

HTML forms

Forms are used very often when the user needs to provide information to the web server:

- Entering keywords in a search box
- Placing an order
- Subscribing to a mailing list
- Posting a comment
- Filling out a survey
- etc.

An HTML `<form>` element (block-level) contains and organizes **form controls** such as text boxes, check boxes, and buttons that can accept information from web site visitors.

Useful forms require two components:

1. the user interface, that is, the web page containing the form and its controls
2. the server-side processing

Form control elements

- The <form> tag contains the form elements on a web page
- The <input> tag configures a variety of form controls including text boxes, radio buttons, check boxes, and buttons
- The <textarea> tag configures a scrolling text box
- The <select> tag configures a select box (drop down list)
- The <option> tag configures an option in the select box

Example:

```
<form action="what is this?">  
  Email: <input type="text"  
          name="CustEmail"  
          id="CustEmail" >  
  
  <br>  
  <input type="submit" value="Submit">  
</form>
```

Email:

Submit

The <form> tag attributes

- **action:** specifies the server-side program or script that will process your form data; in other words, this attribute specifies *where* to send the form data when the form is submitted
- **method:** specifies *how* to send the form data
 - “get”: (default value) form data passed in URL
 - “post”: form data passed in HTTP Entity Body (*a little bit* more secure)
- **name:** provides a way to reference the form in a script
- **id:** uniquely identifies the form

Now, we'll consider in turn several form elements.

First, we'll look at the <input> element, which accepts input from the user in a variety of ways, depending on the value of its `type` attribute. For example, an input element can be a text box, a checkbox, a password box, a radio button, a button, and more.

Text box: <input> element with type “text”

Accepts textual input from the user.

Attributes:

- type="text"
- name: Only form elements with a “name” attribute will have their values sent to the server when the form is submitted
- id
- size: specifies the width (in characters) of the text box shown on the screen
- maxlength: specifies the maximum length (in characters) of the string that the user is allowed to type in, which could be more or less than the size of the text box
- value: defines the initial (default) value of the input box

Example:

```
<form action="form_action.asp" method="get">  
  First name: <input type="text" name="fname"><br>  
  Last name: <input type="text" name="lname"><br>  
  <input type="submit" value="Submit">  
</form>
```

First name:

Last name:

Password box: <input> element with type “password”

Accepts textual input that needs to be hidden as it is entered.

Attributes:

- type="password"
- name
- id
- size: specifies the width (in characters) of the text box shown on the screen
- maxlength: specifies the maximum length (in characters) of the string that the user is allowed to type in, which could be more or less than the size of the text box
- value: defines the initial (default) value of the password box

Example:

```
<form action="form_action.asp" method="get">  
  Password:  
  <input type="password" name="pwd" size="20">  
</form>  
<p>  
  Notice that the browser displays asterisks or bullets  
  instead of characters in a password box.  
</p>
```

Password:

Notice that the browser displays asterisks or bullets instead of characters in a password box.

Check box: <input> element with type “checkbox”

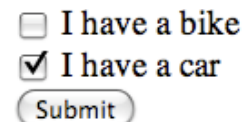
Allows the user to select one or more of a predetermined list of items.

Attributes:

- type="checkbox"
- name
- id
- value: defines the value sent to the server if the box is checked
- checked: specifies that the input element should be preselected when the page loads

Example:

```
<form action="form_action.asp" method="get">  
  <input type="checkbox" name="vehicle1" value="Bike">  
  I have a bike<br>  
  <input type="checkbox" name="vehicle2" value="Car"  
    checked="checked">  
  I have a car <br>  
  <input type="submit" value="Submit">  
</form>
```



Click on the submit button, and the input will be sent to a page on the server called "form_action.asp".

<p>Click on the submit button, and the input will be sent to a page on the server called "form_action.asp".</p>

Radio button: <input> element with type “radio”

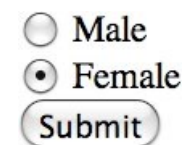
Allows the user to select exactly one from a predetermined list of items.

Attributes:

- type="radio"
- name: must be the same for all the radio buttons in the group
- id
- value: defines the value sent to the server if the radio button is selected
- checked: specifies that the input element should be preselected when the page loads

Example:

```
<form action="form_action.asp" method="get">  
  <input type="radio" name="sex" value="Male"> Male<br>  
  <input type="radio" name="sex" value="Female"  
    checked="checked">  
  Female<br>  
  <input type="submit" value="Submit">  
</form>
```



Submit/reset buttons: <input> element with type “submit/reset”

The submit button sends the form data (that is, the “name=value” pair for each form element) to the web server: when clicked, it triggers the “action” method set in the <form> tag.

Attributes:

- type="submit"
- name
- id
- value: defines the text on the button

The reset button resets the form fields to their initial values.

Attributes:

- type="reset"
- name
- id
- value: defines the text on the button

Button: <input> element with type “button”

Creates a generic button without a default action when the button is clicked. Usually a JavaScript function is defined and invoked when the button is clicked.

