HTML Forms & Input Elements

HTML Forms

An HTML form is a webpage section usually used for collecting data from the users and then sent to a server for further processing.

HTML forms are collections of interactive controls and various input types, such as text, numbers, email, password, radio buttons, checkboxes, buttons, etc., that collect user information. HTML forms are created by using the HTML <form> tag. All user input-related tags are placed inside the <form> tag.

The <form> Element

The HTML <form> element defines a form to collect user information in the HTML document.

Syntax:

The <form> element has the following syntax; you can place any input-related element inside it:

HTML
```
<form>
  </form>
```

The following syntax contains all necessary elements:

HTML
```
<form action="url" method="method_type" target="target_value" enctype="enctype_value">
   </form>
```


Why Use HTML Forms?

HTML forms are used to collect user information from the webpage and send it to the server. The common uses for HTML forms are:

- Creating registration forms so that users can sign up with their information and authenticate further to access the functionalities of the websites/web applications.
- Collect data through the different types of surveys, feedback, etc.
- Uploading the images, resumes, or any other type of files.

Creating an HTML Form

To create an HTML form, use the <form> element along with the other required elements based on the information you want to collect, such as input boxes, buttons, radio buttons, checkboxes, etc. These elements are known as form controls (form elements).

Example

The following is an example of an HTML form having some required elements:

HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML Form Example</title>
</head>
<body>
  <h1>HTML Form Example</h1>
  <form>
    <label for="name"><strong>Name:</strong></label>
    <input type="text" id="name" name="name" placeholder="Enter your name" required>
    <br/><br/>

    <label><strong>Gender:</strong></label>
    <input type="radio" id="male" name="gender" value="male">
    <label for="male">Male</label>
    <input type="radio" id="female" name="gender" value="female">
    <label for="female">Female</label>
    <br/><br/>

    <label><strong>Hobbies:</strong></label>
    <input type="checkbox" id="reading" name="hobbies" value="reading">
    <label for="reading">Reading</label>
    <input type="checkbox" id="traveling" name="hobbies" value="traveling">
    <label for="traveling">Traveling</label>
    <input type="checkbox" id="sports" name="hobbies" value="sports">
    <label for="sports">Sports</label>
    <br/><br/>

    <button type="submit">Submit</button>
  </form>
</body>
</html>
```

Output:

In the previous example, we designed a form that accepts user input but doesn't process the data. In this example, users will be redirected to Tutorialspoint's HTML Tutorial upon form submission. The redirection only happens if both the first name and last name fields are filled out; otherwise, the form prompts the user to provide the required information.

Example

HTML

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8">
   <title>Sample HTML Form</title>
</head>
<body>
   <form action="" method="post">
      <label for="first_name">First Name:</label>
      <input type="text" id="first_name" name="first_name" required />
      <br><br>

      <label for="last_name">Last Name:</label>
      <input type="text" id="last_name" name="last_name" required />
      <br><br>

      <input type="submit" value="Submit">
   </form>
</body>
</html>
```

Output:

First Name:

Last Name:

Submit

Form Elements

There is a list of elements that can be used within the form element. All the elements are briefly described below:

1. **The <form> Element:** HTML <form> tag is used to create the <form> element. This element is the container for all other form elements. The form element does not create the form; it's the container that holds the other form elements.
    - Example: <form>.....</form>
2. **The <input> Element:** HTML <input> tag is an essential element of form control for gathering user input from websites. We can use this tag to create an input element.
    - Example: <input type = ".."/>
3. **The <label> Element:** HTML <label> tag is used to create a label element that represents 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.
    - Example: <label>.....</label>
4. **The <legend> Element:** HTML <legend> tag is the element's first child and specifies the caption or title for the <fieldset> tag.
    - Example: <legend>.....</legend>
5. **HTML <select> Element:** HTML <select> tag is used to create the dropdown in HTML forms. We can use this tag to create a dropdown anywhere we want.
    - Example: <select>....</select>
6. **The <button> Element:** HTML <button> tag is an interactive element used to create a button in HTML.
    - Example: <button>Button</button>
7. **The <fieldset> Element:** HTML <fieldset> tag is used to group several controls within a web form. By using the <fieldset> tag and <legend> tag, a form can be much easier for users to understand.
    - Example: <fieldset>....</fieldset>
8. **The <datalist> Element:** HTML <datalist> tag contains a set of <option> elements that represent recommended options available to choose from among others.
    - Example: <datalist>....</datalist>
9. **The <output> Element:** HTML <output> tag is a flexible and underused component that enables programmers to dynamically show the outcomes of calculations or scripts inside the content.
    - Example: <output> Results... </output>

10. **The <option> Element:** HTML <option> tag 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.
    - Example: <option>.....</option>
11. **The <optgroup> Element:** HTML <optgroup> tag is used in the <select> element to group together relevant <option> elements.
    - Example: <optgroup><option>..</option>.....</optgroup>
12. **The <textarea> Element:** HTML <textarea> tag is used to represent a multiline plain-text editing control.
    - Example: <textarea>.....</textarea>

Form Attributes:

HTML form attributes provide specific functionalities, such as redirection to other web pages, auto-completion of text, etc.

The below table lists out some of the common form attributes:

Attribute	Description
action	It is used to specify a URL that processes the form submission.
method	It is used to define which HTTP method to use when submitting the form.
target	It is used to specify where to open the linked document.
autocomplete	It allows you to set whether the autocomplete for the form should be on or off.
enctype	It is used to specify how the form input data should be encoded before sending it to the server.

