

# Form Handling in PHP

## Session 2



# Objectives

- ◆ *Explain the use of the GET method*
- ◆ *Explain the use of the POST method*
- ◆ *Retrieve data from forms using the Form methods*
- ◆ *Explain the use of hidden fields*

- ◆ Form data is passed to the Web server using following methods:
  - ◆ GET
  - ◆ POST
- ◆ Web server
  - ◆ Accepts the information
  - ◆ Processes the application data
  - ◆ Stores it to the database

- ◆ It is a Web page containing fields
- ◆ It is used by users to enter information
- ◆ It passes entered data from a client to a server

**Contact us!**

Name \* :

Website \* :

E-Mail \* :

Phone\_Number \* :

Subject :

Message \* :

Captcha:  »

- ◆ Steps for handling HTML forms and process information are as follows:
  - ◆ User enters information in an HTML form and sends it to the Web server
  - ◆ Web server passes the information to the PHP script engine for:
    - ◆ Processing the information
    - ◆ Sending output back to the Web browser

- ◆ HTML <FORM> tag is:
  - ◆ Used to create HTML form
  - ◆ Included within the <FORM> and </FORM> tag
- ◆ Attributes of an HTML form tag are:
  - ◆ **Action** - defines URI where the form data is sent after it has been submitted
  - ◆ **Method** - defines protocols that are used to submit the form data set
- ◆ Method protocols are of two types:
  - ◆ GET
  - ◆ POST

- ◆ GET method
  - ◆ Directs the Web browser to send the encoded user information to the processing agent
  - ◆ Appends the encoded information at the end of the URL by a question mark (?) which separates URL and form information
- ◆ The form data sent in the URL is a stream of name/value pair separated by ampersand (&)

- ◆ An input variable will have following structure:
  - ◆ `Name=value`
- ◆ Multiple input variables are grouped as follows:
  - ◆ `Name1=value1&Name2=value2&Name3=value3`
- ◆ The following example shows the multiple name/value pair separated by the & sign:
  - ◆ `Name=john&age=18`
- ◆ The query string is appended with the following URL:
  - ◆ `http://www.information.com/text.php?Name=john&age=18`



- ◆ The restrictions of GET method are as follows:
  - ◆ Form data set values are restricted to American Standard Code for Information Interchange (ASCII) characters
  - ◆ Amount of information transferred is limited
  - ◆ Length of the query string is restricted to 255 characters

- ◆ **POST method**
  - ◆ Directs the Web browser to send all the user information to the processing agent
  - ◆ Uses message body of an HTTP request to send the information
  - ◆ Has capacity to transmit more information as:
    - ◆ No physical limit on the amount of information passed in the HTTP request message body
  - ◆ Uses variables to pass form information
- ◆ **The drawback of POST method is as follows:**
  - ◆ Information sent is not encrypted, so hackers can easily access it

# Difference in the GET and POST Method

- ◆ GET and POST methods work almost identically

Table lists the difference between the GET and POST method

GET	POST
Encodes the form data as a stream of name/value pairs and appends it in the URL making it visible in the browser	Sends the encoded form data through the body of an HTTP request
Form submissions can be bookmarked	Form submissions cannot be bookmarked
Is less secure as the information is displayed in the URL	Is more secure for transmitting passwords and other sensitive information, as the form data is embedded in the body of the HTTP request
The amount of data that can be sent is limited depending on the browser used	Does not have size limitations
This method is mainly used for displaying data such as searching, sorting, and pagination	This method is mainly used for data manipulation such as adding and editing data

- ◆ Retrieving data from an HTML form using the GET method

## Syntax

```
$varname = $_GET["variable"];
```

Where,

- ◆ **varname** - specifies the name of the variable in which data is to be stored
- ◆ **\$\_GET["variable"]** - specifies the name of the input variable

- ◆ Steps for creating an HTML form to retrieve data using the `GET` method are as follows:
  1. Open a new file in the **gedit** text editor
  2. Enter the code and save the file as **Details.html** in the `/usr/local/apache2/htdocs` directory

