />

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

<input type="reset" name="reset"  value="Reset" />

</form>

</body>

</html>
```

This will produce following result:

This is page 10.

HTML Anchor

HTML Anchor

The HTML anchor tag defines a hyperlink that links one page to another page. It can create hyperlinks to other web pages as well as files, location, or any URL. The "href" attribute is the most important attribute of the HTML a tag. and which links to the destination page or URL.

href attribute of HTML anchor tag The href attribute is used to define the address of the file to be linked. In other words, it points out the destination page.

The syntax of HTML anchor tag is given below.

 Link Text

Let's see an example of an HTML anchor tag.

Click for Second Page

Specify a location for Link using target attribute If we want to open that link to another page then we can use the target attribute of <a> tag. With the help of this link will be open on the next page.

Example:

<!DOCTYPE html>

<html>

<head>

<title>

</title>

</head>

<body>

Click on this-link to go on home page of Javatpoint.

</body>

</html>

HTML BUTTON

```
<!DOCTYPE html>
<html>
<head>
<style>
.button {
border: none;
color: white;
padding: 16px 32px;
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
margin: 4px 2px;
transition-duration: 0.4s;
cursor: pointer;
}
button1 {
background-color: white;
color: black;
border: 2px solid #4CAF50;
}
.button1:hover {
background-color: #4CAF50;
color: white;
}
.button2 {
```

```
background-color: white;

color: black;

border: 2px solid #008CBA;

}

.button2:hover {

background-color: #008CBA;

color: white;

}

</style>

</head>

<body>

<button class="button button1">Green</button> <button class="button button2">Blue </button>

</body>

</html>
```

Output

THE HTML GENERIC CONTROL

Some HTML tags don't correspond to specific control like a button, hyperlink, or textbox. In fact, controls in this particular category don't have a visual repre-sentation by themselves. The HTML tags , ,<body>,and <font> are examples of this. The HTML Generic Control is used as follows.

<span | body| div| font| others

runat= “server”

id= “programmaticID”>

Working with ASP.NET

Your HTML goes here

</span| body| div| font| others>

The typical use for HTML Generic Control is to display dynamic text in the page. For example, you may want to display the results of a query that exe-cuted on the server in a certain area on the page. That area can be defined by using the tag and can be server-side enabled by supplying the runat= “server” attribute. Here’s an example.

```
<%@ page Language= “vb” %>

<html>

<head>

<TITLE>HTMLGenericControl Example </TITLE>

<script language=vb runat= “server” >

Sub page_load(Source As Object, E As EventArgs)

  Dim strOutputString as String

  Dim intCount as Integer

  strOutputString = “<OL>”

  For intCountString = 1 To 5

     strOutputString &= “<LI>Number red list” Next

strOutputString & = “</OL>”

lblMyText.InnerHTML = strOutput String

End sub

</script>

</head>

<body>

<form id= “HTMLGenericControl”

method = “post”

