

## FORM

- HTML forms are used to select different kinds of user input.
- HTML forms are used to pass data to a server.
- There are two parts of working form
  - The first part is the form that you see on the page itself. Forms are made up of buttons, text fields, and pull-down menus (collectively known as form controls) used to collect information from the user
  - The second component of a web form is an application or script on the server that processes the information collected by the form and returns an appropriate response

## Form Elements

### Form tag

- The <form> tag is used to create an HTML form:
- HTML forms are used to select different kinds of user input.
- HTML forms are used to pass data to a server.
- There are two parts of working form
  - The first part is the form that you see on the page itself. Forms are made up of buttons, text fields, and pull-down menus (collectively known as form controls) used to collect information from the user
  - The second component of a web form is an application or script on the server that processes the information collected by the form and returns an appropriate response

Example:-

```
<html>
<body>
<h1>Mailing List Signup</h1>
<form action="register.php" method="post">
</form>
</body>
```

</html>

## Most frequently used form attributes are:

- **Name:** This is the name of the form.
- **Action:**
  - The action attribute provides the location (URL) of the application or script (sometimes called the action page) that will be used to process the form (which will receive uploaded data).
  - The action attribute in above example sends the data to a script called **register.php**
  - .php suffix indicates that this form is processed by a php script
- **Method:** Here you will specify method to be used to upload data (The method attribute specifies how the information should be sent to the server). It can take various values but most frequently used are GET and POST.
- **Target:** It specifies the target page 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 like:
  - **application/x-www-form-urlencoded-** This is the standard method most forms use. It converts spaces to the plus sign and non-alphanumeric characters into the hexadecimal code for that character in ASCII text.
  - **multipart/form-data-** This allows the data to be sent in parts, with each consecutive part corresponding to a form control, in the order they appear in the form. Each part can have an optional content-type header of its own indicating the type of data for that form control.

## Form Accessibility Features

- **Labels element**

- The label element is used to associate descriptive text with its respective form field.
- This provides important context for users with speech-based browsers.
- Each label element is associated with exactly one form control

`<label>Male</label>`

`<label>Female</label>`

- **Fieldset element:-** is used to indicate a logical group of form controls
- **Legend element:-**A fieldset may also include a legend element that provides a caption for the enclosed fields.

## Form controls

There are different types of form controls that you can use to collect data from a visitor to your site.

- Text input controls
- Buttons
- Checkboxes and radio buttons
- Select boxes
- File select boxes
- Hidden controls
- Submit and reset button

## Text Input Controls:

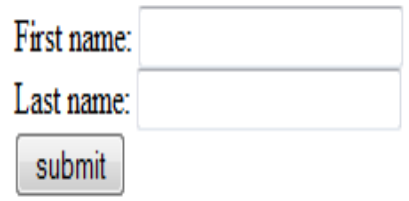
There are actually three types of text input used on forms:

- **Single-line text input controls:** Used for items that require only one line of user input, such as search boxes or names. They are created using the `<input>` element.
- **Password input controls:** Single-line text input that mask the characters a user enters.
- **Multi-line text input controls:** Used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created with the `<textarea>` element.

### Single-line text input controls:

- Single-line text input controls are created using an `<input>` element whose `type` attribute has a value of `text`.

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

<pre> &lt;form action="abc.php" method="get"&gt; First name: &lt;input type="text" name="first_name"/&gt;&lt;br&gt; Last name: &lt;input type="text" name="last_name"/&gt;&lt;br&gt; &lt;input type="submit" value="submit" /&gt; &lt;/form&gt; </pre>	<p>Will result</p> 
--	--

Following is the list of attributes for `<input>` tag.

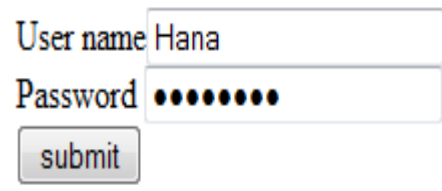
- **Type:** Indicates the type of input control you want to create. This element is also used to create other form controls such as radio buttons and checkboxes.
- **Name:** Used to give the name part of the name/value pair that is sent to the server, representing each form control and the value the user entered.
- **Value:** Provides an initial value for the text input control that the user will see when the form loads.

- **Size:** Allows you to specify the width of the text-input control in terms of characters.
- **maxlength:** Allows you to specify the maximum number of characters a user can enter into the text box.

## Password input controls

- This is also a form of single-line text input controls are created using an `<input>` element whose type attribute has a value of password.

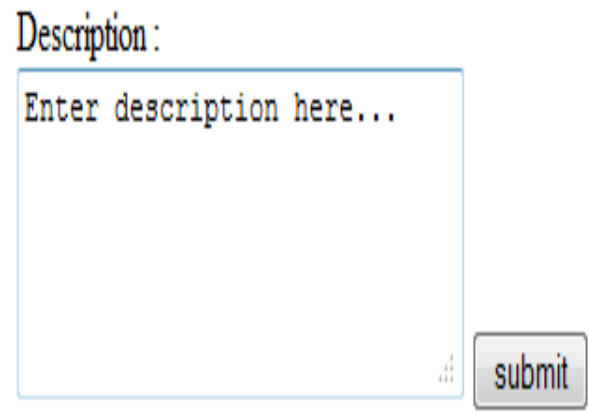
Here is a basic example of a single-line password input used to take user password

<pre> &lt;form action="abc.php" method="get"&gt; User name&lt;input type="text" name="user_name"/&gt;&lt;br&gt; Password &lt;input type="password" name="password"/&gt;&lt;br&gt; &lt;input type="submit" value="submit" /&gt; &lt;/form&gt; </pre>	<p>Will result</p> 
---	---

## Multiple-Line Text Input Controls:

- If you want to allow a visitor to your site to enter more than one line of text, you should create a multiple-line text input control using the `<textarea>` element.

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

<pre> &lt;form action="abc.php" method="get"&gt; Description : &lt;br /&gt; &lt;textarea rows="5" cols="50" name="abc"&gt; Enter description here... &lt;/textarea&gt; &lt;input type="submit" value="submit" /&gt; &lt;/form&gt; </pre>	<p>Will result:-</p> 
--	---

Following is the detail of above used attributes for <textarea> tag

- **Name:** The name of the control. This is used in the name/value pair that is sent to the server.
- **Rows:** Indicates the number of rows of text area box.
- **Cols:** Indicates the number of columns of text area box.


## Creating Button

- There are various ways in HTML to create clickable buttons. You can create clickable button using <input> tag.
- When you use the <input> element to create a button, the type of button you create is specified using the type attribute.

The type attribute can take the following values:

- **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.

Here is the example:

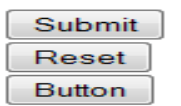
<pre>&lt;form action="abc.php" method="get"&gt; &lt;input type="submit" name="Submit" value="Submit" /&gt; &lt;br/&gt; &lt;input type="reset" value="Reset" /&gt;&lt;br/&gt; &lt;input type="button" value="Button" /&gt; &lt;/form&gt;</pre>	
---	--

- You can use an image to create a button. Here is the syntax:  

```
<form action="http://www.example.com/test.asp" method="get">
<input type="image" name="imagebutton" src="URL" />
```

</form>

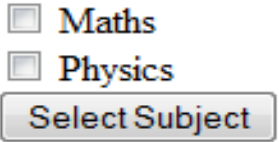
- Here *src* attribute specifies a location of the image on your webserver.
- You can use <button> element to create various buttons. Here is the syntax:

<pre> &lt;form&gt;   &lt;button type="submit"&gt;Submit&lt;/button&gt;   &lt;br /&gt;&lt;br /&gt;   &lt;button type="reset"&gt; Reset &lt;/button&gt;   &lt;button type="button"&gt; Button &lt;/button&gt;&lt;/form&gt; </pre>	<p>Will result</p> 
---	---

### Checkboxes Control:

- Checkboxes are used when more than one option is required to be selected.
- They are created using <input> tag

Here is example HTML code for a form with two checkboxes

<pre> &lt;form&gt; &lt;input type="checkbox" name="maths" value="on"&gt; Maths&lt;br&gt; &lt;input type="checkbox" name="physics" value="on"&gt; Physics&lt;br&gt; &lt;input type="submit" value="Select Subject"/&gt; &lt;/form&gt; </pre>	<p>Will result</p> 
---	--

### Following is the list of important checkbox attributes:

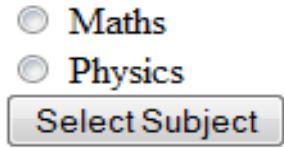
- **Type:** Indicates that you want to create a checkbox.
- **Name:** Name of the control.
- **Value:** The value that will be used if the checkbox is selected. More than one checkbox should share the same name only if you want to allow users to select several items from the same list.
- **Checked:** Indicates that when the page loads, the checkbox should be selected.

### Radio Control:

- Radio Buttons are used when only one option is required to be selected. They are created using

<input> tag as shown below:

Here is example HTML code for a form with two radio button

<pre> &lt;form action="abc.php" method="post"&gt; &lt;input type="radio" name="subject" value="maths" /&gt;Maths&lt;br&gt; &lt;input type="radio" name="subject" value="physics" /&gt; Physics&lt;br&gt; &lt;input type="submit" value="Select Subject" /&gt; &lt;/form&gt; </pre>	<p>Will result</p> 
--	--


Following is the list of important radiobox attributes:

- **Type:** Indicates that you want to create a radiobox.
- **Name:** Name of the control.
- **Value:** Used to indicate the value that will be sent to the server if this option is selected.
- **Checked:** Indicates that this option should be selected by default when the page loads.

## HTML Forms - Select box Control:

Drop down Box is used when we have many options available to be selected but only one or two will be selected.

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

<pre> &lt;form&gt; &lt;select name="dropdown"&gt; &lt;option value="Maths" selected&gt;Maths&lt;/option&gt; &lt;option value="Physics"&gt;Physics&lt;/option&gt; &lt;/select&gt; &lt;input type="submit" value="Submit" /&gt; &lt;/form&gt; </pre>	<p>Will result:-</p> 
--	---

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

- **Name:** This is the name for the control.
- **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>:

- **Value:** The value that is sent to the server if this option 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 Select Boxes:

- If you want to allow a user to upload a file to your web site from his computer, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element.

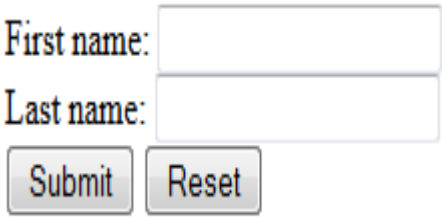
Here is example HTML code for a form with one file select box

<pre>&lt;form name="fileupload" enctype="multipart/form-data"&gt; &lt;input type="file" name="fileupload" accept="image/*" /&gt; &lt;/form&gt;</pre>	<p>will result</p> 
--	---

## Submit and Reset Button:

-These are special buttons which can be created using <input> When submit button is clicked then Forms data is submitted to the back-end application. When reset button is clicked then all the forms control are reset to default state.

Example

<pre>&lt;form&gt; First name: &lt;input type="text" name="first_name" /&gt; &lt;br&gt; Last name: &lt;input type="text" name="last_name" /&gt; &lt;br&gt; &lt;input type="submit" value="Submit" /&gt; &lt;input type="reset" value="Reset" /&gt;</pre>	<p>will result:-</p> 
---	---

</form>	
---------	--

## Page Layout and Design Considerations

### Technical Design Considerations

#### Screen Resolution

- Screen resolution is dependent on the hardware and the settings on the computer that is going to view your page. It is measured in pixels. When designing your pages you should make sure that they look good at the lowest resolution.

#### Color Depth

- Another factor that is determined by the hardware of the surfer's computer is the number of colors supported; the video card or the current video settings can limit this. On a PC, the video driver defaults to 16 colors and in many cases, older video cards only support 16 colors.
- Reducing the colors used in graphics has the additional benefit of reducing the file size, which allows faster downloading.

#### Document Size vs. Download Time

- Probably the biggest technical design consideration is the overall size of the page versus the time it takes to load on the surfer's computer.
- Several things affect the speed at which a browser receives your page. The primary one is the speed of the user's connection to the Internet; usually controlled by the surfer's modem speed. This is another factor that is out of the control of the designer.
- As a designer you should add up the following components to calculate the overall size of your web page:

Page Size = HTML page + Graphics + audio + other (ActiveX and/or Java)

A 50 KB (Kilo Byte) graphics file will take (50000/1800) 27 seconds to download.

