ଓଡ଼ିଶା ରାଜ୍ୟ ମୁକ୍ତ ବିଶ୍ୱବିଦ୍ୟାଳୟ, ସମ୍ବଲପୁର, ଓଡ଼ିଶା
Odisha State Open University, Sambalpur, Odisha
Established by an Act of Government of Odisha.

DCA-06

APPLICATION DEVELOPMENT

Block

2

Unit -3

Php Basics-2

Unit -4

Form Management

ଓଡ଼ିଶା ରାଜ୍ୟ ମୁକ୍ତ ବିଶ୍ୱବିଦ୍ୟାଳୟ, ସମ୍ବଲପୁର, ଓଡ଼ିଶା
Odisha State Open University, Sambalpur, Odisha
Established by an Act of Government of Odisha.

EXPERT COMMITTEE

Dr P.K.Behera **(Chairman)**
Reader in Computer Science
Utkal University
Bhubaneswar, Odisha

Dr. J.R.Mohanty **(Member)**
Professor and HOD
KIIT University
Bhubaneswar, Odisha

Sh Pabitrananda Pattnaik **(Member)**
Scientist –E,NIC
Bhubaneswar, Odisha

Sh Malaya Kumar Das **(Member)**
Scientist –E,NIC
Bhubaneswar, Odisha

Dr. Bhagirathi Nayak **(Member)**
Professor and Head(IT & System)
Sri Sri University
Bhubaneswar, Odisha

Dr. Manoranjan Pradhan **(Member)**
Professor and Head(IT & System)
G.I.T.A
Bhubaneswar, Odisha

Sri V.S.Sandilya **(Convener)**
Academic Consultant (I.T),
Odisha State Open University,
Sambalpur, Odisha

DIPLOMA IN COMPUTER APPLICATION

Course Writer

V.S.Sandilya

Academic Consultant (IT)
Odisha State Open University, Sambalpur, Odisha

Unit -3
Php Basics -2

Learning objectives

After the Completion of this unit you should be able to know

1) What are condition statements.
2) What is a loop and why use loops.
3) What are functions and why do we use functions.
4) What are arrays and what is the use of arrays.
5) What is super global and what is its use.

Structure

3.1 Introduction

When it comes to programming we depend on condition statements so that we overcome against statement conditions. The condition statement is used to tell the computer what to do if such condition arises. As we all know a computer does not have IQ. It works on instructions given by the programmer. The loop statements are used to run a part of program defined number of times. Arrays are used to store data in a sequential manner. Arrays improve the speed of the programming by many times. The advantages of arrays are data in arrays can be manipulated as many times as possible before writing the data permanently into the computer. Arrays can be cleaned up from memory once the job is over.

3.2 Definition

1. Condition Statement :- A conditional statement is a set of rules performed if a certain condition is met.
2. Loop :- A loop is a sequence of instructions that is continually repeated until a certain condition is reached.
3. Array :- A special variable used to hold multiple values at the same time in a continuous memory.

3.3 Condition Statement

3.3.1 If Statement

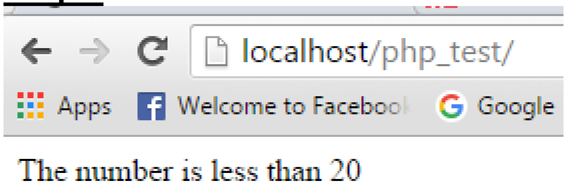
Syntax

```
if (condition)
{
Code to be executed starts here
}
```

Example

```
<?php
$i =10;
if($i< 20)
{
echo ('The number is less than 20');
}
?>
```

Output



The number is less than 20

3.3.2 If else statement

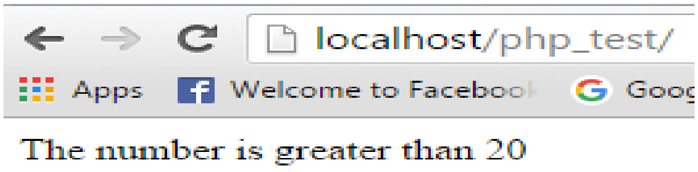
Syntax

```
if (condition) {
    code to be executed if condition is true;
} else {
    code to be executed if condition is false;
}
```

Example

```
<?php
$i =21;
if($i< 20)
{
echo ('The number is less than 20');
}
else
{
echo("The number is greater than 20");
}
?>
```

Output



The number is greater than 20

3.3.3 If else if else statement

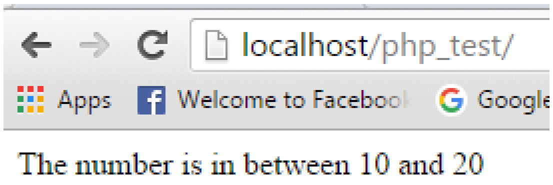
Syntax

```
if (condition) {
    code to be executed if this condition is true;
} elseif (condition) {
    code to be executed if this condition is true;
} else {
    code to be executed if all conditions are false;
}
```

Example

```
<?php
$i = 15;
if($i> 20)
{
echo ('The number is greater than 20');
}
else if($i > 10 && $i < 20)
{
echo('The number is in between 10 and 20');
}
else
{
echo ('Not a valid number');
}
?>
```

Output



```
The number is in between 10 and 20
```

3.3.4 Switch Statement

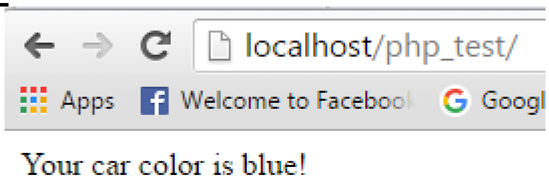
Syntax

```
      switch (c) {
          case label1:
              code to be executed if c=label1;
              break;
          case label2:
              code to be executed if c=label2;
              break;
          case label3:
              code to be executed if c=label3;
              break;
          ...
          default:
              code to be executed if c is different from
      all labels;
      }
```

Example

```
<?php
$carcolor = "blue";
switch ($carcolor) {
    case "red":
        echo "Your car color is red!";
        break;
    case "blue":
        echo "Your car color is blue!";
        break;
    case "green":
        echo "Your car color is green!";
        break;
    default:
        echo "Your car color is neither red, blue, nor green!";
}
?>
```

Output



```
Your car color is blue!
```

3.4 Loop

The loop is of 4 types in PHP

1. While loop
2. Do While loop
3. For loop
4. For each loop /* We shall discuss about this in arrays */

3.4.1 While loop

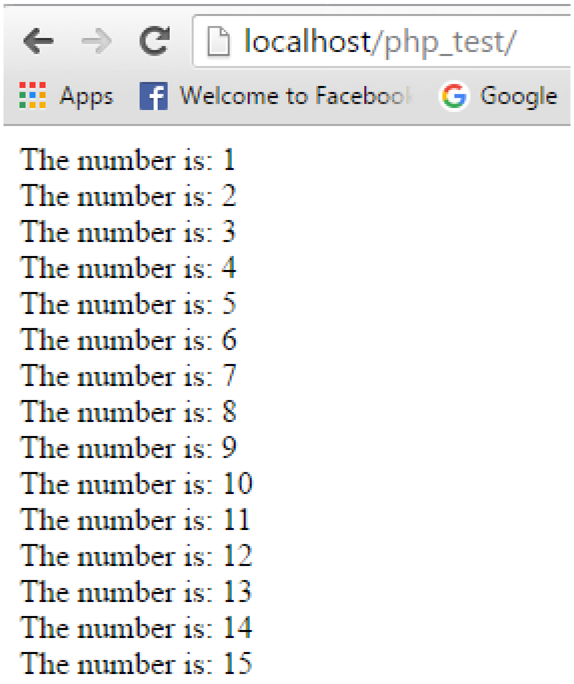
Syntax

```
while (condition is true) {
    code to be executed;
}
```

Example

```
<?php
$n = 1;
while($n <= 15) {
   echo "The number is: $n <br>";
   $n++;
}
?>
```

Output



```
The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9
The number is: 10
The number is: 11
The number is: 12
The number is: 13
The number is: 14
The number is: 15
```

3.4.2 Do While loop

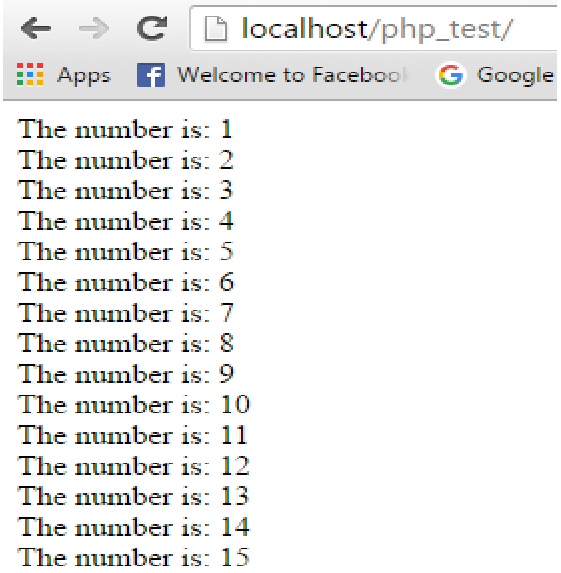
Syntax

```
do {
    code to be executed;
} while (condition is true);
```

Example

```
<?php
$n = 1;
do {
   echo "The number is: $n <br>";
   $n++;
} while ($n <= 15);
?>
```

Output



```
The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9
The number is: 10
The number is: 11
The number is: 12
The number is: 13
The number is: 14
The number is: 15
```

Description

In the above example the do loop runs as long as $n<=15

3.4.3 For loop

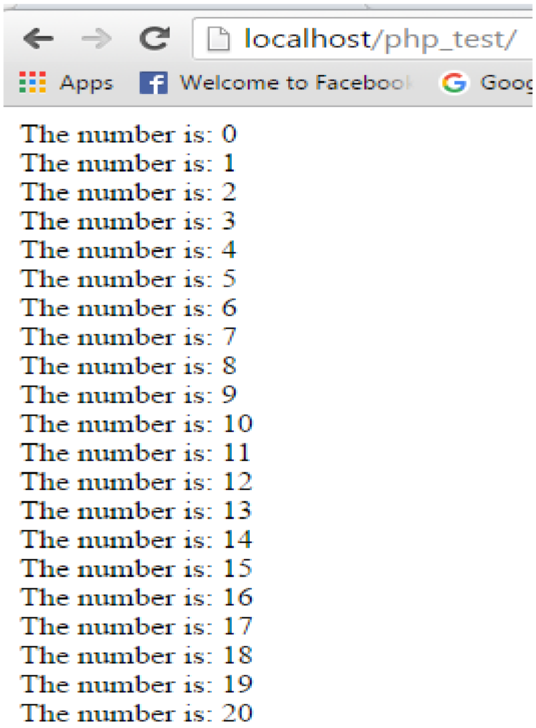
Syntax

```
for (init counter; test counter; increment
counter) {
    code to be executed;
}
```

Example

```
<?php
for ($n = 0; $n <= 20; $n++) {
   echo "The number is: $n <br>";
}
?>
```

Output



```
The number is: 0
The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9
The number is: 10
The number is: 11
The number is: 12
The number is: 13
The number is: 14
The number is: 15
The number is: 16
The number is: 17
The number is: 18
The number is: 19
The number is: 20
```

Description

In the above example the for loops runs from 0 to 20 and displays the output as shown above.

3.4.3 For each loop

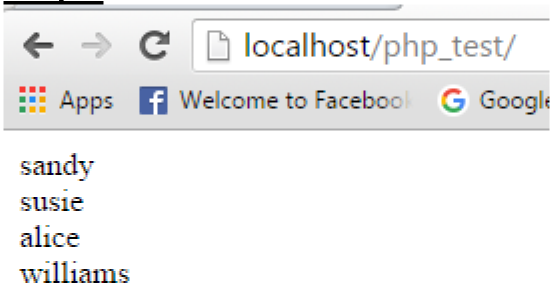
Syntax

```
foreach ($array as $value) {
    code to be executed;
}
```

Example

```
<?php
$names = array("sandy","susie","alice","williams");
foreach ($names as $value) {
  echo "$value <br>";
}
?>
```

Output



```
sandy
susie
alice
williams
```

Description

In the above example we have defined an array $names where we defined 4 variables sandy, Susie, Alice and Williams. foreach ($names as $value) retrieves each name from the array names and assigns it to variable $value. echo “$value” displays the names in loop fashion.

3.5 Super Global

Several predefined variables in PHP are super global, which means they are available in all scopes throughout a script.

The super global are

1. $_POST
2. $_GET
3. $_FILES
4. $_REQUEST
5. $GLOBALS
6. S_SERVER
7. $_ENV
8. $_COOKIE
9. $_SESSION

We shall be discussing about these super global in detail in the forthcoming unit.

Check your progress -1

Q.1 What is a condition Statement? What are the various kinds of condition statements?

