**Creating an HTML Form**

The HTML [**<form>**](https://www.tutorialspoint.com/html/html_form_tag.htm) tag is used to create an HTML form. There are a few more tags which are required to create an actual form - all those tags are briefly desribed in this post. The **<form>** element contains various predefined tags and elements termed as form controls.

Syntax

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

form controls like input, textarea etc.

</form>

## HTML Form Elements

There are list of elements which can be used within the form element. All the elements are briefy described bellow.

### 1. HTML <form> Element

[**HTML form tag**](https://www.tutorialspoint.com/html/html_form_tag.htm) is used to create the form element, this element is the container for all other form elements. The form element soes not create the form it's container that keeps the other form elements.

### Syntax

<form>.....</form>

### 2. HTML <input> Element

[**HTML input tag**](https://www.tutorialspoint.com/html/html_input_tag.htm) is an essential element of form control for gathering user input from the web sites, we can use this tag to create an input element.

### Syntax

<input type = ".."/>

### 3. HTML <label> Element

[**HTML label tag**](https://www.tutorialspoint.com/html/html_label_tag.htm) is used to create label element that represent a caption for an item in a UI(user interface), or to add labels to a form control like text, textarea, checkbox, radio button, etc.

### Syntax

<label>.......</label>

### 4. HTML <legend> Element

[**HTML legend tag**](https://www.tutorialspoint.com/html/html_legend_tag.htm) is the element's first child, specifies the caption or title for the <fieldset> tag.

### Syntax

<legend>.......</legend>

### 5. HTML <select> Element

[**HTML select tag**](https://www.tutorialspoint.com/html/html_select_tag.htm) is used to create the dropdon in HTML forms, we can use this tag to create dropdown anywhere we want.

### Syntax

<select>....</select>

### 6. HTML <button> Element

[**HTML button Tag**](https://www.tutorialspoint.com/html/html_button_tag.htm) is an interactive element that is used to create a button in HTML.

### Syntax

<button>Button</button>

### 7. HTML <fieldset> Element

[**HTML fieldset tag**](https://www.tutorialspoint.com/html/html_fieldset_tag.htm) is used to group several controls within a web form. By using the <fieldset> tag and <legend> tag a form can be much eaiser to understand to the users.

### Syntax

<fieldset>....</fieldset>

### 8. HTML <datalist> Element

[**HTML datalist tag**](https://www.tutorialspoint.com/html/html_datalist_tag.htm) contains a set of <option> elements that represent recommended options available to choose from among others.

### Syntax

<datalist>....</datalist>

### 9. HTML <output> Element

[**HTML output tag**](https://www.tutorialspoint.com/html/html_output_tag.htm) is a flexible and under used component that enables programmers to dynamically show the outcomes of calculations or scripts inside the content.

### Syntax

<output> Results... </output>

### 10. HTML <option> Element

[**HTML option tag**](https://www.tutorialspoint.com/html/html_option_tag.htm) defines either the elements of the data list for autocomplete, specified by the <datalist> tag, or the items of a drop-down list, defined by the <select> tag.

### Syntax

<option>.....</option>

### 11. HTML <optgroup> Element

[**HTML optgroup tag**](https://www.tutorialspoint.com/html/html_optgroup_tag.htm) is used in the <select> element to group together relevant <option> elements.

### Syntax

<optgroup>

<option>..</option>

.

.

</optgroup>

### 12. HTML <textarea> Element

[**HTML textarea tag**](https://www.tutorialspoint.com/html/html_textarea_tag.htm) is used to represent a multiline plain-text editing control.

### Syntax

<textarea>.......</textarea>

### Creating Simple HTML Form

In the following example, we will create a simple HTML form(login form) where a user can enter his/her name with the help of **<form>** element and form controls like input, type and name.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Sample HTML Form </title>

</head>

<body>

<!-- Start of the form element -->

<form action = " ">

<!-- various form controls -->

<label for="first\_name">First name:</label>

<input type = "text" name = "first\_name" />

<br><br>

<label for="first\_name">Last name:</lable>

<input type = "text" name = "last\_name" />

<br><br>

<input type = "submit">

</form>

</body>

</html>

## What are Form Attributes?

In **HTML**, each element has its own **attributes** that are used to define the characteristics of that particular HTML element and are placed inside the element's opening tag. The **<form>** element also has attributes that provide different functionalities like redirection on other web pages and auto completion of text.

