



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

DEPARTMENT OF COMPUTER SCIENCE

E-NOTES

OPEN SOURCE SOFTWARE

PREPARED BY

Mrs. M.SILAMBARSI, M.Sc., M.Phil.,

**HEAD, DEPARTMENT OF COMPUTER SCIENCE & INFORMATION
SYSTEM MANAGEMENT**

Mrs. R. SANGEETHA, M.C.A., M.Phil.,

ASSISTANT PROFESSOR, DEPARTMENT OF COMPUTER SCIENCE



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

OPEN SOURCE SOFTWARE

Course Objectives

To understand the basic concepts of PHP.

Course Outcome:

The student will be able to build dynamic pages using PHP.

CONTENTS

1. PHP Control Statement
2. Array
3. Functions
4. Basic Form Processing
5. File and Folder Access
6. Cookies
7. Sessions

Text Books: "The Complete Reference Linux", Peterson, Tata McGraw HILL–2010



CONTROL STATEMENTS, ARRAYS & FUNCTIONS

1. CONTROL STATEMENTS

A control statement is a block of code that decides the execution path of a program depending on the value of the set condition.

1.1 CONDITIONAL STATEMENTS

Conditional statements are used to perform different actions based on different conditions.

In PHP we have the following conditional statements:

- | | |
|--|---|
| 1. if statement | - Executes some code if one condition is true |
| 2. if...else statement | - Executes some code if a condition is true and another code if that condition is false |
| 3. if...elseif...else statement | - Executes different codes for more than two conditions |
| 4. switch statement | - Selects one of many blocks of code to be executed |

if statement

The if statement executes some code if one condition is true.

Syntax:

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

Example:

```
<?php
$bank_balance=50;
if ($bank_balance < 100)
{
    $bank_balance += 50;
}
echo "Bank Balance=", $bank_balance;
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

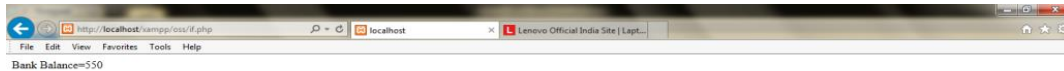
Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

?>

Output:



if...else statement

The if...else statement executes the conditional statement first if the condition is TRUE. But if it's FALSE, the second one is executed. One of the two choices must be executed. Under no circumstance can both (or neither) be executed.

Syntax:

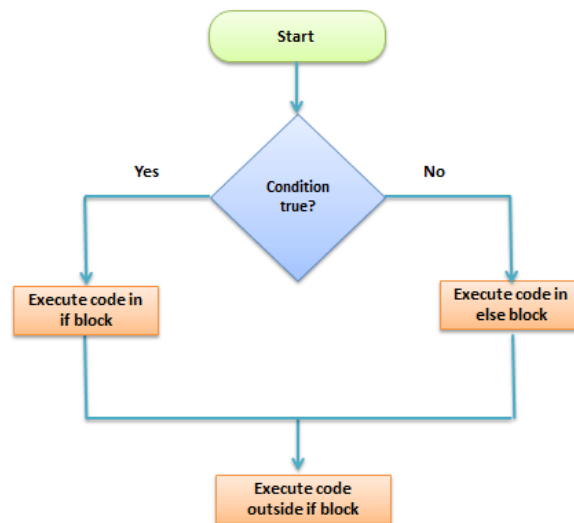
```
if (condition)
{
    code to be executed if condition is true;
}
else
```



```
{  
    code to be executed if condition is false;  
}
```

Whereas,

- “**if (condition is true)**” is the control structure
- “**block one**” is the code to be executed if the condition is true
- {...**else...**} is the fallback if the condition is false
- “**block two**” is the block of code executed if the condition is false



Example:

```
<?php  
$bank_balance=90;  
if ($bank_balance < 100)  
{  
    $bank_balance += 500;  
}  
else  
{  
    $bank_balance +=1000;  
}  
echo "Bank Balance=", $bank_balance;
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

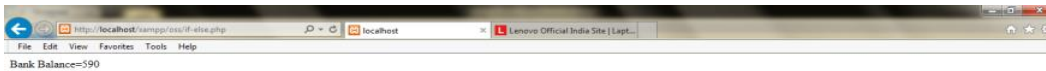
Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

?>

Output:



if...elseif...else Statement

The if...elseif...else statement executes different codes for more than two conditions.