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.

HTML form attributes provide different functionalities, such as redirection to other web pages, auto-completion of text, specifying data validation rules, controlling the behavior of form submissions, etc.

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 the action attribute.

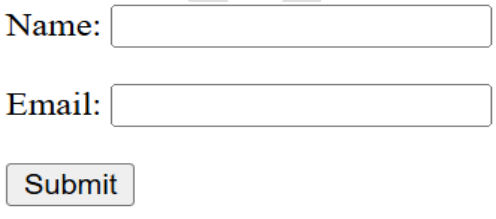
Example

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

HTML

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8">
   <title> The action Attribute </title>
</head>
<body>
   <form action = "https://www.tutorialspoint.com">
      Name: <input type = "text" name = "your_name" required/>
      <br><br>
      Email: <input type = "email" name = "mail" required/>
      <br><br>
      <input type = "submit">
   </form>
</body>
</html>
```

Output:



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:

- **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.
- **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 demonstrates how to use the method attribute of the <form> element.

HTML

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> The method Attribute </title>
</head>
<body>
  <form action = "https://www.tutorialspoint.com" method = "post">
    Name: <input type = "text" name = "your_name" required/>
    <br><br>
    Email: <input type = "email" name = "mail" required/>
    <br><br>
    <input type = "submit">
  </form>
</body>
</html>
```

Output:

Name:

Email:

Submit

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:

- **_self:** It opens the response in the same frame as it was clicked.
- **_blank:** It opens the response in the new window or tab.
- **_parent:** It opens the response in the parent frame.
- **_top:** It opens the response in the full body of window.
- **framename:** It 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:

HTML

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> The target Attribute </title>
</head>
<body>
  <form action = "https://www.tutorialspoint.com" target = "_self">
    Name: <input type = "text" name = "your_name" required/>
    <br><br>
    Email: <input type = "email" name = "mail" required/>
    <br><br>
    <input type = "submit">
  </form>
</body>
</html>
```

Output:

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.

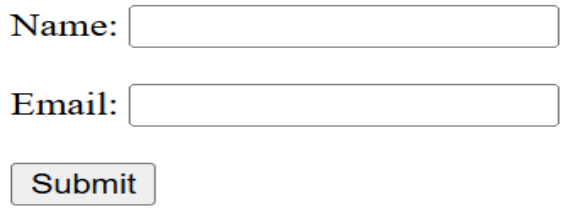
Example

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 entering any information:

HTML

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> The novalidate Attribute </title>
</head>
<body>
  <form action = "https://www.tutorialspoint.com" target = "_blank" autocomplete="off" method = "get" novalidate>
    Name: <input type = "text" name = "your_name" required/>
    <br><br>
    Email: <input type = "email" name = "mail" required/>
    <br><br>
    <input type = "submit">
  </form>
</body>
</html>
```

Output:



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.

- **on:** By default, the autocomplete attribute is set to on, enabling the autocomplete functionality.
- **off:** The autocomplete attribute can be toggled to off to disable this feature as per the requirements of the web application.

Instance

HTML
<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 −

- **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 images, Word files etc.
- **text/plain:** It only encodes the spaces into + symbol.

HTML - Form Controls

HTML Form Controls (Elements)

HTML form controls (elements) are the elements used within the <form> element to collect the user information.

Form Controls (Elements)

The form elements create controls for the user interaction within the webpage; these elements are also termed as form controls. The form elements 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 Control
- Radio Buttons Control
- Select Box Control
- File Select Box
- Button Control
- Hidden Form Control
- Datetime Controls
- Date Control
- Month Control

- Week Control
- Time Control
- Number Control
- Range Control
- Email Control
- URL Control

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 Control
- Password Input Control
- Multi-line Text Input Control

Single-line Text Input Control

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 the <input> tag.

Example

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

HTML

```
<!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>
```

Output:

First name:

Last name:

Password Input Control

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 the HTML <input> tag, but the type attribute is set to password:

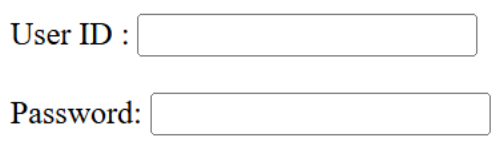
Example

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

HTML

```
<!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>
```

Output:



Multiple-line Text Input Control

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 the HTML <textarea> tag.

Example

The following example demonstrates how to use multi-line text input to take item descriptions:

HTML

```
<!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>
```

Output:

Checkboxes Control

Checkboxes are used when more than one option is required to be selected. They are also created using the <input> tag, but the type attribute is set to checkbox.

Example

In this HTML code, we are creating a form with two checkboxes:

HTML

```
<!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>
```

Output:

Radio Buttons Control

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

Example

In this example code, we are creating a form with two radio buttons:

HTML
```
<!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>
```

Output:

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:

HTML
```
<!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>
```

Output:

File Select Box

If we want to allow a user to upload a file to our website, 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 the 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:

HTML
```
<!DOCTYPE html>
<html>
<head>
  <title>File Upload Box</title>
</head>
<body>
  <form>
    <input type = "file" name = "fileupload" accept = "image/*" />
  </form>
</body>
</html>
```

Output:

Button Control

There are various ways in HTML to create clickable buttons. We can create a clickable button using the <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:

HTML
```
<!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>
```

Output:

Hidden Form Control

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.

Example

The following example shows the usage of hidden control:

HTML

```
<!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>
```

Output: