

Web Applications

Front-End : HTML Form

Overview

	Web Applications
Front-End	Part 1a : Basic HTML Part 1b : Basic CSS Part 2 : HTML - Forms
Back-End	Part 3 : Form with Flask Part 4 : SQLite with Flask

Part 2 : HTML Forms

Sending data with HTML

Forms

Traditionally, the term 'form' refers to a printed document for users to fill in their information.

HTML borrows the concept of a form to refer to different elements that allow you to collect information from users visiting our website.

- HOW TO COLLECT INFORMATION FROM USERS
- DIFFERENT KINDS OF FORM CONTROLS

How Form Works?

1. A user fills in a form and submits the information to the server.

VOTE FOR YOUR FAVORITE SONG

Username:

I vote for:

- ☐ Despacito
- ☐ Shape of You
- ☐ Something Just Like This
- ☐ I'm the One
- ☐ Attention
- ☒ Symphony
- ☐ Stay
- ☐ That's What I like
- ☐ There's Nothing Holdin' Me Back
- ☐ It Ain't Me

Name / Value
"username" /
"John"

Name / Value
"vote" / "Symphony"

2. The name of each form control and its value, which the user entered or selected, is sent to the server.

3. The server processes the information using some programming language. It may also store the information in a database.

4. The server creates a new webpage and send it back to the user to read on a browser.

THANKS FOR SUBMITTING YOUR VOTE
Hi John, thanks for voting "Symphony" as your favorite song.

GET or POST ?

Forms can be sent using one of the two methods: get or post.

The GET Method

Note that the query string (name/value pairs) is sent in the URL of a GET request:

```
/test/demo_form.php?name1=value1&name2=value2
```

With the get method, the values from the form are added to the end of the URL specified in the action attribute. The get method is ideal for:

- short forms, such as search boxes.
- when you are just retrieving data from the web server, not sending information that should be added to or deleted from a database.

The POST Method

The data sent to the server with POST is stored in the request body of the HTTP request:

```
POST /test/demo_form.php HTTP/1.1  
Host: w3schools.com  
name1=value1&name2=value2
```

With the post method, the values are sent in what are known as HTTP headers. As a rule of thumb, you should use the post method if your form:

- allows users to upload a file;
- is very long;
- contain sensitive data, eg passwords;
- adds information to or deletes from a database.

If the method attribute is not specified, the form data will be sent using the get method.

Compare GET vs. POST

The following table compares the two HTTP methods: GET and POST.

	GET	POST
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

https://www.w3schools.com/tags/ref_httpmethods.asp

0. Form Structure

- **Using the GET method**

```
<form action="http://127.0.0.1:5000">
```

```
<p>This is where the form controls will appear.</p>
```

```
</form>
```

- **Using the POST method**

```
<form action="http://127.0.0.1:5000" method="post">
```

```
<p>This is where the form controls will appear.</p>
```

```
</form>
```

1. Text Input

```
<form action="http://127.0.0.1:5000/">
```

```
<p>Username:<input type="text" name="username"  
size="15" maxlength="30"></p>
```

```
<p>Username:</p>
```

```
<input type="text" name="username" size="15"  
maxlength="30">
```

```
</form>
```

Username:

Username:

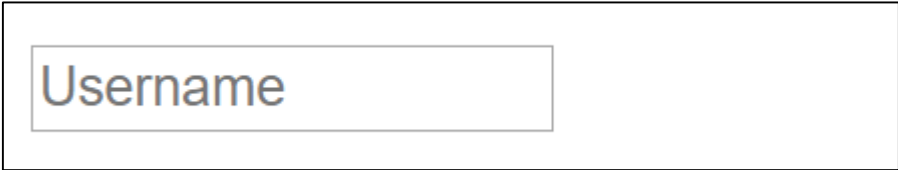
1. Text Input

- Using placeholder

```
<form action="http://127.0.0.1:5000/">
```

```
<p><input type="text" name="username" size="15"  
maxlength="30" placeholder="Username"></p>
```

```
</form>
```



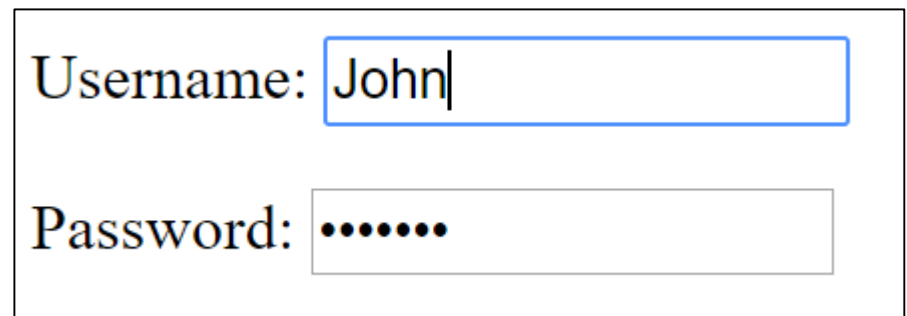
2. Password Input

```
<form action="http://127.0.0.1:5000/" method="POST">
```

```
<p>Username: <input type="text" name="username"  
size="15" maxlength="30"></p>
```

```
<p>Password: <input type="password" name="password"  
size="15" maxlength="30"></p>
```

```
</form>
```

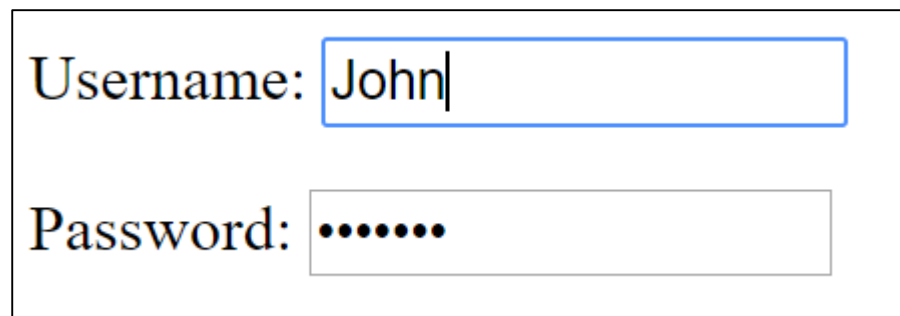


Username: John|

Password:

2. Password Input

- Although the password is hidden on the screen, this does not mean that the data in a password is sent securely to the server.
- For full security, the server needs to be set up to communicate with the users' browsers using Secure Sockets Layer (SSL)



A login form with two input fields. The first field is labeled 'Username:' and contains the text 'John'. The second field is labeled 'Password:' and contains seven dots, indicating a masked password.

Username:	<input type="text" value="John"/>
Password:	<input type="password" value="....."/>

3. Text Area

```
<form action="http://127.0.0.1:5000/" method="Post">
```

```
<p>What do you think about the movie?</p>
```

```
<br>
```

```
<textarea name="comments" cols="20" rows="4">Enter your  
comments ...</textarea>
```

```
</form>
```

What do you think about the movie?

Enter your comments

...|

4. Radio Button

- Once a radio button is selected, it **cannot be deselected**.
- The user can only select a different option.

```
<form action="http://127.0.0.1:5000/" method="Post">
```

```
<Please select your favorite genre:</p>
```

```
<input type="radio" name="genre" value="rock"  
      checked="checked"> Rock
```

```
<input type="radio" name="genre" value="pop">Pop
```

```
<input type="radio" name="genre" value="jazz">Jazz
```

```
</form>
```

Please select your favorite genre:

☒ Rock ☐ Pop ☐ Jazz

5a. Checkbox – only one option

- If you want to allow the user to select only one option and **be able to deselect**, you should use a checkbox instead.

```
<form action="http://127.0.0.1:5000/" method="Post">
```

```
<p>Terms and Conditions:</p>
```

```
<input type="checkbox" name="TnC" value="Yes">I accept  
all the Terms and Conditions.
```

```
</form>
```

Terms and Conditions:

☒ I accept all the Terms and Conditions.

5b. Checkbox – with few options

- Checkbox allow users to select and deselect one or more options

```
<form action="http://127.0.0.1:5000/">
```

```
<p>Please select your favorite music service(s):</p>
```

```
<input type="checkbox" name="service" value="itunes"  
      checked="checked">iTunes
```

```
<input type="checkbox" name="service"  
      value="lastfm">Last.fm
```

```
<input type="checkbox" name="service"  
      value="spotify">Spotify
```

```
</form>
```

Please select your favorite music service(s):

☒ iTunes ☐ Last.fm ☐ Spotify

6. Drop Down List Box

```
<form action="http://127.0.0.1:5000/">
```

```
<p>What device do you listen music on?</p>
```

```
<select name="devices">
```

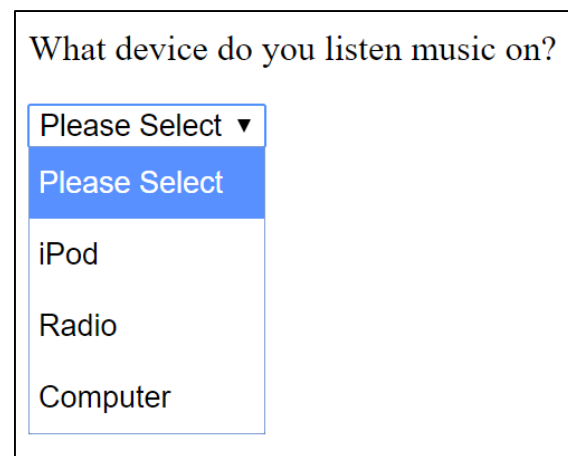
```
<option value="" selected="selected">Please Select</option>
```

```
<option value="ipod">iPod</option>
```

```
<option value="radio">Radio</option>
```

```
<option value="computer">Computer</option>
```

```
</select> </form>
```



What device do you listen music on?

Please Select ▼

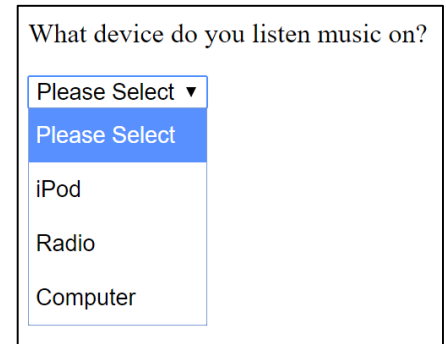
Please Select

iPod

Radio

Computer

6. Drop Down List Box



What device do you listen music on?

Please Select ▼

Please Select

iPod

Radio

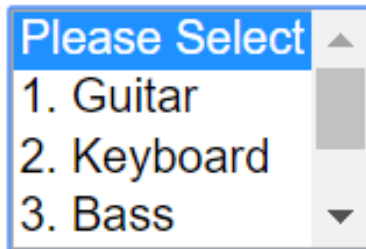
Computer

1. If the selected attribute is not used, the first option will be shown when the page loads. If the user does not select an option, then the first item will be sent to the server as the value for this control.
 2. The user can only select one option.
- if users need to see all the options at a glance, use radio buttons instead.
 - if there is a very long list of options, such as a list of countries, drop down list boxes should be used.

7. Multiple Select Box

Do you play any musical instruments?