A.____

____

Q.2 What is a loop? What are the various kinds of loops?

A.____

____

Q.3 What is a super global? What are the various kinds of SuperGlobal?

A.____

____

____

3.6 Let us sum up

A conditional statement is a set of rules performed if a certain condition is met. Conditional statements are if, if elseif, if elseif else, switch. A loop is a sequence of instructions that is continually repeated until a certain condition is reached. Typically, a certain process is done. The various kinds of loops are While loop, Do While loop, For loop and For each loop. Super global are those variables that can be accessed from any part of the program irrespective of the scope of the variable. The super global are $_POST, $_GET, $_FILES, $_REQUEST, $GLOBALS, S_SERVER,$_ENV,$_COOKIE, $_SESSION

3.7 References

1. W3schools.com
2. Google.com.

3.8 Check your progress –possible answers

Check your progress –possible answers -1

A.1 A conditional statement is a set of rules performed if a certain condition is met. Conditional statements are if, if elseif, if elseif else, switch.

A.2 A loop is a sequence of instructions that is continually repeated until a certain condition is reached. Typically, a certain process is done. The various kinds of loops are While loop, Do While loop, For loop and For each loop.

A.3 Super global are those variables that can be accessed from any part of the program irrespective of the scope of the variable. The super global are $_POST, $_GET, $_FILES, $_REQUEST, $GLOBALS, $_SERVER,$_ENV,$_COOKIE, $_SESSION

Unit -4
Form Management

Learning objectives

After the Completion of this unit you should be able to know

1) Form Design
2) Form Validation.
3) Form navigation

Structure

4.1 Introduction

Forms are used to input, view data, search data basing on conditions. Forms are used to create an easy user interface for the end user to enter data. Example to enter data of a person are person name, person date of birth, gender, address, mobile number, email etc. Forms help an end user to enter appropriate data so that he/she does not commit a mistake. This process is called form validation. Form validation are of two types and they are client side validation and server side validation. The client side validation is done by Java Script and the server side validation is done by Php. Programmers do both side validations so that in case the client side validation fails the server side validation works. Client side validation fails at times because this may be disabled by the browser by the end user of the browser might be an old browser which does not understand the Javascript validation techniques. We shall also learn about the form navigation in this unit.

4.2 Definition

1. Form :- A form is a window or screen that contains numerous fields, or spaces to enter or view or edit or delete data . Each field holds a field label so that any user who views the form gets an idea of its contents.
2. Form validation :- Form validation is the process of checking the data entered by the end user by the computer that the data entered is correct before the computer processes the data.
3. Form Navigation :- Form navigation is a process used in form to navigate to the required form upon clicking of a button or a hyperlink.

4.3 Form Design

In this section we shall be dealing with creation of forms, form validation, form entry using Html, Javascript and Php. We shall be learning this section by very small examples.

Data is entered by Input types in HTML5. These are

1. Textbox
2. Text area
3. Password
4. Check boxes
5. Radio buttons
6. Drop down box
7. File
8. Submit
9. Reset
10. Button

11. Color
12. Date
13. Date time
14. Date time-local
15. Time
16. Week
17. Month
18. Email
19. Number
20. Range
21. Search
22. Tel
23. Url

The super global used in form management are

1. $_POST
2. $_GET
3. $_FILES
4. $_REQUEST
5. $GLOBALS
6. S_SERVER
7. $_ENV
8. $_COOKIE
9. $_SESSION

4.3.1 Working With Textbox

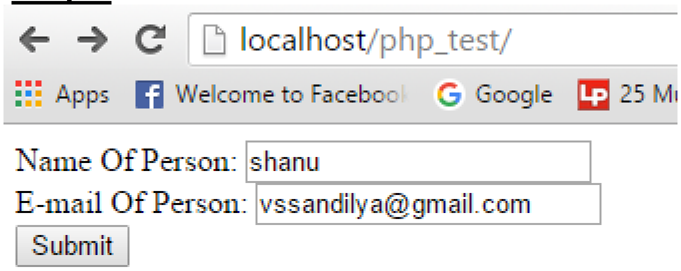
4.3.1.1 Working With Textbox and $_POST method

In this section we shall be working on Text box and a super global $_POST.

Example :-

```
<html>
<body>
<form action="welcome.php" method="post">
Name   Of Person: <input type="text" name="person_name"><br>
E-mail Of Person: <input type="text" name="person_email"><br>
<input type="submit">
</form>
</body>
</html>
```

Output



Name Of Person: shanu
E-mail Of Person: vssandilya@gmail.com

Description

In the above example we have defined

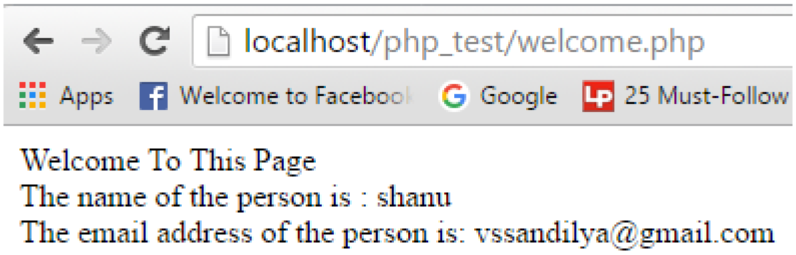
<form action="welcome.php" method="post">

This means that once a person clicks on the submit button the person is directed to the page welcome.php page and the method is post.

Creating welcome.php with $_POST

```
<html>
<body>
Welcome  To This Page <br>
The name of the person is : <?php echo $_POST["person_name"]; ?><br>
The  email address of the person is: <?php echo $_POST["person_email"]; ?>
</body>
</html>
```

Output



Welcome To This Page
The name of the person is : shanu
The email address of the person is: vssandilya@gmail.com

Description

The data sent from index.php is received by welcome.php and this is receive by command $_POST[ ].

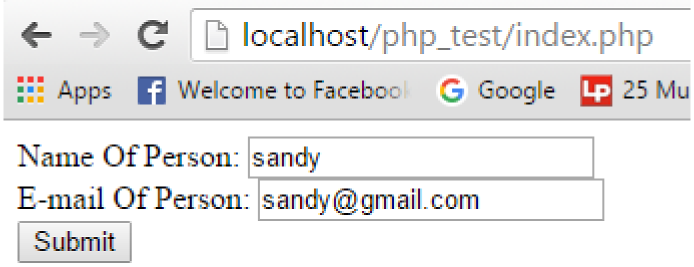
4.3.1.2 Working With Textbox an $_GET method

In this section we shall be working on Text box and a super global $_GET

Example :-

```
<html>
<body>
<form action="welcome.php" method="get">
Name   Of Person: <input type="text" name="person_name"><br>
E-mail Of Person: <input type="text" name="person_email"><br>
<input type="submit">
</form>
</body>
</html>
```

Output



localhost/php_test/index.php

Name Of Person: sandy
E-mail Of Person: sandy@gmail.com
Submit

Description

In the above example we have defined

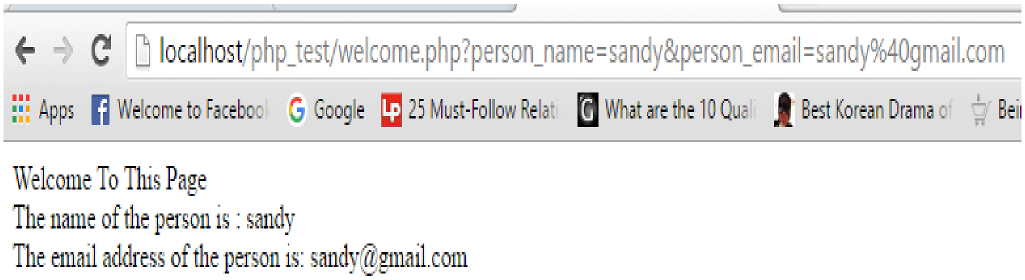
```
<form action="welcome.php" method="get">
```

This means that once a person clicks on the submit button the person is directed to the page welcome.php page and the method is get.

Creating welcome.php with $_GET

```
<html>
<body>
Welcome  To This Page <br>
The name of the person is : <?php echo $_GET["person_name"]; ?><br>
The  email address of the person is: <?php echo $_GET["person_email"]; ?>
</body>
</html>
```

Output



localhost/php_test/welcome.php?person_name=sandy&person_email=sandy%40gmail.com

```
Welcome To This Page
The name of the person is : sandy
The email address of the person is: sandy@gmail.com
```

Difference between $_GET and $_POST

Sl.No	$_GET Method	$_POST Method
1	Data sent via $_GET method is visible across the URL	Data sent via $_POST method is not visible across the URL
2	It is an array of variables passed to the current script via URL parameters	It is an array of variables passed to the current script via HTTP post parameters
3	It is confined to 2000 characters through this method	Infinite amount of data is transmitted through this method
4	Should not be used when passing sensitive information such as user name and password	Should be used when passing sensitive information such as user name and password

Check your progress -1

Q.1 What is a form ? What are input types and their use ?

A.____

____

Q.2 What is the difference between $_GET and $_POST?

A.____

____

4.3.2 Working With Text Area

index.php

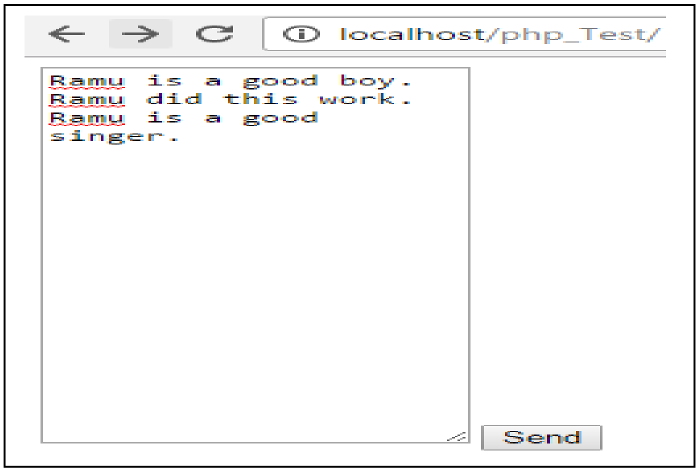
```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>test</title>
</head>
<body>
<form method="post" action="welcome.php">
<textarea id="work_list" name="work_list" rows="20" cols="20"></textarea>
<input type="submit" name="submit" value="Send" id="submit"/>
</form>
</body>
</html>
```

welcome.php

```
<table>
<tr>
<td width="140">
<?php
   if ($_POST['submit']) {
            echo $_POST['work_list'];
   }
?>
</td>
</tr>
</table>
```

Output

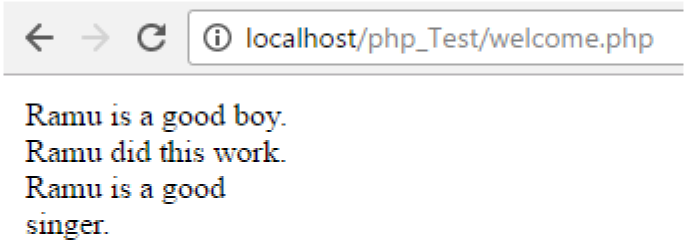
Step-1



```
Ramu is a good boy.
Ramu did this work.
Ramu is a good
singer.
```

Step-2

Upon clicking the send button the output is displayed in welcome. php as shown below.



```
Ramu is a good boy.
Ramu did this work.
Ramu is a good
singer.
```

Description

In the above example we have defined

<form action="welcome.php" method="post">

This means that once a person clicks on the send button the person is directed to the page welcome.php page and the method is post.

