HTML Forms

HTML Form Example

Top of Form

First name:  
  
Last name:  


Bottom of Form

The <form> Element

The HTML <form> element defines a form that is used to collect user input:

<form>  
.  
*form elements*  
.  
</form>

An HTML form contains form elements.

Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

The <input> Element

The <input> element is the most important form element.

The <input> element can be displayed in several ways, depending on the type attribute.

Here are some examples:

|  |  |
| --- | --- |
| Type | Description |
| <input type="text"> | Defines a one-line text input field |
| <input type="radio"> | Defines a radio button (for selecting one of many choices) |
| <input type="submit"> | Defines a submit button (for submitting the form) |

You will learn a lot more about input types later in this tutorial.

Text Input

<input type="text"> defines a one-line input field for text input:

Example

<form>  
  First name:<br>  
  <input type="text" name="firstname"><br>  
  Last name:<br>  
  <input type="text" name="lastname">  
</form>

This is how it will look like in a browser:

First name:  
  
Last name:  


Note: The form itself is not visible. Also note that the default width of a text field is 20 characters.

Radio Button Input

<input type="radio"> defines a radio button.

Radio buttons let a user select ONE of a limited number of choices:

Example

<form>  
  <input type="radio" name="gender" value="male" checked> Male<br>  
  <input type="radio" name="gender" value="female"> Female<br>  
  <input type="radio" name="gender" value="other"> Other  
</form>

This is how the HTML code above will be displayed in a browser:

Male  
Female  
Other

The Submit Button

<input type="submit"> defines a button for submitting the form data to a form-handler.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's action attribute:

Example

<form action="/action\_page.php">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br><br>  
  <input type="submit" value="Submit">  
</form>

This is how the HTML code above will be displayed in a browser:

Top of Form

First name:  
  
Last name:  


Bottom of Form

The Action Attribute

The action attribute defines the action to be performed when the form is submitted.

Normally, the form data is sent to a web page on the server when the user clicks on the submit button.

In the example above, the form data is sent to a page on the server called "/action\_page.php". This page contains a server-side script that handles the form data:

<form action="/action\_page.php">

If the action attribute is omitted, the action is set to the current page.

The Target Attribute

The target attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.

The default value is "\_self" which means the form will be submitted in the current window.

To make the form result open in a new browser tab, use the value "\_blank":

Example

<form action="/action\_page.php" target="\_blank">

Other legal values are "\_parent", "\_top", or a name representing the name of an iframe.

The Method Attribute

The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data:

Example

<form action="/action\_page.php" method="get">

or:

Example

<form action="/action\_page.php" method="post">

When to Use GET?

The default method when submitting form data is GET.

However, when GET is used, the submitted form data will be visible in the page address field:

/action\_page.php?firstname=Mickey&lastname=Mouse

Notes on GET:

Appends form-data into the URL in name/value pairs

The length of a URL is limited (about 3000 characters)

Never use GET to send sensitive data! (will be visible in the URL)

Useful for form submissions where a user wants to bookmark the result

GET is better for non-secure data, like query strings in Google

When to Use POST?

Always use POST if the form data contains sensitive or personal information. The POST method does not display the submitted form data in the page address field.

Notes on POST:

POST has no size limitations, and can be used to send large amounts of data.

Form submissions with POST cannot be bookmarked

The Name Attribute

Each input field must have a name attribute to be submitted.

If the name attribute is omitted, the data of that input field will not be sent at all.

This example will only submit the "Last name" input field:

Example

<form action="/action\_page.php">  
  First name:<br>  
  <input type="text" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br><br>  
  <input type="submit" value="Submit">  
</form>

Grouping Form Data with <fieldset>

The <fieldset> element is used to group related data in a form.

The <legend> element defines a caption for the <fieldset> element.

Example

<form action="/action\_page.php">  
  <fieldset>  
    <legend>Personal information:</legend>  
    First name:<br>  
    <input type="text" name="firstname" value="Mickey"><br>  
    Last name:<br>  
    <input type="text" name="lastname" value="Mouse"><br><br>  
    <input type="submit" value="Submit">  
  </fieldset>  
</form>

This is how the HTML code above will be displayed in a browser:

Top of Form

Personal information: First name:  
  
Last name:  


Bottom of Form

Bottom of Form

Here is the list of all <form> attributes:

|  |  |
| --- | --- |
| Attribute | Description |
| accept-charset | Specifies the charset used in the submitted form (default: the page charset). |
| action | Specifies an address (url) where to submit the form (default: the submitting page). |
| autocomplete | Specifies if the browser should autocomplete the form (default: on). |
| enctype | Specifies the encoding of the submitted data (default: is url-encoded). |
| method | Specifies the HTTP method used when submitting the form (default: GET). |
| name | Specifies a name used to identify the form (for DOM usage: document.forms.name). |
| novalidate | Specifies that the browser should not validate the form. |
| target | Specifies the target of the address in the action attribute (default: \_self). |

You will learn more about the form attributes in the next chapters.