- Keep graphics less than 50KB usually
- Keep total page size less than 75KB (45 seconds @ 14.4)
- Use interlaced graphics (don't use for backgrounds)

#### Page Loading – HTTP 1.0 Differences with HTTP 1.1

- Many questions come up over how a page is loaded, and a lot of that can be answered by the way the protocol that transports the page from the server to the web client works.
- Web servers and web client software will support HTTP 1.1, which has several enhancements over HTTP 1.0. The major enhancement is that only one connection will be established for the page and all the elements that need to be downloaded. This will reduce the number of connections that must be opened and maintained, at the server and will decrease download time.

## **Browser Compatibility**

- Web client software is written by a number of companies and as such there are a number of differences between the browsers in the way that they interpret standard HTML code.
- One difference is the way in which the browsers draw tables. Internet Explorer will redisplay the table as new elements fill in the cells. Navigator 4 downloads all the elements before displaying the table and so does not perform multiple redraws as the table fills with data.

## **Page layout**

### Page Layout Guidelines

1. In most cases your pages should not require the viewer to scroll to the right. The maximum width of a page should be 600 pixels in order to be displayed properly, regardless of the display resolution.
2. Recommended components of a home page:
  - i. E-mail address (i.e. webmaster@whatever.com)
  - ii. Mailing address
  - iii. Phone numbers
  - iv. Interactive component that encourages repeat visits by your target audience
  - v. Look, feel and message should reflect the organization
  - vi. Information that is relevant to the target audience
  - vii. Directions to what your target audience wants to know or do
3. To control a page you may want to create a table with a maximum size of 600 pixels and contain the page inside the table.

4. When deciding to use a Background Image that has a distinct left side you should make sure it is 1280 pixels wide, because the browser tiles the background image, and anything smaller will appear again on the right of the page when viewed at high resolutions.
5. Use the DIV and CENTER element to control layout. (Div creates divisions within the document that do not have preformatted spacing associated with them, like paragraphs) You can control the alignment of objects with the alignment attribute of the DIV element.
6. To create attractive looking forms you should use a table to control the layout and alignment of the form elements.
7. Web pages should use white space at the borders. This makes the pages easier to look at and to read. Printed pages usually have whitespace at the left and right hand sides; as well, at the top and bottom of the page. This effect can be achieved by centering the contents of the page.
8. Frames are good for keeping people in your site. Careful consideration must be given to using Frames, as they require Frames-capable browsers. Always remember to provide valuable information in the NoFrames section of the Frameset.
9. You should be consistent in your layout across multiple pages, people become comfortable with the format, and begin to expect it.
10. Bigger is better. If things are too small they will strain the viewer's eyes and it will not be comfortable for them.
11. You should vary sizes for text. When everything is the same size it becomes very boring.
12. Blinking text should only be used in situations where the information is changing frequently.
13. be sure to use case properly. Mix upper and lower case as required or for effect. Documents done in all UPPER case are hard to read.

## Site Design Factors and Criteria

- When developing a web site from scratch you should first answer the questions related to why people will be visiting the site in the first place. Once you know what problem you are trying to solve with the web site you can then start your design.
- In general a site should be intuitive for the person visiting it. It should be very easy for them to contact your organization at any time and they should be able to locate the information they are looking for in three to five mouse clicks.
  1. The Home page should be simple and fast.
  2. The Home page should answer some basic questions for the surfer:
    - i. Is this the organization I was looking for?
    - ii. Do they have what I am looking for?
    - iii. How can I contact them?
    - iv. How do I get what I want now?

## Site Layout and Navigation

1. Navigation controls should be located across the top or the right hand side of the page. This will locate your navigation controls close to the browser's controls for movement and up and down scrolling.
2. As previously mentioned you should only have 3 to 5 mouse clicks to move from start to destination.
3. Create sub sites to simplify navigation, and for marketability.  
E.g. [education.novell.com](http://education.novell.com), [insight.netscape.com](http://insight.netscape.com)