4.3.3 Working With Password

We use the password option so that the data being entered into the computer is seen by others as *****. The password utility is security.

index.php

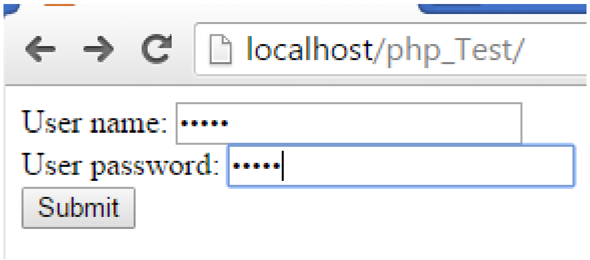
```
<!DOCTYPE html>
<html>
<body>
<form action="welcome.php" method="POST">
User name: <input  type="password" name="user_name"><br>
User password: <input  type="password" name="passwd"><br>
<input type="submit">
</form>
</body>
</html>
```

Creating welcome.php with $_POST

```
<html>
<body>
Welcome  To User Login <br>
User name  is : <?php echo $_POST["user_name"]; ?><br>
Password   is : <?php echo $_POST["passwd"]; ?>
</body>
</html>
```

Output

Step-1 :- Open google chrome browser

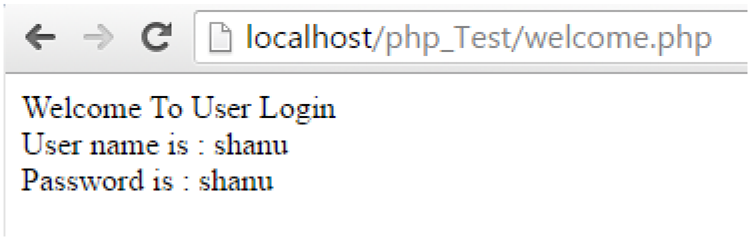


User name: •••••

User password: •••••

Step-2

Upon clicking the submit button



Welcome To User Login
User name is : shanu
Password is : shanu

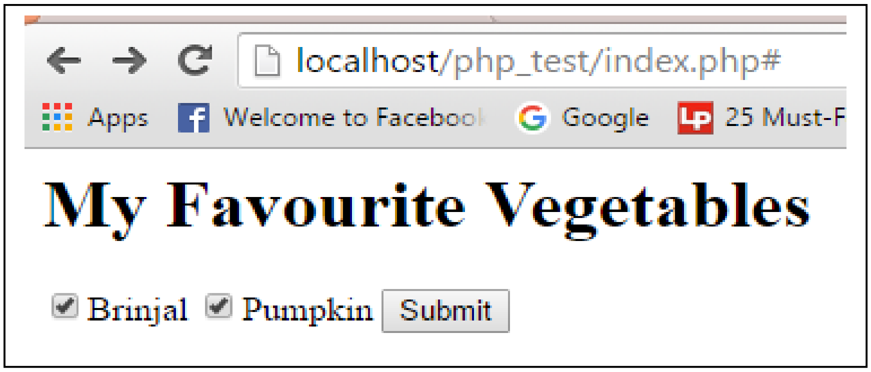
Description

In the above example we have defined a file named index.php which is comprised of the code as <input type="password" name="user_name"> which means input type is password and the name is user_name.

4.3.4 Working With Checkboxes and $_POST

```
<form action="welcome.php" method="post">
<H1> My Favourite Vegetables </H1>
<input type="checkbox" name="brinjal" value="Brinjal">Brinjal</input>
<input type="checkbox" name="pumpkin" value="Pumpkin">Pumpkin</input>
<input type="submit" name="submit" value="Submit"/>
</form>
```

Output



My Favourite Vegetables

☑ Brinjal ☑ Pumpkin Submit

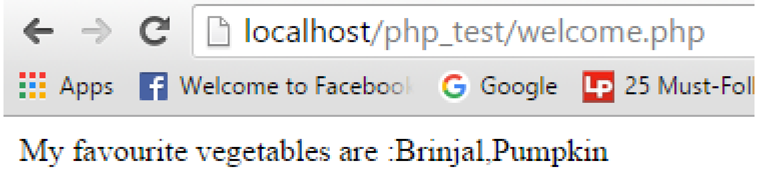
Description

In the above example we set up 2 checkboxes and set action to welcome.php and the method as post. The checkboxes are given different names such as brinjal and pumpkin. Upon clicking the submit button data is sent to the welome.php.

Creating welcome.php with $_POST

```
<?php
echo('My favourite vegetables are  :');
if (isset($_POST['brinjal'])){
echo $_POST['brinjal']; // Displays value of checked checkbox.
echo(",");
}
if (isset($_POST['pumpkin'])){
echo $_POST['pumpkin']; // Displays value of checked checkbox.
}
?>
```

Output

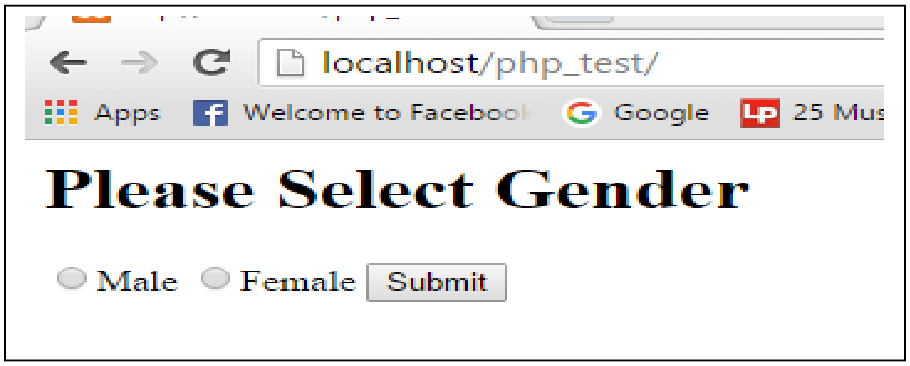


```
My favourite vegetables are :Brinjal,Pumpkin
```

4.3.5 Working With Radio Button and $_POST

```
<form action="welcome.php" method="post">
<H1> Please Select Gender  </H1>
<input type="radio" name="gender" value="Male">Male</input>
<input type="radio" name="gender" value="Female">Female</input>
<input type="submit" name="submit" value="Submit"/>
</form>
```

Output



Please Select Gender

○ Male ○ Female Submit

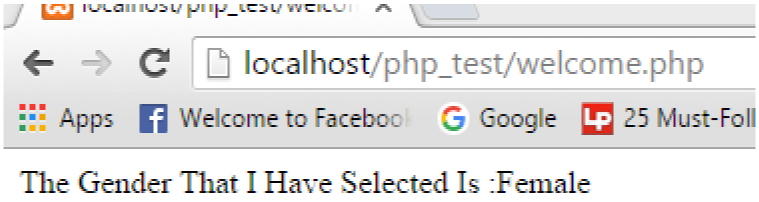
Description

In the above example we have created two radio buttons namely male and female. Since we have to select any one gender male or female. We have a submit button which is used to submit the content of the radio buttons.

Creating welcome.php with $_POST

```
<?php
echo('The Gender That I Have Selected Is  :');
if (isset($_POST['gender'])){
echo $_POST['gender']; // Displays value of radiobox .
}
?>
```

Output

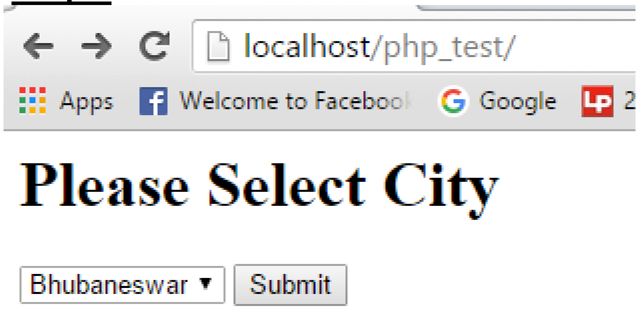


```
The Gender That I Have Selected Is :Female
```

4.3.6 Working With Drop Down Box and $_POST

```
<form action="welcome.php" method="post">
<H1> Please Select City  </H1>
<select id='city' name='city'>
<option value="Bhubaneswar" >Bhubaneswar</option>
<option value="Cuttack">Cuttack</option>
<option value="Calcutta">Calcutta</option>
</select>
<input type="submit" name="submit" value="Submit"/></form>
```

Output



Please Select City

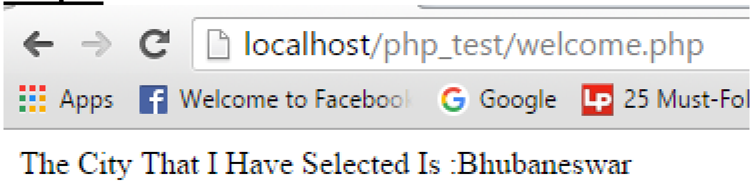
Description

In the above example the drop down is generated with 3 cities Bhubaneswar, Cuttack and Calcutta and upon clicking the submit button values of the drop down box are sent to welcome.php.

Creating welcome.php with $_POST

```
<?php

echo('The City That I Have Selected Is  :');

if (isset($_POST['city'])){

echo $_POST['city']; // Displays value of drop down

}

?>
```

Output



The City That I Have Selected Is :Bhubaneswar

Description

The output is received by $_POST[‘CITY’] which we received from the drop down box.

4.3.7 Working With Files and $_POST

Source:-

https://www.developphp.com/video/PHP/Build-an-Image-Upload-Application-to-Enable-Working-With-Images-On-the-Fly

Images are a kind of data that are uploaded to the folder of the server.

Step-1:

Here we have defined a folder named uploads in e:\ drive and the folder path comes to

e:\uploads.

Step-2:

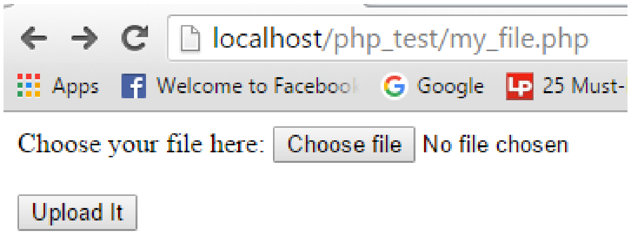
Now at C:\xampp\htdocs folder create a file named my_file.php

Step-3:

```
<?php
// Access the $_FILES global variable for this specific file being uploaded
// and create local PHP variables from the $_FILES array of information
$fileName = $_FILES["my_file"]["name"]; // The file name
$fileTmpLoc = $_FILES["my_file"]["tmp_name"]; // File in the PHP tmp folder
$fileType = $_FILES["my_file"]["type"]; // The type of file it is
$fileSize = $_FILES["my_file"]["size"]; // File size in bytes
$fileErrorMsg = $_FILES["my_file"]["error"]; // 0 = false | 1 = true
$moveResult = move_uploaded_file($fileTmpLoc, "e:/uploads/$fileName");
// Check to make sure the move result is true before continuing
if ($moveResult != true) {
    echo "ERROR: File not uploaded. Try again.";
    //unlink($fileTmpLoc);
// Remove the uploaded file from the PHP temp folder
    exit();
}
// Display things to the page so you can see what is happening for testing purposes
echo "The file named <strong>$fileName</strong> uploaded successfully.<br /><br />";
echo "It is <strong>$fileSize</strong> bytes in size.<br /><br />";
echo "It is an <strong>$fileType</strong> type of file.<br /><br />";
//echo "The Error Message output for this upload is: $fileErrorMsg";
?>
```

Output

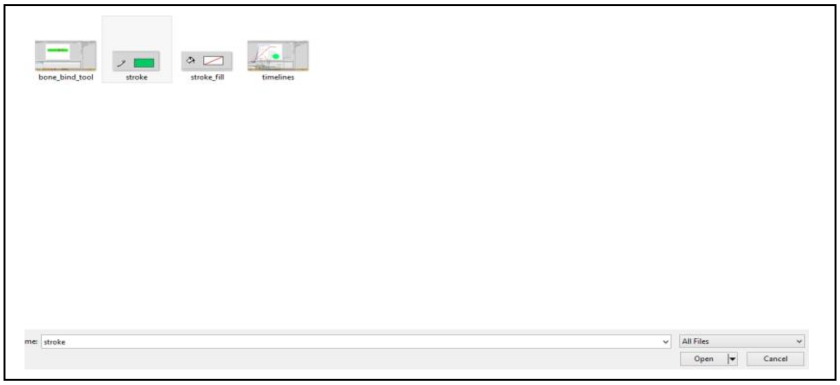
Step-1:-



localhost/php_test/my_file.php

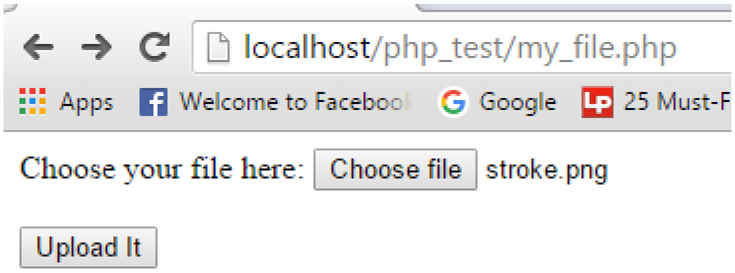
Choose your file here: Choose file No file chosen

Step-2 :-



me: stroke

Step-3 :-

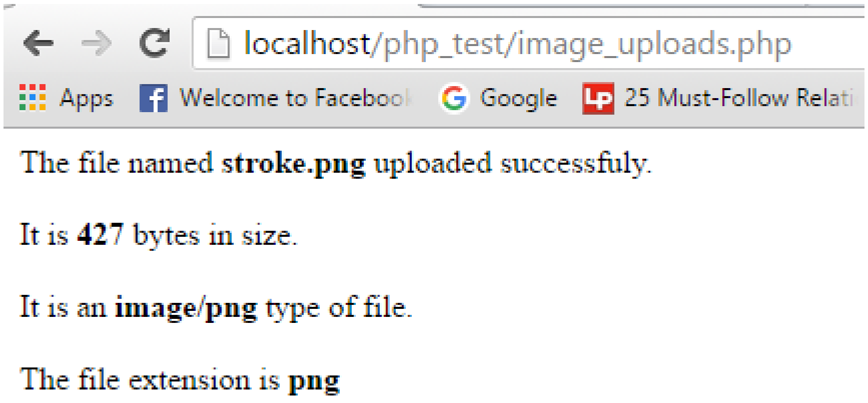


Choose your file here: Choose file stroke.png

Upload It

Step-4

Clicking on the upload it button the image is uploaded into it we are directed to the screen as shown below.



