



MARK AS COMPLETE

VIEW CALENDAR 

Select and Multi-select

Input Element in HTML Forms

An HTML form collects information from elements. You will specify an addition **type** attribute to indicate which field to display. Various fields can be created such as a text field, checkbox, password field, or radio button.

Text Field

A text field is a one-line input field that allows the user to input a line of text. Text Fields are created by specifying the type attribute value as "text".

The below example will display a text field with the label *Email Id*:

Example

```
<form>
  <label for="email-input">Email Id:</label><br>
  <input type="text" name="email-input" />
</form>
```

Password Field

Password fields are a type of text field in which the text entered is masked using asterisk or dots. This prevents others from viewing the screen to see what is typed in. Also, its created by specifying the type attribute value as "password".

Example

```
<form>
  <label for="user-password">Password:</label><br>
  <input type="password" name="user-password" />
</form>
```

NOTE: Although a password field is masked, it is NOT encrypted. You will have to use other measures such as HTTPS to encrypt data once the HTML form is submitted.

Radio Buttons

Radio Buttons are used to let the user select exactly one value from a list of predefined options. It is created by specifying the type attribute value as "radio".

Example

```
<form>
  SELECT GENDER
  <br>
  <input type="radio" name="gender" id="male">
  <label for="male">Male</label><br>
  <input type="radio" name="gender" id="female">
  <label for="female">Female</label>
</form>
```

NOTE: A form may have multiple sets of radio buttons. In order to make sure the user only selects one option from a given set, each radio ☐ element must have matching **name** attributes. In the example above, both buttons have a **name** attribute value as **gender**.

Checkboxes

Checkboxes are used to let the user select one or more values from a pre-defined set of options. The type attribute value for checkboxes input control is "checkbox".

Example

```

<form>
  <b>SELECT SUBJECTS</b>
  <br>
  <input type="checkbox" name="subject" id="math" />
  <label for="math">Math</label>
  <input type="checkbox" name="subject" id="science" />
  <label for="science">Science</label>
  <input type="checkbox" name="subject" id="english" />
  <label for="english">English</label>
</form>

```

NOTE: Just like radio buttons, a form may have multiple sets of checkboxes. In order to make sure the user selects options related to a given set, each checkbox ☐ element must have matching `name` attributes. In the example above, each checkbox has a `name` attribute value as **subject**.

File select boxes

File select boxes are used to allow the user to select a local file on their computer and send it as an attachment to the webserver. It is similar to a text box with a button that allows the user to browse for a file. Instead of browsing for the file, the path and the name of the file can also be written. They are created by specifying a type attribute value as "file".

Example

```

<form>
  <label for="fileselect">Upload:</label>
  <input type="file" />
</form>

```

Text area

A text area is a multi-line text input control which allows users to provide a paragraph or multiple lines of text. It is created by using the "textarea" element. This is one of the few input controls that DO NOT use the `<input>` element.

You can control the size of a text area by adding attributes "rows" and "cols" to specify the number of rows and columns of text it supports. Most often text area elements are resizable, but the default size is managed by those two attributes.

Example

```

<form>
  <label for="Description">Description:</label>
  <textarea rows="5" cols="50" name="Description"
            id="Description"></textarea>
</form>

```

Select Boxes (Drop-downs)

Select boxes are used to allow users to select one or more options from a drop-down list. Select boxes are created by using two elements: `<select>` and `<option>`. The `<select>` element defines a drop-down while list items are defined within the select element using `<option>` elements.

Example

```

<form>
  <label for="country">Country:</label>
  <select name="country" id="country">
    <option value="United States">United States</option>
    <option value="Canada">Canada</option>
    <option value="Mexico">Mexico</option>
  </select>
</form>

```

Reset And Submit Buttons

The **submit** button allows the user to send the form data to the web server. You can define a submit button by specifying the type attribute as "submit".

The **reset** button is used to reset the form data and will display any default values. You can define a reset button by specifying the type attribute as "reset".

Example

```

<form action="test.php" method="post" id="users">
  <label for="username">Username:</label>
  <input type="text" name="username" id="Username" />
  <input type="submit" value="Submit" />
  <input type="reset" value="Reset" />
</form>

```

[← Previous](#)
Input Elements ...

[Next →](#)
Submitting Forms