runat = “server”>
```

<h1>HTMLGenericControl Example</h1>

Span text;

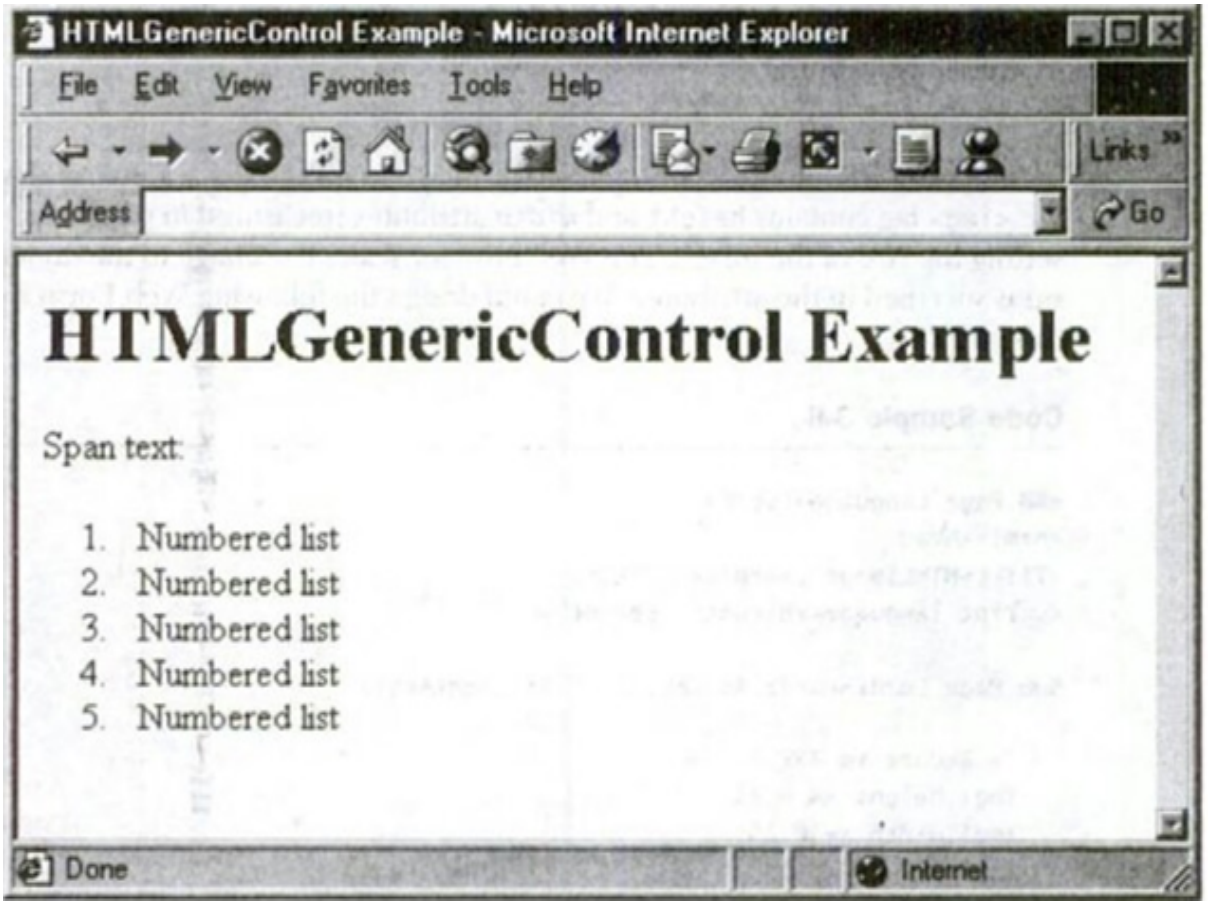
<span id = lblMyText

runat = “server”>

</form>

</body>

This example builds a simple HTML ordered list and puts the HTML text into a string when the **page_Load** event fires. Then the **InnerHTML** property is set for the HTMLgenericControl lblMyText. This dynamically inserts the ordered list between the tags. The output appears in Figure.



HTML IMAGE

Earlier, the webpages used to comprise of only texts, which made them appear quite boring and uninteresting.

Fortunately, it wasn't long enough that the ability to embed images on web pages was added for users.

Let's see how to add images on a webpage.

Adding images on a webpage

The "img" tag is used to add images on a webpage.

The "img" tag is an empty tag, which means it can contain only a list of attributes and it has no closing tag.

The images are often subject to change with each page visit.

This is especially true of banner advertisement, which must change according to user demographics or product popularity. Whatever the reaction for manipulating inline image on a web page.

ASP.NET provides the HTML IMAGE control for changing < img> tag characteristic on the server.

All of the <img> tag attributes are made available on the server. Below is the complete syntax showing those attributes.

<img

id="programmatic ID"

runat= " server"

alt = "alttext"

align =top | middle | bottom | left | right

border= "borderwidth"

Height= "imageheight"

Src = "imageURI"

Width= "Image width" >

Syntax: <img src="url" alt="some_text".

Attribute

src:

src stands for source. Every image has a src attribute which tells the browser where to find the image you want to display.

The URL of the image provided points to the location where the image is stored.

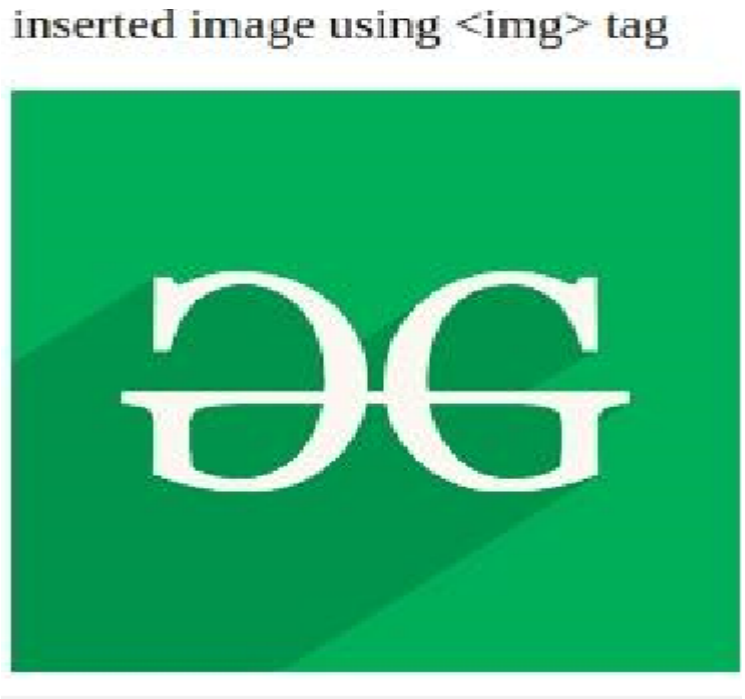
Alt:

If the image cannot be displayed then the alt attribute acts as an alternative description for the image. The value of the alt attribute is an user-defined text.

Coding:

```
<html>
<head>
 <title>Inserting an image using "img" tag</title>
</head>
<body>
    <p>inserted image using <img> tag: </p>
                                                                    <img src=
"https://media.geeksforgeeks.org/wp-content/cdnuploads/20190710102234/download3.png"
      alt="GeeksforGeeks logo">
</body>
</html>
```

Output:



HTML INPUT BUTTON

A push button that activates a javascript when is clicked: <INPUT TYPE = “BUTTON” VALUE = “CLICK ME” ONCLICK = “MSG( )”>

The <INPUT TYPE = “BUTTON”> Defines a clickable button) mostly used with a javascript to activate a script).

Attribute					
type="button"	1.0	Yes	1.0	1.0	Yes

Syntax:

```
<input type="button">
```

< HTML <input> type attribute

HTML INPUT CHECKBOX

If you don’t know what an element is or how you must use it, I recommend you read the “HTML tags and attributes” tutorial that you can find in the HTML tutorials section

DESCRIPTION

The <input> element having the ‘checkbox’ value in its type attribute, represents a two-states control that allows users to mark it as selected or deselected.s

THE HTML INPUT FILE

Many websites can accept files transmitted from users' computers. Examples of such sites include those that host photo albums, provide backup facilities, or accept file attachments for web-based e-mail. In ASP.NET, the HTMLInputFile Control enables this capability.

Uploading a file using the HTMLInputFile Control is a three – step process.

1. The control allows the user to select a file to upload using a file-brosing dialog box.
2. Once the user selects the file, it is encoded for trasmisson and sent along with other from data with the server post – back.

3. The file is saved to disk on the server.

Here's the syntax for the HTML Input File Control.

```
<input
   type=file
   runat="server"
   id="programmaticID"
   accept="MIMEencodings"
   maxlength="maxfilepathlength"
   size="widthoffilepathtextbox"
   postedfile="uploadedfile"
>
```

Sample Code

```
<%@ page Language = "vb" %>

<html><head>

<TITLE>HTMLInputFile Example</TITLE>

<script language = vb runat= "sever">

   Function GetFileNameFromPath(Byva1 pathName as String)

Dim intFirstPos As Integer

intFirstPos = InStrRev(pathName, "\") + 1

GetFileNameFromPath = Mid(pathName, intFirstPos)

End Function

Sub btnSubmit_Click(Source As Object, E As EventArgs)

Dim strSavePath as String

Dim strFileName as String

strFileName = GetFileNameFrompath( _filselectFile.PostedFile.FileName)

lblFileInfo.InnerHTML = "You selected:" & strFileName

strSavePath = Server.MapPath("\Samples2")

filSelectFile.PostedFile.SaveAs(strSavePath & "\" & _ strFileName)

