

## HTML5 form attributes

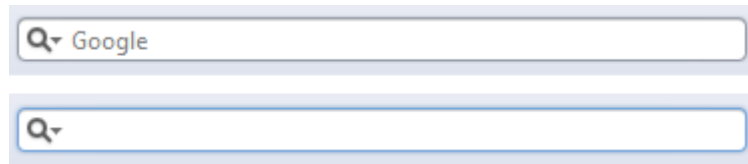
There are 14 new attributes

- placeholder
- autofocus
- autocomplete
- required
- pattern
- list
- multiple
- novalidate
- formnovalidate
- height and width Attributes
- form
- formaction
- formenctype
- formmethod
- formtarget

### placeholder

- 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.

- It's very similar to the search box



Browser search box in Safari without and with focus

```
<input type="text" name="user-name" id="user-name" placeholder="at least 3
characters">
```



Figure 2. Placeholder attribute support in Chrome, unfocused and focused

## autofocus

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

```
<input type="text" name="first-name" id="first-name" autofocus>
```

## autocomplete

- The autocomplete attribute helps users complete forms based on earlier input.

- 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.

Ex 1.

```
<input type="text" name="tracking-code" id="tracking-code"
autocomplete="off">
```

## required

- 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 1

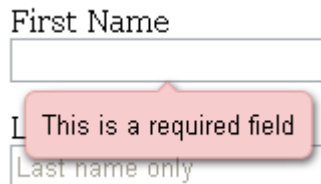
A required input field:

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

Ex 2.

```
<input type="text" id="given-name" name="given-name" required>
```

In browser : The default rendering of “required” error messages depends on the individual browser



First Name

This is a required field

Last name only

## pattern

- 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.

simple example using a product number.

```
<label>Product Number:
```

```
<input pattern="[0-9][A-Z]{3}" name="product" type="text" title="Single digit followed by three uppercase letters."/>
```

```
</label>
```

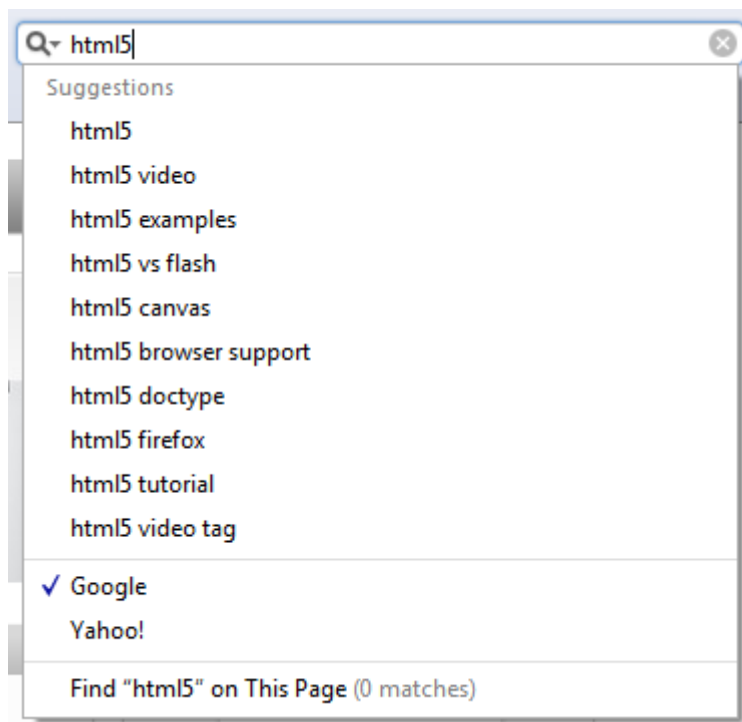
This pattern prescribes that the product number should be a single digit [0-9] followed by three uppercase letters [A-Z]{3}.

Ex 2 : 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">`

### list and the datalist element

- The list attribute enables the user to associate a list of options with a particular field.
- The **list** attribute refers to a <datalist> element that contains pre-defined options for an <input> element.



Ex 1. `<input list="browsers">`

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

Ex 2.

`<label>`Your favorite fruit:

```
<datalist id="fruits">
  <select name="fruits">
    <option value="Blackberry">Blackberry</option>
    <option value="Blackcurrant">Blackcurrant</option>
    <option value="Blueberry">Blueberry</option>
    <!-- ... -->
  </select>
```

If other, please specify:

```
</datalist>
  <input type="text" name="fruit" list="fruits">
</label>
```

Your favourite fruit:

Bla
Blackberry
Blackcurrent
Blueberry

## multiple

- 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.

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

## novalidate and formnovalidate

- The novalidate and formnovalidate attributes indicate that the form shouldn't be validated when submitted.

Ex 1.

- 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>
```

Ex 2.

```
<form action="process.php" novalidate>
  <label for="email">Email:</label>
  <input type="text" name="email" value="gordo@example.com">
  <input type="submit" value="Submit">
</form>
```

## The height and width Attributes

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

### **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.

An input field with a specified legal number intervals:

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

### **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".

### 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>
```

### *formaction*

formaction specifies the file or application that will submit the form. It has the same effect as the action attribute on the form element and can only be used with a submit or image button (type="submit" or type="image"). When the form

is submitted, the browser first checks for a formaction attribute; if that isn't present, it proceeds to look for an action attribute on the form.

```
<input type="submit" value="Submit" formaction="process.php">
```

## 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**:

```
<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:

## 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:

First name:

Last name:

## 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>
```

### Try it Yourself »

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

First name:

Last name:

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!"
```

# HTML5 Input Types

HTML5 added several new input types:

- color
- date
- datetime-local
- email
- month
- number
- range

- search
- tel
- time
- url
- week

## 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.



### 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.

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

You can also add restrictions to dates:

```
<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.

```
<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 adds ".com" to the keyboard to match email input.

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

# Input Type Month

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

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

```
<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:

```
<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 (some are new in HTML5):

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

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:

```
<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:

```
<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).

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

```
<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.

```
<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.

```
<form>  
  Select a week:  
  <input type="week" name="week_year">  
</form>
```