The file named **stroke.png** uploaded successfuly.

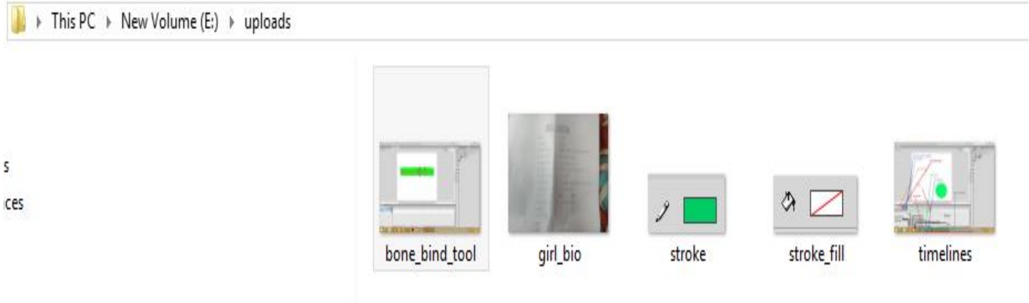
It is **427** bytes in size.

It is an **image/png** type of file.

The file extension is **png**

Step-5

We find in E:\uploads the images as shown below



4.3.8 Working with the Submit Button

The submit button is used to submit the form as above. This   button is clicked after all the fields in the form are appropriately filled up.

4.3.9 Working with the Reset Button

The Reset button is used to reset all the input fields. Used to clear the form

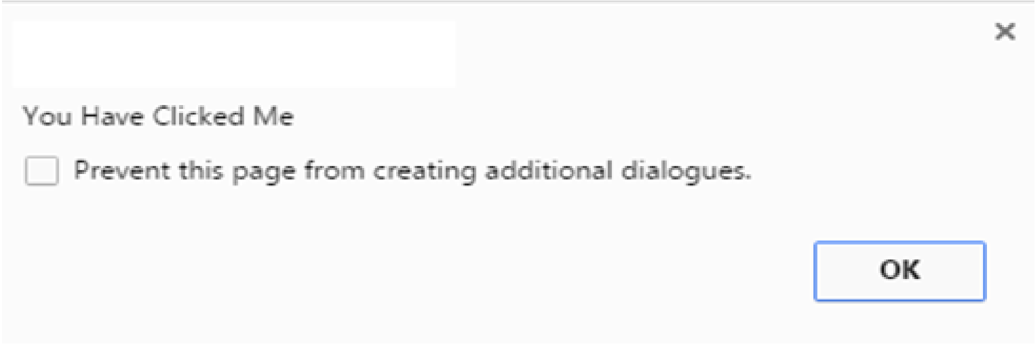
```
<!DOCTYPE html>
<html>
<body>
<form action="welcome.php" method="POST">
User name:
<input  type="text" name="user_name">
<br>
User password:
<input  type="text" name="passwd">
<br>
<input type="submit">
<input type="reset">
</form>
</body>
</html>
```

4.3.10 Working With Button

The button is a form control that is used to be clicked as shown below

```
<!DOCTYPE html>
<html>
<body>
<button type="button" onclick="alert('You Have Clicked Me')">
Click Me!
</button>
 </body>
</html>
```

Output



You Have Clicked Me

□ Prevent this page from creating additional dialogues.

Description

In the above example we prepared a button and on click an event occurs and shows an alert message You Have Clicked Me.

4.3.11 Working with Color control

Let us write the code for index.php as shown below

Index.php

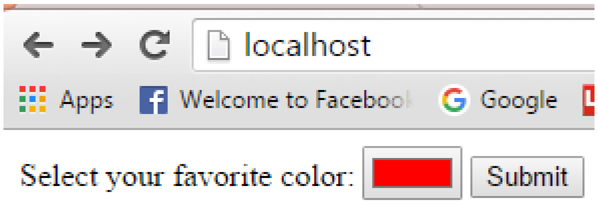
```
<!DOCTYPE html>
<html><body>
<form action= 'php_test.php' method='post'>
  Select your favorite color:
 <input type=’color’ name=’mycolor’   value=’#ff0000’>
 <input type=’submit’>
 </form>
 </body>
</html>
```

Let us write the code for php_test.php as shown below

php_test.php

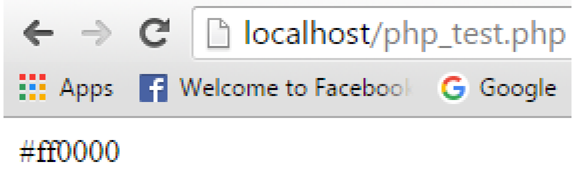
```
<html>
<head>
<Title> My first PHP program </title>
</head>
<body>
<?php
$p= $_POST['mycolor'];
echo $p;
?>
</body> </html>
```

Output



Select your favorite color: 🟥 Submit

Now clicking on the submit button the control goes to php_test.php and the output is as shown below



```
#ff0000
```

Description

In the above example we have defined a color control as shown below

```
<input type=’color’ name=’mycolor’   value=’#ff0000’>
```

In the above code we have defined a input type as ‘color’ and gave it a name as ‘mycolor’ and defined the value as ‘ff0000’ which is red and on clicking the submit button we are directed to php_test.php which is #ff0000.

4.3.12 Working with Date control

This is a form control of html5 which is used to enter the date as shown below.

index.php

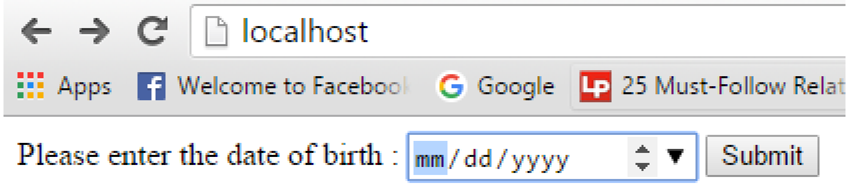
```
<!DOCTYPE html>
<html>
<body>
<form action="php_test.php" method="post">
  Please enter the date of birth :
  <input type="date" name="dob" >
  <input type="submit"> </form> </body></html>
```

php_test.php

```
<html>
<head>
<Title> My first PHP program </title>
</head>
<body>
<?php
$p= $_POST['dob'];
echo $p;
?>
</body>
</html>
```

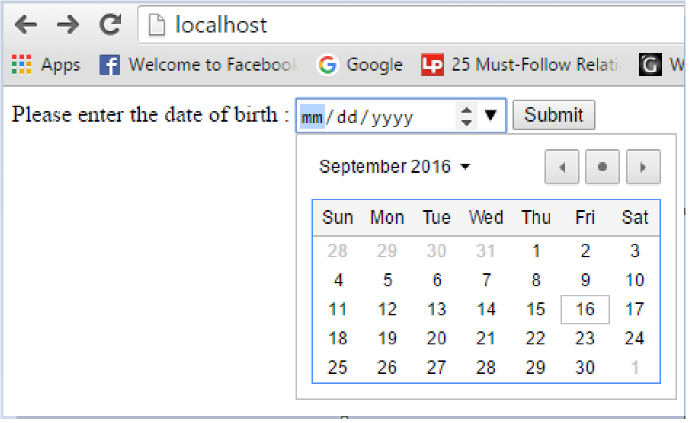
Output

Step-1



Please enter the date of birth : mm / dd / yyyy

Step-2



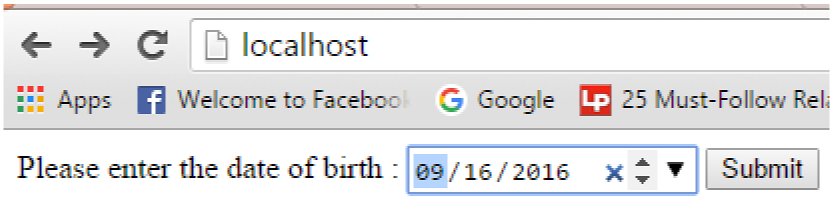
Please enter the date of birth : mm/dd/yyyy

September 2016

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

Step-3

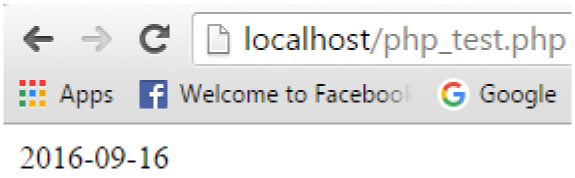
Upon selecting the date in the date control



Please enter the date of birth : 09/16/2016

Step-4

Upon clicking on the submit button the contol is directed to php_test.php as shown below



2016-09-16

4.3.13 Working with datetime control

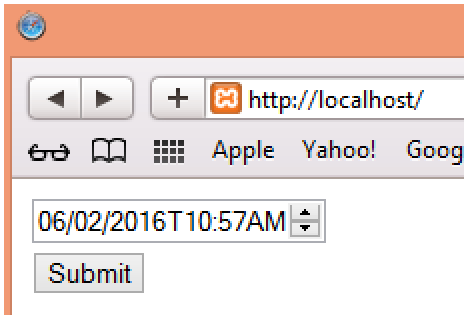
This is a form control used in HTML5 and is used for entering date and time. It can be used in web browsers like Safari as shown below.

```
<!DOCTYPE html>

<html>

<body>

<form action="php_test.php" method="post">
```

```
<input type="datetime" id="myDatetime" name="myDatetime"
value="06/02/2016T10:57AM"><br>

<input type="submit">

</form>

</body>

</html>
```

Output

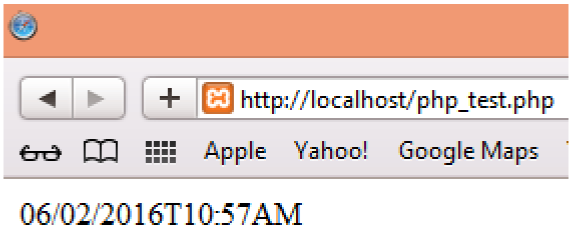
Step-1



06/02/2016T10:57AM

Step-2

After clicking the submit button we are directed to page php_test.php as shown below.



```
06/02/2016T10:57AM
```

4.3.14 Working with datetime-local control

index.php

```
<!DOCTYPE html>
<html>
<body>
<form action="php_test.php" method="post">
<h3>A demonstration of how to access a Datetime field</h3>
<input type="datetime-local" id="myDatetime" name= "myDatetime">
<input type="submit">
```

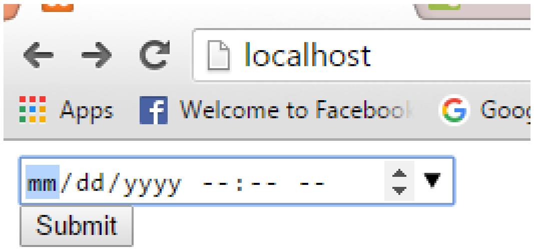
```
</form>
</body>
</html>
```

php_test.php

```
------------------
<head>
<Title> My first PHP program </title>
</head>
<body>
<?php
$p= $_POST['myDatetime'];
echo ($p);
?>
</body>
</html>
```

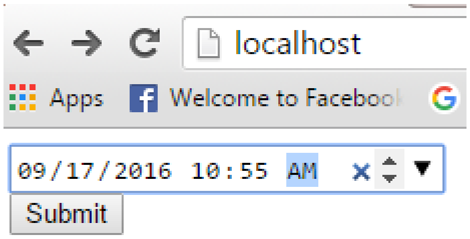
<u>Output</u>

<u>Step-1</u>



mm/dd/yyyy --:-- --

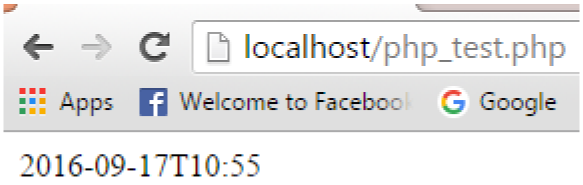
<u>Step-2</u>



09/17/2016 10:55 AM

<u>Step-3</u>

Clicking on submit button



```
2016-09-17T10:55
```

4.3.15 Working with time control

index.php

```
<!DOCTYPE html>
<html>
<body>
<form action="php_test.php" method="post">
<input type="time" id="myTime" name="myTime"><br>
<input type="submit">
</form>
</body>
</html>
```