Syntax:

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

Example:

```
<?php
$bank_balance=150;
if ($bank_balance < 100)
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
{
    $bank_balance += 500;
}
elseif (($bank_balance >=100) && ($bank_balance < 500) )
{
    $bank_balance += 1000;
}
else
{
    $bank_balance +=2000;
    echo "Bank Balance=", $bank_balance;
}
?>
```

Output:



Switch Case Statement

Switch... case is similar to the **if then... else** control structure. It only **executes** a single block of code depending on the **value** of the condition. If no condition has been met then the default block of code is executed.

Syntax:

```
<?php

switch(condition)
{
case value:

//block of code to be executed

break;
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
case value2:  
  
    //block of code to be executed  
  
    break;  
  
default:  
  
    //default block code  
  
    break;  
}
```

?>

Whereas,

- “**switch(...){...}**” is the control structure block code
- “**case value: case...**” are the blocks of code to be executed depending on the value of the condition
- “**default:**” is the block of code to be executed when no value matches with the condition

Working Principle:

The flow chart shown below illustrates how the switch control structure works.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

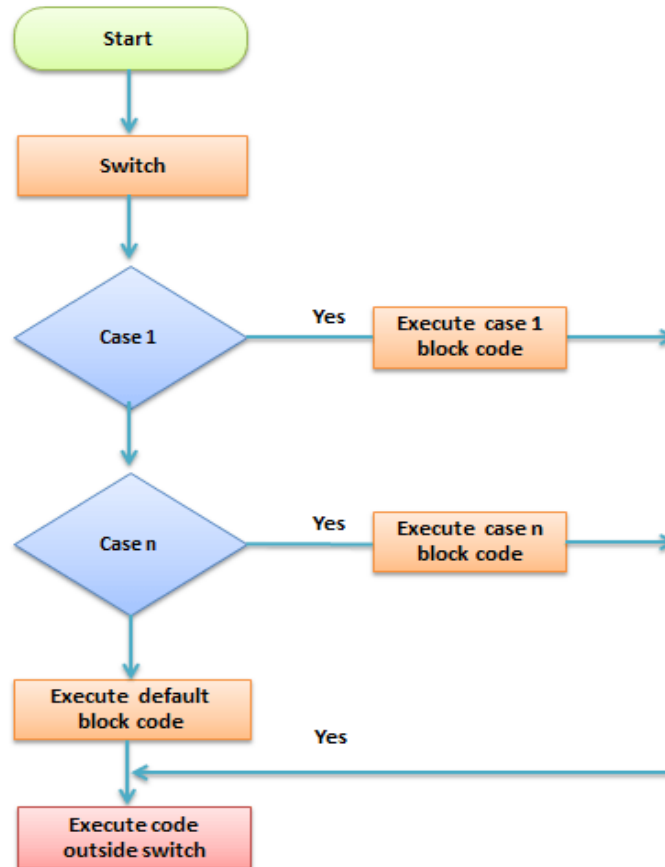
(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS



Example:

```
<?php
$page="Home";
switch ($page)
{
case "Home":
echo "You selected Home";
break;
case "About":
echo "You selected About";
break;
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463

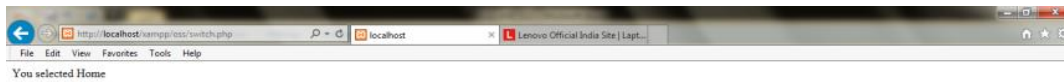


E-NOTES / CS

```
case "News":
    echo "You selected News";
    break;
case "Login":
    echo "You selected Login";
    break;
case "Links":
    echo "You selected Links";
    break;
}
```

?>

Output:



1.2 LOOPING STATEMENT

Loops in PHP are used to execute the same block of code a specified number of times. PHP supports following four loop types.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

- **for** – loops through a block of code a specified number of times.
- **while** – loops through a block of code if and as long as a specified condition is true.
- **do...while** – loops through a block of code once, and then repeats the loop as long as a special condition is true.
- **foreach** – loops through a block of code for each element in an array.

We will discuss about **continue** and **break** keywords used to control the loops execution.

The for loop statement

PHP for loop is very similar to a while loop in that it continues to process a block of code until a statement becomes false, and everything is defined in a single line.

Syntax:

```
for (initialization; condition; increment)
{
    code to be executed;
}
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

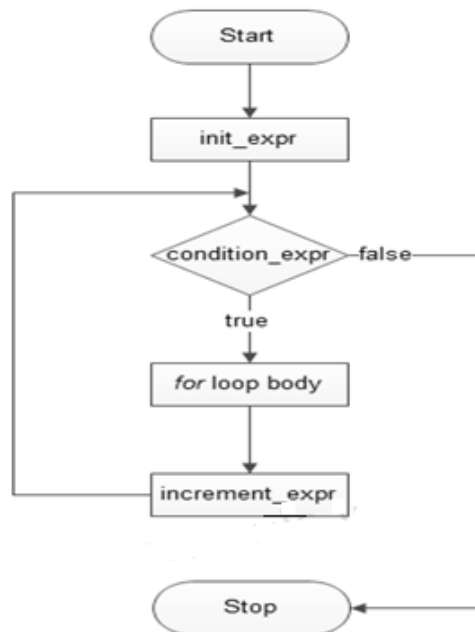
(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS



The initializer is used to set the start value for the counter of the number of loop iterations. A variable may be declared here for this purpose and it is traditional to name it \$i.

Example:

The following example makes five iterations and changes the assigned value of two variables on each pass of the loop.

```
<?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" );
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

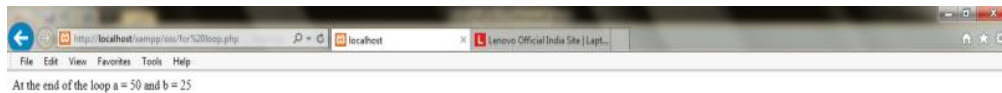
Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

?>

OUTPUT:



The while loop statement

The while statement will execute a block of code if and as long as a test expression is true. If the test expression is true then the code block will be executed. After the code has executed the test expression will again be evaluated and the loop will continue until the test expression is found to be false.

Syntax:

```
while (condition)
{
    code to be executed;
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

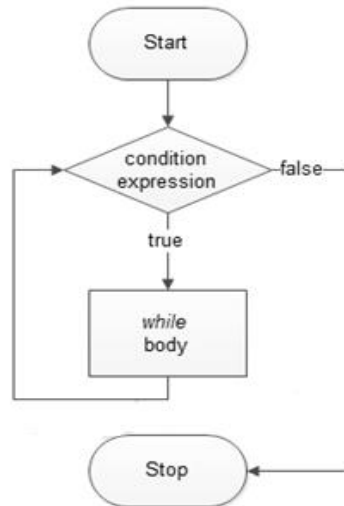
HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

}



Example:

These example decrements a variable value on each iteration of the loop and the counter increments until it reaches 10 when the evaluation is false and the loop ends.

```
<html>
  <body>
    <?php
      $i = 0;
      $num = 50;
      while( $i < 10)
      {
        $num--;
        $i++;
      }
      echo ("Loop stopped at i = $i and num = $num" );
    ?>
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463

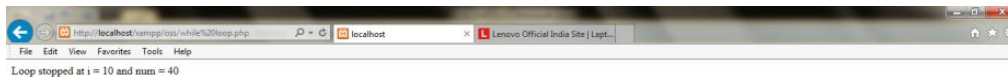


E-NOTES / CS

```
</body>
```

```
</html>
```

OUTPUT:



The do...while loop statement

The do...while statement will execute a block of code at least once - it then will repeat the loop as long as a condition is true.

Syntax:

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



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

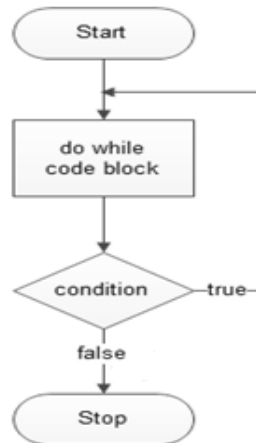
(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS



Example:

The following example will increment the value of *i* at least once, and it will continue incrementing the variable *i* as long as it has a value of less than 10.

