

HTML5 Forms

| | |
|--|---|
| Introduction..... | 1 |
| The Basics of HTML Forms..... | 2 |
| method | 2 |
| action..... | 2 |
| Form tag example..... | 2 |
| Marking up form elements..... | 2 |
| Form example..... | 3 |
| The <code><input></code> element | 3 |
| <code><input></code> element attributes/values..... | 3 |
| <code><input></code> attributes/values introduced in HTML5 | 4 |
| Form examples | 5 |
| A Radio Button Group | 5 |
| A Check Box Group | 5 |
| A File Chooser | 5 |
| A Drop-Down List..... | 5 |
| A Text Area | 6 |
| A Data List..... | 6 |
| A Colour Picker | 6 |
| Date and Time | 6 |
| Range | 6 |
| Email, tel and URL..... | 7 |
| Numbers | 7 |
| Client-side Validation..... | 7 |
| <i>required</i> | 7 |
| <i>maxlength</i> | 7 |
| <i>pattern</i> | 7 |

Introduction

The purpose of this tutorial is to provide a brief guide to creating HTML forms and introduce some of the new features offered by HTML5.

Forms are an important part of web development. They provide the primary means by which the user can interact with a web application on the server. They offer the user boxes in which to type, lists from which to select items, checkboxes to check and buttons to press. They then allow all this user submitted data to be sent to your server-side script so that it can be processed.

The Basics of HTML Forms

An HTML form element is marked up using the `<form>` tag. The `<form>` tag has two important attributes (amongst others). Those attributes are *method* and *action*. Providing your `<form>` tag with an *id* attribute is also useful so that you can easily reference it from other elements, CSS or JavaScript.

method

The *method* attribute allows you to specify the way in which the form data is sent to the server. The value of this attribute specifies the HTTP method by which this is achieved. Usually this is limited to the HTTP methods GET or POST.

action

The *action* attribute specifies where the data will be sent. This can be a relative address (relative to the URL for the original page request) or it may be a URL. If this attribute is left empty or the attribute is omitted the form will generally send its data to the URL for the original page request. However, it is always best to specify the URL to avoid potential unintended results.

Form tag example

This example form uses the GET method. It will send its data to the page “process.htm” on the server as a relative URL.

```
<form name="order" method="get" action="process.htm">
  ...
</form>
```

Marking up form elements

Forms contain a number of controls. Typically these controls will be come in the form of `<input>` elements. Other control elements are `<button>`, `<keygen>`, `<object>`, `<select>` and `<textarea>`.

Other elements that can be associated with a form are `<fieldset>`, `<label>` and `<output>`.

Each part of a form is considered as a paragraph and should be marked up using the `<p>` tag¹.

Controls can also be labelled using the `<label>` tag. The control element will go inside the `<label>` along with any text to be associated with it.

Controls can be grouped together using the `<fieldset>` tag. The title of controls grouped in this way is given by the `<legend>` element which must be the first element within `<fieldset>`.

¹ <http://www.w3.org/TR/html5/forms.html#forms>

Form example

```
<form id="order" method="get" action="process.htm">
  <fieldset>
    <legend> Order Form </legend>
    <p><label>First name: <input type="text" name="fname"></label></p>
    <p><label>Last name: <input type="text" name="lname"></label></p>
    <p><input type="submit" value="Submit"></p>
    <p><input type="reset" value="Reset"></p>
  </fieldset>
</form>
```

The `<input>` element

As you have seen from the examples above that one of the main items on a HTML form is an input element.

`<input>` element attributes/values

| Attribute | Value | Description |
|-----------|--|---|
| checked | checked | Shows if an input element should be preselected when the page loads (for type="checkbox" or type="radio") |
| disabled | disabled | Specifies that an input element should be disabled when the page loads |
| maxlength | number | Specifies the maximum length (in characters) of an input field (for type="text" or type="password") |
| multiple | multiple | Specifies that a user can enter more than one value in an <code><input></code> element |
| name | name | Specifies a name for an input element |
| size | number | Specifies the width of an input field |
| type | checkbox file hidden password radio reset submit text | Specifies the type of an input element |

| | | |
|-------|-------|---|
| value | value | Specifies the value of an input element |
|-------|-------|---|

<input> attributes/values introduced in HTML5

| Attribute | Value | Description |
|--------------|--|--|
| autocomplete | on off | Specifies whether an <input> element should have autocomplete enabled |
| autofocus | autofocus | Specifies that an <input> element should automatically get focus when the page loads |
| max | number date | Specifies the maximum value for an <input> element |
| min | number date | Specifies the minimum value for an <input> element |
| pattern | regexp | Specifies a regular expression that an <input> element's value is checked against |
| placeholder | text | Specifies a short hint that describes the expected value of an <input> element |
| required | required | Specifies that an input field must be filled out before submitting the form |
| type | color date datetime datetime-local email month number range tel time url week | Specifies the type of an input element |

Form examples

You should try out the examples below in a browser if you have not encountered them before.