Following is a list of the most frequently used form attributes −

* action
* method
* target
* autocomplete
* enctype
* novalidate

## The action Attribute

The **action** attribute of the **<form>** element transmits the user's input to a backend script for processing. A form is of no use unless it processes the information provided by the user. Therefore, it is important to pass the URL of a program to the action attribute. Note that the **formaction** attribute can override the value of action attribute.

### Example

The following example illustrates the use of **action** attribute. When we click the submit button, the form will redirect us to the home page of Tutorialspoint.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title> The action Attribute </title>

</head>

<body>

<!-- Start of the form element -->

<form action = "https://www.tutorialspoint.com">

<!-- to take input -->

Name: <input type = "text" name = "your\_name" required/>

<br><br>

Email: <input type = "email" name = "mail" required/>

<br><br>

<!-- to submit the data -->

<input type = "submit">

</form>

</body>

</html>

## The method Attribute

The method attribute determines which HTTP method should be used by the browser while uploading the form information. The most commonly used methods are as follows −

|  |  |
| --- | --- |
| **S.No** | **Values & Description** |
| 1 | **GET**  It is the default method for form submission which means if we don't specify the method name explicitly the form will use the GET method to send data. |
| 2 | **POST**  It is used to send form data inside HTTP request body. It is safer than GET method. |

*It is not recommended to use the GET method while sending sensitive information like credit/debit card numbers and passwords because it exposes the submitted data in the URL.*

### Example

The following example demonstrate how to use the method attribute of <form> element.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title> The method Attribute </title>

</head>

<body>

<!-- Start of the form element -->

<form action = "https://www.tutorialspoint.com" method = "post">

<!-- to take input -->

Name: <input type = "text" name = "your\_name" required/>

<br><br>

Email: <input type = "email" name = "mail" required/>

<br><br>

<!-- to submit the data -->

<input type = "submit">

</form>

</body>

</html>

On clicking the submit button, user will be redirected to the home page of Tutorialspoint.

## The target Attribute

The **target** attribute determines the target window or frame where the result of the script will be displayed after submitting the form. The default target is the current window. The target attribute accepts the following values −

|  |  |
| --- | --- |
| **S.No.** | **Values & Description** |
| 1 | **\_self**  It opens the response in the same frame as it was clicked. |
| 2 | **\_blank**  It opens the response in the new window or tab. |
| 3 | **\_parent**  Opens the response in the parent frame. |
| 4 | **\_top**  Opens the response in the full body of window. |
| 5 | **framename**  Opens the response in the named iframe. |

### Example

In the following example, we will use the **target** attribute with the value **\_self**. The response will be open in the current window.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title> The target Attribute </title>

</head>

<body>

<!-- Start of the form element -->

<form action = "https://www.tutorialspoint.com" target = "\_self">

<!-- to take input -->

Name: <input type = "text" name = "your\_name" required/>

<br><br>

Email: <input type = "email" name = "mail" required/>

<br><br>

<!-- to submit the data -->

<input type = "submit">

</form>

</body>

</html>

## The novalidate Attribute

The **novalidate** is a Boolean attribute that indicates the form does not need any kind of validation. The term validation refers to the process of verifying the correctness of user input based on predefined conditions. This attribute, when applied, exempts the form from such checks, allowing user inputs to bypass these conditions.

