

HTML Forms

HTML Forms Definition: are one of the main points of interaction between a user and a web site or application. They allow users to send data to the web site.

Usage: 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.

Procedure: A form will take input from the site visitor and then will post it to a back-end application such as PHP script on the server side. The back-end application will perform required processing on the passed data.

HTML form syntax:

```
<form action = "Script URL" method = "GET|POST">  
  form elements like input, textarea etc.  
</form>
```

<form> tag: defines a form that is used to collect user input.

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. If the action attribute is omitted, the action is set to the current page. Ex. <form action="/action_page.php" >
method	Specifies the HTTP method to be used when submitting the form data. The most frequently used are GET and POST methods. - Get Method: The default method when submitting form data. However, when GET is used, the submitted form data will be visible in the page address field:

	<p>/action_page.php?firstname=Dana&lastname=Obada</p> <p>The length of a URL is limited. Never use GET to send sensitive data, it is better for non-secure data, like query strings in Google.</p> <ul style="list-style-type: none"> - Post Method: Always use POST if the form data contains sensitive or personal information. The POST method does not display the submitted form data in the page address field. it has no size limitations, and can be used to send large amounts of data.
target	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, iframe name etc.
enctype	<p>You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are :</p> <p>application/x-www-form-urlencoded – This is the standard method most forms use in simple scenarios.it is the default.</p> <p>multipart/form-data – This is used when you want to upload binary data in the form of files like image, word file etc.(attachment)</p> <p>The enctype attribute can be used only if method="post".</p>

Example

```
<form action="/action_page.php" target="_blank" method="GET">
```

```
First name:<br>
```

```
<input type="text" name="firstname" value="Mickey">
```

```
<br>
```

```
Last name:<br>
```

```
<input type="text" name="lastname" value="Mouse">
```

```
<br><br>
```

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

```
</form>
```

Form elements:

An HTML form contains **form elements**, which are different types of input elements, like: text fields, checkboxes, radio buttons, submit buttons, and more.

HTML Form Elements

The **<input>** tag is the most important form element. The `<input>` tag can be displayed in several ways, depending on the type attribute.

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

1. 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 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 HTML **<textarea>** tag.

A. 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 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>
```

First name:

Last name:

```
<body>
```

```
<form >
```

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

```
<br>
```

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

```
</form>
```

```
</body>
```

```
</html>
```

value = "Ali"

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 .



HTML Forms

Instructor: Asma'a Khtoom

Handwritten: name Ali

value	This can be used to provide an initial value inside the control .
size	Allows to specify the <u>width of the text-input control</u> in terms of characters.
maxlength	Allows to specify the maximum number of characters a user can enter into the text box .

B. 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 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" />
```

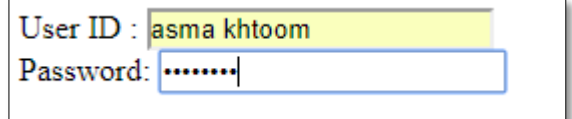
```
<br>
```

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

```
</form>
```

```
</body>
```

```
</html>
```

A screenshot of a web form. It contains two input fields. The first field is labeled 'User ID :' and contains the text 'asma khtoom'. The second field is labeled 'Password:' and contains seven dots, indicating a password mask.

C. Multiple-Line Text Input Controls

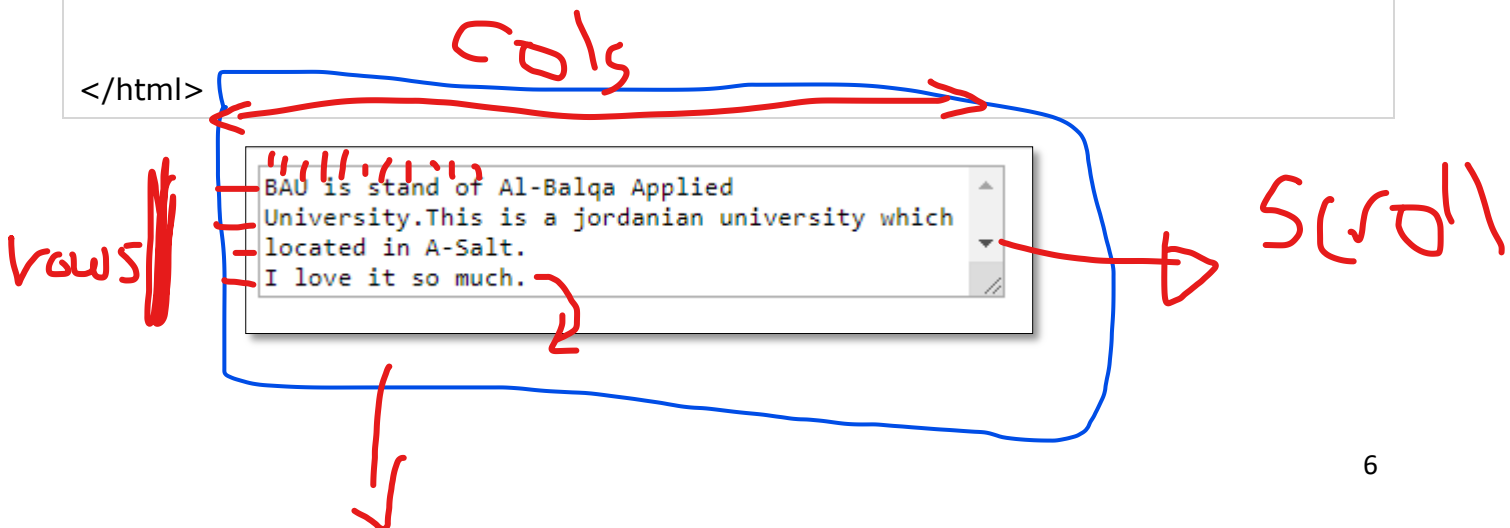
This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag. it can hold an unlimited number of characters.