Attributes:

- type="button"
- name
- id
- value: defines the text on the button

Scrollable text area: <textarea> element

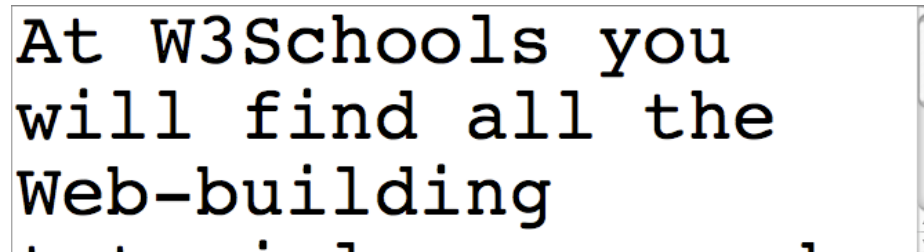
Creates a multi-line text input control in which the user can write an unlimited number of characters.

Attributes:

- name
- id
- cols: specifies the visible width of a text area (better done with CSS)
- rows: specifies the visible height of a text area (better done with CSS)

Example:

```
<textarea rows="3" cols="20"> At W3Schools you will find all the Web-building tutorials you need, from HTML to CSS and JavaScript. </textarea>
```



Drop-down list: <select> element

Creates, together with the <option> tag, a select list (AKA drop-down list/box, select box, option box)

Attributes:

- name
- id
- size: specifies the number of visible options in the drop-down list
- multiple: specifies that multiple options can be selected

The <option> element has attributes "selected" and "value"

Example:

```
<form action="form_action.asp" method="get">  
  <select name="car">  
    <option>Volvo</option>  
    <option>Saab</option>  
    <option>Mercedes</option>  
    <option>Audi</option>  
  </select>  
  <input type="submit" value="Submit">  
</form>
```



Hidden field: <input> element with type “hidden”

Defines a hidden field which is not visible by the user but can store a default value or have its value changed via JavaScript code.

Attributes:

- type="hidden"
- name
- id
- value: defines the value sent to the server

Dave's Pizza - Order Form

Phone number:

Pizza size:

Toppings:

- ☐ Sausage
- ☐ Onion
- ☐ Green pepper

☒ **Delivery** ☐ **Carry out**

© 2014

Dave's Pizza - Order Form

Phone number:

Pizza size:

Toppings:

- ☐ Sausage
- ☐ Onion
- ☐ Green pepper

☒ **Delivery** ☐ **Carry out**

Reset

Place your order

© 2014

MONDAY

Time	Dhara	Dhruv
8-9	IOT F304	Robotics
9-10	Software Eng E201	Software Eng E204
10-11	System Software & Compiler Design E201	-
11-12	-	Computer Network E201
12-1	Distributed Systems E202	Distributed Systems E202
1-2	Lunch	Lunch
2-3	Intro to Web Technology E305	-
3-4	Intro to Web Technology E305	-
4-5	OE E001	-
5-6	OE E001	-

TUESDAY

Time	Dhara	Dhruv
8-9	IOT F304	Robotics
9-10	Software Eng E202	Software Eng E201
10-11	-	System Software & Compiler Design E201
11-12	-	Computer Network E201
12-1	Computer Network E202	System Software & Compiler Design E201
1-2	Lunch	Lunch
2-3	System Software & Compiler Design E201	Computer Network lab E213
3-4	Info security E201	Computer Network lab E213

4-5	OE E001	-
5-6	-	-

WEDNESDAY

Time	Dhara	Dhruv
8-9	IOT F304	Robotics
9-10	-	-
10-11	-	Intro to Web tech E201
11-12	-	Info security E302
12-1	Distributed Systems E202	Distributed Systems E202
1-2	Lunch	Lunch
2-3	Comp Network Lab E302	Intro to Web tech labE215
3-4	Comp Network Lab E302	Intro to Web tech lab E215
4-5	-	-
5-6	-	-

THURSDAY

Time	Dhara	Dhruv
8-9	IOT F204	Robotics
9-10	Computer Network E201	Intro to web techE302
10-11	-	-
11-12	Intro to Web tech E201	Info security E302
12-1	Distributed Systems E202	Distributed Systems E202
1-2	Lunch	Lunch
2-3	Computer Network E202	System Software & Compiler Design E304

3-4	System Software & Compiler Design E202	System Software & Compiler Design E304
4-5	Distributed Sys Lab E 304	-
5-6	Distributed Sys Lab E 304	-

FRIDAY

Time	Dhara	Dhruv
8-9	IOT F204	Robotics
9-10	Software Engineering E202	Software Engineering E201
10-11	Information Security E202	System Software & Compiler Design E201
11-12	Information Security lab E216	Info security lab E303
12-1	Information Security lab E216	Info security lab E303
1-2	Lunch	Lunch
2-3	System Software & Compiler Design - Lab E213	-
3-4	System Software & Compiler Design - Lab E213	-
4-5	-	-
5-6	-	-