*If Boolean Attributes like novalidate are present on an HTML element, it specifies true and in the case of absence, false is assumed. They do not accept any values.*

### Example

In the previous examples, the form redirected us to a new web page when we entered our name and email. For this example, we will use the novalidate attribute which will allow the redirection without enterning any information.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title> The novalidate Attribute </title>

</head>

<body>

<!-- Start of the form element -->

<form action = "https://www.tutorialspoint.com" target = "\_blank" autocomplete="off" method = "get" novalidate>

<!-- to take input -->

Name: <input type = "text" name = "your\_name" required/>

<br><br>

Email: <input type = "email" name = "mail" required/>

<br><br>

<!-- to submit the data -->

<input type = "submit">

</form>

</body>

</html>

## The "autocomplete" Attribute

The **autocomplete** attribute of HTML predicts and suggests the subsequent input based on the initial characters entered in the input field. This attribute primarily has two states namely **on** and **off**.

|  |  |
| --- | --- |
| **S.No.** | **Values & Description** |
| 1 | **on**  By default, the autocomplete attribute is set to **on**, enabling the autocomplete functionality. |
| 2 | **off**  The autocomplete attribute can be toggled to **off** to disable this feature as per the requirements of the web application. |

### Instance

<form action = "https://www.tutorialspoint.com" target = "\_blank" autocomplete="off" method = "get">

## The "enctype" Attribute

We use the **enctype** attribute to specify how the browser encodes the data before it sends it to the server. Its possible values are −

|  |  |
| --- | --- |
| **S.No.** | **Values & Description** |
| 1 | **application/x-www-form-urlencoded**  This is the standard method most forms use in simple scenarios. |
| 2 | **mutlipart/form-data**  This is used when you want to upload binary data in the form of files like image, word file etc. |
| 3 | **text/plain**  It only encodes the spaces into + symbol. |

### Instance

<form action = "https://www.tutorialspoint.com" target = "\_blank" autocomplete="off" method = "get" enctype = "text/plain">

## HTML Form Controls

The form elements that are used to create controls for the user interaction within the browser are termed as **form controls**. They enable users to enter information for the server side processing. The nature of interaction with the server can vary depending on the type of control used while creating the form. For example, radio buttons are typically used to accept gender information. We have used several common form controls in previous discussions, we will now dive into a more detailed exploration of these elements.

There are different types of form controls that we can use to collect data using HTML form −

* Text Input Controls
* Checkboxes Controls
* Radio Box Controls
* Select Box Controls
* File Select boxes
* Button Controls
* Hidden Form Controls
* Datetime Controls
* Date Controls
* Month Controls
* Week Controls
* Time Controls
* Number Controls
* Range Controls
* Email Controls
* URL Controls

Let us look at these controls one by one −

## Text Input Controls

The text input controls are further divided into three main categories −

* Single-line text input controls
* Password input controls
* Multi-line text input controls

### Single-line text input controls

The **single-line text input 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**

The following example illustrates how to take a single-line text input.

<!DOCTYPE html>

<html>

<head>

<title>Text Input Control</title>

</head>

<body>

<form >

First name: <input type = "text" name = "first\_name" />

<br><br>

Last name: <input type = "text" name = "last\_name" />

</form>

</body>

</html>

On running the above code, two single-line text input fields will be displayed.

### Password input controls

The **password input control** 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**

In the following example, we will demonstrate how to take password input from users.

<!DOCTYPE html>

<html>

<head>

<title>Password Input Control</title>

</head>

<body>

<form >

User ID : <input type = "text" name = "user\_id" />

<br><br>

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

</form>

</body>

</html>

The above HTML code will display one text input field and one password input field.

### Multiple-Line Text Input Controls

The **multiple line text input control** 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.

#### **Example**

The following example demonstrates how to use multi-line text input to take item description.

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<form>

Description : <br />

<textarea rows = "5" cols = "50" name = "description">

Enter description here...

</textarea>

</form>

</body>

</html>

The above code will produce a text area where users can provide multiple lines of text.

## 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

In this HTML code, we are creating 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>

On executing the above code, it will create two checkboxes.

## 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**.

### Example

In this example code, we are creating 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>

On running the above HTML code, it will produce two radio buttons.

## Select Box Control

A **select box** provides features to list down various options in the form of drop down list, from where a user can select one or more options.

