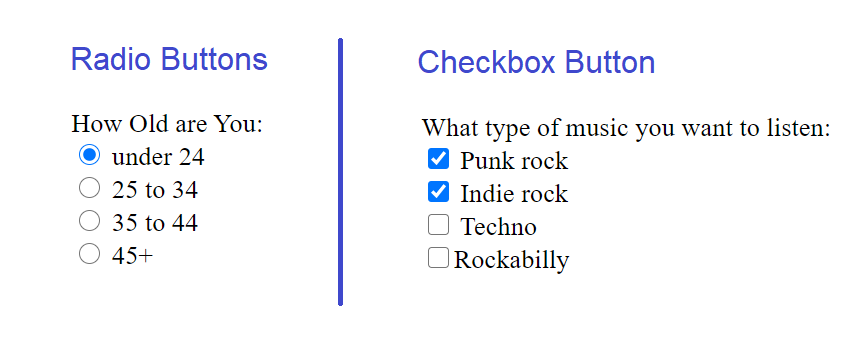
**Working with Radio and Checkbox Buttons**

Both checkbox and radio buttons make it simple for your visitors to choose from a number of provided options. They are similar in that they function like little on/off switches that can be toggled by the user and are added using the **input**element. They serve distinct functions, however. A form control made up of a collection of radio buttons is appropriate when only one option from the group is permitted, or, in other words, when the selections are mutually exclusive (such as Yes or No, or Male or Female).

When one radio button is “on,” all of the others must be “off,” sort of the way buttons used to work on old radios—press one button in and the rest pop out.

When checkboxes are grouped together, however, it is possible to select as many or as few from the group as desired. This makes them the right choice for lists in which more than one selection is okay.



**Radio buttons**

Radio buttons are added to a form with the **input**element with the **type**attribute set to **radio**. The **name**attribute is required. Here is the syntax for a minimal radio button: In this example, radio buttons are used as an interface for users to enter their age group (a person can’t belong to more than one age group, so radio buttons are the right choice). Figure 4 shows how radio buttons are rendered in the browser.

  <form>

      <label for="radio">How Old are You:</label><br />

      <input type="radio"name="age" id="radio" value="under24" checked="checked"/>

     under 24

      <input type="radio" name="age" value="25-34" /> 25 to 34

      <input type="radio" name="age" value="35-44" /> 35 to 44

      <input type="radio" name="age" value="over45" /> 45+

</form>

Notice that all of the **input**elements have the same variable name (“age”), but their values are different. Because these are radio buttons, only one button can be checked at a time, and therefore, only one value will be sent to the server for processing when the form is submitted.

You can decide which button is checked when the form loads by adding the **checked**attribute to the **input**element. In this example, the button next to “under 24” will be checked by default.

**Checkbox buttons**

Checkboxes are added using the **input**element with its **type**set to **checkbox**. As with radio buttons, you create groups of checkboxes by assigning them the same **name**value. The difference, as we’ve already noted, is that more than one checkbox may be checked at a time. The value of every checked button will be sent to the server when the form is submitted. Here is an example of a group of checkbox buttons used to indicate musical interests. Figure shows above how they look in the browser:

<form>

      <label for="radio">What type of music you want to listen:</label><br />

      <input type="checkbox" name="genre" value="punk" checked="checked" /> Punk

      rock

      <input type="checkbox" name="genre" value="indie" checked="checked" />

      Indie rock

      <input type="checkbox" name="genre" value="techno" />Techno

      <input type="checkbox" name="genre" value="rockabilly" />Rockabilly <br />

</form>

**Menus**

Option for providing a list of choices is to put them in a pull-down or scrolling menu. Menus tend to be more compact than groups of buttons and checkboxes.

You  add  both  pull-down  and  scrolling  menus  to  a  form  with  the  select element. Whether the menu pulls down or scrolls is the result of how you specify its size and whether you allow more than one option to be selected. Let’s take a look at both menu types.

**Pull-down menus**

The select element displays as a pull-down menu and it’s a container for a number of option elements. The content of the chosen option element is what gets passed to the web application when the form is submitted. In pull-down menus, only one item may be selected. Here’s an example

<label for="form-fave">What is your favourite 80s Band?</label>

  <select name="EightiesFave" id="form-fave">

        <option>The Cure</option>

        <option>Cocteau Twins</option>

        <option>Tears for Fears</option>

        <option>Thompson Twins</option>

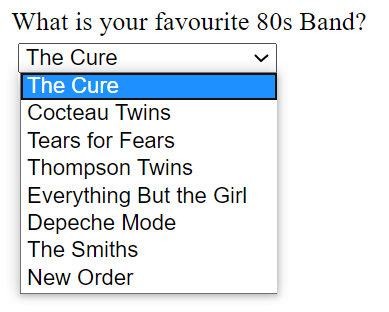
        <option>Everything But the Girl</option>

        <option>Depeche Mode</option>

        <option>The Smiths</option>

        <option>New Order</option>

  </select>



**Text Box controls**

There are three basic types of text box fields in web forms: single-line text boxes, password boxes, and multiline text entry boxes.

* **Single-line text box:**One of the simplest types of form control is the text box field used for entering a single word or line of text. It is added to the form using the **input**element with its **type**attribute set to **text**, and the **name**attribute is required for identifying the variable name as shown here in the Figure.
* **Password text box:**A password box field works just like a text box field, except the characters are obscured from view using asterisk (\*) or bullet (•) characters, or another character determined by the browser.
* **Multiline text entry box:**When you want your users to be able enter more than just one line of text, for these instances, use the **textarea**element that is replaced by a multi-line, scrollable text entry box when displayed by the browser. Unlike the empty **input**element, the **textarea**element has content between its opening and closing tags. The content of the **textarea**element is the initial content of the text box when the form is displayed in the browser.



<form action="index.php" method="post">

      <label for="city">City</label>

      <input type="text" name="City\_name" id="city" /><br /><br /><br />

      <label for="login">Login</label>

      <input type="password" name="login" id="login" /><br /><br /><br />

      <label for="Official">Official Contest Entery</label><br />

      <textarea name="contest\_entry" id="Official" rows="5" cols="100">

      Tell us why you love the band

      in 50 words or less.

      Five winners will

      get backstage passes!

      </textarea>

    </form>