**HTML Form Elements**

This chapter describes all HTML form elements.

The <input> Element

The most important form element is the <input> element.

The <input> element can be displayed in several ways, depending on the type attribute.

Example

<input name="firstname" type="text">

If the type attribute is omitted, the input field gets the default type: "text".

All the different input types are covered in the next chapter.

The <select> Element

The <select> element defines a drop-down list:

Example

<select name="cars">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>

The <option> elements defines an option that can be selected.

By default, the first item in the drop-down list is selected.

To define a pre-selected option, add the selected attribute to the option:

Example

<option value="fiat" selected>Fiat</option>

Visible Values:

Use the size attribute to specify the number of visible values:

Example

<select name="cars" size="3">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>

Allow Multiple Selections:

Use the multiple attribute to allow the user to select more than one value:

Example

<select name="cars" size="4" multiple>  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>

The <textarea> Element

The <textarea> element defines a multi-line input field (a text area):

Example

<textarea name="message" rows="10" cols="30">  
The cat was playing in the garden.  
</textarea>

The rows attribute specifies the visible number of lines in a text area.

The cols attribute specifies the visible width of a text area.

This is how the HTML code above will be displayed in a browser:



You can also define the size of the text area by using CSS:

Example

<textarea name="message" style="width:200px; height:600px;">  
The cat was playing in the garden.  
</textarea>

The <button> Element

The <button> element defines a clickable button:

Example

<button type="button" onclick="alert('Hello World!')">Click Me!</button>

This is how the HTML code above will be displayed in a browser:

Note: Always specify the type attribute for the button element. Different browsers may use different default types for the button element.

HTML5 Form Elements

HTML5 added the following form elements:

<datalist>

<output>

Note: Browsers do not display unknown elements. New elements that are not supported in older browsers will not "destroy" your web page.

HTML5 <datalist> Element

The <datalist> element specifies a list of pre-defined options for an <input> element.

Users will see a drop-down list of the pre-defined options as they input data.

The list attribute of the <input> element, must refer to the id attribute of the <datalist> element.

OperaSafariChromeFirefoxInternet Explorer

Example

<form action="/action\_page.php">  
  <input list="browsers">  
  <datalist id="browsers">  
    <option value="Internet Explorer">  
    <option value="Firefox">  
    <option value="Chrome">  
    <option value="Opera">  
    <option value="Safari">  
  </datalist>   
</form>

HTML5 <output> Element

The <output> element represents the result of a calculation (like one performed by a script).

OperaSafariChromeFirefoxInternet Explorer

Example

Perform a calculation and show the result in an <output> element:

<form action="/action\_page.php"  
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">  
  0  
  <input type="range"  id="a" name="a" value="50">  
  100 +  
  <input type="number" id="b" name="b" value="50">  
  =  
  <output name="x" for="a b"></output>  
  <br><br>  
  <input type="submit">  
</form>

HTML Form Elements

|  |  |
| --- | --- |
| Tag | Description |
| [<form>](https://www.w3schools.com/tags/tag_form.asp) | Defines an HTML form for user input |
| [<input>](https://www.w3schools.com/tags/tag_input.asp) | Defines an input control |
| [<textarea>](https://www.w3schools.com/tags/tag_textarea.asp) | Defines a multiline input control (text area) |
| [<label>](https://www.w3schools.com/tags/tag_label.asp) | Defines a label for an <input> element |
| [<fieldset>](https://www.w3schools.com/tags/tag_fieldset.asp) | Groups related elements in a form |
| [<legend>](https://www.w3schools.com/tags/tag_legend.asp) | Defines a caption for a <fieldset> element |
| [<select>](https://www.w3schools.com/tags/tag_select.asp) | Defines a drop-down list |
| [<optgroup>](https://www.w3schools.com/tags/tag_optgroup.asp) | Defines a group of related options in a drop-down list |
| [<option>](https://www.w3schools.com/tags/tag_option.asp) | Defines an option in a drop-down list |
| [<button>](https://www.w3schools.com/tags/tag_button.asp) | Defines a clickable button |
| [<datalist>](https://www.w3schools.com/tags/tag_datalist.asp) | Specifies a list of pre-defined options for input controls |
| [<output>](https://www.w3schools.com/tags/tag_output.asp) | Defines the result of a calculation |

For a complete list of all available HTML tags, visit our [HTML Tag Reference](https://www.w3schools.com/tags/default.asp).