End Sub

</script>

</head>
```

<body>

<form id = “HTMLInputFileSample”

method = “post”

enctype = “multipart/form-data”

runat = “sever”>

<h1>HTMLInputFile Example</h1>

Please select a file to upload to the server:

<input id=filSelectFile

type=file

runat= “server”>

<INPUT id=btnSubmit

type= “Submit”

value= “Submit”

OnServerClick= “btnSubmit_Click”

runat= “server”>

<span id= lblFileInfo

runat= “server”>

</form>

</body>

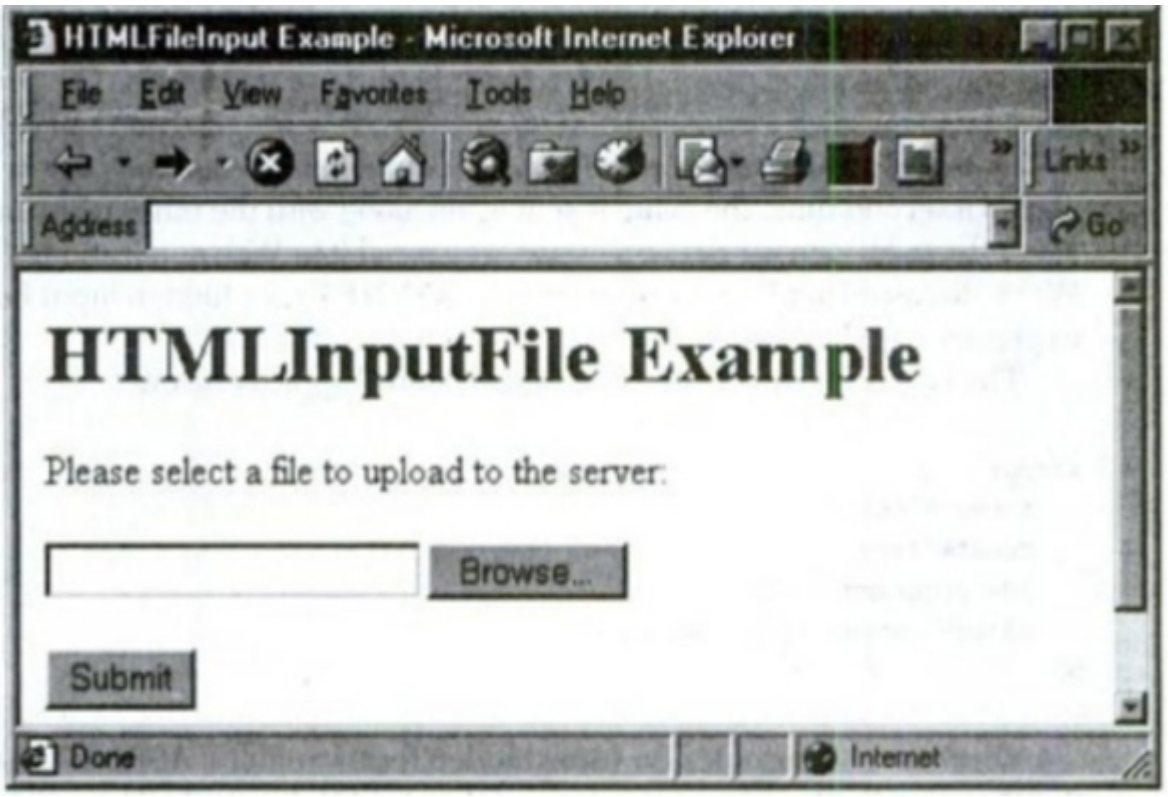
</html>

In the code block that starts at line, we show the declaration of an HTMLInputFile Control. This will display a file browse button that will allow the user to select a file to upload to the server. Note that the enctype for the form is set to “multipart/form-data”, and the method is set to “post” (see the code block beginning at line). This is required because files uploaded via a form post are encoded in a special way that differs from regular form submissions(that is, forms that contain only simple controls such as input boxes, radio buttons, checkboxes, and so on).

The form resulting from this code looks similar to that shown in Figure.

The file is uploaded with a post-back to the server(that is,when the user click the submit button). The btnSubmit_Click event procedure code in line handles that form submission event. In this procedure,retrieving the file data that was posted and saving it to the server’s storage area is a two-step process.

1. Obtain the file name from the posted file.information about the posted file is contained in the postedFile object(see line),which is a member of the HTMLInputFile control.FileName holds the complete pathname of the file selected on the user’s workstation. The GetFileNameFromPath( ) function defined on line extracts the file name from the absolute path in FileName.



2. Save the file on the server. To save the file, we need to obtain the absolute, physical path o the location to which we wish to save the file.we obtain this location by using the MapPath( ) method of the ASP.NET server object. I’ll explain this object in detail in Section 3.23,but for now I’ll just explain the mapPath( ) method. It returns the physicl path for given virtual path(for example, c:\inetpub\wwwroot\images

for\images). Then, the SaveAs( ) method of the postedFile object saves the actual file with the file name extracted from the POST request into the directory obtained from MapPath( ).

HTML Control Name	HTML 3.2 Representation
HTMLAnchor	<a>
HTMLButton	<button>
HTMLForm	<form>
HTMLGenericControl	or <>span>
HTMLImage	<img>
HTMLInputButton	<input type="button">
HTMLInputCheckBox	<input type="check">
HTMLInputFile	<input type="file">
HTMLInputHidden	<input type="hidden">
HTMLInputImage	<input type="image">
HTMLInputRadioButton	<input type="radio">
HTMLInputText	<input type="button"> or <input type="password">
HTMLSelect	<select>
HTMLTable	<table>
HTMLTableCell	<td>
HTMLTableRow	<tr>
HTMLTextArea	<textarea>

HTML INPUT HIDDEN

The HTML<input type=" hidden" >is used to define an input hidden field. A hidden field also includes those data that could not be seen or modified by the users when submitted the form. A hidden field only stores those database records that need to be updated when submitting the form.

Syntax:

<input type = "hidden" >

Example:

```
<! DOCTYPE html >
<html >
<head>
<title> HTML input type hidden </ title>
<style>
h1{
color : green;
}
body{
text align: center ;
}
</ style>
</ head>
<body >
<h1>
Geeks for Geeks
</ h1>
<h3>
HTML <input type = "hidden" >
</ h3>
<form action = "#" >
<input type = "hidden" id = "my File"
value = "1234" >
name: <input type = "text" >
<input type = "submit "value = "Submit" >
```

</ form>

</ body > </ html >

Supported Browsers:

The browsers supported by <input type = “hidden” >are listed below:

Output:

- Google Chrome 1. 0
- InternetExplorer
- Firefox 1. 0
- Safari 1. 0
- Opera 1. 0

HTML INPUT IMAGE

The <input type = “image”> defines an image as a submit button. The path to the image is specified in the src attribute.

Syntax

<input type = “image”>

Example

<input type = “image”

src = “submit.gift”

alt = “Submit”

style = “float:right”

width = “48” height = “48”>

HTML INPUT RADIO BUTTON

HTML radio buttons allow the user to select from a group of mutually exclusive options. In other works, use radio buttons when you want the user to make only one choice form a set of choices. It’s important to group together related radio buttons by using the name attribute. This

instructs the browser to treat the buttons as a group. When the user clicks on one radio button to highlight it, the previous selection becomes unhighlighted.

ASP.NET provides the HTML Input Radio Button Control for programmatically controlling. HTML radio buttons. The code selection below shows the typical syntax for the HTML Input Radio Button control.

```
<input
    type=radio
    runat=”server”
    id=”programmaticID”
    checked
name=”radiobuttongroup”
>
```

Sample code

```
<%@ page Language = “vb” %>
<html><head>
<title> HTMLInputFile Example </title>
<script language = vb runat= “sever”>
Sub cmdSendSurvey_Click(Source As Object, E As EventArgs)
Dim strResults As String = _ “<b> you answered: <br>”
    If rdbyearlyIncome1.Checked Then
        strResults & = “Less than $25,000/yr”
End If
   If rdbyearlyIncome2.Checked Then
        strResults & = “$26,000 - $50,000/yr”
End If
If rdbyearlyIncome3.Checked Then
        strResults & = “$51,000 or more/yr”
End If
```