A Radio Button Group

```
<fieldset>
  <legend>Satisfaction</legend>

  <p><label><input type="radio" name="satisfaction" value="sat2">
Satisfied</label></p>

  <p><label><input type="radio" name="satisfaction" value="sat1"> Not very
satisfied</label></p>

  <p><label><input type="radio" name="satisfaction" value="sat0">
Disappointed</label></p>

</fieldset>
```

A Check Box Group

```
<fieldset>
  <legend>Pick your colour</legend>

  <p><label><input type="checkbox" name="colour" value="red">
red</label></p>

  <p><label><input type="checkbox" name="colour" value="green">
green</label></p>

  <p><label><input type="checkbox" name="colour" value="blue">
blue</label></p>

</fieldset>
```

In order to have the checkbox values available to PHP you would have to append square brackets `[]` to the value of the *name* attribute. In the code above this would be *name="colour[]"* for all three of the checkboxes.

A File Chooser

```
<p><label>Choose file: <input type="file" name="file"></label></p>
```

A Drop-Down List

```
<p>
  <label>Choose your town:
    <select name="town">
      <option value="">Please Select ...</option>
      <option value="Swindon">Swindon</option>
      <option value="London">London</option>
      <option value="Stafford">Stafford</option>
    </select>
  </label>
```

```
</p>
```

A Text Area

```
<p><label>Message: <textarea name="message"></textarea></label></p>
```

A Data List

You use a `<datalist>` to provide a user with 'suggested input' as they type. In the example code below a textbox `<input type="text">` is associated with a `<datalist>`. This is accomplished by the `list` attribute of the textbox referencing the `id` attribute of the `<datalist>`.

```
<p><label>Title: <input type="text" name="title"
list="salutation"></label></p>

<datalist id="salutation">
  <option value="Mr">
  <option value="Ms">
  <option value="Mad Professor">
</datalist>
```

A Colour Picker

```
<p><label>Colour picker: <input type="color" name="colour"></label></p>
```

Date and Time

```
<fieldset>
  <legend>Time</legend>

  <p><label>Time: <input type="time" name="delivery"></label></p>

  <p><label>Date and time: <input type="datetime" name="t1"></label></p>

  <p><label>Date and time (local): <input type="datetime-local"
name="t2"></label></p>

  <p><label>Month: <input type="month" name="orderTimeMonth"></label></p>

  <p><label>Week: <input type="week" name="orderTimeWeek"></label></p>
</fieldset>
```

Range

```
<p><label><input type="range" min="0" max="30" step="10" value="0"
name="points"></label></p>
```

or for a slightly clearer display

```
<p><label><input type="range" min="0" max="30" step="10" value="0"
name="points" list="powers"></label></p>

<datalist id="powers">
  <option value="0">
  <option value="10">
```

```
<option value="20">
<option value="30">
</datalist>
```

Email, tel and URL

```
<fieldset>
  <legend>Email, telephone and URL</legend>
  <p><label>Email: <input type="email" name="email"></label></p>
  <p><label>Telephone: <input type="tel" name="tel"></label></p>
  <p><label>URL: <input type="url" name="url"></label></p>
</fieldset>
```

Numbers

```
<p><label>Age: <input type="number" min="18" max="21"
name="age"></label></p>
```

Client-side Validation

Client-side validation can be useful to improve the users experience; if they make a mistake then feedback can be immediate rather than waiting for a server response. It can also cut down on server traffic if a form only needs to be processed once.

Client-side validation cannot replace server-side validation. Server-side validation must always be performed.

required

The *required* attribute can be specified on *<input>* elements. It checks for a value in the field.

```
<p><label>First name: <input type="text" required name="fname"
></label></p>
```

maxlength

The *maxlength* attribute limits the number of characters that a value can have.

```
<p><label>First name: <input type="text" required maxlength="10"
name="fname"></label></p>
```

pattern

The *pattern* attribute allows the use of regular expressions to pattern match an input. If the input does not conform to the pattern it will show a validation error.

In the code below the input expects a three letter country code; for example "GBR".

```
<p><label>Country code (3 letters): <input type="text" name="country_code"
pattern="[A-Za-z]{3}"></label></p>
```

For more details on regular expressions please see

http://gnosis.cx/publish/programming/regular_expressions.html

and

<http://html5pattern.com/>.