

Form

The `<form>` tag is used to create a container for collecting and submitting data from user input on a web page. It is used to define a section of a web page that contains input fields, buttons, and other form elements, which allow users to enter and submit data to a web server.

The `<form>` element has several attributes that can be used to specify how the form should behave, such as *action*, *method*, *enctype*, and *target*. Here's a brief overview of these attributes:

- *action*: This attribute specifies the URL where the form data should be submitted when the form is submitted. It can be a relative or absolute URL, or a JavaScript function to handle form submission.
- *method*: This attribute specifies the HTTP method to be used for submitting the form data. It can be either "GET" or "POST". "GET" method appends the form data to the URL as query parameters, while "POST" method sends the form data in the request body.
- *enctype*: This attribute specifies the encoding type for the form data when it is submitted. It is used with the *method* attribute set to "POST". Common values include "application/x-www-form-urlencoded" for URL-encoded form data and "multipart/form-data" for binary or large form data.
- *target*: This attribute specifies where the form response should be displayed after form submission. It can take values like "_self" (default), "_blank", "_parent", or "_top", or a custom target name.

Here's a brief overview of some common tags used in conjunction with the `<form>` tag to create interactive and user-friendly forms on web pages :

1. `<label>`: The `<label>` tag is used to associate a label with an input field in a form. It is typically used to provide a text description for an input field, such as a text input, checkbox, or radio button. The *for* attribute of the `<label>` tag should match the *id* attribute of the associated input field, so that clicking on the label will focus or activate the associated input field.

2. `<select>` and `<option>`: The `<select>` tag is used to create a drop-down list of options in a form. The `<option>` tag is used to define the individual options within the `<select>` tag. Users can select one or more options from the list, depending on the *multiple* attribute on the `<select>` tag. The selected option(s) will be included in the form data when the form is submitted.

3. `<textarea>`: The `<textarea>` tag is used to create a multi-line text input field in a form. It allows users to enter and edit text that can span multiple lines. The text entered in the `<textarea>` field will be included in the form data when the form is submitted.

4. `<button>`: The `<button>` tag is used to create a button in a form. It can be used for various purposes, such as submitting a form, resetting a form, or triggering a JavaScript function. The `type` attribute on the `<button>` tag can be set to "submit", "reset", or "button" to specify the button's behavior.

5. `<fieldset>` and `<legend>`: The `<fieldset>` tag is used to group related form elements together and create a visual grouping effect. The `<legend>` tag is used to provide a caption or title for the `<fieldset>` element, describing the purpose of the grouped form elements. `<fieldset>` and `<legend>` are commonly used to group and organize related form elements.

6. `<datalist>`: The `<datalist>` tag is used to provide a pre-defined list of options for an input field in a form. The `<datalist>` tag is used in conjunction with an `<input>` field of type "text" or "search", and the list of options is defined using `<option>` tags within the `<datalist>` tag. The user can select an option from the list or enter a custom value.

Example :

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Form</title>
</head>
<body>
  <form action="/submituserdata" method="get" target="_blank" autocomplete="off"
id="myform1" enctype="multipart/form-data">
    <input type="text" name="email" placeholder="enter your email">
    <input type="password" name="pwd" placeholder="enter your password">
    <input type="submit" value="login">
```

```
<br><br>
<label>What is your age ?</label>
<select name="usage" size="1">
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="3">3</option>
  <option value="4">4</option>
</select>
<br><br>
<textarea name="message" rows="10" cols="50" style="width:100px;height:100px;max-width:300px;min-width:150px;"></textarea>

<br><br>
<button type="button" onclick="alert('hey there')">Don't Click</button>

<br><br>
<fieldset>
  <legend>Login</legend>
  <input type="text" name="email" placeholder="enter your email" list="myname">
  <datalist id="myname">
    <option value="name1">name1</option>
    <option value="name2">name2</option>
    <option value="name3">name3</option>
  </datalist>
  <input type="password" name="pwd" placeholder="enter your password">
  <input type="submit" value="login" formaction="/someotherpath"
formenctype="multipart/form-data" formmethod="get" formtarget="_blank">
</fieldset>

</form>
<input type="submit" value="login" form="myform1">
</body>
```

</html>

Output :

enter your email enter your password login

What is your age ? 1 ▾

Don't Click

Login

enter your email enter your password login

login

Output

Explanation :

The `<form>` tag defines a form that is used to collect user input. It has several attributes:

- `action="/submituserdata"`: Specifies the URL to which the form data will be submitted.
- `method="get"`: Specifies the HTTP method to be used for submitting the form data. In this case, it is set to "get", which means the form data will be appended to the URL as query parameters.
- `target="_blank"`: Specifies the target window or frame where the form response will be displayed. In this case, it is set to "_blank", which means the form response will open in a new window.
- `autocomplete="off"`: Specifies whether the form should have autocomplete enabled or disabled. In this case, it is set to "off", which means autocomplete is disabled.
- `id="myform1"`: Specifies a unique identifier for the form, which can be used to reference the form in JavaScript or other parts of the HTML document.
- `enctype="multipart/form-data"`: Specifies the encoding type for the form data. In this case, it is set to "multipart/form-data", which is used for uploading files.

Inside the `<form>` tag, there are several form elements:

1. `<input>` elements: These are used to collect user input. There are three `<input>` elements in this form:

- One with `type="text"` and `name="email"`: This is a text input field where users can enter their email address.

- One with *type*="password" and *name*="pwd": This is a password input field where users can enter their password.
- One with *type*="submit" and *value*="login": This is a submit button that users can click to submit the form data.

2. *<select>* element: This is a dropdown list where users can select an option. It has the *name* attribute set to "userage" and the *size* attribute set to "1", which means it will display one option at a time.

3. *<textarea>* element: This is a multi-line text input field where users can enter a longer message. It has the *name* attribute set to "message" and the *rows* and *cols* attributes set to control the size of the textarea.

4. *<button>* element: This is a button that users can click. It has the *type* attribute set to "button" and an *onclick* attribute that specifies a JavaScript function to be executed when the button is clicked. In this case, it displays an alert with the message "hey there".

5. *<fieldset>* element: This is used to group related form elements together. It has a *<legend>* element inside that specifies a caption for the fieldset, which in this case is "Login". Inside the fieldset, there are two *<input>* elements:

- One with *type*="text" and *name*="email": This is a text input field where users can enter their email address. It also has a **list** attribute that references a *<datalist>* element with the id "myname", which provides suggestions for the input field.
- One with *type*="password" and *name*="pwd": This is a password input field where users can enter their password.
- There is also an *<input>* element with *type*="submit", which is a submit button that users can click to submit the form data. It has additional attributes *formaction*="/someotherpath", and *formenctype*="multipart/form-data". These attributes override the form-level action, method, and enctype attributes for this specific submit button. When this submit button is clicked, the form data will be submitted to the URL specified in the *formaction* attribute, using the encoding type specified in the *formenctype* attribute, and the HTTP method specified in the *formmethod* attribute.