(You can select more than one option by holding down control on a PC or command key on a Mac while selecting different options)



Please Select

- 1. Guitar
- 2. Keyboard
- 3. Bass

7. Multiple Select Box

```
<p>Do you play any musical instruments? <br>
```

```
(You can select more than one option by holding down  
control on a PC or command key on a Mac while selecting  
different options)</p>
```

```
<select name="instruments" size="4" multiple="multiple">  
  <option value="" selected="selected">  
    Please Select</option>  
  <option value="guitar">1. Guitar</option>  
  <option value="keyboard">2. Keyboard</option>  
  <option value="bass">3. Bass</option>  
  <option value="piano">4. Piano</option>  
  <option value="flute">5. Flute</option>  
</select>  
</form>
```

7. Multiple Select Box

- Size attribute is the number of options shown at once.
 - In our example, we show 4 out of the 6 options.
- Multiple attribute allows the user to select more than one option.
- It is a good practice to tell the user they can use the control/command key on the PC/Mac while selecting the options.

8. File Input Box

```
<form action="http://127.0.0.1:5000/" method="post">
```

```
<p>Upload your song in MP3 format:</p>
```

```
<input type="file" name="user-song">
```

```
<br>
```

```
<input type="submit" value="Upload">
```

```
</form>
```

Upload your song in MP3 fromat:

Choose File No file chosen

Upload

8. File Input Box

Upload your song in MP3 format:

Choose File

No file chosen

Upload

1. To upload files, we must use the POST method. We cannot send files using the HTTP GET method.
2. This type of input may appear differently on different browsers:
 - some creates a box that looks like a text input box with a "Browse" button;
 - on Chrome and Safari, they display a "Choose Files" button with a text showing "No file chosen" or the name of the file chosen by user.

9. Submit Button

```
<form action="http://127.0.0.1:5000/" method="post">
```

```
<p>Subscribe to our email list:</p>
```

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

```
<input type="submit" name="subscribe"  
      value="Count me in">
```

```
</form>
```

Type your email address to subscribe to our email list:

9. Submit Button

The value attribute for the submit button is used to specify the words to appear on the button; if not specify, the default value of the button is "Submit" on some browsers.

Type your email address to subscribe to our email list:

10. Image Button

```
<form action="http://127.0.0.1:5000/">
```

```
<p>Subscribe to our email list:</p>
```

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


```
<input type="image" src="image/submit.jpg" width="100"  
      height="25" alt="Submit">
```

```
</form>
```

Subscribe to our email list:

Submit

Subscribe to our email list:

 Submit form

11. Date Input

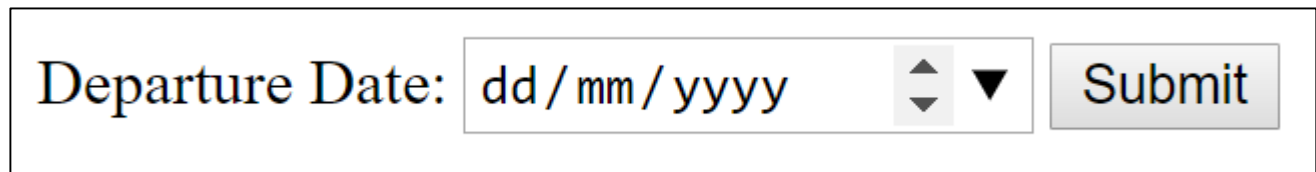
```
<form action="http://127.0.0.1:5000/" method="post">
```


```
<label for="depart">Departure Date:</label>
```

```
<input type="date" name="depart">
```

```
<input type="submit" value="Submit">
```

```
</form>
```



Departure Date: 

12. Grouping Forms Elements

```
<fieldset>
```

```
<legend>Contact Details</legend>
```

```
<label>Name:<br>
```

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

```
</label><br>
```

```
<label>Email:<br>
```

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

```
</label><br>
```

```
<label>Mobile:<br>
```

```
<input type="text" name="mobile">
```

```
</label><br>
```

```
</fieldset>
```

Contact Details

Name:

Email:

Mobile:

Design a form with all these form controls

Your Details:

Name:

Email:

Your Review:

How did you hear about us?

Would you visit again?

☐ Yes ☐ No ☐ Maybe

Comments:

☒ Sign me up for updates

Submit review