### Example

The following HTML code illustrates how to create a form with a 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>

<option value = "Chemistry">Chemistry</option>

</select>

</form>

</body>

</html>

The above HTML code will create a dropdown menu with three values.

## File Upload Box

If we want to allow a user to upload a file to our web site, we 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

In the following example, we will create a form with one file upload box that accepts only images.

<!DOCTYPE html>

<html>

<head>

<title>File Upload Box</title>

</head>

<body>

<form>

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

</form>

</body>

</html>

## Button Controls

There are various ways in HTML to create **clickable buttons**. We can create a clickable button using **<input>** tag by setting its type attribute to **button**.

### Example

In this HTML code, we are creating a form with three different 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 = "imagebutton" src = "/html/images/logo.png" />

</form>

</body>

</html>

## Hidden Form Controls

The **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

The following example shows 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>

## Datetime Controls

In HTML, the **datetime control** represents date and time (year, month, day, hour, minute, second, fractions of a second) together encoded according to ISO 8601 with the time zone set to UTC. If we use the **datetime-local**, it will dispaly date and time with no time zone information.

### Example

<!DOCTYPE html>

<html>

<body>

<form action = "/cgi-bin/html5.cgi" method = "get">

Date and Time : <input type = "datetime" name = "newinput" />

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

</form>

</body>

</html>

## Date Controls

The HTML **date control** is used to specify a date (year, month, day) encoded according to ISO 8601.

### Example

<!DOCTYPE html>

<html>

<body>

<form action = "/cgi-bin/html5.cgi" method = "get">

Date : <input type = "date" name = "newinput" />

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

</form>

</body>

</html>

## The Month Control

In HTML, the **month control** is used to display a date consisting of only a year and a month encoded according to ISO 8601.

### Example

<!DOCTYPE html>

<html>

<body>

<form action = "/cgi-bin/html5.cgi" method = "get">

Month : <input type = "month" name = "newinput" />

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

</form>

</body>

</html>

## Week Controls

As the name suggests, the **week control** displays a date consisting of only a year and a week number encoded according to ISO 8601.

### Example

<!DOCTYPE html>

<html>

<body>

<form action = "/cgi-bin/html5.cgi" method = "get">

Week : <input type = "week" name = "newinput" />

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

</form>

</body>

</html>

## Time Controls

The HTML **time control** specify the hours, minutes, seconds, and fractional seconds encoded according to ISO 8601.

### Example

<!DOCTYPE html>

<html>

<body>

<form action = "/cgi-bin/html5.cgi" method = "get">

Time : <input type = "time" name = "newinput" />

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

</form>

</body>

</html>

## Number Controls

The **number controls** accepts only numerical value. The step attribute specifies the precision and its default values is 1.

### Example

<!DOCTYPE html>

<html>

<body>

<form action = "/cgi-bin/html5.cgi" method = "get">

Select Number : <input type = "number" min = "0" max = "10" step "1"

value = "5" name = "newinput" />

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

</form>

</body>

</html>

## Range Controls

The **range** type is used for input fields that should contain a value from a range of numbers.

### Example

<!DOCTYPE html>

<html>

<body>

<form action = "/cgi-bin/html5.cgi" method = "get">

Select Range : <input type = "range" min = "0" max = "10" step "1"

value = "5" name = "newinput" />

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

</form>

</body>

</html>

## Email Controls

The **email** control accepts only email value. This type is used for input fields that should contain an e-mail address. If you try to submit a simple text, it forces to enter only email address in email@example.com format.

### Example

<!DOCTYPE html>

<html>

<body>

<form action = "/cgi-bin/html5.cgi" method = "get">

Enter email : <input type = "email" name = "newinput" />

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

</form>

</body>

</html>

## URL Controls

The HTML **URL control** accepts only URL value. This type is used for input fields that should contain a URL address. If you try to submit a simple text, it forces to enter only URL address either in **http://www.example.com** format or in **http://example.com** format.