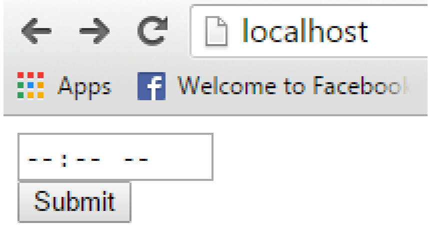
php_test.php

```
-----------------
<html>
<head>
<Title> My first PHP program </title>
</head>
<body>
<?php
$date= $_POST['myTime'];
echo date('h:i:s a', strtotime($date));
?></body></html>
```

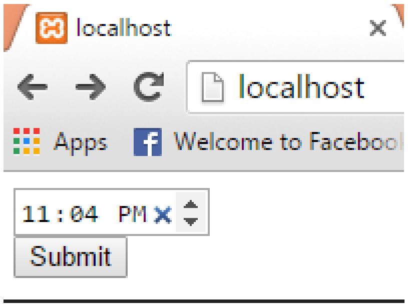
Output

Open the chrome browser

Step-1



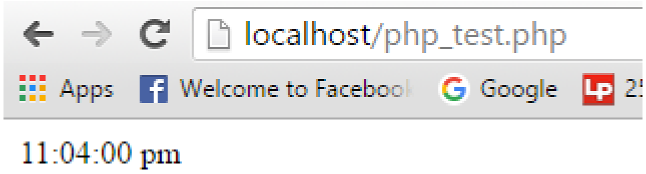
Step-2



11:04 PM

Step-3

Upon clicking the submit button



```
11:04:00 pm
```

4.3.16 Working with week control

index.php

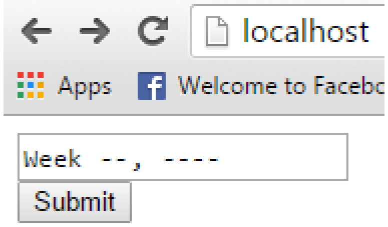
```
<!DOCTYPE html>
<html>
<body>
<form action="php_test.php" method="post">
<input type="week" id="myWeek" name="myWeek"><br>
<input type="submit">
</form>
</body>
</html>
```

php_test.php

```
----------------
<html>
<head> <Title> My first PHP program </title></head>
<body>
<?php
echo $_POST['myWeek'];
?> </body></html>
```

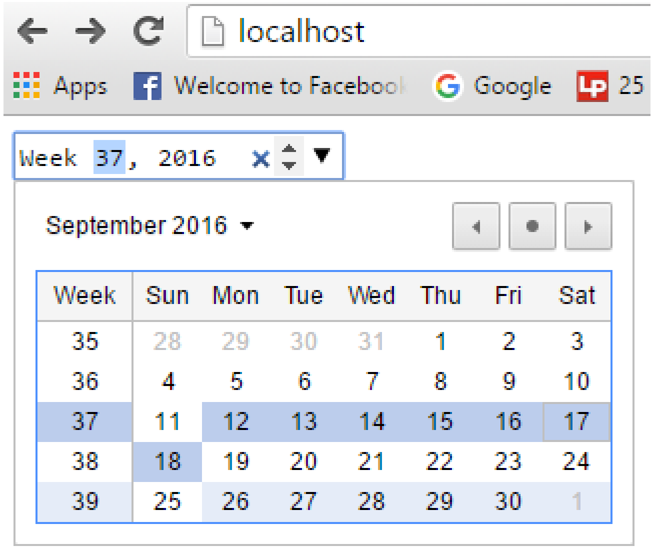
Output

Step-1



Week --, ----

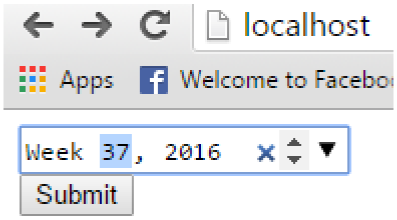
Step-2



Week 37, 2016

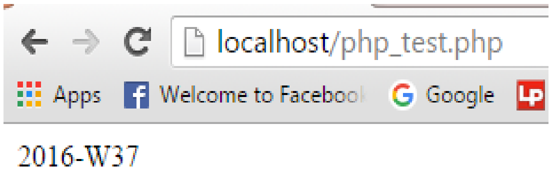
Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
35	28	29	30	31	1	2	3
36	4	5	6	7	8	9	10
37	11	12	13	14	15	16	17
38	18	19	20	21	22	23	24
39	25	26	27	28	29	30	1

Step-3



Week 37, 2016

Step-4 On clicking on the submit button we are directed to php_test.php as shown below.



```
2016-W37
```

4.3.17 Working with month control

The month control is used to enter the month name and the year.

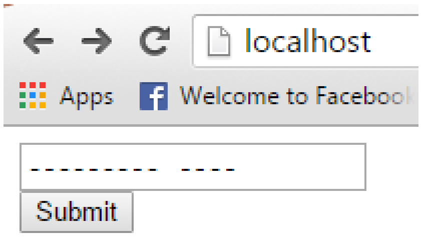
index.php

```
<!DOCTYPE html>

<html>

<body>

<form action="php_test.php" method="post">

<input type="month" id="mymonth" name="mymonth"><br>

<input type="submit">

</form>

</body>

</html>
```

php_test.php

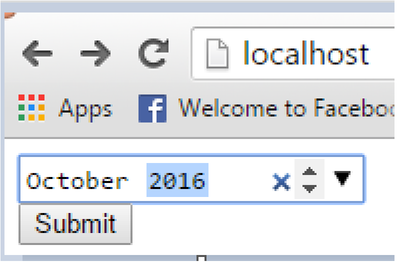
```
-----------------

<html>

<head> <Title> My first PHP program </title></head>

<body>

<?php

echo $_POST['mymonth'];

?> </body></html>
```

Step-1



```
--------- ----
```

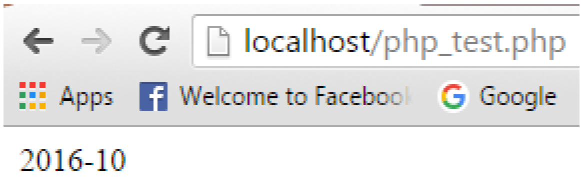
Step-2



```
October 2016
```

Step-3

On clicking the submit button we get the output as shown below.



```
2016-10
```

4.3.18 Working with email control

index.php

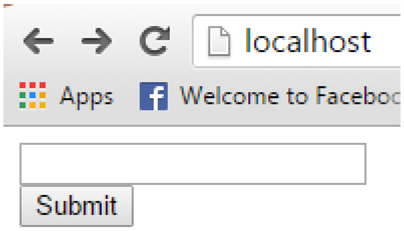
```
<!DOCTYPE html>
<html>
<body>
<form action="php_test.php" method="post">
<input type="email" id="myemail" name="myemail"><br>
<input type="submit">
</form>
</body>
</html>
```

php_test.php

```
-----------------
<html>
<head> <Title> My first PHP program </title></head>
<body>
<?php
echo $_POST['myemail'];
?> </body></html>
```

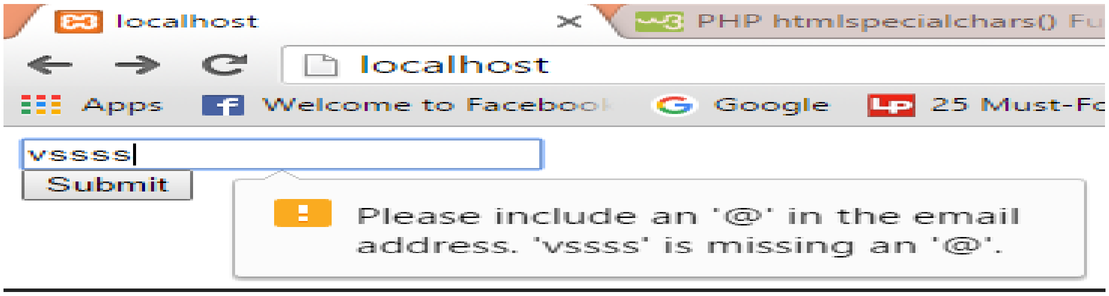
The email control is used to allow email entry only as shown below.

Step-1



Step-2

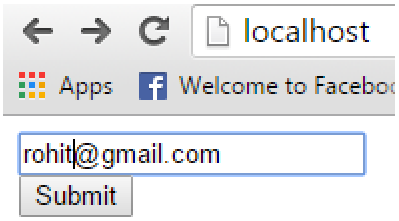
Below I entered a wrong email id and clicked the submit button and therefore got an warning message that email address is not proper as shown below



Please include an '@' in the email address. 'vssss' is missing an '@'.

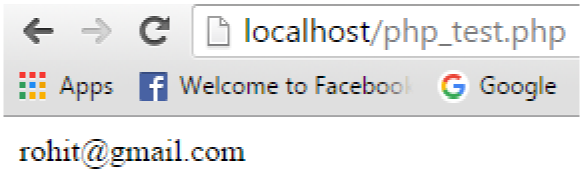
Step-3

Now typing the right email id when I click a submit button.



rohit@gmail.com

Step-4



```
rohit@gmail.com
```

4.3.19 Working with numbers

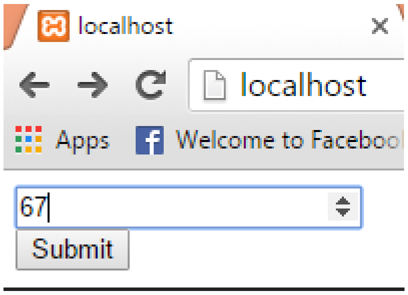
index.php

```
<!DOCTYPE html>
<html>
<body>
<form action="php_test.php" method="post">
<input type="number" id="mynumber" name="mynumber"><br>
<input type="submit">
</form>
</body>
</html>
```

php_test.php

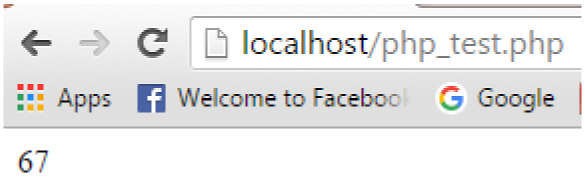
```
<html>
<head> <Title> My first PHP program </title></head>
<body>
<?php
echo $_POST['mynumber'];
?> </body></html>
```

Step-1



67

Step-2



```
67
```

4.3.20 Working with range

index.php

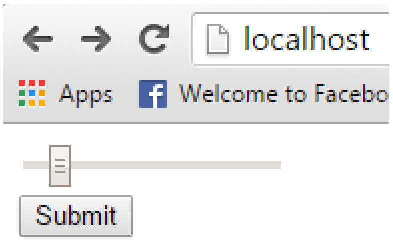
```
<!DOCTYPE html>
<html>
<body>
<form action="php_test.php" method="post">
<input type="range" id="myrange" name="myrange"><br>
<input type="submit">
</form></body></html>
```

php_test.php

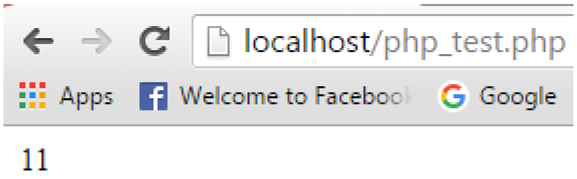
```
----------------
<html>
<head>
<Title> My first PHP program </title>
</head>
<body>
<?php
echo $_POST['myrange'];
?>
</body>
</html>
```

Output

Step-1



Step-2



```
11
```

4.3.21 Working with search

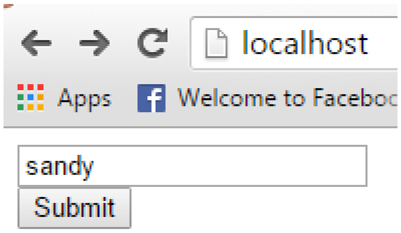
index.php

```
<!DOCTYPE html>
<html>
<body>
<form action="php_test.php" method="post">
<input type="search" id="mysearch" name="mysearch"><br>
<input type="submit">
</form>
</body>
</html>
```

php_test.php

```
<html>
<head> <Title> My first PHP program </title></head>
<body>
<?php
echo $_POST['mysearch'];
?> </body></html>
```

Step-1

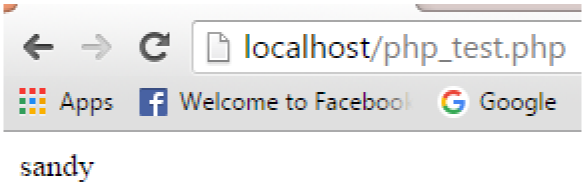


localhost

sandy

Step-2

Clicking on the submit button directs us to the php_test.php. Clicking on the blue cross clears the search box



localhost/php_test.php

sandy

4.3.22 Working with tel

tel control is used to enter only telephone numbers.

index.php

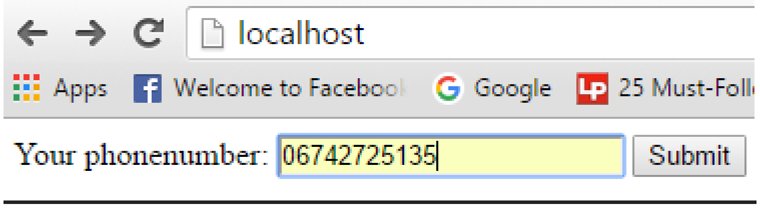
```
<!DOCTYPE html>
<html>
<body>
<form method="post" action="php_test.php">
        Your phonenumber: <input type="tel" id='tel1' name='tel1' required />
        <input type="submit" value="Submit" />
</form>
</body>
</html>
```

php_test.php

```
------------------
<html>
<head>
<Title> My first PHP program </title>
</head>
<body>
<?php
echo $_POST['tel1'];
?>
</body>
</html>
```

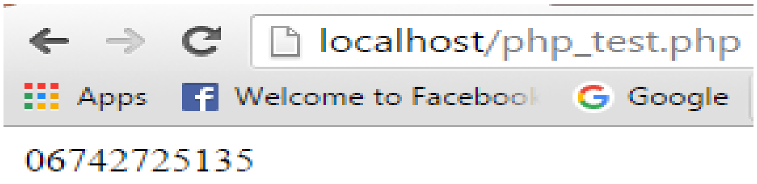
Output

Step-1



Your phonenumber: 06742725135

Step-2



06742725135

4.3.23 Working with url

index.php

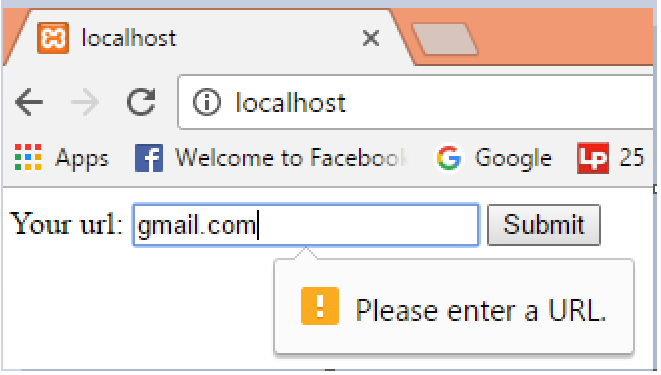
```
<!DOCTYPE html>
<html>
<body>
<form method="post" action="php_test.php">
        Your url: <input type="url" id='url1' name='url1' required />
        <input type="submit" value="Submit" />
</form></body></html>
```

php_test.php

```
----------------
<html>
<head><Title> My first PHP program </title></head>
<body>
<?php
echo $_POST['url1'];
?>
</body>
</html>
```

Output

Step-1



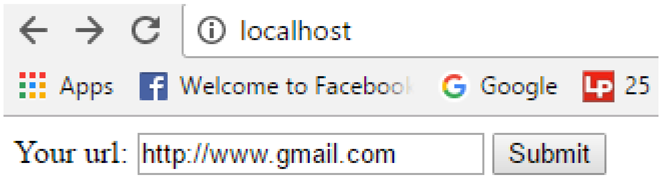
Your url: gmail.com

Please enter a URL.

When I enter gmail.com web page shows an error that I have to enter the url.

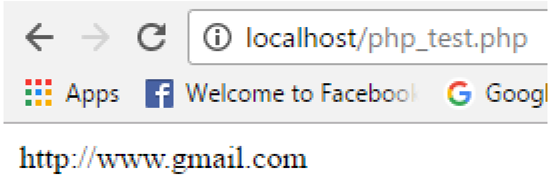
Step-2

Now I enter a valid url as shown below.



Your url: http://www.gmail.com

Now clicking on the submit button



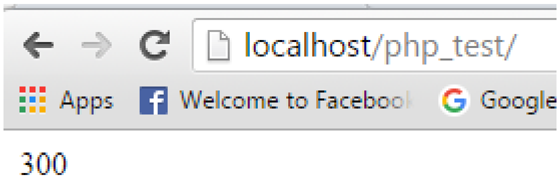
```
http://www.gmail.com
```

Working with the $Globals

The $Globals is a super global that is used to access the global variables.

```
<?php
$x = 100;
$y = 200;
function addition() {
   $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}
addition();
echo $z; ?>
```

Output



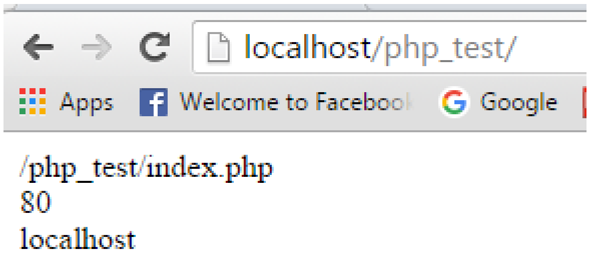
```
300
```

Working with the $_Server

The $Server is a Super Global that is used to hold information about the filename where the php script is being run, the port address of the apache server, server name. The $server is used for a number of purposes which is beyond the scope of this book.

```
<?php
echo ($_SERVER['PHP_SELF']);
echo("<br>");
echo($_SERVER['SERVER_PORT']);
echo("<br>");
echo $_SERVER['SERVER_NAME'];
?>
```

Output



```
/php_test/index.php
80
localhost
```

Working with the $_ENV

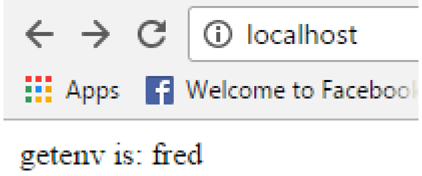
$_ENV is a superglobal that checks environment variables. Environment variables are given by the shell in which **PHP** is running, so they may change according to different shells.

The environment variable can be set but putenv command.

The environment variable can be retrieved by getenv command.

```
<?php

putenv("USER=fred");

print "getenv is: ".getenv("USER")."\n";

?>
```

Output



```
getenv is: fred
```

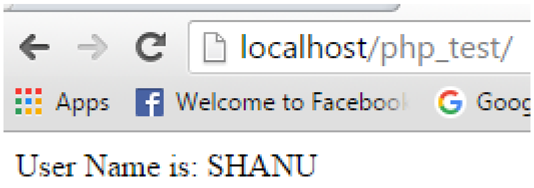
Working with the $_SESSION

The session variable is used as long as the user is logged into the system and working on the system. The session variable is lost once the user has logged out of the system on closed the browser due to some instance. The best example of session variable use is done in case of user name and password. The user logs into the system and the user name and user password is stored in the session variables user_name and passwd which we define in the php program. These session variables can be used throughout the application when the user navigates from one web page to the other. Why is the session variable used. The session variable is used to check what privileges the user can use basing on the security conditions. For example a stores in charge cannot have access to financial records as he/she is not given the privilege to financial data.

In index.php write the following code

```
<?php

$_SESSION["USER_NAME"]="SHANU";

print "User Name is: ".$_SESSION["USER_NAME"]."\n";

?>
```

Output



```
User Name is: SHANU
```

Working with the $_COOKIE

Cookie is a small file created by the user to identify the use. Cookie is used to store small information for a particular period of time. After the time span the cookie expires. Any number of cookies can be created, modified, enabled, disabled, deleted. A cookie is identified by

1. Cookie name
2. Cookie value
3. Time span of the Cookie

Create A Cookie

To create a Cookie we use the following syntax as shown below.

```
setcookie(name, value, expire, path, domain, secure, httponly);

Setcookie('sandy',1000, time()+86400*10,/,True,False);
```

name :- This parameter defines the name of the cookie.

value :- This parameter defines the value of the cookie.

expire :- This parameter defines when the cookie will expire. The above cookie will expire after 10 days. As 1 day = 86400 seconds and time()+86400*10 sets the cookie to 10 days from the current time.

path (optional) :- This option is used to set the path. If / is written over here it means that cookie is available for the entire domain or website that we create like abc.com. If /process is written it means that the cookie is available for the folder /process and is accessible by all the sub folders of the /process. If this is not defined it means that it is accessible by the current folder.

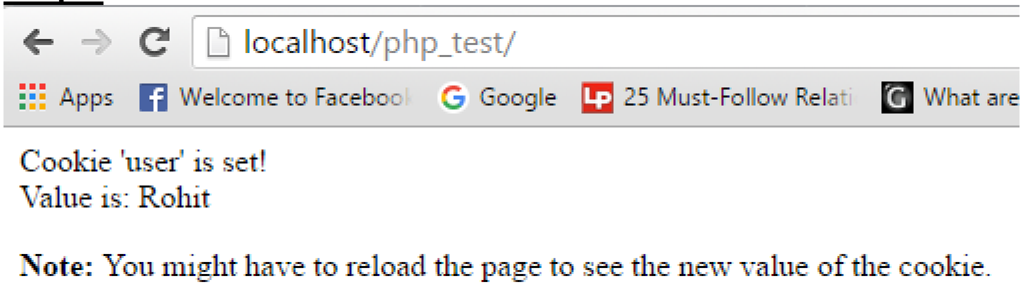
Secure(optional) :- This option if set to true accepts only https secure connection. If set to false data can be accessed by http and https. If this is set to blank it means false

Httponly (optional) :- If set to true it can be accessed by http protocol only. default is false

Example :-

```
<!DOCTYPE html>

<?php

$cookie_name = "user";

$cookie_value = "Rohit";

setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");

?>

<html>

<body>

<?php

if(!isset($_COOKIE[$cookie_name])) {

    echo "Cookie named '" . $cookie_name . "' is not set!";

} else {

    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];

}

?>

<p><strong>Note:</strong> You might have to reload the page to see the new
value of the cookie.</p>

</body>

</html>
```

Output



Cookie 'user' is set!
Value is: Rohit

Note: You might have to reload the page to see the new value of the cookie.

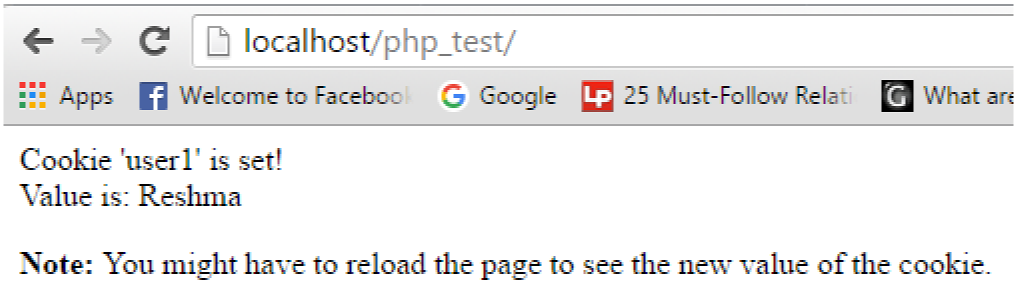
Modify A Cookie

To modify a Cookie we change the name and the value of the Cookie as shown below.

```
<!DOCTYPE html>
<?php
$cookie_name = "user1";
$cookie_value = "Reshma";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
?>
<html>
<body>
<?php
if(!isset($_COOKIE[$cookie_name]))
{
    echo "Cookie named '" . $cookie_name . "' is not set!";
}
else
{
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
```

```
?>
<p><strong>Note:</strong> You might have to reload the page to see the new
value of the cookie.</p>
</body> </html>
```

Output



```
Cookie 'user1' is set!
Value is: Reshma

Note: You might have to reload the page to see the new value of the cookie.
```

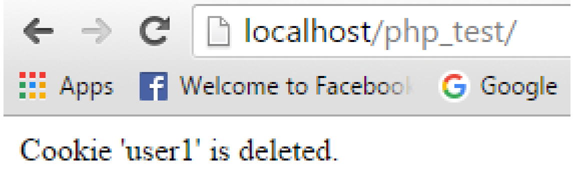
Delete A Cookie

Delete a cookie option is used to remove the cookie from the domain or the folder.

To delete a cookie set the time value to a time which was 1 hour before. This means that set the time to 2:30 PM if it is now 3:30 PM. Example

```
<!DOCTYPE html>
<?php
// set the expiration date to one hour ago
setcookie("user1", "", time() - 3600);
?>
<html>
<body>
<?php
echo "Cookie 'user1' is deleted.";
?>
</body>
</html>
```

Output



```
Cookie 'user1' is deleted.
```

Check your progress -2

Q.1 what are the super global used in php

A._____________ ____

____

____

4.4 Form Validation

This is a process to check the form and check the values entered in the form are valid before they are sent for further processing by php. Form validation is the most important part of the PHP technology. We shall be covering the following.

1. Validating Text Box.
2. Validating Text Area
3. Validating Drop down boxes
4. Validating Radio buttons
5. Validating checkboxes.