```
strResults & = “<br>”
If Radio1.Checked Then
        strResults & = “E-mail newsletter”
End If
If Radio2.Checked Then
        strResults & = “Magazine Ad”
End If
If Radio3.Checked Then
        strResults & = “Other”
End If
End Sub
</script>
</head>
<body>
<form id = “HTMLInputRadio”
method = “post”
runat = “server”>
<h1>HTMLInputRadio Example</h1>
<p>Please answer our short survey: </p>
<ol>
<li><b> what is your yearly income?</b><br>
<input id = rdbyearlyIncome1
Checked name = “yearlyIncome”
type = radio
runat= “server”>Less than $25,000/yr <br>
<input id = rdbyearlyIncome2
name = “yearlyIncome”
```

type = radio

runat= “server”>$26,000 – $50,000/yr

<input id = rdbyearlyIncome3

name = “yearlyIncome”

type = radio

runat= “server”>$51,000 – or more/yr

 How did you hear about our web site?

<input id = Radio1

name = “HearAboutUs”

type = radio

runat = “server”>E-mail newsletter

<input id = Radio2

name = “HearAboutUs”

type = radio

runat = “server”>Magazine Ad

<input id = radio3

name = “HearAboutUs”

type = radio

runat = “server”>Other

<INPUT id = cmdSendSurvey

type = “submit”

value = “Send Survey”

runat = “server”

onserverclick = “cmdsendsurvey_Click”>

</form>

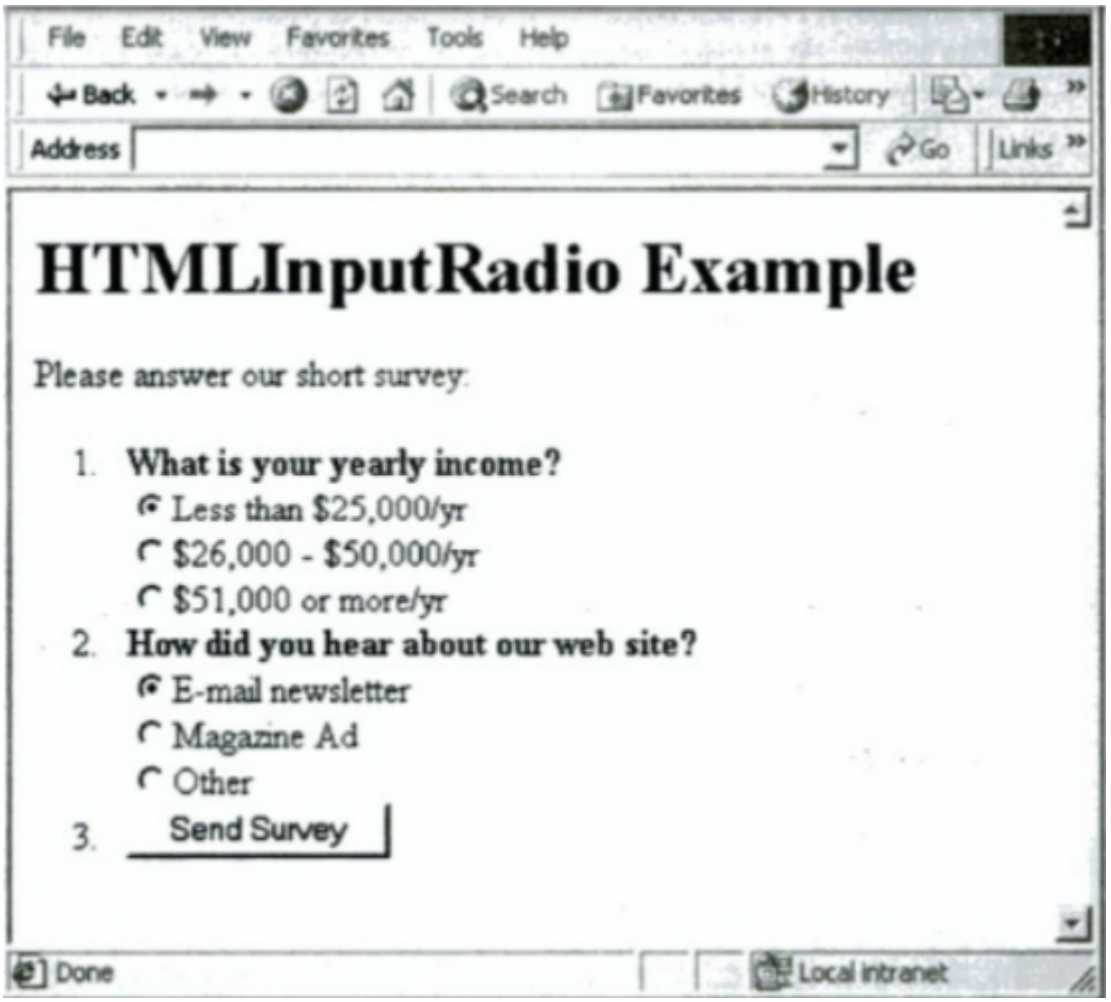
</LI>

</OL>

</body>

</html>

The code has two groups of radio buttons: yearly Income and About Us. This way, only one choice can be made for each question. We set the subroutine cmd Send Survey_Click to be the event handler for the subroutine event for the Send Survey button.



HTML INPUT TEXT

The HTML<Input type = “text”> is used to define a single_line text field. The default width of the text field is 20 characters.

Syntax

<input type = “text”>

Example

<html>

<head>

<title> Basic Form</title>

</head>

<body>

<h1> Login Form</h1>

<form>

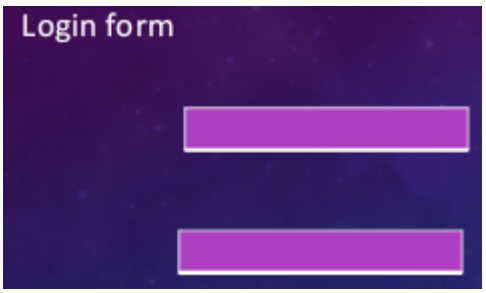
<input type = “text”>

<input type = “text”>

</from>

</body>

</html>



<html>

<head>

<title> Basic Form</title>

</head>

<body>

<h1> Login Form</h1>

<form>

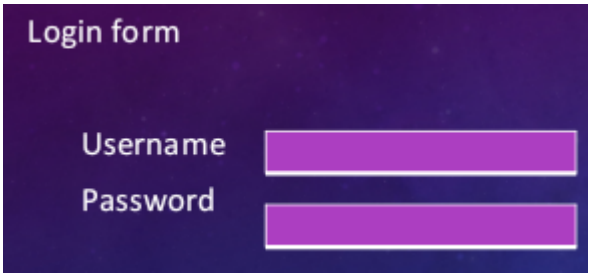
Username : <input type = “text”

Password : <input type = “text”

</from>

</body>

</html>



HTML SELECT

<select> Element is used to create a "drop down list." The <select> element is most often used in a form, to collect user input. The name attribute is needed to reference the form data after the form is submitted. (If you omit the name attributes, no data from the drop down list will be submitted).

The <option> tag inside the <select> element defines the available option in the drop down list. Html server controls are basically the standard html controls enhanced to enable server side processing.

Example

<label for="cars"> choose a car:</label>

<select name="cars"id="cars>"

<option value="volvo">

volvo <\option>

<option value="saab">

saab <\option> <\select>

Output

Choose a car: Volvo

Saab

Attributes

Attributes	Description
Auto Focus	Specifies that the drop down list should automatically get focus when the page loads.
Disabled	Specifies that a drop down list should be disabled.
Form	Form_id define which form the drop down list belongs to.
Multiple	Specifies that multiple option can be selected at once.
Name	Define a name for the drop down list.
Required	Specifies that the user is required to select a value before submitting the form.
Size	Define the number of visible option in a drop down list.

Tag

Control Name	Html Tag
Html head	<head> element
Html Input Button	<input type = button |submit | reset |>
Html Input Checkbox	<input type = checkbox>
Html Input File	< input type= file>
Html Input Hidden	<input type= hidden>
Html Input Image	<input type= image>
Html Input Password	<input type= password>
Html Input Radio Button	<input type= radio>
Html Input Reset	<input type= reset>
Html Text	<input type= text |password>
Html Image	<img> element
Html Link	<link> element
Html Anchor	<a> element
Html Button	<button> element
Html Select	<select> element
Html Form	<form> element
Html Table	<table> element
Html Table Cell	<td> and <th>
Html Table Row	<tr> element
Html Title	<title> element

HTML TABLE DEFINITION

The **<table>** tag defines an HTML table. Each table row is defined with a **<tr>** tag. Each table header is defined with a **<th>** tag. Each table data/cell is defined with a **<td>** tag. The text in **<th>** elements are bold and centered. The text in **<td>** elements are regular and left-aligned.

EXAMPLE OF HTML TABLE

