

HTML Input Attributes

[< Previous](#)[Next >](#)

The value Attribute

The **value** attribute specifies the initial value for an input field:

Example

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John">
</form>
```

[Try it Yourself »](#)

The readonly Attribute

The **readonly** attribute specifies that the input field is read only (cannot be changed):

Example

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" value="John" readonly>
</form>
```

Try it Yourself »

The disabled Attribute

The **disabled** attribute specifies that the input field is disabled.

A disabled input field is unusable and un-clickable, and its value will not be sent when submitting the form:

Example

```
<form action="">  
  First name:<br>  
  <input type="text" name="firstname" value="John" disabled>  
</form>
```

Try it Yourself »

The size Attribute

The **size** attribute specifies the size (in characters) for the input field:

Example

```
<form action="">  
  First name:<br>  
  <input type="text" name="firstname" value="John" size="40">  
</form>
```

Try it Yourself »

The maxlength Attribute

The **maxlength** attribute specifies the maximum allowed length for the input field:

Example

```
<form action="">
  First name:<br>
  <input type="text" name="firstname" maxlength="10">
</form>
```

Try it Yourself »

With a **maxlength** attribute, the input field will not accept more than the allowed number of characters.

The **maxlength** attribute does not provide any feedback. If you want to alert the user, you must write JavaScript code.

Note: Input restrictions are not foolproof, and JavaScript provides many ways to add illegal input. To safely restrict input, it must be checked by the receiver (the server) as well!

HTML5 Attributes

HTML5 added the following attributes for **<input>** :

- autocomplete
- autofocus
- form

- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

and the following attributes for `<form>` :

- autocomplete
- novalidate

The autocomplete Attribute

The `autocomplete` attribute specifies whether a form or input field should have autocomplete on or off.

When autocomplete is on, the browser automatically completes the input values based on values that the user has entered before.

Tip: It is possible to have autocomplete "on" for the form, and "off" for specific input fields, or vice versa.

The `autocomplete` attribute works with `<form>` and the following `<input>` types: text, search, url, tel, email, password, datepickers, range, and color.



Example

An HTML form with autocomplete on (and off for one input field):

```
<form action="/action_page.php" autocomplete="on">
  First name:<input type="text" name="fname"><br>
```

```
Last name: <input type="text" name="lname"><br>
E-mail: <input type="email" name="email" autocomplete="off"><br>
<input type="submit">
</form>
```

Try it Yourself »

Tip: In some browsers you may need to activate the autocomplete function for this to work.

The novalidate Attribute

The `novalidate` attribute is a `<form>` attribute.

When present, novalidate specifies that the form data should not be validated when submitted.



Example

Indicates that the form is not to be validated on submit:

```
<form action="/action_page.php" novalidate>
  E-mail: <input type="email" name="user_email">
  <input type="submit">
</form>
```

Try it Yourself »

The autofocus Attribute

The `autofocus` attribute specifies that the input field should automatically get focus when the page loads.



Example

Let the "First name" input field automatically get focus when the page loads:

```
First name:<input type="text" name="fname" autofocus>
```

[Try it Yourself »](#)

The form Attribute

The **form** attribute specifies one or more forms an **<input>** element belongs to.

Example

An input field located outside the HTML form (but still a part of the form):

```
<form action="/action_page.php" id="form1">
  First name: <input type="text" name="fname"><br>
  <input type="submit" value="Submit">
</form>

Last name: <input type="text" name="lname" form="form1">
```

[Try it Yourself »](#)

The formation Attribute

The **formation** attribute specifies the URL of a file that will process the input control when the form is submitted.

The formation attribute overrides the action attribute of the **<form>** element.

The formation attribute is used with **type="submit"** and **type="image"**.



Example

An HTML form with two submit buttons, with different actions:

```
<form action="/action_page.php">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit"><br>  
  <input type="submit" formaction="/action_page2.php"  
    value="Submit as admin">  
</form>
```

Try it Yourself »

The formenctype Attribute

The **formenctype** attribute specifies how the form data should be encoded when submitted (only for forms with method="post").

The **formenctype** attribute overrides the enctype attribute of the **<form>** element.

The **formenctype** attribute is used with **type="submit"** and **type="image"**.



Example

Send form-data that is default encoded (the first submit button), and encoded as "multipart/form-data" (the second submit button):

```
<form action="/action_page_binary.asp" method="post">  
  First name: <input type="text" name="fname"><br>  
  <input type="submit" value="Submit">  
  <input type="submit" formenctype="multipart/form-data"  
    value="Submit as Multipart/form-data">  
</form>
```

Try it Yourself »

The formmethod Attribute

The `formmethod` attribute defines the HTTP method for sending form-data to the action URL.

The `formmethod` attribute overrides the method attribute of the `<form>` element.

