**If..else…**

<html>

<body>

<?php

$d=date("D");

if ($d=="Fri")

echo "Have a nice weekend!";

else

echo "Have a nice day!";

?>

</body>

</html>

**Switch case**

<html>

<body>

<?php

$d=date("D");

switch ($d)

{

case "Mon":

echo "Today is Monday";

break;

case "Tue":

echo "Today is Tuesday";

break;

case "Wed":

echo "Today is Wednesday";

break;

case "Thu":

echo "Today is Thursday";

break;

case "Fri":

echo "Today is Friday";

break;

case "Sat":

echo "Today is Saturday";

break;

case "Sun":

echo "Today is Sunday";

break;

default:

echo "Wonder which day is this ?";

}

?>

</body>

</html>

**While loop:**

<html>

<body>

<?php

$i = 0;

$num = 50;

while( $i < 10)

{

$num--;

$i++;

}

echo ("Loop stopped at i = $i and num = $num" );

?>

</body>

</html>

**Do while loop :**

<html>

<body>

<?php

$i = 0;

$num = 0;

do

{

$i++;

}while( $i < 10 );

echo ("Loop stopped at i = $i" );

?>

</body>

</html>

**For loop:**

<html>

<body>

<?php

$a = 0;

$b = 0;

for( $i=0; $i<5; $i++ )

{

$a += 10;

$b += 5;

}

echo ("At the end of the loop a=$a and b=$b" );

?>

</body>

</html>

**For each**

<html>

<body>

<?php

$array = array( 1, 2, 3, 4, 5);

foreach( $array as $value )

{

echo "Value is $value <br />";

}

?>

</body>

</html>

**Array and and typs of array**

An array is a data structure that stores one or more similar type of values in a single value. For example if you want to store 100 numbers then instead of defining 100 variables its easy to define an array of 100 length.

There are three different kind of arrays and each array value is accessed using an ID c which is called array index.

* **Numeric array** - An array with a numeric index. Values are stored and accessed in linear fashion. These arrays can store numbers, strings and any object but their index will be prepresented by numbers. By default array index starts from zero.

**Associative array** - An array with strings as index. This stores element values in association with key values rather than in a strict linear index order. The associative arrays are very similar to numeric arrays in term of functionality but they are different in terms of their index. Associative array will have their index as string so that you can establish a strong association between key and values.

To store the salaries of employees in an array, a numerically indexed array would not be the best choice. Instead, we could use the employees names as the keys in our associative array, and the value would be their respective salary.

* **Multidimensional array** - An array containing one or more arrays and values are accessed using multiple indices. A multi-dimensional array each element in the main array can also be an array. And each element in the sub-array can be an array, and so on. Values in the multi-dimensional array are accessed using multiple index.

**Program for numeric array:**

<html>

<body>

<?php

/\* First method to create array. \*/

$numbers = array( 1, 2, 3, 4, 5);

foreach( $numbers as $value )

{

echo "Value is $value <br />";

}

/\* Second method to create array. \*/

$numbers[0] = "one";

$numbers[1] = "two";

$numbers[2] = "three";

$numbers[3] = "four";

$numbers[4] = "five";

foreach( $numbers as $value )

{

echo "Value is $value <br />";

}

?>

</body>

</html>

**Program for Associative array** :

<html>

<body>

<?php

/\* First method to associate create array. \*/

$salaries = array(

"mohammad" => 2000,

"qadir" => 1000,

"zara" => 500

);

echo "Salary of mohammad is ". $salaries['mohammad'] . "<br />";

echo "Salary of qadir is ". $salaries['qadir']. "<br />";

echo "Salary of zara is ". $salaries['zara']. "<br />";

/\* Second method to create array. \*/

$salaries['mohammad'] = "high";

$salaries['qadir'] = "medium";

$salaries['zara'] = "low";

echo "Salary of mohammad is ". $salaries['mohammad'] . "<br />";

echo "Salary of qadir is ". $salaries['qadir']. "<br />";

echo "Salary of zara is ". $salaries['zara']. "<br />";

?>

</body>

</html>

**$Get method:**

The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the **?** character.

|  |
| --- |
| http://www.test.com/index.htm?name1=value1&name2=value2 |

* The GET method produces a long string that appears in your server logs, in the browser's Location: box.
* The GET method is restricted to send upto 1024 characters only.
* Never use GET method if you have password or other sensitive information to be sent to the server.
* GET can't be used to send binary data, like images or word documents, to the server.
* The data sent by GET method can be accessed using QUERY\_STRING environment variable.
* The PHP provides **$\_GET** associative array to access all the sent information using GET method.

Try out following example by putting the source code in test.php script.

|  |
| --- |
| <?php  if( $\_GET["name"] || $\_GET["age"] )  {  echo "Welcome ". $\_GET['name']. "<br />";  echo "You are ". $\_GET['age']. " years old.";  exit();  }  ?>  <html>  <body>  <form action="<?php **$\_PHP\_SELF** ?>" method="GET">  Name: <input type="text" name="name" />  Age: <input type="text" name="age" />  <input type="submit" />  </form>  </body>  </html> |

**The POST Method**

The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY\_STRING.

* The POST method does not have any restriction on data size to be sent.
* The POST method can be used to send ASCII as well as binary data.
* The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
* The PHP provides **$\_POST** associative array to access all the sent information using POST method.

Try out following example by putting the source code in test.php script.

|  |
| --- |
| <?php  if( $\_POST["name"] || $\_POST["age"] )  {  echo "Welcome ". $\_POST['name']. "<br />";  echo "You are ". $\_POST['age']. " years old.";  exit();  }  ?>  <html>  <body>  <form action="<?php **$\_PHP\_SELF** ?>" method="POST">  Name: <input type="text" name="name" />  Age: <input type="text" name="age" />  <input type="submit" />  </form>  </body>  </html> |