```
<html>

<body>

<h2> Basic HTML Table</h2>

<table

style="width:100%">

<tr>

<th>Rollno</th>

<th>Name</th>

<th>Age</th>

</tr>

<tr>

<td>19MCS414</td>

<td>Varsha</td>

<td>22</td>

</tr>

<tr>

<td>19MCS415</td>

<td>Vinitha</td>

<td>23</td>

</tr>

</table>

</body>
```

```
</html>
```

OUTPUT

RollNo	Name	Age
19MCS414	Varsha	22
19MCS415	Vinitha	23

ADD A BORDER

```
<html>
<head>
<style>
table , th, td {
border=1px solid black;   }
</style>
</head>
<body>
<h2> Add a border</h2>
<table
style=“width:100%”>
<tr>
<th>Rollno</th>
<th>Name</th>
<th>Age</th>
</tr>
<tr>
<td>19MCS414</td>
<td>Varsha</td>
<td>22</td>
```

```
</tr>
<tr>
<td>19MCS415</td>
<td>Vinitha</td>
<td>23</td>
</tr>
</table>
</body>
</html>
```

OUTPUT OF ADD A BORDER

Add a border

Rollno	Name	Age
19MCS414	Varsha	22
19MCS415	Vinitha	23

COLLAPSED BORDER

```
<html>
<head>
<style>
table , th, td {
border=1px solid black;
border_collapse: collapse; }
</style>
</head>
<body>
<h2> Add a border</h2>
<table
```

```
style="width:100%">
<tr>
<th>Rollno</th>
<th>Name</th>
<th>Age</th>
</tr>
<tr>
<td>19MCS414</td>
<td>Varsha</td>
<td>22</td>
</tr>
<tr>
<td>19MCS415</td>
<td>Vinitha</td>
<td>23</td>
</tr>
</table>
</body>
</html>
```

OUTPUT OF COLLAPSED BORDER

RollNo	Name	Age
19MCS414	Varsha	22
19MCS415	Vinitha	23

CELL PADDING

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