The `formmethod` attribute can be used with `type="submit"` and `type="image"`.



Example

The second submit button overrides the HTTP method of the form:

```
<form action="/action_page.php" method="get">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit">
  <input type="submit" formmethod="post" value="Submit using
POST">
</form>
```

Try it Yourself »

The formnovalidate Attribute

The `formnovalidate` attribute overrides the novalidate attribute of the `<form>` element.

The `formnovalidate` attribute can be used with `type="submit"`.



Example

A form with two submit buttons (with and without validation):

```
<form action="/action_page.php">  
  E-mail: <input type="email" name="userid"><br>  
  <input type="submit" value="Submit"><br>  
  <input type="submit" formnovalidate value="Submit without  
validation">  
</form>
```

Try it Yourself »

The formtarget Attribute

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

The **formtarget** attribute overrides the target attribute of the **<form>** element.

The **formtarget** attribute can be used with **type="submit"** and **type="image"**.



Example

A form with two submit buttons, with different target windows:

```
<form action="/action_page.php">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit as normal">  
  <input type="submit" formtarget="_blank"  
  value="Submit to a new window">  
</form>
```

Try it Yourself »

The height and width Attributes

The `height` and `width` attributes specify the height and width of an `<input type="image">` element.

Always specify the size of images. If the browser does not know the size, the page will flicker while images load.



Example

Define an image as the submit button, with height and width attributes:

```
<input type="image" src="img_submit.gif" alt="Submit" width="48"
height="48">
```

Try it Yourself »

The list Attribute

The `list` attribute refers to a `<datalist>` element that contains pre-defined options for an `<input>` element.



Example

An `<input>` element with pre-defined values in a `<datalist>`:

```
<input list="browsers">

<datalist id="browsers">
```

```
<option value="Internet Explorer">
<option value="Firefox">
<option value="Chrome">
<option value="Opera">
<option value="Safari">
</datalist>
```

Try it Yourself »

The min and max Attributes

The **min** and **max** attributes specify the minimum and maximum values for an **<input>** element.

The **min** and **max** attributes work with the following input types: number, range, date, datetime-local, month, time and week.



Example

<input> elements with min and max values:

Enter a date before 1980-01-01:

```
<input type="date" name="bday" max="1979-12-31">
```

Enter a date after 2000-01-01:

```
<input type="date" name="bday" min="2000-01-02">
```

Quantity (between 1 and 5):

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

Try it Yourself »

The multiple Attribute

The **multiple** attribute specifies that the user is allowed to enter more than one value in the `<input>` element.

The **multiple** attribute works with the following input types: email, and file.



Example

A file upload field that accepts multiple values:

```
Select images: <input type="file" name="img" multiple>
```

Try it Yourself »

The pattern Attribute

The **pattern** attribute specifies a regular expression that the `<input>` element's value is checked against.

The **pattern** attribute works with the following input types: text, search, url, tel, email, and password.

Tip: Use the global [title](#) attribute to describe the pattern to help the user.

Tip: Learn more about [regular expressions](#) in our JavaScript tutorial.



Example

An input field that can contain only three letters (no numbers or special characters):

```
Country code: <input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code">
```

Try it Yourself »

The placeholder Attribute

The **placeholder** attribute specifies a hint that describes the expected value of an input field (a sample value or a short description of the format).

The hint is displayed in the input field before the user enters a value.

The **placeholder** attribute works with the following input types: text, search, url, tel, email, and password.



Example

An input field with a placeholder text:

```
<input type="text" name="fname" placeholder="First name">
```

Try it Yourself »

The required Attribute

The **required** attribute 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

A required input field:

```
Username: <input type="text" name="username" required>
```

Try it Yourself »

The step Attribute

The **step** attribute specifies the legal number intervals for an `<input>` element.

Example: if `step="3"`, legal numbers could be -3, 0, 3, 6, etc.

Tip: The step attribute can be used together with the max and min attributes to create a range of legal values.

The **step** attribute works with the following input types: number, range, date, datetime-local, month, time and week.



Example

An input field with a specified legal number intervals:

```
<input type="number" name="points" step="3">
```

Try it Yourself »

HTML Exercises

Test Yourself With Exercises

Exercise:

In the input field below, add placeholder that says "Your name here".

```
<form action="/action_page.php">  
<input type="text" >  
</form>
```

[Submit Answer »](#)

[Start the Exercise](#)

HTML Form and Input Elements

Tag	Description
<u><form></u>	Defines an HTML form for user input
<u><input></u>	Defines an input control

For a complete list of all available HTML tags, visit our [HTML Tag Reference](#).

[◀ Previous](#)

[Next ▶](#)

[Copyright 1999-2019](#) by Refsnes Data. All Rights Reserved.