4.4.1 Validating Text Fields

Text field validation is done to check

1. Text fields do not contain blank fields in case names are being entered
2. Text files contain valid email address in case email address is being entered is a valid email.

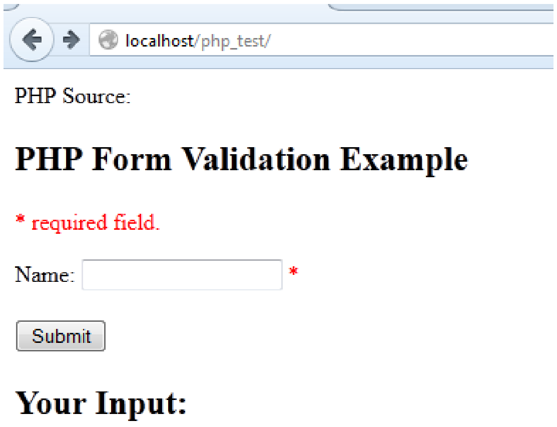
4.4.1.1 Validating Text Fields to prevent blank fields

Check
http://www.w3schools.com/php/showphp.asp?filename=demo_form_validation_required

PHP Source:

```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<?php
// define variables and set to empty values
$nameErr ="";
$name ="";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  if (empty($_POST["name"])) {
    echo("<p><span class='error'>* required field.</span></p>");
    $nameErr = "Name is required";
  }
else
{
    $name = test_input($_POST["name"]);
  }
  }
function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
?>
<h2>PHP Form Validation Example</h2>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']);?>">
Name: <input type="text" name="name" value=<?php echo($name); ?>>
<?php
  if($name=='')
  {
        echo('<span class="error">* <?php echo $nameErr;?></span>');
  }
  ?>
  <br><br>
  <input type="submit" name="submit" value="Submit">
</form>
<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
?>
</body>
</html>
```

Output



localhost/php_test/

PHP Source:

PHP Form Validation Example

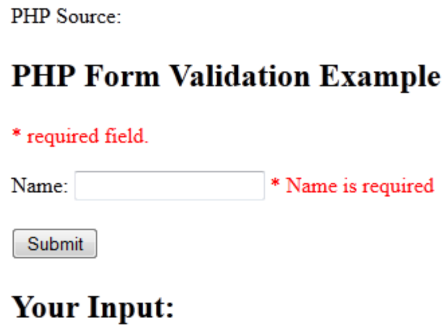
* required field.

Name: *

Submit

Your Input:

Upon clicking the submit button without the text field being entered



PHP Source:

PHP Form Validation Example

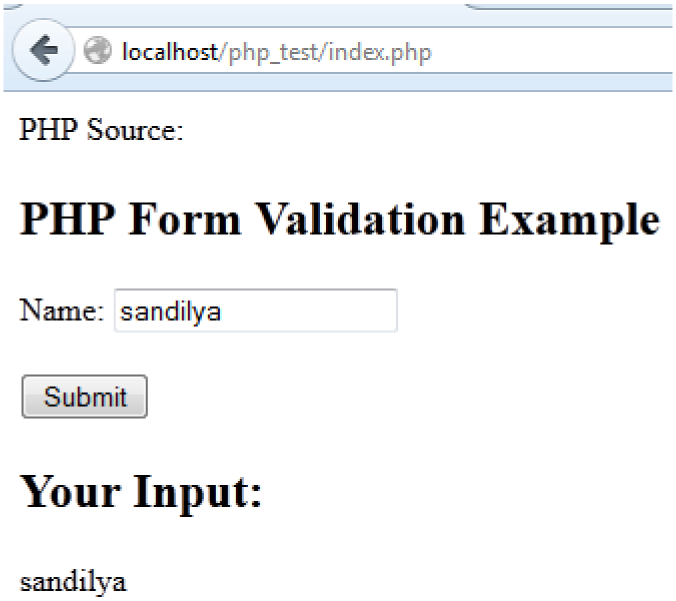
* required field.

Name: * Name is required

Submit

Your Input:

Upon clicking the submit button with the text field being entered.



PHP Source:

PHP Form Validation Example

Name: sandilya

Your Input:

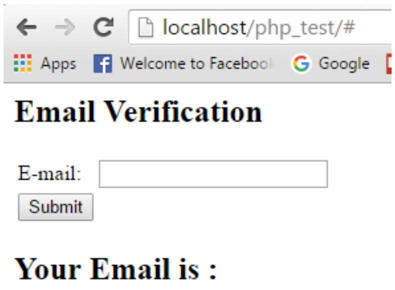
sandilya

4.4.1.2 Validating Text Fields to send valid email address

```
<html>
<head>
<style>
.error {color: #FF0000;}
 </style>
 </head>
 <body>
 <?php
 // define variables and set to empty values
  $nameErr = $emailErr = $genderErr = $websiteErr = "";
  $name = $email = $gender = $comment = $website = "";

   if ($_SERVER["REQUEST_METHOD"] == "POST")
 {
  if (empty($_POST["email"]))
{
   $emailErr = "Email is required";
    }
    else
    {
    $email = test_input_email($_POST["email"]);

    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL))
    {
     $emailErr = "Invalid email format";
     }
     }
    }
      function test_input_email($data)
          {
          $data = trim($data);
          $data = stripslashes($data);
          $data = htmlspecialchars($data);
          return $data;
          }
         ?>

     <h2>Email Verification</h2>

     <?php
     if ($emailErr)
          {
      $rr = "<p><span class = 'error'>* required field.</span></p>";
            echo($rr);
            }
            ?>
 <form method = "post" action = "#">
        <table>
          <tr>
            <td>E-mail: </td>
            <td>
            <input type = "text" name = "email" value=<?php echo($email); ?> >
             <?php
                     if ($emailErr)
                              {
                              $ss = "<span class = 'error'>* <?php echo
                                $emailErr;?></span>";
                              echo($ss);
                                 }
                    ?>
            </td>
          </tr>
            <tr>
            <td>
            <input type = "submit" name = "submit" value = "Submit">
             </td>
             </tr>
             </table>
             </form>
   <?php
             if ($emailErr)
           {
            }
          else
          {
       echo "<h2>Your Email is :</h2>";
      echo $email;
       echo "<br>";
          }
           ?>
    </body>
</html>
```

Output

Step-1

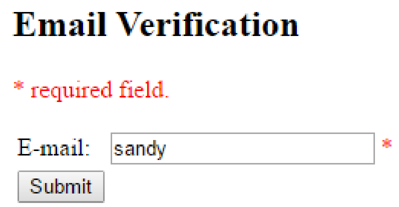


Email Verification

E-mail:

Your Email is :

Step-2

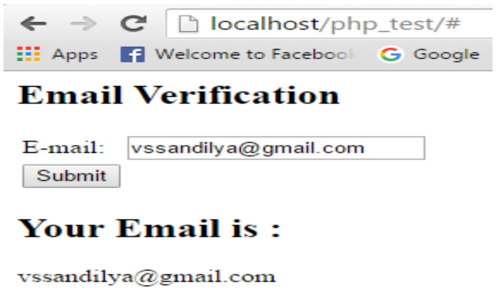


Email Verification

* required field.

E-mail: sandy *

Step-3



Email Verification

E-mail: vssandilya@gmail.com

Your Email is :

vssandilya@gmail.com

4.4.2 Validating TextArea

The program defined below is used to validate the Text Area so that blank fields are

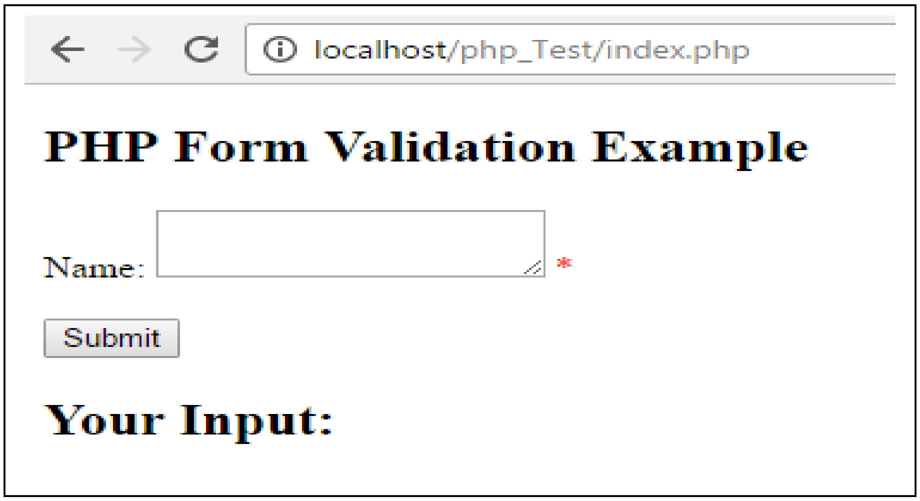
not allowed.

```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<?php
// define variables and set to empty values
$nameErr ="";
$name ="";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  if (empty($_POST["name"])) {
    echo("<p><span class='error'>* required field.</span></p>");
    $nameErr = "Name is required";
  }

else
{
    $name = test_input($_POST["name"]);
  }
  }
function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
?>
<h2>PHP Form Validation Example</h2>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']);?>">
Name: <textarea name="name" value=<?php echo($name); ?>></textarea>
<?php
  if($name=='')
  {
        echo('<span class="error">* <?php echo $nameErr;?></span>');
  }
  ?>
  <br><br>
  <input type="submit" name="submit" value="Submit">
```

```
</form>
<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
?>
</body>
</html>
```

Output

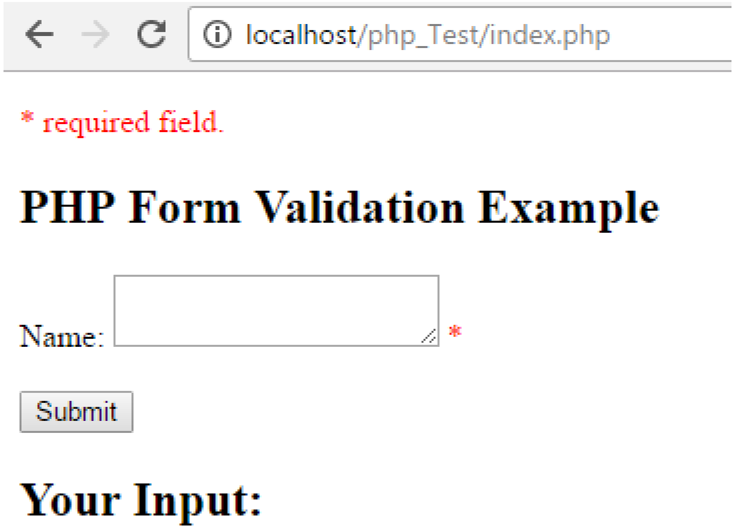
Step-1



```
PHP Form Validation Example

Name:  *

Your Input:
```

Step-2

Upon clicking the submit button without entering anything in the text area



```
* required field.

PHP Form Validation Example

Name:  *

Your Input:
```

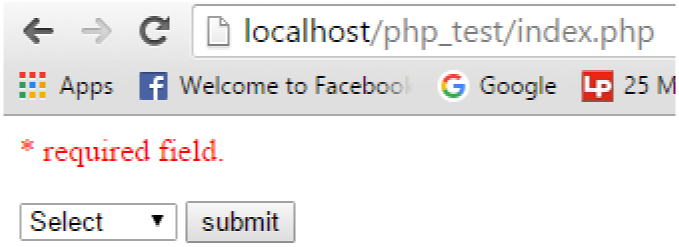
4.4.3 Validating Drop Down Boxes

This option is used to validate the Drop Down Box contain a valid field

```
 <html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<?php
 (isset($_POST["company"])) ? $company = $_POST["company"] : $company=0;
if($company==0)
{
echo("<p><span class='error'>* required field.</span></p>");
}
 ?>
 <form method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']);?>">
<select id="company" name="company">
<option <?php if ($company == 0 ) echo 'selected' ; ?> value="0">Select</option>
<option <?php if ($company == 1 ) echo 'selected' ; ?> value="1">Apple</option>
<option <?php if ($company == 2 ) echo 'selected' ; ?> value="2">Samsung</option>
<option <?php if ($company == 3 ) echo 'selected' ; ?> value="3">HTC</option>
</select>
<input type='submit' name='submit' value="submit">
</form>
</body>
</html>
```

