

Revision Tutorial

This Tutorial covers

- forms
- radio buttons
- posting a form
- input validation
- session handling
- creating a confirmation page

Creating the form

Let's begin by creating a simple html form

Open a new file in notepad or your favourite text editor

We'll create the following in the new file

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  </head>
5  <body>
6  <form action="form.php" method="POST">
7      First Name<input type='text' name='username' ></input><br/><br/>
8      <input type='radio' name='gender' value='Male'>Male</input>
9      <input type='radio' name='gender' value='Female'>Female</input><br/>
10     <input type='submit' </input>
11 </form>
12 </body>
13 </html>
```

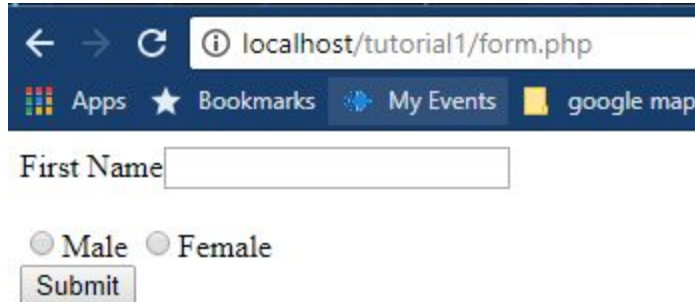
Lets save it under the htdocs folder as tutorial1/form.php where tutorial1 is a new folder. So the location of the file should be

C:\xampp\htdocs\tutorial1\form.php

Bring up your Apache server and load the file in the browser i.e. in the address bar type

localhost/tutorial1/form.php

You should see something like



First Name

☐ Male ☐ Female

Let's go back to the form and examine some of the code

```
<form action="form.php" method="POST">
```

The form has both an action and a method

action - tells us what php file to execute when we submit the form

method - whether to do a form **post** or **get**

When we submit the form we will execute the same php file. At the moment nothing much will happen if we submit as we haven't added any php code.

We have two input fields one for a user name and a radio button for selecting gender

```
First Name<input type='text' name='username' ></input><br/><br/>
<input type='radio' name='gender' value='Male'>Male</input>
<input type='radio' name='gender' value='Female'>Female</input><br/>
```

Note the **name** is how we store the value that the user has input. We will use this name when we post the form and retrieve these values. For the radio button we set the two choices with a value for each choice.

Adding the PHP code

Lets add some PHP code. At the top of the file add the following

```
<?php  
    print_r($_POST);  
?>
```

When we do a form POST PHP uses a global variable `$_POST` which stores the values from the form based on the name we have used for each input field. The code above just prints out this array when we post the form. Go ahead and add the php code and bring the form back up in the browser. Enter some values for each of the input fields and submit the form.

Depending on what you entered you should see something like

```
Array ( [username] => seanc [gender] => Female )  
First Name   
  
☐ Male ☒ Female  

```

Here we are printing the value of `$_POST` which is an associative array where the key is the name we gave to each of the input fields and the value is what the user entered on the form.

Let's make a change so that we only print out the `$_POST` variable if a POST has occurred. To do this we will use an if condition. So our code now looks like

```

1  <?php
2      if ($_POST) {
3          print_r($_POST);
4      }
5  ?>
6
7  <!DOCTYPE html>
8  <html>
9  <head>
10 </head>
11 <body>
12 <form action="form.php" method="POST">
13     First Name<input type='text' name='username' >/input><br/><br/>
14     <input type='radio' name='gender' value='Male'>Male</input>
15     <input type='radio' name='gender' value='Female'>Female</input><br/>
16     <input type='submit' </input>
17 </form>
18 </body>
19 </html>

```

Let's change our php code to store the \$_POST values to some variables and print them out.

```

<?php
    if ($_POST) {
        $username = $_POST['username'];
        $gender = $_POST['gender'];
        echo "<br/> User name: ".$username;
        echo "<br/> Gender: ".$gender;
    }
?>

```

Adding validation

Lets add validation to both our input fields. We'll start with the user name. Let's make the username be allowable between 4 and 20 characters. We'll use the strlen() function to help us with this.

We need to add an error string to store an error message which we'll print to the screen if the users has not met the validation. So we'll add the code

```
if (strlen($username) < 4 || strlen($username) > 20)
    $usernameErr = "name must be between 4 and 20 characters";
```

We need to add to the html form to print out the error message if the form fails validation. After the input field for username we'll add a span element to print the error if it exists.

```
1  <?php
2      $usernameErr = "";
3  if ($_POST) {
4      $username = $_POST['username'];
5      $gender = $_POST['gender'];
6      if (strlen($username) < 4 || strlen($username) > 20)
7          $usernameErr = "name must be between 4 and 20 characters";
8      echo "<br/> User name: ".$username;
9      echo "<br/> Gender: ".$gender; }
10 ?>
11 <!DOCTYPE html>
12 <html>
13 <head></head>
14 <body>
15 <form action="form.php" method="POST">
16     First Name<input type='text' name='username' >/input>
17     <span><?php echo $usernameErr ?></span><br/><br/>
18     <input type='radio' name='gender' value='Male'>Male</input>
19     <input type='radio' name='gender' value='Female'>Female</input><br/>
20     <input type='submit' </input>
21 </form>
22 </body>
23 </html>
```

Now let's move to the radio button. Notice if you submit the form without selecting a gender we get an error on line 5 as the `$_POST['gender']` has not been set. We'll add validation to fix this. We need another error message string to hold this message and print it to the screen. We'll use the `isset()` function to help us here. Change the php code so it looks like (we've removed the echo statements).

