<form method="value-goes-here" action="url-goes-here">

<!-- inputs go inside here -->

</form>

The action attribute is used to specify where the form data should be sent when the form is submitted.

The method attribute is used to specify the HTTP method to use when sending the form data. The most common methods are GET and POST

Forms consist of inputs where users can input their data. You can group related inputs together using the fieldset element. When working with fieldset elements, it is common to use a caption to describe the group of inputs. You can use the legend element for this.

Here is an example of using a fieldset element:

<form action="/example-url">

<fieldset>

<legend>Personal Information</legend>

<!-- inputs go inside here-->

</fieldset>

</form>

<label for="name">Name:</label>

<input type="text" id="name" name="name">

The for attribute on the label element should match the id attribute on the input element. This is known as an explicit association.

The name attribute is used to identify form data after it has been submitted to the server.

Here is an example of how to use the name attribute:

<input type="email" name="email">

input elements can have a size attribute. This attribute defines the number of characters that should be visible as the user types into the input. The value of size should be a non-negative integer greater than zero. If size is not specified, or is specified with an invalid value, the input will have the default width set by the browser.

<label for="lastName">Last Name:</label>

<input id="lastName" name="lastName" type="text" size="10”>

The number input is used to create a numeric input field.

Here is an example of a number input field:

<input type="number" id="age" name="age" min="18" max="100">

The min and max attributes are used to set the minimum and maximum values that can be entered in the input field.

If you want users to select one option from a list of options, you can use a set of radio buttons.

Here is an example of two radio buttons:

<input type="radio" id="yes" name="first-time">

<label for="yes">Yes</label>

<input type="radio" id="no" name="first-time">

<label for="no">No</label>

In this example, the radio buttons are grouped together by using the same name attribute value. This means that only one radio button can be selected at a time.

When you want a user to select multiple options from a list, you can use checkboxes.

Here is an example of how to work with checkboxes dealing with food options:

<fieldset>

<legend>Food Options</legend>

<label for="pizza">Pizza</label>

<input type="checkbox" id="pizza" name="food" value="pizza">

<label for="burger">Burger</label>

<input type="checkbox" id="burger" name="food" value="burger">

</fieldset>

The value attribute is used to specify the value that will be sent to the server when the form is submitted.

When you want users to make selections from a dropdown menu, you can use the select and option elements.

Here is an example of using the select and option elements to create a dropdown for different cities:

<label for="city">Choose a City: </label>

<select id="city" name="city">

<option value="new-york">New York</option>

<option value="los-angeles">Los Angeles</option>

<option value="chicago">Chicago</option>

<option value="miami">Miami</option>

</select>

To make an option selected by default, you can add the selected attribute to the option element you want to be selected.

Here is an example using the selected attribute:

<option selected value="amazing">Amazing</option>

If you want users to have more space to write their comments, you can use a textarea element.

The textarea element is a multi-line text input control that allows users to enter text that is longer than a single line. It can be used to create a comment box, a message input, or other text input that requires multiple lines.

The rows attribute is used to specify the visible height of the textarea, and the cols attribute is used to specify the visible width of the textarea.