```
<html>
<body>
<?php
    $i = 0;
    $num = 0;
    do
    {
        $i++;
    }
    while( $i < 10 );
    echo ("Loop stopped at i = $i" );
?>
</body>
```




M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

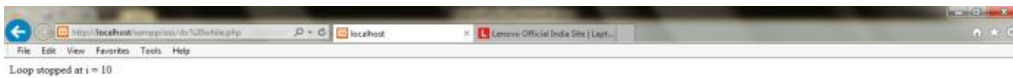
Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

</html>

OUTPUT:



The foreach loop statement

The foreach loop is mainly used for looping through the values of an array. It loops over the array, and each value for the current array element is assigned to \$value, and the array pointer is advanced by one to go the next element in the array.

Syntax:

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

Example:

<html>



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

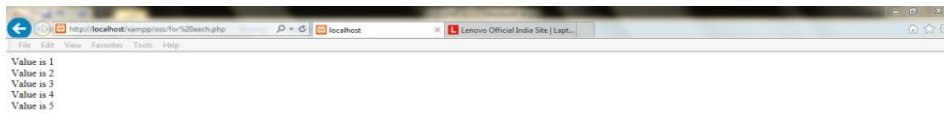
Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
<body>
  <?php
    $array = array( 1, 2, 3, 4, 5);
    foreach( $array as $value )
    {
      echo "Value is $value <br />";
    }
  ?>
</body>
</html>
```

OUTPUT:



The break statement

The PHP **break** keyword is used to terminate the execution of a loop prematurely.

The **break** statement is situated inside the statement block. It gives you full control and whenever you want to exit from the loop you can come out. After coming out of a loop immediate statement to the loop will be executed.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

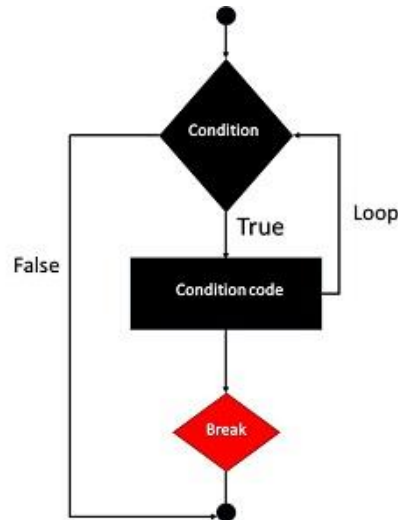
(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS



Example:

In the following example condition test becomes true when the counter value reaches 3 and loop terminates.

```
<html>
<body>
<?php
    $i = 0;
    while( $i < 10)
    {
        $i++;
        if( $i == 3 )
        break;
    }
    echo ("Loop stopped at i = $i" );
?>
</body>
</html>
```

OUTPUT:



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

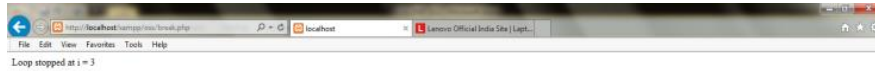
(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS



The continue statement

The PHP **continue** keyword is used to halt the current iteration of a loop but it does not terminate the loop.

Just like the **break** statement the **continue** statement is situated inside the statement block containing the code that the loop executes, preceded by a conditional test. For the pass encountering **continue** statement, rest of the loop code is skipped and next pass starts.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

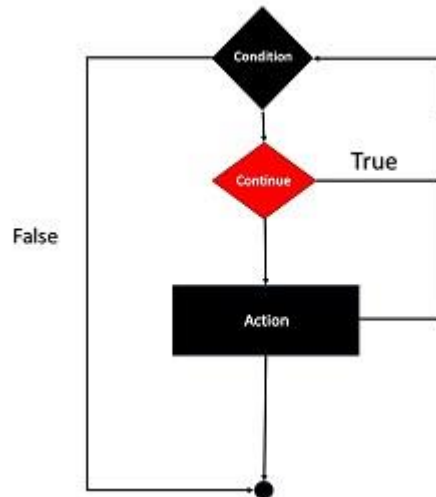
(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS



Example:

In the following example loop prints the value of array but for which condition becomes true it just skip the code and next value is printed.

```
<html>
<body>
<?php
$array = array( 1, 2, 3, 4, 5);
foreach( $array as $value )
{
    if( $value == 3 )continue;
    echo "Value is $value <br />";
}
?>
</body>
</html>
```

OUTPUT:



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

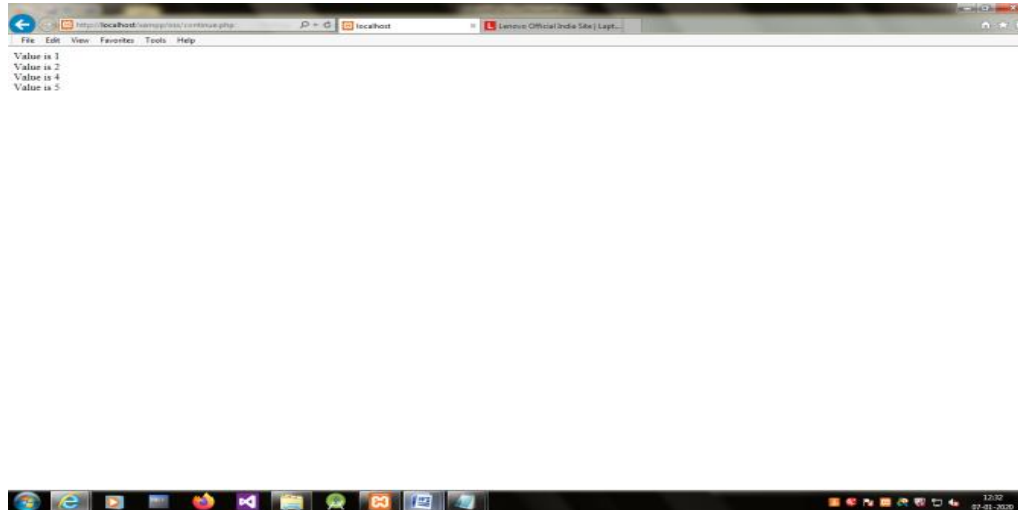
(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS



2. PHP ARRAYS

An array is a data structure in PHP that stores multiple values in one memory location.

Advantages of PHP Arrays:

1. **Less Code:** We don't need to define multiple variables.
2. **Easy to traverse:** By the help of single loop, we can traverse all the elements of an array.
3. **Sorting:** We can sort the elements of array.

There are three types of arrays in PHP:

- **Numeric array (Indexed array)** - An array with a numeric index value.
- **Associative array** - Where each ID key is associated with a value of an array.
- **Multidimensional array** - Array containing one or more arrays.

NUMERIC ARRAY



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

In an indexed/ Numeric array, the keys are numeric and start with 0, and the values can be any data type. The following shows two ways of assigning values to an indexed array:

1) Adding items to an array using explicit locations

Example:

```
<?php
    $friends[0] = 'Jhon';
    $friends[1] = 'Ramson';
    $friends[2] = 'Nikita';
?>
```

or

array():

This is another function which is used to create an array.

Example:

```
<?php
    $friends = array('Jhon','Ramson','Nikita');
?>
```

2) Adding items to an array and retrieving them

We can easily add items to an array and retrieve them in PHP. Let's see a simple example to traverse all the elements of PHP array.

```
<?php
    $paper[] = "Copier";
    $paper[] = "Inkjet";
    $paper[] = "Laser";
    $paper[] = "Photo";
    for ($j = 0 ; $j < 4 ; ++$j)
        echo "$j: $paper[$j]<br>";
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

?>

This example prints out the following:

0: Copier

1: Inkjet

2: Laser

3: Photo

Advantages of Indexed Array

1. Numeric index values make it easier to use and having numeric indexes are common to almost all the programming languages, hence it also makes the code more readable for others who go through your code.

ASSOCIATIVE ARRAY

Keeping track of array elements by index works just fine, but can require extra work in terms of remembering which number refers to which product. It can also make code hard for other programmers to follow.

This is where associative arrays come into their own. Using them, you can reference the items in an array by name rather than by number. PHP allows you to associate name/label with each array elements in PHP using => symbol. Such way, you can easily remember the element because each element is represented by label than an incremented number.

Example 1:

```
<?php
    $paper['copier'] = "Copier & Multipurpose";
    $paper['inkjet'] = "Inkjet Printer";
    $paper['laser'] = "Laser Printer";
    $paper['photo'] = "Photographic Paper";
    echo $paper['laser'];
?>
```




M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

In place of a number (which doesn't convey any useful information, aside from the position of the item in the array), each item now has a unique name that you can use to reference it elsewhere, as with the echo statement—which simply prints out **Laser Printer**. The names (copier, inkjet, etc.) are called **indexes or keys** and the items assigned to them (such as "Laser Printer") are called **values**.

Or

array():

This is another way to create an array.

Example 2:

```
<?php
    $salary = array('John'=>15000, 'Ramson'=>25000, 'Nikita'=>27000);
?>
```

Advantages of Associative Array

1. We can provide more meaningful key or index values to our array elements.
2. We can save more data, as we can have a string as key to the array element, where we can have associated data to the value to be stored.

MULTIDIMENSIONAL ARRAY

A multidimensional array is an array which stores another array at each index rather than storing a single value. In simple words, a multidimensional array is an array of arrays. In general practice, associative arrays are stored inside multidimensional arrays.

Example:

```
<?php
    $multiDArray = array(
        "A" => array(0 => "red", 2 => "blue", 3 => "green"),
        "B" => array(1 => "orange", 2 => "black"),
        "C" => array(0 => "white", 4 => "purple", 8 => "grey")
    );
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463

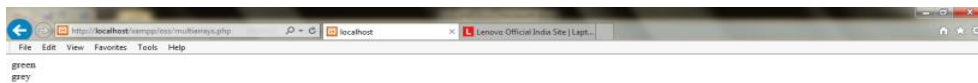


E-NOTES / CS

```
echo $multiDArray["A"][3];   echo "<br>";  
echo $multiDArray["C"][8];
```

?>

OUTPUT:



Traversing PHP Multidimensional Array

Traversing an array means to iterate it starting from the first index till the last element of the array. We can traverse a multidimensional array either using two for loops or two foreach or one for loop and one foreach.

Using one for loop and one foreach

In case of multidimensional array, we have to traverse the main array and then the arrays stored inside the main arrays, hence we need two loops. While using the for loop to traverse a multidimensional array we must know the size/length of the array, which can be found using the **count()** function.

Let's take the array defined above for the first example of traversing multidimensional array.

```
<?php  
/*  
    multidimensional array initialization
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
*/
$cars = array(
    array(
        "name"=>"Urus",
        "type"=>"SUV",
        "brand"=>"Lamborghini"
    ),
    array(
        "name"=>"Cayenne",
        "type"=>"SUV",
        "brand"=>"Porsche"
    ),
    array(
        "name"=>"Bentayga",
        "type"=>"SUV",
        "brand"=>"Bentley"
    ),
);
// array traversal
// find size of the array
$size = count($lamborghinis);
// using the for loop
for($i = 0; $i < $size; $i++)
{
    foreach($cars[$i] as $key => $value) {
        echo $key . " : " . $value . "\n";
    }
    echo "\n";
}
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

?>

Output:

name : Urus

type : SUV

brand : Lamborghini

name : Cayenne

type : SUV

brand : Prosche

name : Bentayga

type : SUV

brand : Bentley

In the above multidimensional array, we have the main array as indexed array and the arrays stored as elements are associative.

Advantages of Multidimensional Array

Here are a few advantages of using multidimensional array in our program/script:

1. Detailed information can be stored in multidimensional array.
2. On top level, it can either be kept indexed or associative, which makes it more user-friendly, as they can use it as per their requirements.

3. FUNCTIONS

The function is a self-contained block of statements that can repeatedly be executed whenever we need it.

The concept of the function is the same as in other languages like C. There are more than 1,000 *inbuilt functions* into the standard PHP distribution. Apart from these, we can define own functions according to the requirements.

Syntax:



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
function functionName()  
{  
    statement 1 :  
    statement 2 :  
    statement 3 :  
    .....  
}
```