## Snippet

```
<HTML>
<BODY>
<B>ENTER YOUR PERSONAL DETAILS</B>
<FORM METHOD=GET ACTION="Details.php">
FIRST NAME:
<INPUT NAME="n1text" TYPE="TEXT"><BR>
LAST NAME:
<INPUT NAME="n2text" TYPE="TEXT"><BR>
ADDRESS:
<TEXTAREA NAME="n3text" ROWS=1, COLUMNS=1000></TEXTAREA>
<BR>
CONTACT NO:
<INPUT NAME="n4text" TYPE="TEXT"><br>
<INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="SUBMIT">
</FORM>
</BODY>
</HTML>
```

- ◆ Steps for creating PHP script to retrieve and process the data entered in the HTML form are as follows:
  1. Open a new file in the **gedit** text editor
  2. Enter the code and save the file as **Details.php** in `/usr/local/apache2/htdocs` directory

## Snippet

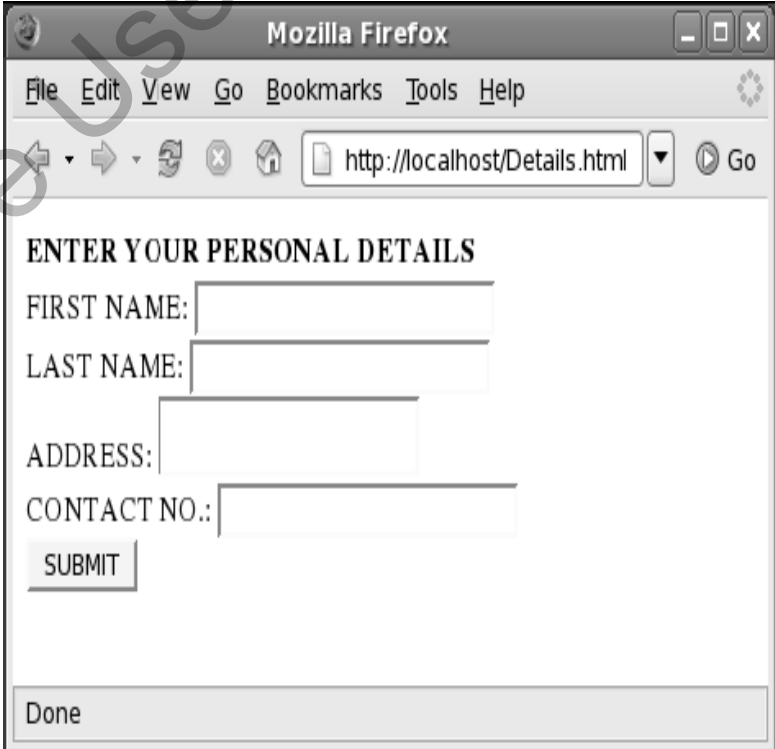
```
<?php
error_reporting(0);
$A = $_GET["n1text"];

$B = $_GET["n2text"];
$C = $_GET["n3text"];
$D = $_GET["n4text"];
echo "YOUR PERSONAL DETAILS";
echo "<BR><BR>";
echo "FIRST NAME: $A <BR>";
echo "LAST NAME: $B <BR>";
echo "ADDRESS: $C <BR>";
echo "CONTACT NO.: $D <BR>";
?>
```



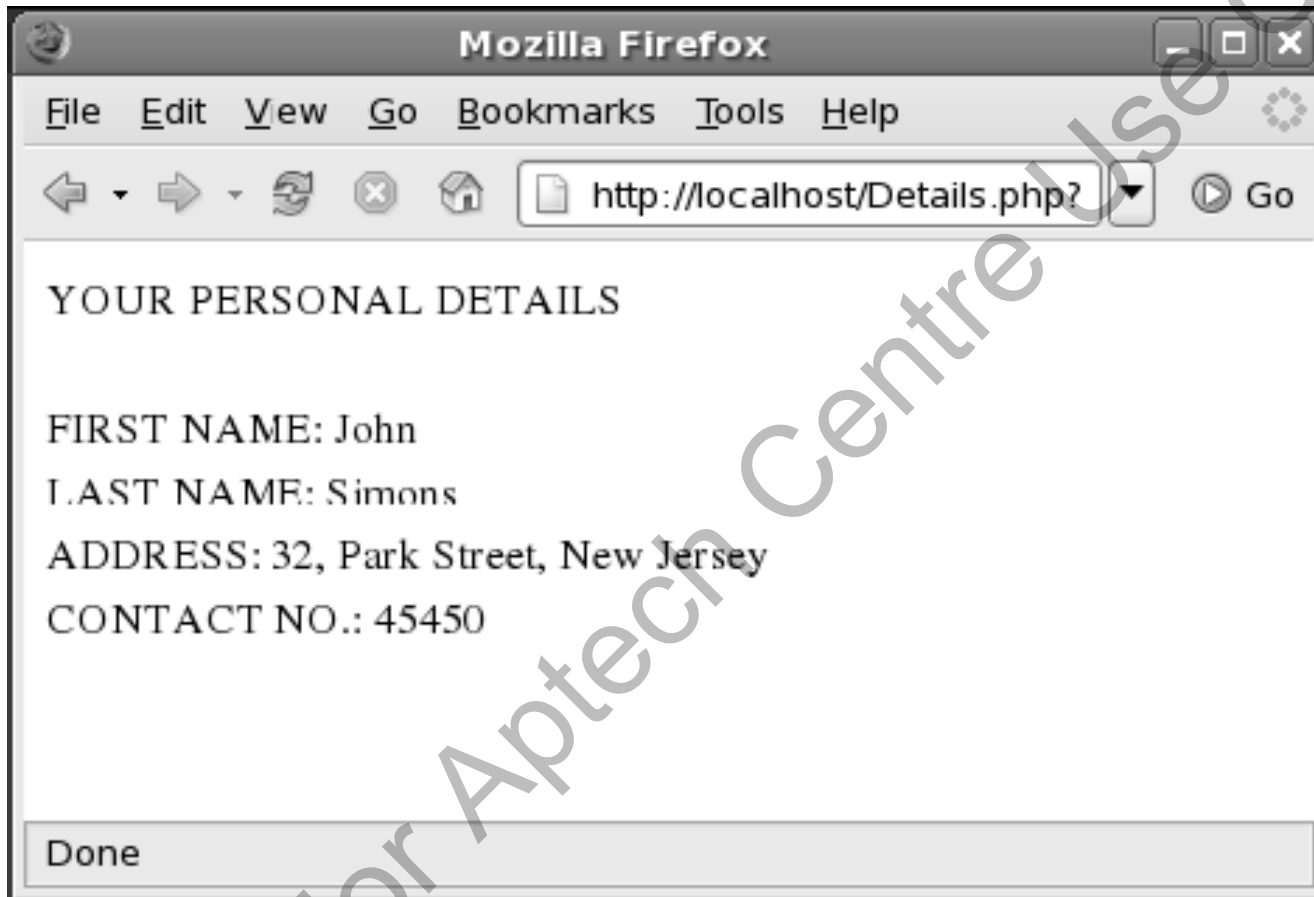
- ◆ Steps for displaying **details.html** page are as follows:

1. Open the Mozilla Firefox Web browser
2. Enter `http://localhost/Details.html` in the Address bar and press **Enter**
3. Enter **John** in the FIRST NAME box
4. Enter **Simons** in the LAST NAME box
5. Enter **32, Park Street, New Jersey** in the ADDRESS box
6. Enter **45450** in the CONTACT NO. box
7. Click **SUBMIT**



The screenshot shows a Mozilla Firefox browser window with the address bar displaying `http://localhost/Details.html`. The page content is a form titled "ENTER YOUR PERSONAL DETAILS". The form contains four text input fields: "FIRST NAME:", "LAST NAME:", "ADDRESS:", and "CONTACT NO.:". Below these fields is a "SUBMIT" button. The status bar at the bottom of the browser window shows "Done".

Displays the following output:



- ◆ Retrieving data from an HTML form using the POST method

## Syntax

```
$varname = $_POST["variable"];
```

Where,

- ◆ **varname** - specifies the name of the variable in which the data is to be stored
- ◆ **\$\_POST["variable"]** - specifies the name of the input variable

- ◆ Steps for creating an HTML form to retrieve data using the `POST` method are as follows:
  1. Open a new file in the **gedit** text editor
  2. Enter the code and save the file as **EMP\_DETAILS.html** in the `/usr/local/apache2/htdocs` directory

## Snippet

```
<HTML>
<HEAD>
<TITLE>Employee Details</TITLE>
</HEAD>
<BODY>
<H4>Enter your details</H4>
<FORM METHOD=POST ACTION="EMP_DETAILS.php">
<TABLE>
<TR>
<TD>Employee ID</TD>
<TD><INPUT TYPE="text" NAME="empid"></TD>
</TR>
<TR>
<TD>Name</TD>
<TD><INPUT TYPE="text" NAME="Name"></TD>
</TR>
<TR>
```

## Snippet

```
<TD>Department</TD>
<TD>
<INPUT TYPE="radio" NAME="dept" VALUE="Finance">Finance
<INPUT TYPE="radio" NAME="dept" VALUE="Marketing">Marketing
<INPUT TYPE="radio" NAME="dept" VALUE="IT">IT
</TD>
</TR>
<TR>
<TD>Email</TD>
<TD><INPUT TYPE="text" NAME="email"></TD>
</TR>
</TABLE>
<BR>
<TD><INPUT TYPE="submit" VALUE="SUBMIT"></TD>
</FORM>
</BODY>
</HTML>
```

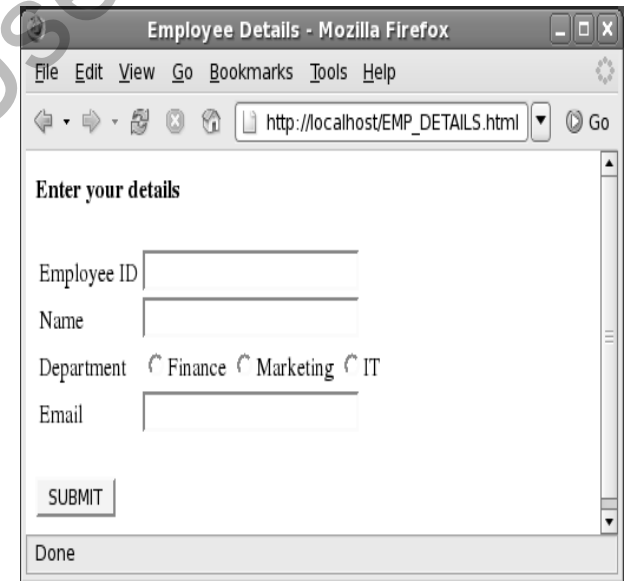
- ◆ Steps for creating a PHP script to retrieve and process the data entered in the HTML form are as follows:
  1. Open a new file in the **gedit** text editor
  2. Enter the code and save the file as **EMP\_DETAILS.php** in the `/usr/local/apache2/htdocs` directory

## Snippet

```
<?php
error_reporting(0);
$A=$_POST["empid"];
$B=$_POST["Name"];
$C=$_POST["dept"];
$D=$_POST["email"];
echo "YOUR PERSONAL DETAILS";
echo "<BR><BR>";
echo "EMPID: $A <BR>";
echo "NAME: $B <BR>";
echo "DEPARTMENT NAME: $C <BR>";
echo "EMAIL: $D <BR>";
?>
```

- ◆ Steps for displaying **EMP\_DETAILS.html** page are as follows:

1. Open the Mozilla Firefox Web browser
2. Enter `http://localhost/EMP_DETAILS.html` in the Address bar and press **Enter**
3. Enter **A05** in the Employee ID textbox
4. Enter **Peter Taylor** in the Name box
5. Select **IT** as the Department
6. Enter **peterA05@finfotech.com** in the Email box
7. Click **SUBMIT**