### Example

<!DOCTYPE html>

<html>

<body>

<form action = "/cgi-bin/html5.cgi" method = "get">

Enter URL : <input type = "url" name = "newinput" />

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

</form>

</body>

</html>

Print Page

HTML Input Attributes

The HTML **input attributes** are used to define the characteristics and behavior of the **<input>** element. These attributes are used with the different types of input fields such as text, email, password, date, number and so forth. Note that the Input element is used to create interactive controls for the web-based forms so that it can accept data from the user.

The <input> element requires only an opening tag and it will work only if we add it in between the **<form>** tags. In this tutorial, we are going to explore the attributes that are used with <input> element.

The attributes of the <input> element are as follows −

* name
* value
* type
* size
* maxlength
* readonly
* disabled
* min
* max
* accept
* multiple
* placeholder
* required
* autofocus
* list

The "type" and "name" Attribute

The **type** attribute indicates the type of input control like text, password, email and so on. The **name** attribute of an **input element** assigns an identifier to the form control that enables the server to recognize and retrieve the value.

Example

The following HTML code illustrates the use of type and name Attributes.

<!DOCTYPE html>

<html>

<head>

<title>The type and name Attributes</title>

</head>

<body>

<form >

First name: <input type = "text" name = "first\_name" />

<br><br>

Last name: <input type = "text" name = "last\_name" />

</form>

</body>

</html>

The "value" Attribute

The **value** attribute is used to provide an initial value inside the input control.

Example

In the following example, we are creating two input fields with initial value as " first name..." and " last name...".

<!DOCTYPE html>

<html>

<head>

<title>The value Attribute</title>

</head>

<body>

<form >

First name: <input type = "text" name = "first\_name" value = "first name..." />

<br><br>

Last name: <input type = "text" name = "last\_name" value = "last name..."/>

</form>

</body>

</html>

The "size" Attribute

The **size** attribute allows to specify the width of the text-input control in terms of characters. The default size is 20 characters.

Example

In this example, the size of text-input control is set to 40.

<!DOCTYPE html>

<html>

<head>

<title>The size Attribute</title>

</head>

<body>

<form >

First name: <input type = "text" name = "first\_name" size = "40" />

<br><br>

Last name: <input type = "text" name = "last\_name" size = "40"/>

</form>

</body>

</html>

The "maxlength" Attribute

The **maxlength** attribute allows to specify the maximum number of characters a user can enter into the text box.

Example

The following example demonstrates how to set the **maxlength** of an input field.

<!DOCTYPE html>

<html>

<head>

<title>The maxlength Attribute</title>

</head>

<body>

<form >

First name: <input type = "text" name = "first\_name" />

<br><br>

Last name: <input type = "text" name = "last\_name" />

<br><br>

Contact: <input type = "text" name = "phone" maxlength = "10"/>

</form>

</body>

</html>

The "readonly" Attribute

The **"readonly"** attribute of an input field indicates the field as read-only. Although the content of a read-only field cannot be altered, users can still select it and copy the text. Also, the value of a read-only field is included when the form is submitted.

Example

Following example shows the use of **"readonly"** attribute of **<input>** element −

<!DOCTYPE html>

<html>

<head>

<title>The readonly Attribute</title>

</head>

<body>

<form >

Emp. Name: <input type = "text" name = " your\_name" value = "your name..."/>

<br><br>

Emp. Email: <input type = "text" name = "mail" value = "your email..."/>

<br><br>

Organization: <input type = "text" name = "organization" value = "Tutorialspoint" readonly/>

</form>

</body>

</html>

The "disabled" Attribute

The **disabled** attribute of an input field indicates the field as disabled. Unlike readonly, the value of a disabled field will not be included when the form is submitted.

Example

In this example, the field containing Organization name is marked as disabled.

<!DOCTYPE html>

<html>

<head>

<title>The disabled Attribute</title>

</head>

<body>

<form >

Emp. Name: <input type = "text" name = "your\_name" value = "your name..."/>

<br><br>

Emp. Email: <input type = "email" name = "mail" value = "your email..."/>

<br><br>

Organization: <input type = "text" name = "organization" value = "Tutorialspoint" disabled/>

</form>

</body>

</html>

The "min" and "max" Attributes

The **min** and **max** attributes determine the minimum and maximum values respectively of an input field like number, date, week and so on. If we use them together, they will allow users to enter an input within a predefined range.