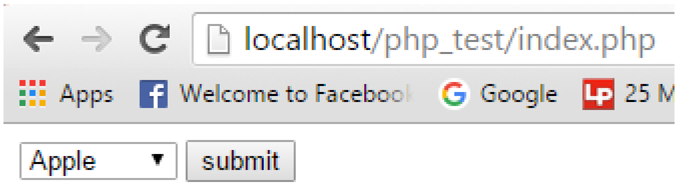
Output

Step-1



```
* required field.
```

Step-2

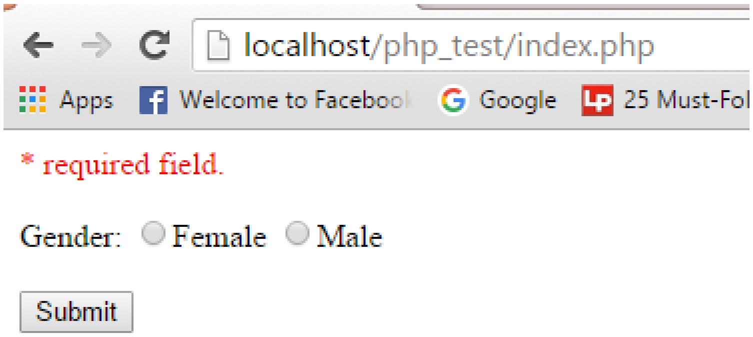


4.4.4 Validating Radio Buttons

```
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<?php

(isset($_POST["gender"])) ? $gender = $_POST["gender"] : $gender="";
if($gender==")
{
echo("<p><span class='error'>* required field.</span></p>");
}
 ?>

<form method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']);?>">
Gender:
  <input type="radio" name="gender" value="female" <?php if ($gender=='female') { echo
('checked=checked');} ?> >Female
  <input type="radio" name="gender" value="male"   <?php if ($gender=='male') { echo
('checked=checked');} ?>>Male
  <br><br>
  <input type="submit" name="submit" value="Submit">
</form>
</body>
</html>
```

Output

Step-1

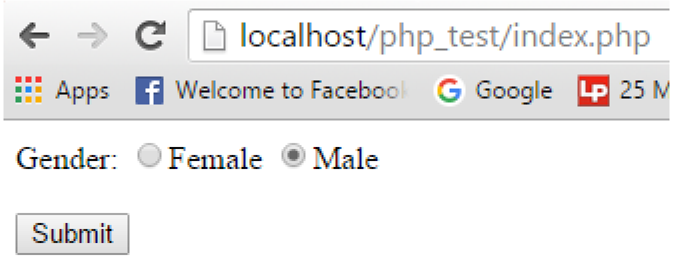


* required field.

Gender: ○ Female ○ Male

Submit

Step-2



Gender: ○ Female ● Male

Submit

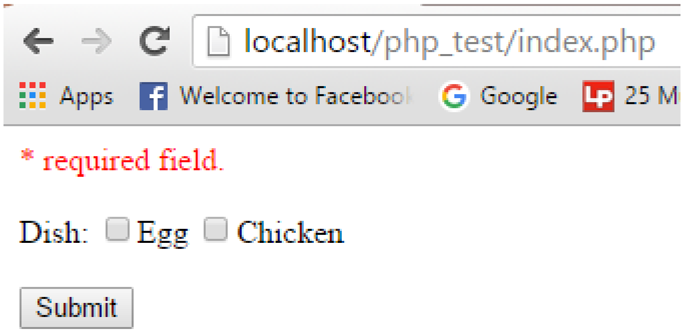
4.4.5 Validating Check Boxes

```
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>
<?php
 (isset($_POST["egg"])) ? $egg = $_POST["egg"] : $egg="";
(isset($_POST["chicken"])) ? $chicken = $_POST["chicken"] : $chicken="";
if(($egg==")&&($chicken=="))
{
echo("<p><span class='error'>* required field.</span></p>");
}
 ?>
 <form method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']);?>">
Dish:
  <input type = "checkbox"  name="egg" value="egg" <?php if ($egg=='egg') { echo
('checked=checked');} ?> >Egg
```

```
  <input type="checkbox" name="chicken" value="chicken"   <?php if
($chicken=='chicken') { echo ('checked=checked');} ?>>Chicken
  <br><br>
  <input type="submit" name="submit" value="Submit">
</form>
</body>
</html>
```

Output

Step-1

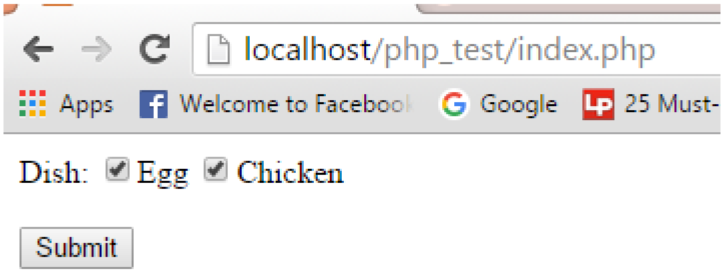


* required field.

Dish: □ Egg □ Chicken

Submit

Step-2



Dish: ☑ Egg ☑ Chicken

Submit

Description

We shall be discussing about the code what we have written. First thing we shall do is dissect the code into very small parts.

```
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
```

In the above part of the code we have defined a CSS class named error by the colour red. We have defined it as color:#FF0000.

```
<body>
<?php
 (isset($_POST["egg"])) ? $egg = $_POST["egg"] : $egg="";
```

```
(isset($_POST["chicken"])) ? $chicken = $_POST["chicken"] : $chicken="";
if(($egg=='')&&($chicken==''))
{
echo("<p><span class='error'>* required field.</span></p>");
}
 ?>
```

In the above code we have defined

```
isset($_POST["egg"])) ? $egg = $_POST["egg"] : $egg=""
```

which means if we receive some value by $_POST["egg"] then define a variable $egg and assign the value by $_POST["egg"] else set the $egg to “”

Now we come across another part of the code as shown below

```
if(($egg=='')&&($chicken==''))
{
echo("<p><span class='error'>* required field.</span></p>");
}
```

The above statement is a conditional statement which means that if $egg contains a blank and $chicken contains a blank then it would show *required field in red colour.

```
<form method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']);?>">
```

The above code shows that the method is post

and

```
 action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']);?>">
```

calls the form by itself.

```
<input type = "checkbox"  name="egg" value="egg" <?php if ($egg=='egg') { echo
('checked=checked');} ?> >Egg
```

The above code creates a checkbox and sets it a name as egg. The value of the checkbox is egg. If ($egg==’egg’) checks if the variable $egg is comprised of ‘egg ’ and if so then echo ('checked=checked'); sets the checkbox to checked.

Check your progress -3

Q.1 What is Form Validation? What are the various types of Form Validation

done at the server end?

A.____

____

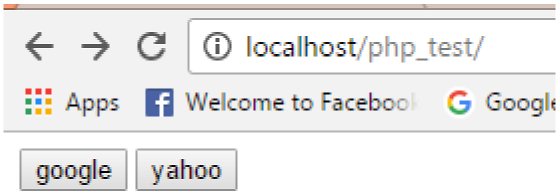
____

4.5 Form Navigation in PHP

```
<?php
if(isset($_POST['google']))
{
header('Location: http://google.com');
exit;
}
if(isset($_POST['yahoo']))
{
header('Location: http://yahoo.com');
exit;
}
?>
<html>
<head></head>
<body>
<form action="index.php" method="post">
<input type='submit' name='google' value='google' />
<input type='submit' name='yahoo' value ='yahoo' />
</form>
</body>
</html>
```

Output

Step-1

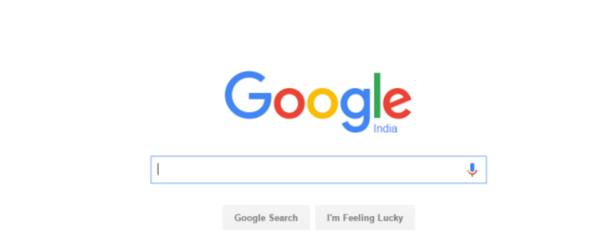


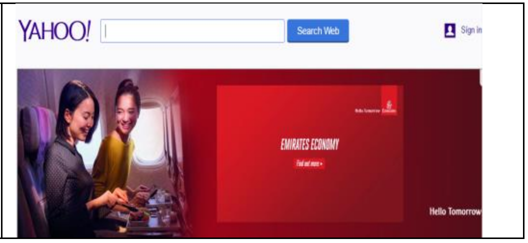
Step-2

Upon clicking on google we are navigated to google.com

Upon clicking on yahoo we are navigated to yahoo.com.

as shown below.





Description

In the above example we have defined two submit buttons. Upon clicking these two submit buttons we are directed to the websites google.com and yahoo.com.

4.6 Let us sum up

In this unit we have learnt that data is entered into the form using the form controls. The form controls are 23 in number. These are Textbox, Text Area, Password, Check boxes, Radio buttons, Drop down box, file, Submit, Reset, Button, Color, Date, Date time, Date time-local, Time, Week, Month, Email, Number, Range, Search, Tel, Url. All these controls are used for various purposes. We use Textbox controls to enter text values such as name of the person. The password control is used to enter passwords and other confidential matters such as atm pin number etc. The check boxes are used to enter multiple values such as vegetables that you like. The radio buttons are used to select one value among multiple values such as gender of the person can be male or female. The drop down box is used to select one item among multiple items that are visible in a drop box such country, state and city where country is a drop down, state is a drop down and city is a drop down. To upload images and other files from the computer that we are using the file control. To submit the form the submit button is used. The reset button is used to clear the form so that fresh data can be entered into the form. The button control is used for clicking various kinds of buttons. The color control is used to select a colour among multiple colours that are seen. The color control is used to set the background colour of the form. The Date control is used to enter dates such as date of birth, The date time control is used to enter date and the time such as date of birth and time of birth but this control is confined to Safari. The date time-local control is also used to enter the data and time control. The time control is used to enter the time values only. The week control is used to enter the week of the year. The month control is used to select the month and the year. The email control is used to enter the emails only. The numbers control is used enter numbers only. The range control is used to a value between two ranges like 1 to 100.The search control is used to enter a value to be searched. The tel control is used to enter telephone numbers. The url control is used to enter the url. We have learnt about super globals used in form management which are $_POST,$_GET,$_FILES, $_REQUEST, $GLOBALS,S_SERVER,$_ENV,$_COOKIE and $_SESSION. $_POST and $_GET are the methods used to transfer data from form to server. The $_GET,$_POST and $_REQUEST are used to receive data at the server level. The $Globals is a super global that is used to access the global variables. The $Server is a Super Global that is used to hold information about the filename where the php script is being run, the port address of the apache server, server name. $**_ENV** is a superglobal that checks environment variables. Environment variables are given by the shell in which **PHP** is running, so they may change according to different shells.The environment variable can be set but putenv command.The environment variable can be retrieved by getenv command.. The session variable is used as long as the user is logged into the system and working on the system. The session variable is lost once the user has logged out of the system on closed the browser due to some instance. The best example of session variable use is done in case of user name and password. The user logs into the system and the user name and user password is stored in the session variables user_name and passwd which we define in the php program. Cookie is a small file created by the user to identify the use. Cookie is used to store small information for a particular period of time. After the time span the cookie expires. Any number of cookies can be created, modified, enabled, disabled, deleted. A cookie is identified by Cookie name, Cookie value ,Time span of the Cookie. We have learnt about server form validation using PHP. This is done using validating text fields, drop down boxes, radio buttons, check boxes. We have learn to enter valid emails and not allow blank text fields. The drop down boxes are validated to enter only valid fields. Validating radio buttons is used to select any one of the radio buttons. Validating check boxes is used select any one check box. We have learnt about navigating to websites upon clicking the google and yahoo.

4.7 References

1. W3schools.com
2. Google.com.

4.8 Check your progress –possible answers

Check your progress –possible answers -1

A.1 A form is a window or screen that contains numerous fields,

or spaces to enter or view or edit or delete data. Each field holds a field label

so that any user who views the form gets an idea of its contents.These fields

are called Form controls. The Form controls are 23 in number and they are

Textbox, Text area, Password, Check boxes, Radio buttons, Drop down box,file, Submit, Reset, Button, Color, Date, Date time, Date time-local, Time, Week, Month, Email, Number, Range, Search, Tel, Url.

A.2 The difference between $_GET and $_POST are

Sl.No	**$_GET Method**	**$_POST Method**
1	Data sent via $_GET method is visible across the URL	Data sent via $_POST method is not visible across the URL
2	It is an array of variables passed to the current script via URL parameters	It is an array of variables passed to the current script via HTTP post parameters
3	It is confined to 2000 characters through this method	Infinite amount of data is transmitted through this method
4	Should not be used when passing sensitive information such as user name and password	Should be used when passing sensitive information such as user name and password

Check your progress –possible answers -2

Super globals those that are used are 9 in number and they are

1. $_POST
2. $_GET
3. $_FILES
4. $_REQUEST
5. $GLOBALS
6. S_SERVER
7. $_ENV

8. $_COOKIE
9. $_SESSION

Check your progress –possible answers -3

Form validation is the process of checking the data entered by the end user by the computer that the data entered is correct before the computer processes the data. Form validation at the server end are classified into 5 types and they are validating text box, text area , drop down boxes, radio buttons, check boxes.