

Form Element:

- The HTML <form> element defines a form that is used to collect user input.
- General syntax:

```
<form>
... form elements ...
</form>
```
- An HTML form contains form elements.
- Form elements are different types of input elements, like: text fields, checkboxes, radio buttons, submit buttons, and more.

Action Attribute:

- The action attribute defines the action to be performed when the form is submitted.
- Usually, the form data is sent to a page on the server when the user clicks on the submit button.
- For Flask, you should set the action attribute to be the same as app.route. For example, if you have

```
@app.route('/')
def index:
    return render_template("index.html")
```

Then, in your index.html file, you need to set action="/" in the form tag.

I.e.

```
<form action="/">
```

```
...
```

```
</form>
```

Method Attribute:

- The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data.
 - **Notes on GET:**
 - In the GET method, after the submission of the form, the form values will be visible in the address bar of the new browser tab.
I.e. It appends form-data into the URL in name/value pairs.
 - It has a limited size of 2048 characters.
 - It is only useful for non-secure data not for sensitive information. Never use GET to send sensitive data as it will be visible in the URL. GET is better for non-secure data, like query strings in Google.
 - It is useful for form submissions where a user wants to bookmark the result.
 - The default HTTP method when submitting form data is GET.
I.e. If you do not specify which HTTP method you want to use, it will be GET.
 - **Notes on POST:**
 - In the post method, after the submission of the form, the form values will not be visible in the address bar of the new browser tab.
 - It appends form data inside the body of the HTTP request.
 - It has no size limitation.
 - This method does not support bookmark the result.
- General Syntax:

```
<form method="GET|POST">
```

...
 </form>

- **Note:** In Flask, the default HTTP method is also GET. If you want to use POST, you must also specify it in your app.route() and in your function.

Label Element:

- The <label> tag defines a label for many form elements.
 I.e. The <label> element represents a caption for an item in a user interface.
- The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user is focused on the input element.
- The <label> element also helps users who have difficulty clicking on very small regions, such as radio buttons or checkboxes, because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.
- To associate the <label> with an <input> element, you need to give the <input> an id attribute. The <label> then needs a for attribute whose value is the same as the input's id. Alternatively, you can nest the <input> directly inside the <label>, in which case the for and id attributes are not needed because the association is implicit.
- General syntax:
 <label> ... </label>
- E.g. You can use the label tag like such:
 1.
 <label for="FirstName"> Enter your first name: </label>
 <input type="text" id="FirstName" name="first name">
 2.
 <label> Enter your first name:
 <input type="text" name="first name">
 </label>

Name Attribute:

- Each input field, explained below, must have a name attribute to be submitted.
- If the name attribute is omitted, the data of that input field will not be sent at all.

Input Element:

- **Text:**
 - General syntax:
 <input type="text">
Note: There is no closing tag for input.
 - It defines a single-line text input field.

- E.g. Consider the code below:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title> Text </title>
5 </head>
6 <body>
7   <h1> This shows how to create and use a text box. </h1>
8
9   <!-- Uses the HTTP GET request as no specific request was stated. -->
10  <form action="/">
11
12    <!-- The value of the for attribute in label must match the id attribute attribute of input. -->
13    <label for="text"> Enter something below: </label> <br>
14    <input type="text" id="text" name="text" placeholder="Enter something"> <br>
15
16    <!-- Creates a submit button. -->
17    <input type="submit" value="submit">
18  </form>
19 </body>
20 </html>

```

This is what the HTML page looks like:

The screenshot shows a web page with a large heading "This shows how to create and use a text box." Below the heading is a form. The form contains a label "Enter something below:" followed by a text input field with the placeholder text "Enter something". Below the input field is a submit button labeled "submit".

- Radio:

- General syntax:

<input type="radio">

- It defines a radio button. Radio buttons let a user select ONLY ONE of a limited number of choices.
- E.g. Consider the code below:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title> Radio </title>
5 </head>
6 <body>
7   <h1> This shows how to create and use the input type radio. </h1>
8   <!-- Uses the HTTP GET request as no specific request was stated. -->
9   <form action="/">
10
11     <!-- The value of the for attribute in label must match the id attribute attribute of input. -->
12     <input type="radio" id="male" name="gender" value="male">
13     <label for="male">Male</label><br>
14
15     <input type="radio" id="female" name="gender" value="female">
16     <label for="female">Female</label><br>
17
18     <input type="radio" id="other" name="gender" value="other">
19     <label for="other">Other</label> <br> <br>
20
21     <!-- Creates a submit button. -->
22     <input type="submit" value="submit">
23   </form>
24 </body>
25 </html>

