

Chapter 18: Forms

Attribute	Description
<code>accept-charset</code>	Specifies the character encodings that are to be used for the form submission.
<code>action</code>	Specifies where to send the form-data when a form is submitted.
<code>autocomplete</code>	Specifies whether a form should have autocomplete on or off.
<code>enctype</code>	Specifies how the form-data should be encoded when submitting it to the server (only for <code>method="post"</code>).
<code>method</code>	Specifies the HTTP method to use when sending form-data (POST or GET).
<code>name</code>	Specifies the name of a form.
<code>novalidate</code>	Specifies that the form should not be validated when submitted.
<code>target</code>	Specifies where to display the response that is received after submitting the form.

In order to group input elements and submit data, HTML uses a form element to encapsulate input and submission elements. These forms handle sending the data in the specified method to a page handled by a server or handler. This topic explains and demonstrates the usage of HTML forms in collecting and submitting input data.

Section 18.1: Submitting

The Action Attribute

The action attribute defines the action to be performed when the form is submitted, which usually leads to a script that collects the information submitted and works with it. if you leave it blank, it will send it to the same file

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

The Method Attribute

The method attribute is used to define the HTTP method of the form which is either GET or POST.

```
<form action="action.php" method="get">
<form action="action.php" method="post">
```

The GET method is mostly used to *get* data, for example to receive a post by its ID or name, or to submit a search query. The GET method will append the form data to the URL specified in the action attribute.

```
www.example.com/action.php?firstname=Mickey&lastname=Mouse
```

The POST method is used when submitting data to a script. The POST method does not append the form data to the action URL but sends using the request body.

To submit the data from the form correctly, a name attribute name must be specified. As an example let's send the value of the field and set its name to *lastname*:

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

More attributes

```
<form action="action.php" method="post" target="_blank" accept-charset="UTF-8"
enctype="application/x-www-form-urlencoded" autocomplete="off" novalidate>
```

```
<!-- form elements -->
```

```
</form>
```

Section 18.2: Target attribute in form tag

The target attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form.

The target attribute defines a name of, or keyword for, a browsing context (e.g. tab, window, or inline frame).

Form Tag with a target attribute:

```
<form target="_blank">
```

Attribute Values

Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the same frame (this is default)
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window
framename	The response is displayed in a named iframe

Note: The target attribute was **deprecated** in **HTML 4.01**. The target attribute is **supported** in **HTML5**.

Frames and framesets are not supported in **HTML5**, so the **_parent, _top and framename values are now mostly used with iframes**.

Section 18.3: Uploading Files

Images and files can be uploaded/submitted to server by setting enctype attribute of form tag to multipart/form-data. enctype specifies how form data would be encoded while submitting to the server.

Example

```
<form method="post" enctype="multipart/form-data" action="upload.php">
  <input type="file" name="pic" />
  <input type="submit" value="Upload" />
</form>
```

Section 18.4: Grouping a few input fields

While designing a form, you might like to group a few input fields into a group to help organise the form layout. This can be done by using the tag . Here is an example for using it.

For each fieldset, you can set a legend for the set using the tag LEGEND TEXT

Example

```
<form>
  <fieldset>
    <legend>1st field set:</legend>
    Field one:<br>
    <input type="text"><br>
```

```
Field two:<br>
  <input type="text"><br>
</fieldset><br>
<fieldset>
  <legend>2nd field set:</legend>
  Field three:<br>
  <input type="text"><br>
  Field four:<br>
  <input type="text"><br>
</fieldset><br>
<input type="submit" value="Submit">
</form>
```

Result

1st field set:

Field one:

Field two:

2nd field set:

Field three:

Field four:

Submit

Browser Support

Chrome, IE, Edge, FireFox, Safari and Opera's latest versions also supports the tag