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 : <br />
      <textarea rows = "4" cols = "50" name = "description">
        BAU is stand of Al-Balqa Applied University. This is a jordanian university which
        located in A-Salt.
        I love it so much.
      </textarea>
    </form>
  </body>
</html>
```



2. Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML `<input>` tag but type attribute is set to **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" checked> Maths
```

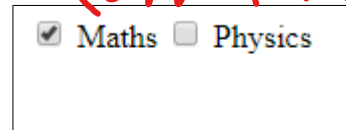
```
<input type = "checkbox" name = "physics" value = "on"> Physics
```

```
</form>
```

```
</body>
```

```
</html>
```

① Multiple choice
② Limited choices (small)



Check box Attributes: Following is the list of attributes for `<checkbox>` tag.

1. **type:** it will be set to **checkbox**
2. **name:** name to the control which is sent to the server to be recognized and get the value.
3. **value:** The value that will be used if the checkbox is selected. This attribute if not set the default will be set to on. (value=on) if the option is selected.
4. **Checked:** Set to *checked* if you want to select it by default.

3. 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 HTML `<input>` tag but type attribute is set to **radio**. Only one radio button in a group can be selected at the same time.

value: this attribute if not set the default will be set to on.(value=on) if the option is selected.

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" checked > Maths

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

    </form>

  </body>

</html>
```



Attributes

Attributes for radio button are the same as checkbox: but the **name attribute** should **be the same for all radio buttons in the same group**.

Example2.

Please select your preferred contact method:

☐ Email ☐ Phone ☒ Mail

4. Select Box Control

A select box, also called drop down box, which provides 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" size="2" multiple="multiple">

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

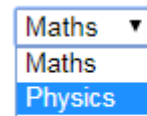
        <option value = "Physics "> Physics</option>

      </select>

    </form>

  </body>

</html>
```



select Attributes:

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

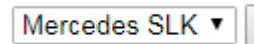
1. **name:** Used to give a name to the control which is sent to the server to be recognized and get the value.
2. **Size:** specify how many options should be shown at one. **The default is 1.**
But if multiple attribute is determined the default value of the size would be 4.
3. **multiple:** If set to "multiple" then allows a user to select multiple items from the menu at once.by pressing the ctrl key and choose the another choice.

<option> tag Attribute: defines an option in a select list. The <option> tag can be used without any attributes, but you usually need the **value** attribute, which indicates what is sent to the server.

1. **value:** Specifies the value to be sent to a server, if **no value** attribute is included, the value defaults to the text contained inside the element.
2. **selected:** Specifies that this option should be the initially selected value when the page loads.

Example1.

```
<form action="/action_page.php">
  <select name="cars">
    <option value="volvo">VolvoXC90</option>
    <option value="saab">Saab95</option>
    <option value="mercedes"selected>MercedesSLK</option>
    <option value="audi">AudiTT</option>
  </select>
  <input type="submit" value="Submit">
</form>
```



Example 2.

```
<select name="pet-select">

  <option value="">--Please choose an option--</option>

  <option value="dog">Dog</option>

  <option value="cat">Cat</option>

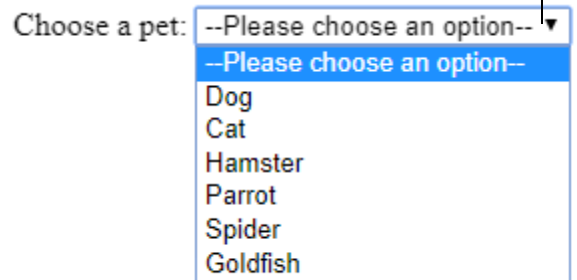
  <option value="hamster">Hamster</option>

  <option value="parrot">Parrot</option>

  <option value="spider">Spider</option>

  <option value="goldfish">Goldfish</option>

</select>
```



5. File Upload Box

If you want to allow a user to upload a file to your web site, 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.

```
<form method="post" enctype="multipart/form-data">
<fieldset>
```

```
<legend>Profile</legend>
```

Display name:

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

```
<br><br>
```

Profile picture:

```
<input type="file" name="prof" multiple="multiple" accept="image/*, .jpg, .doc, .docx" />
```

```
</fieldset>
```

```
</form>
```

File upload box **Attributes**:

- **accept**: Specifies the types of files that the server accepts.