```

This is what the HTML page looks like:

This shows how to create and use the input type radio.

☐ Male
☐ Female
☐ Other

- **Number:**

- General syntax:
`<input type="number">`
- It defines a numeric input field.
- E.g. Consider the code below:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title> Number </title>
5  </head>
6  <body>
7    <h1> This shows how to create and use the input type number. </h1>
8    <!-- Uses the HTTP GET request as no specific request was stated. -->
9    <form action="/">
10
11      <!-- The value of the for attribute in label must match the id attribute attribute of input. -->
12      <label for="quantity">Choose a quantity below:</label><br>
13      <input type="number" id="quantity" name="quantity"><br><br>
14
15      <!-- Creates a submit button. -->
16      <input type="submit" value="submit">
17    </form>
18  </body>
19  </html>

```

This is what the HTML page looks like:

This shows how to create and use the input type number.

Choose a quantity below:

- **Password:**

- General syntax:
`<input type="password">`
- It defines a password field.
- Note: The characters in a password field are masked (shown as asterisks or circles).

- E.g. Consider the code below:

```
<!DOCTYPE html>
<html>
<head>
  <title> Password </title>
</head>
<body>
  <h1> This shows how to create and use the input type password. </h1>
  <!-- Uses the HTTP GET request as no specific request was stated. -->
  <form action="/">

    <!-- The value of the for attribute in label must match the id attribute attribute of input. -->
    <label for="text"> Enter something below: </label> <br>
    <input type="text" id="text" name="text" placeholder="Enter something"> <br>

    <label for="password"> Enter your password below: </label> <br>
    <input type="password" id="password" name="password" placeholder="Password"> <br>

    <!-- Creates a submit button. -->
    <input type="submit" value="submit">
  </form>
</body>
</html>
```

This is what the HTML page looks like:

This shows how to create and use the input type password.

Enter something below:

Enter your password below:

- **Submit:**
 - General syntax:
<input type="submit">
 - It defines a button for submitting form data to a form-handler.
 - The form-handler is typically a server page with a script for processing input data.
 - The form-handler is specified in the form's action attribute.

Input Attributes:

- **Maxlength:**
 - The input maxlength attribute specifies the maximum number of characters allowed in an input field.
 - **Note:** When a maxlength is set, the input field will not accept more than the specified number of characters. However, this attribute does not provide any feedback. So, if you want to alert the user, you must write JavaScript code.
- **Min and Max:**
 - The input min and max attributes specify the minimum and maximum values for an input field.
 - The min and max attributes work with the following input types: number, range, date, datetime-local, month, time and week.
 - **Note:** Use the max and min attributes together to create a range of legal values.
- **Pattern and Title:**
 - The input pattern attribute specifies a regular expression that the input field's value is checked against, when the form is submitted.

- The pattern attribute works with the following input types: text, date, search, url, tel, email, and password.
- **Note:** Use the title attribute to describe the pattern to help the user.
- The title attribute specifies extra information about an element.
- The information is most often shown as a tooltip text when the mouse moves over the element.
- **Placeholder:**
 - The input placeholder attribute specifies a short hint that describes the expected value of an input field (a sample value or a short description of the expected format).
 - The short hint is displayed in the input field before the user enters a value.
 - The placeholder attribute works with the following input types: text, search, url, tel, email, and password.
- **Required:**
 - The input required attribute specifies that an input field must be filled out before submitting the form.
 - The required attribute works with the following input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file.
- **Size:**
 - The input size attribute specifies the visible width, in characters, of an input field.
 - The default value for size is 20.
 - **Note:** The size attribute works with the following input types: text, search, tel, url, email, and password.
- **Value:**
 - The value attribute in HTML is used to specify the value of the element with which it is used. It has different meanings for different HTML elements. It can be used with the following elements: <input>, <button>, <meter>, , <option>, <progress> and <param>. The input value attribute specifies an initial value for an input field.

Select Element:

- The <select> element defines a drop-down list.
- The <option> element defines an option that can be selected.
- By default, the first item in the drop-down list is selected.
- To define a pre-selected option, add the selected attribute to the option.
- You can use the size attribute to specify the number of visible values.
- You can use the multiple attribute to allow the user to select more than one value.
- General syntax:

```
<select>
  <option> Option 1 </option>
  <option> Option 2 </option>
  ...
</select>
```


- E.g. Consider the code below:

```
<!DOCTYPE html>
<html>
<head>
  <title> Select </title>
</head>
<body>
  <h1> This shows how to create and use the select element. </h1>
  <!-- Uses the HTTP GET request as no specific request was stated. -->
  <form action="/">

    <!-- The value of the for attribute in label must match the id attribute attribute of select. -->
    <!-- Here, you can only choose one of the items. -->
    <label for="fruits"> Choose a fruit below: </label> <br>
    <select id="fruits" name="fruit" size="2">
      <option value="apple">Apple</option>
      <option value="banana">Banana</option>
      <option value="grape">Grape</option>
      <option value="orange">Orange</option>
    </select><br><br>

    <!-- Here, you can only more than one of the items. -->
    <!-- To select multiple options, hold down the Ctrl (windows) / Command (Mac) button. -->
    <label for="fruits"> Choose one or more fruits below: </label> <br>
    <select id="fruits" name="fruit" size="2" multiple>
      <option value="apple">Apple</option>
      <option value="banana">Banana</option>
      <option value="grape">Grape</option>
      <option value="orange">Orange</option>
    </select><br><br>

    <!-- Creates a submit button. -->
    <input type="submit" value="submit">
  </form>
</body>
</html>
```

This is what the HTML page looks like:

This shows how to create and use the select element.

Choose a fruit below:

Apple

Banana

Choose one or more fruits below:

Apple

Banana

submit

Textarea Element:

- General syntax:

```
<textarea>
...
</textarea>
```
- The <textarea> element defines a multi-line input field.
- The rows attribute specifies the visible number of lines in a text area.
- The cols attribute specifies the visible width of a text area.

- Note: By default, the size of the textarea box is changeable. You can use CSS to make it so that the size of the textarea box cannot be changed.

E.g. Consider the code below:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title> Textarea </title>
5  </head>
6  <body>
7  |   <h1> This shows how to create and use a textarea box. </h1>
8  |   <!-- Uses the HTTP GET request as no specific request was stated. -->
9  |   <form action="/">
10 |
11 |       <!-- The value of the for attribute in label must match the id attribute attribute of input. -->
12 |       <label for="textarea"> Enter something below: </label> <br>
13 |       <textarea id="textarea" name="text" placeholder="Enter something" rows="10" cols="30">
14 |
15 |       </textarea><br>
16 |
17 |       <!-- Creates a submit button. -->
18 |       <input type="submit" value="submit">
19 |   </form>
20 </body>
21 </html>
```

This is what the HTML page looks like:

This shows how to create and use a textarea.


Enter something below:

submit

The user can drag the 2 lines on the bottom right corner to change the size of the textarea box, as such:

This shows how to create and use a textarea.

Enter something below:



submit