**HTML Input Types**

This chapter describes the different input types for the <input> element.

HTML Input Types

Here are the different input types you can use in HTML:

<input type="button">

<input type="checkbox">

<input type="color">

<input type="date">

<input type="datetime-local">

<input type="email">

<input type="file">

<input type="hidden">

<input type="image">

<input type="month">

<input type="number">

<input type="password">

<input type="radio">

<input type="range">

<input type="reset">

<input type="search">

<input type="submit">

<input type="tel">

<input type="text">

<input type="time">

<input type="url">

<input type="week">

Input Type Text

<input type="text"> defines a one-line text input field:

Example

<form>  
  First name:<br>  
  <input type="text" name="firstname"><br>  
  Last name:<br>  
  <input type="text" name="lastname">  
</form>

This is how the HTML code above will be displayed in a browser:

First name:  
  
Last name:  


Input Type Password

<input type="password"> defines a password field:

Example

<form>  
  User name:<br>  
  <input type="text" name="username"><br>  
  User password:<br>  
  <input type="password" name="psw">  
</form>

This is how the HTML code above will be displayed in a browser:

User name:  
  
User password:  


The characters in a password field are masked (shown as asterisks or circles).

Input Type Submit

<input type="submit"> defines a button for submitting form data to a form-handler.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's action attribute:

Example

<form action="/action\_page.php">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br><br>  
  <input type="submit" value="Submit">  
</form>

This is how the HTML code above will be displayed in a browser:

Top of Form

First name:  
  
Last name:  


Bottom of Form

If you omit the submit button's value attribute, the button will get a default text:

Example

<form action="/action\_page.php">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br><br>  
  <input type="submit">  
</form>

Input Type Reset

<input type="reset"> defines a reset button that will reset all form values to their default values:

Example

<form action="/action\_page.php">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br><br>  
  <input type="submit" value="Submit">  
  <input type="reset">  
</form>

This is how the HTML code above will be displayed in a browser:

Top of Form

First name:  
  
Last name:  


Bottom of Form

If you change the input values and then click the "Reset" button, the form-data will be reset to the default values.

Input Type Radio

<input type="radio"> defines a radio button.

Radio buttons let a user select ONLY ONE of a limited number of choices:

Example

<form>  
  <input type="radio" name="gender" value="male" checked> Male<br>  
  <input type="radio" name="gender" value="female"> Female<br>  
  <input type="radio" name="gender" value="other"> Other  
</form>

This is how the HTML code above will be displayed in a browser:

Male  
Female  
Other

Input Type Checkbox

<input type="checkbox"> defines a checkbox.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example

<form>  
  <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>  
  <input type="checkbox" name="vehicle2" value="Car"> I have a car   
</form>

This is how the HTML code above will be displayed in a browser:

I have a bike   
I have a car

Input Type Button

<input type="button"> defines a button:

Example

<input type="button" onclick="alert('Hello World!')" value="Click Me!">

This is how the HTML code above will be displayed in a browser:

HTML5 Input Types

HTML5 added several new input types:

color

date

datetime-local

email

month

number

range

search

tel

time

url

week

New input types that are not supported by older web browsers, will behave as <input type="text">.

Input Type Color

The <input type="color"> is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  Select your favorite color:  
  <input type="color" name="favcolor">  
</form>

Input Type Date

The <input type="date"> is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  Birthday:  
  <input type="date" name="bday">  
</form>

You can also use the min and max attributes to add restrictions to dates:

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  Enter a date before 1980-01-01:  
  <input type="date" name="bday" max="1979-12-31"><br>  
  Enter a date after 2000-01-01:  
  <input type="date" name="bday" min="2000-01-02"><br>  
</form>

Input Type Datetime-local

The <input type="datetime-local"> specifies a date and time input field, with no time zone.

Depending on browser support, a date picker can show up in the input field.

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  Birthday (date and time):  
  <input type="datetime-local" name="bdaytime">  
</form>

Input Type Email

The <input type="email"> is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and add ".com" to the keyboard to match email input.