```

1  <?php
2      $usernameErr = "";
3      $genderErr = "";
4      if ($_POST) {
5          $username = $_POST['username'];
6          if (!isset($_POST['gender']))
7              $genderErr = "Please select a gender";
8          else
9              $gender = $_POST['gender'];
10         if (strlen($username) < 4 || strlen($username) > 20)
11             $usernameErr = "name must be between 4 and 20 characters";
12     }
13  ?>

```

Now lets add the error message to the form like we did for the username.
So the html code now looks like.

```

14  <!DOCTYPE html>
15  <html>
16      <head></head>
17      <body>
18          <form action="form.php" method="POST">
19              First Name<input type='text' name='username' ></input>
20              <span><?php echo $usernameErr ?></span><br/><br/>
21              <input type='radio' name='gender' value='Male'>Male</input>
22              <input type='radio' name='gender' value='Female'>Female</input>
23              <span><?php echo $genderErr ?></span><br/><br/><br/>
24              <input type='submit' </input>
25          </form>
26      </body>
27  </html>

```

Now test the app, press submit without entering anything to the form and you should see both errors appear.

Now let's add a condition to print out a message if the form is valid. For the form to be valid both the error messages should be empty, so we'll use the **empty()** function to help us here.

```
1 <?php
2     $usernameErr = "";
3     $genderErr = "";
4     if ($_POST) {
5         $username = $_POST['username'];
6         if (!isset($_POST['gender']))
7             $genderErr = "Please select a gender";
8         else
9             $gender = $_POST['gender'];
10        if (strlen($username) < 4 || strlen($username) > 20)
11            $usernameErr = "name must be between 4 and 20 characters";
12        if (empty($usernameErr) and empty($genderErr)) {
13            echo "<br/>All valid we're good to go";
14        }
15    }
16 }
17 ?>
```

Now let's print a message if the form is valid we want to print out the gender selected. Remember the value for the radio button was set by the 'value' in the input field. We'll add the following (in red)

```
if (empty($usernameErr) and empty($genderErr)) {
    echo "<br/>All valid we're good to go";
    if ($gender == 'Male')
        echo "<br/> You selected male as gender";
    if ($gender == 'Female')
        echo "<br/> You selected female as gender";
}
```

Adding a session

We're going to start a session and save some session variables. To start a session we use the

session_start() function

This needs to be placed at the very top of the php file. Once we have done this we now have available to us the global variable `$_SESSION` this is an array that we can use to store session data that makes the data available in another php file within the same session. We will store the `$username` and `$gender` variables to the session when we've validated the data.

```
1  <?php
2  session_start();
3  $usernameErr = "";
4  $genderErr = "";
5  if ($_POST) {
6      $username = $_POST['username'];
7      if (!isset($_POST['gender']))
8          $genderErr = "Please select a gender";
9      else
10         $gender = $_POST['gender'];
11     if (strlen($username) < 4 || strlen($username) > 20)
12         $usernameErr = "name must be between 4 and 20 characters";
13     if (empty($usernameErr) and empty($genderErr)) {
14         echo "<br/>All valid we're good to go";
15         if ($gender == 'Male')
16             echo "<br/> You selected male as gender";
17         if ($gender == 'Female')
18             echo "<br/> You selected female as gender";
19         $_SESSION['gender'] = $gender;
20         $_SESSION['username'] = $username;
21     }
22 }
23 ?>
```


Now once the data is valid we'll move to a new php file, depending on gender selected.

We'll create two files **maleconfirm.php** and **femaleconfirm.php**. In these files we'll print out the session variables.

First we need to make a change to the form.php to redirect to these files. We'll remove all the echo statements and add a header() function to do the redirection

```
1 <?php
2     session_start();
3     $usernameErr = "";
4     $genderErr = "";
5     if ($_POST) {
6         $username = $_POST['username'];
7         if (!isset($_POST['gender']))
8             $genderErr = "Please select a gender";
9         else
10            $gender = $_POST['gender'];
11        if (strlen($username) < 4 || strlen($username) > 20)
12            $usernameErr = "name must be between 4 and 20 characters";
13        if (empty($usernameErr) and empty($genderErr)) {
14            $_SESSION['gender'] = $gender;
15            $_SESSION['username'] = $username;
16            if ($gender == 'Male')
17                header('location: maleconfirm.php');
18            if ($gender == 'Female')
19                header('location: femaleconfirm.php');
20        }
21    }
22 ?>
```

The two confirm files look like (Remember we need the session_start() at the top of each file.

maleconfirm.php

```

1  <?php
2      session_start();
3
4      echo "<br/> gender is ".$_SESSION['gender'];
5      echo "<br/> user is ".$_SESSION['username']
6
7  ?>
8  <!DOCTYPE html>
9  <html>
10     <head></head>
11     <body>
12         <h1>You are in the male confirm file</h1>
13     </body>
14 </html>
15

```

femaleconfirm.php

```

1  <?php
2      session_start();
3
4      echo "<br/> gender is ".$_SESSION['gender'];
5      echo "<br/> user is ".$_SESSION['username']
6
7  ?>
8  <!DOCTYPE html>
9  <html>
10     <head></head>
11     <body>
12         <h1>You are in the female confirm file</h1>
13     </body>
14 </html>
15

```

Test the above in localhost.

Finally

In the next tutorial we'll look at adding a logout function (to destroy the session)