The **accept attribute** values is a string containing one or more of these unique content specifiers, separated by commas. Each unique content specifier may take one of the following forms:

- A valid case-insensitive filename extension, starting with a period (".") character. For example: ".jpg", ".pdf", or ".doc".
- A valid MIME type string, with no extensions.
 - The string "audio/*" meaning "any audio file".
 - The string "video/*" meaning "any video file".
 - The string "image/*" meaning "any image file".

Attention: If you want to send files, you need to take extra three steps:

1. Set the method attribute to POST because file content can't be put inside URL parameters.
2. Set the value of enctype to multipart/form-data because the data will be split into multiple parts, one for each file plus one for the text data included in the form body (if text is also entered into the form).
3. To include one or more file to allow your users to select the file(s) that will be uploaded use **accept and multiple attributes**.

6. Button Controls

There are various ways in HTML to create clickable buttons.

1. **You can create a clickable button using <input>tag by setting its type attribute to button.** 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.

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

```
<html>

<head>
    <title>File Upload Box</title>
</head> <body>

    <form>

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

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

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

    </form>

</body></html>
```

2. **You can use button element to create html buttons:**

Example1: <button type="submit" value="Send"> Send </button>

Example2: <button type="reset" value="reset"> Reset </button>

Example3: Create submit button as an image

```
<button type="submit" value="Submit">
    
</button>
```

Example4: Create normal button

```
<button type="button" value="Color">Color </button>
```

7. 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, following hidden form is being used to keep current page number. When a user will click next page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

Example

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

```
<!DOCTYPE html>

<html>

  <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>
```

8. The HTML <fieldset> element: is used to group several controls within a web form.

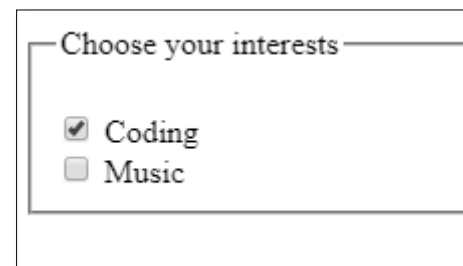
9. HTML <legend> element: represents a caption for the content of its parent <fieldset>.

Example.

```
<form>
  <fieldset>
    <legend>Choose your interests</legend><br>

    <input type="checkbox" name="interest" value="coding" checked> Coding <br>

    <input type="checkbox" name="interest" value="music"> Music
  </fieldset>
</form>
```



10. Form <label> element: represents a caption for an item in a user interface.

Hint: Use label to **providing a bigger hit area** for your checkboxes. HTML form labels that makes it easier to click the option you want, especially on small-screen devices like smartphones.

Example.

```
<form>
```

```
<label>Do you like peas?
```

```
<input type="checkbox" name="peas">
```

Do you like peas? ☒

```
</label>
```

```
</form>
```

11. <input type="email">

The <input type="email"> is used for input fields that should contain an e-mail address. Depending on browser support, the e-mail address **can be automatically validated** when submitted.

Some smartphones recognize the email type, and adds ".com" to the keyboard to match email input.

Example1.

```
<form>
```

```
E-mail:
```

```
<input type="email" name="email">
```

```
</form>
```

Email Field

E-mail:

You must enter a valid email address

If you don't enter @ this caused an error.

12. <input type="number">

The <input type="number"> defines a **numeric** input field. You can also set restrictions on what numbers are accepted. The following example displays a **numeric input field, where you can enter a value from 1 to 5:**

Example1.

```
<form action="/action_page.php">
```

Quantity (between 1 and 5):

```
<input type="number" name="quantity" min="1" max="5"> <input  
type="submit">
```

```
</form>
```

Number Field

Quantity (between 1 and 5): x

You must enter a value between 1 and 5

Example2.

```
<input type="number" name="quant" min="1" max="5" value="30" step="10">
```

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

Quantity:

After one click down:(step10)

Quantity:

Different Types Of Form Validation

There are two different types of form validation which you'll encounter on the web:

- **Client-side validation** is validation that occurs in the browser before the data has been submitted to the server. This is more user-friendly than server-side validation as it gives an instant response. This can be further subdivided:
 - **JavaScript** validation is coded using JavaScript. It is completely customizable.
 - **Built-in form validation** using HTML5 form validation features.
- **Server-side validation** is validation which occurs on the server after the data has been submitted. Server-side code is used to validate the data before it is saved into the database. If the data fails authentication, a response is sent back to the client to tell the user what corrections to make.

Form Data Validation Attributes

There is two important attributes that's allows you to add basic **data validation** without resorting to JavaScript.

1. Required attribute (means an input mandatory)

This attribute is a Boolean attribute. When present, it specifies that an input **field must be filled out before submitting the form**.

The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.

Example.

```
<form action="/action_page.php">
```

```
E-mail: <input type="email" name="email" >
```

```
E-pass: <input type="password" name="psw" required="required">
```

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

```
</form>
```

If the user does not enter password and press submit button, next message will be shown:

2. Pattern attribute

The pattern attribute of the `<input>` element allows you to add basic data validation without resorting to JavaScript. It works by matching the input value against a regular expression. A **regular expression** is a formalized string of characters that define a pattern.