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  E-mail:  
  <input type="email" name="email">  
</form>

Input Type File

The <input type="file"> defines a file-select field and a "Browse" button for file uploads.

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  Select a file: <input type="file" name="myFile">  
</form>

Input Type Month

The <input type="month"> allows the user to select a month and year.

Example

<form>  
  Birthday (month and year):  
  <input type="month" name="bdaymonth">  
</form>

Input Type Number

The <input type="number"> defines a numeric input field.

You can also set restrictions on what numbers are accepted.

The following example displays a numeric input field, where you can enter a value from 1 to 5:

Example

<form>  
  Quantity (between 1 and 5):  
  <input type="number" name="quantity" min="1" max="5">  
</form>

Input Restrictions

Here is a list of some common input restrictions:

|  |  |
| --- | --- |
| Attribute | Description |
| disabled | Specifies that an input field should be disabled |
| max | Specifies the maximum value for an input field |
| maxlength | Specifies the maximum number of character for an input field |
| Min | Specifies the minimum value for an input field |
| pattern | Specifies a regular expression to check the input value against |
| readonly | Specifies that an input field is read only (cannot be changed) |
| required | Specifies that an input field is required (must be filled out) |
| Size | Specifies the width (in characters) of an input field |
| Step | Specifies the legal number intervals for an input field |
| value | Specifies the default value for an input field |

You will learn more about input restrictions in the next chapter.

The following example displays a numeric input field, where you can enter a value from 0 to 100, in steps of 10. The default value is 30:

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  Quantity:  
  <input type="number" name="points" min="0" max="100" step="10" value="30">  
</form>

Input Type Range

The <input type="range"> defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min, max, and step attributes:

Example

<form>  
  <input type="range" name="points" min="0" max="10">  
</form>

Input Type Search

The <input type="search"> is used for search fields (a search field behaves like a regular text field).

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  Search Google:  
  <input type="search" name="googlesearch">  
</form>

Input Type Tel

The <input type="tel"> is used for input fields that should contain a telephone number.

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  Telephone:  
  <input type="tel" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">  
</form>

Input Type Time

The <input type="time"> allows the user to select a time (no time zone).

Depending on browser support, a time picker can show up in the input field.

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  Select a time:  
  <input type="time" name="usr\_time">  
</form>

Input Type Url

The <input type="url"> is used for input fields that should contain a URL address.

Depending on browser support, the url field can be automatically validated when submitted.

Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  Add your homepage:  
  <input type="url" name="homepage">  
</form>

Input Type Week

The <input type="week"> allows the user to select a week and year.

Depending on browser support, a date picker can show up in the input field.

OperaSafariChromeFirefoxInternet Explorer

Example

<form>  
  Select a week:  
  <input type="week" name="week\_year">  
</form>

**HTML Input Attributes**

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>

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>

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>

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>

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>

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>

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.

OperaSafariChromeFirefoxInternet Explorer

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>

The autofocus Attribute

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

OperaSafariChromeFirefoxInternet Explorer

Example

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

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

The form Attribute

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

Tip: To refer to more than one form, use a space-separated list of form ids.

OperaSafariChromeFirefoxInternet Explorer

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">

The formaction Attribute

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

The formaction attribute overrides the action attribute of the <form> element.

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

OperaSafariChromeFirefoxInternet Explorer

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>

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".

OperaSafariChromeFirefoxInternet Explorer

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>

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>

The formnovalidate Attribute

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

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

OperaSafariChromeFirefoxInternet Explorer

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>

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".

OperaSafariChromeFirefoxInternet Explorer

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>

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.

OperaSafariChromeFirefoxInternet Explorer

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">

The list Attribute

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

OperaSafariChromeFirefoxInternet Explorer

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>

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.

OperaSafariChromeFirefoxInternet Explorer

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">

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.

OperaSafariChromeFirefoxInternet Explorer

Example

A file upload field that accepts multiple values:

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

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](https://www.w3schools.com/tags/att_global_title.asp) attribute to describe the pattern to help the user.

Tip: Learn more about [regular expressions](https://www.w3schools.com/js/js_regexp.asp) in our JavaScript tutorial.

OperaSafariChromeFirefoxInternet Explorer

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">

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.

OperaSafariChromeFirefoxInternet Explorer

Example

An input field with a placeholder text:

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

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.

OperaSafariChromeFirefoxInternet Explorer

Example

A required input field:

Username: <input type="text" name="usrname" required>

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.

OperaSafariChromeFirefoxInternet Explorer

Example

An input field with a specified legal number intervals:

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