```
table , th, td {
border=1px solid black;
border_collapse: collapse; }
th, td {  padding:15px; }
</style>
</head>
<body>
<h2> Add a border</h2>
<table
style="width:100%">
<tr>
<th>Rollno</th>
<th>Name</th>
<th>Age</th>
</tr>
<tr>
<td>19MCS414</td>
<td>Varsha</td>
<td>22</td>
</tr>
<tr>
<td>19MCS415</td>
<td>Vinitha</td>
<td>23</td>
</tr>
</table>
</body>
```

</html>

OUTPUT OF CELLPADDING

RollNo	Name	Age
19MCS414	Varsha	22
19MCS415	Vinitha	23

LEFT-ALIGN HEADING

```
<html>
<head>
<style>
table , th, td {
border=1px solid black;
border_collapse: collapse; }
th, td {  padding:15px; }
th  { text_align: left;  }
</style>
</head>
<body>
<h2> Add a border</h2>
<table
style="width:100%">
<tr>
<th>Rollno</th>
<th>Name</th>
<th>Age</th>
</tr>
<tr>
```

<td>19MCS414</td>

<td>Varsha</td>

<td>22</td>

</tr>

<tr>

<td>19MCS415</td>

<td>Vinitha</td>

<td>23</td>

</tr>

</table>

</body>

</html>

OUTPUT OF LEFT-ALIGN

RollNo	Name	Age
19MCS414	Varsha	22
19MCS415	Vinitha	23

HTML TABLE CELL AND TABLE ROW

The <td> tag defines a standard data cell in an HTML table.

An HTML table has two kinds of cells:

- Header cells – contains header information(created with the <th> element)
- Data cells- contains data(created with the <td> element)

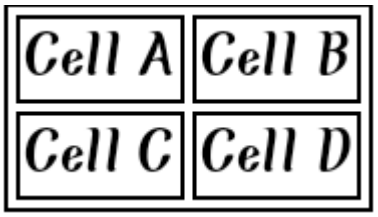
The text in <td> elements are regular and left-aligned by default. The text in <th> elements are bold and centered by default.

<html>

<head>

<style>

```
table, th, td {
  border: 1px solid black;
}
</style>
</head>
<body>
<h1> The td element defines a cell in a table: </p>
<table>
<tr>
<td> Cell A</td>
<td> Cell B</td>
</tr>
<tr>
<td> Cell C</td>
<td> Cell D</td>
</tr>
</table>
</body>
</html>
```



HTML <tr> Tag

The <tr> tag specifies a row in an HTML table. then cells inside it are defined using <th>(a header cell) or <td> (a standard cell) elements.

Example

```
<html>
<head>
<title> Title of the document</title>
<style>
table {
    width: 80%;
    margin: 30px auto;
border-collapse: collapse;
}
th, td {
   padding: 10px;
   border: 1px solid #666;
}
</style>
</head>
<body>
<table>
<tr>
 <th> Month </th>
<th> Date </th>
</tr>
<tr>
<td> June </td>
<td> 18.07.2018 </td>
</tr>
</table>
</body>
```

</html>

Output

Month	Date
March	10.09.2018
June	18.07.2018

HTML <Textarea> Tag

The <textarea> tag defines a multi-line text input control. The <textarea> element is often used in a form, to collect user inputs like comments or reviews. A text area can hold an unlimited number of characters, and the text renders in a fixed-width font. The size of a text area is specified by the <cols> and <rows> attributes. The name attribute is needed to reference the form data after the form is submitted. If you omit the name attribute, no data from the text area will be submitted. The id attribute is needed to associate the text area with a label.

ATTRIBUTES

Attribute	Value	Description
Autofocus	Autofocus	Specifies that a text area should automatically get focus when the page loads.
Cols	Number	Specifies the visible width of a text area.
Dirname	Textareaname.dir	Specifies that the text direction of the textarea will be submitted.
Disabled	Disabled	Specifies that a text area should be disabled.
Form	form-id	Specifies which form the text area belongs to
Maxlength	Number	Specifies the maximum number of characters allowed in the text area
Name	Text	Specifies a name for a text area.

Placeholder	Text	Specifies a short hint that describes the expected value of a text area.
Readonly	Readonly	Specifies that a text area should be read-only.
Required	Required	Specifies that a text area is required/must be filled out.
Rows	Number	Specifies the visible number of lines in a text area
Wrap	Hardsoft	Specifies how the text in a text area is to be wrapped when submitted in a form

Global Attributes

- The <textarea> tag also supports the Global Attributes in HTML.

Event Attributes

- The <textarea> tag also supports the Event Attributes in HTML.

1. Matt J Crouch, “ASP.NET and VB.NET web programming”, Pearson Education, 2005.
2. https://www.geeksforgeeks.org/html-tutorials/
3. https://www.tutorialspoint.com/html/