Example

In the following example, we are mentioning the minimum working hours as 3 and maximum as 8 by using the min and max attributes.

<!DOCTYPE html>

<html>

<head>

<title>The min and max Attribute</title>

</head>

<body>

<form >

Emp. Name: <input type = "text" name = "your\_name" value = "your name..."/>

<br><br>

Emp. Email: <input type = "email" name = "mail" value = "your email..."/>

<br><br>

Organization: <input type = "text" name = "organization" value = "Tutorialspoint" readonly/>

<br><br>

Working Hrs: <input type = "number" name = "working\_hours" min="3" max="8"/>

</form>

</body>

</html>

The "accept" and "multiple" Attribute

The **accept** attribute specifies the types of files that the server will take in. If we use the **multiple** attribute, it will allow the users to upload more than one file.

Example

The following HTML code can accept multiple image files.

<!DOCTYPE html>

<html>

<head>

<title>The accept and multiple Attributes</title>

</head>

<body>

<form>

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

</form>

</body>

</html>

The "placeholder" Attribute

The **placeholder attribute** of an input field like text, search and email briefly outlines the desired value of the field. Its predefined value is displayed in the input field until the user begins to enter their own value.

Example

In the following example, we are using the placeholder attribute for the email input field.

<!DOCTYPE html>

<html>

<head>

<title>The placeholder Attribute</title>

</head>

<body>

<form>

Emp. Name: <input type = "text" name = "your\_name"/>

<br><br>

Emp. Email: <input type = "email" name = "mail" placeholder = "example@email.com"/>

</form>

</body>

</html>

The "required" Attribute

The required attribute in an input field like text, search, password and email signifies that the field must contain some values for the form to be successfully submitted. In other words, it indicates the mandatory field.

Example

The following example illustrates the use of required attribute. Without filling the mandatory fields, users will not be able to submit the form.

<!DOCTYPE html>

<html>

<head>

<title>The required Attribute</title>

</head>

<body>

<form >

<p>The \* Star represents mandatory field</p>

Emp. Name: <input type = "text" name = "your\_name" required/>\*

<br><br>

Emp. Email: <input type = "email" name = "mail" placeholder = "example@email.com" required/>\*

<br><br>

<input type = "submit">

</form>

</body>

</html>

The "autofocus" Attribute

The **autofocus attribute** in an input field ensures that the field must be selected automatically once the webpage loads completely. It implies that the cursor will be positioned to the specified input field. In cases where multiple elements use the autofocus attribute, only the first element will acquire the focus.

Example

Following is the example of autofocus attribute.

<!DOCTYPE html>

<html>

<head>

<title>The autofocus Attribute</title>

</head>

<body>

<form >

Emp. Name: <input type = "text" name = "your\_name" autofocus/>

<br><br>

Emp. Email: <input type = "email" name = "mail" placeholder = "example@email.com" />

<br><br>

<input type = "submit">

</form>

</body>

</html>

The "list" Attribute

The **list** attribute defines a set of predefined options for an <input> element, which are defined within a **<datalist>** element. The <input> element uses a specific string as an ID to create a link to the corresponding <datalist> element.

Example

In this example, we are creating a list of cities with the help of list attribute.

<!DOCTYPE html>

<html>

<head>

<title>The list Attribute</title>

</head>

<body>

<form >

Emp. Name: <input type = "text" name = "your\_name" autofocus/>

<br><br>

Emp. Email: <input type = "email" name = "mail" placeholder = "example@email.com" />

<br><br>

Location −

<input list="location" name="cities">

<datalist id = "location">

<option value="Banglore">

<option value="Hyderabad">

<option value="Patna">

<option value="Delhi">

</datalist>

<input type = "submit">

</form>

</body>

</html>

Print Page