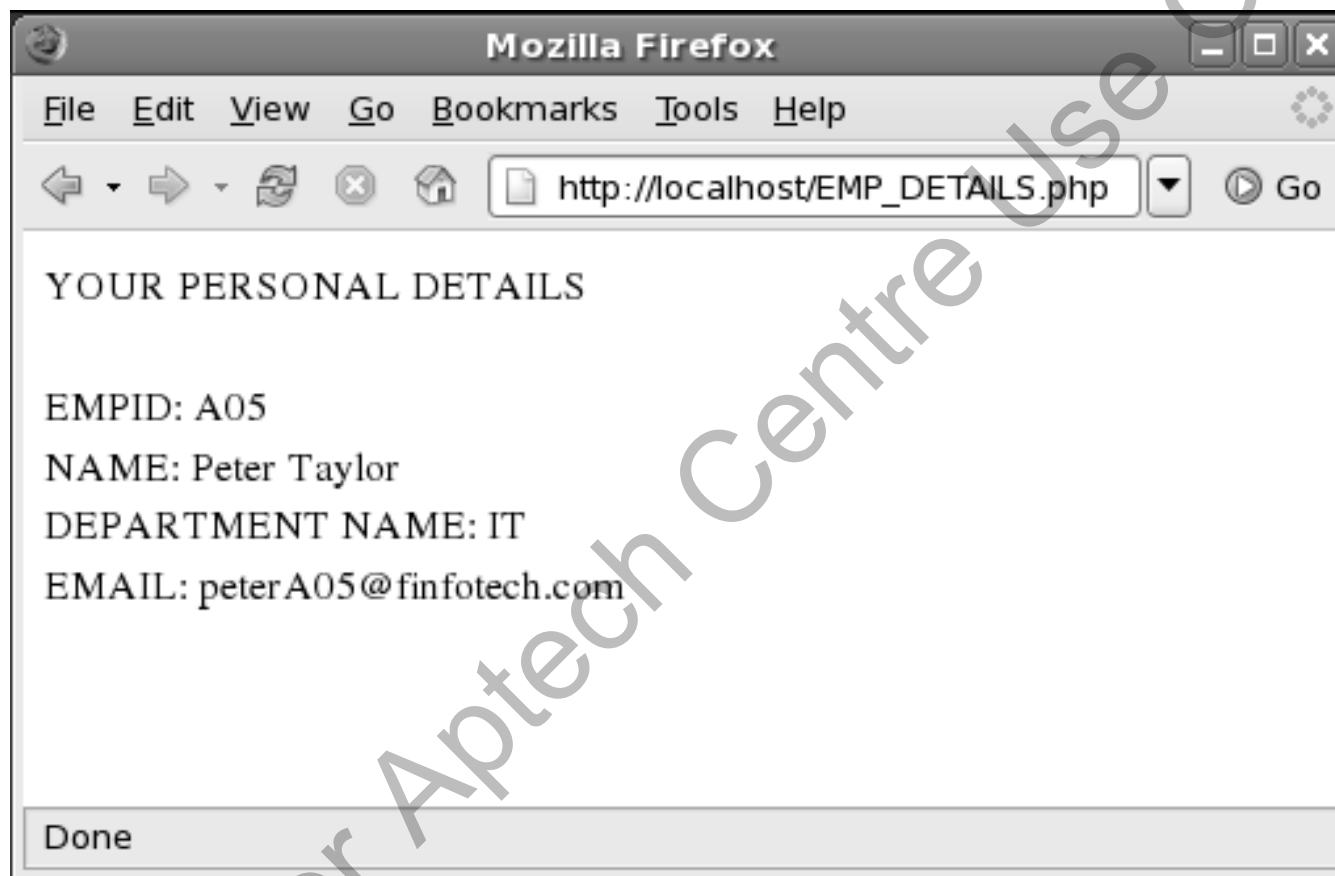
The screenshot shows a Mozilla Firefox browser window titled "Employee Details - Mozilla Firefox". The address bar displays "http://localhost/EMP\_DETAILS.html". The page content is a form titled "Enter your details" with the following fields:

- Employee ID:
- Name:
- Department: ☒ Finance ☐ Marketing ☒ IT
- Email:

Below the fields is a "SUBMIT" button. The status bar at the bottom of the browser window shows "Done".



Displays the following output:



## ◆ Hidden Field

- ◆ Is embedded in the HTML source code of the form
- ◆ Enables the user to pass variables with values from one form to another without requiring to re-enter the information
- ◆ Contents cannot be viewed by the user

### Syntax

```
<INPUT TYPE=HIDDEN NAME=hidden1 VALUE="PHP MESSAGE">
```

Where,

- ◆ **INPUT TYPE** - specifies that the field is hidden
- ◆ **NAME** - specifies the name of the hidden field
- ◆ **VALUE** - specifies the value as it appears on the form

- ◆ Steps for passing the names of the continents in a PHP script using hidden fields are as follows:
  1. Open a new file in the **gedit** text editor
  2. Enter the code and save the file as **continent.html** in the `/usr/local/apache2/htdocs` directory

## Snippet

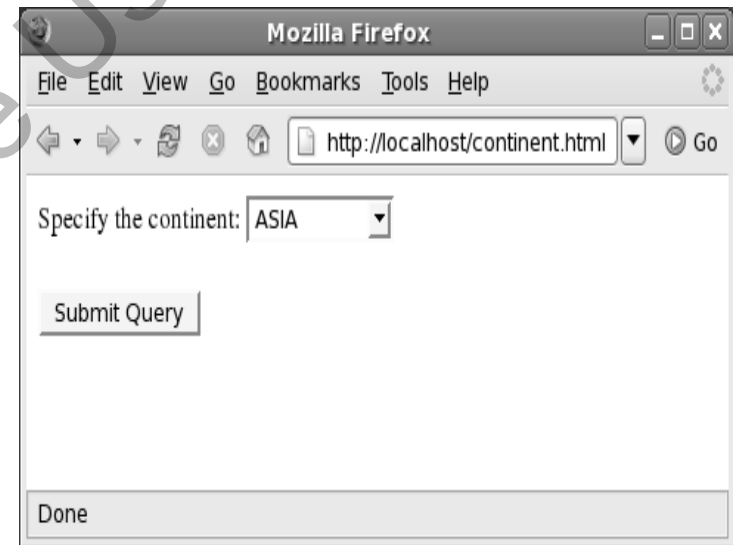
```
<html>
<FORM METHOD='get' action='continent.php'>
Specify the continent:
<SELECT TYPE='LISTBOX' NAME='continent'>
<OPTION>ASIA</OPTION>
  <OPTION>AUSTRALIA</OPTION>
<OPTION>EUROPE</OPTION>
</SELECT><BR><BR>
<INPUT TYPE=HIDDEN NAME=Asia>
<INPUT TYPE=HIDDEN NAME=Australia>
<INPUT TYPE=HIDDEN NAME=Europe>
<BR><INPUT TYPE=SUBMIT>
</FORM> </html>
```

- ◆ Steps for creating a PHP script to retrieve and process the data entered in the HTML form are as follows:
  1. Open a new file in the **gedit** text editor
  2. Enter the code and save the file as **continent.php** in the `/usr/local/apache2/htdocs` directory

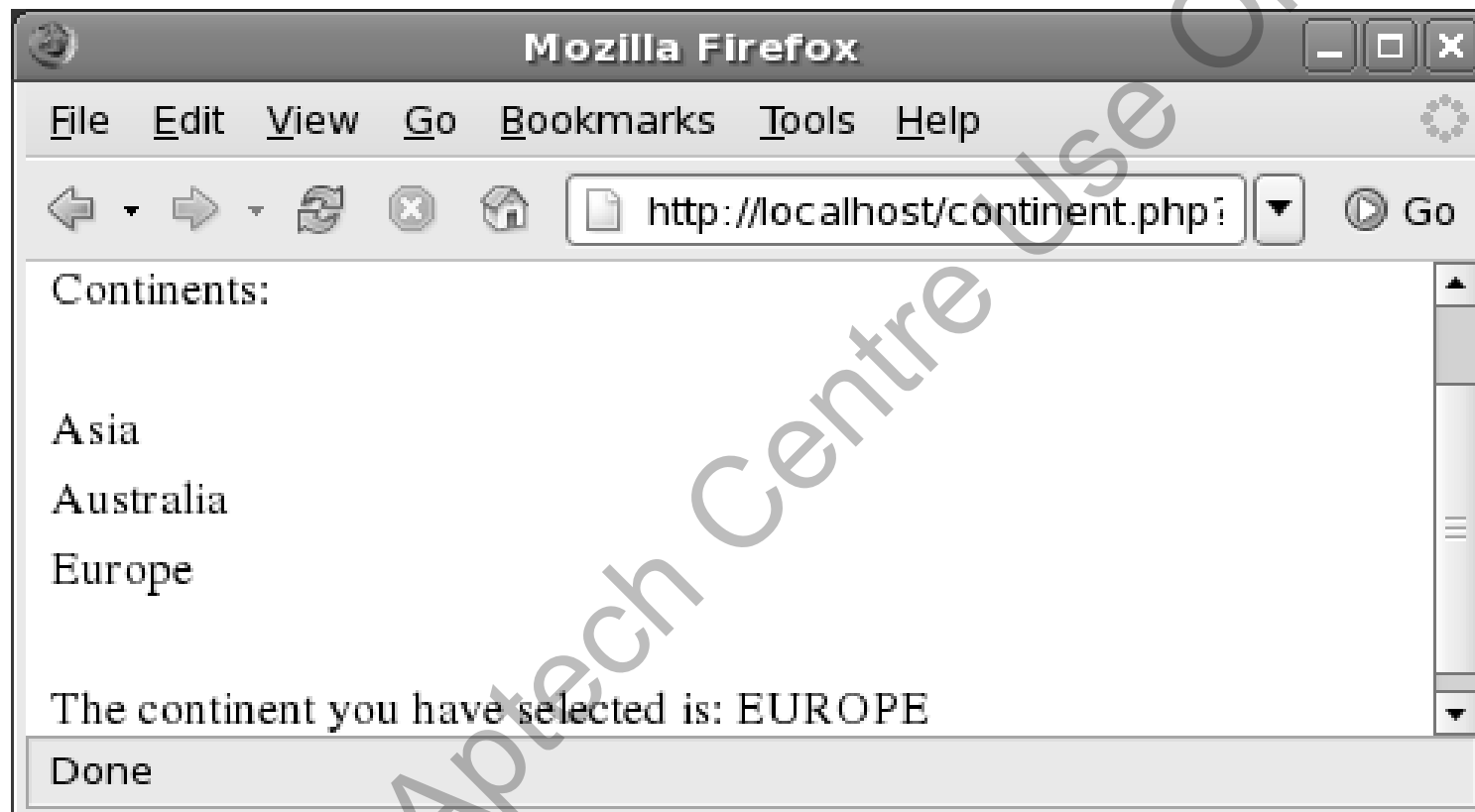
## Snippet

```
<?php
$A=$_GET['Asia'];
$B=$_GET['Australia'];
$C=$_GET['Europe'];
$Name=$_GET['continent'];
echo "<BR>";
echo "Continents:<BR> <BR> Asia <BR> Australia <BR> Europe <BR> <BR>";
echo "The continent you have selected is: $Name";
?>
```

- ◆ Steps for displaying **continent.html** page are as follows:
  1. Open the Mozilla Firefox Web browser
  2. Enter `http://localhost/continent.html` in the address bar and press **Enter**
  3. Select the required continent from the drop-down menu
  4. Click **Submit Query**



Displays the following output:



- ◆ A form is a Web page that is used to pass data from a client to a server
- ◆ PHP has a built-in support for collecting data from an HTML form
- ◆ The attributes of a form are namely, action and method
- ◆ The action attribute of a form specifies the URL that will process the form data and provide the feedback
- ◆ The method attribute of the form defines the method of transmitting information to the URL

- ◆ The GET method directs the Web browser to send the encrypted user information appended at the end of the URL, to the processing agent
- ◆ The POST method directs the Web browser to send all the user information to the processing agent, through the message body of an HTTP request
- ◆ Hidden form fields are not visible to users and enable form developers to pass information from a form to a script or from one form to another, before being passed to a script