ADVANTAGES OF FUNCTIONS:

- **Reusability:** If we have a common code that we would like to use at various parts of a program, we can simply contain it within a function and call it whenever required. This reduces the time and effort of repetition of a single code. This can be done both within a program and also by importing the PHP file, containing the function, in some other program
- **Easier error detection:** Since, our code is divided into functions, we can easily detect in which function, the error could lie and fix them fast and easily.
- **Easily maintained:** As we have used functions in our program, so if anything or any line of code needs to be changed, we can easily change it inside the function and the change will be reflected everywhere, where the function is called. Hence, easy to maintain.

TYPES OF FUNCTIONS IN PHP:

PHP provides us with two major types of functions:

- **Built-in functions :** PHP provides us with huge collection of built-in library functions. These functions are already coded and stored in form of functions. To use those we just need to call them as per our requirement like, var_dump, fopen(), print_r(), gettype() and so on.
- **User Defined Functions :** Apart from the built-in functions, PHP allows us to create our own customised functions called the user-defined functions.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

Using this we can create our own packages of code and use it wherever necessary by simply calling it.

Internal (Built-in) Functions

Built in functions are functions that exist in PHP installation package. These built in functions are what make PHP a very efficient and productive scripting language. The built in functions can be classified into many categories. Below is the list of the categories.

1) String Function

strtolower	converts all characters of the string to lower case
ucfirst()	converts first letter to upper case
strlen	counts number of characters in a string and returns integer value
ltrim()	trims unnecessary spaces from left
str_repeat()	repeats the string
strpos()	finds position of the string
strtoupper()	converts all characters of the string to upper case
ucwords()	converts first letter of each word to upper case
trim()	trims unnecessary spaces
rtrim()	trims unnecessary spaces from right
str_word_count()	count the number of words
str_replace()	search and replace characters from the string
substr(int,int)	prints a string from defined initial character number to defined last number

2) Numeric Function

abs()	returns positive value of a number
sqrt()	returns square root of a number
round()	rounds a floating number
floor()	rounds a number down to a nearest integer



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

ceil()	rounds a number up to a nearest integer
rand()	generates a random integer
mt_rand()	generates random number between defined initial and end number
pow()	returns x raised to the power of y
pi()	returns the value of pi
min()	returns the lowest value from an array
max()	returns the highest value from an array
fmod()	returns the remainder from x/y { % }
bindec()	converts a binary number to a decimal number
decbin()	converts a decimal number to a binary number
deg2rad()	converts a degree value to a radian value

3) Date & Time Function

checkdate()	Validates a Gregorian date
date_diff()	Returns the difference between two dates
date_format()	Returns a date formatted according to a specified format
date_modify()	Modifies the timestamp
date_time_set()	Sets the time
date_timestamp_get()	Returns the Unix timestamp representing the date
date_timestamp_set()	Sets the date and time based on an Unix timestamp
date_timezone_get()	Return time zone relative to given DateTime
date_timezone_set()	Sets the time zone for the DateTime object
date()	Formats a local date and time
getdate()	Returns date/time information of the timestamp or the current local date/time
gettimeofday()	Returns the current time
gmdate()	Formats a GMT/UTC date and time
gmmktime()	Get Unix timestamp for a GMT date
gmstrftime()	Formats a GMT/UTC date and time according to locale settings
idate()	Formats a local time/date as integer
localtime()	Returns the local time
mktime()	Returns the Unix timestamp for a date



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

strftime()	Formats a local time/date according to locale settings
strptime()	Parses a time/date generated with strftime()
strtotime()	Parses an English textual datetime into a Unix timestamp
time()	Returns the current time as a Unix timestamp

User Defined Functions (Using Your own PHP Function)

- You can also create your own functions.
- First, you need to give it a name.
- All functions in PHP start with function().
- Functions can also accept parameters.
- A function can also be used to return values.

Example 1:

```
<?php
    function myFunction()
    {
        echo "Example PHP function";
    }
    myFunction();
?>
```

OUTPUT:



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

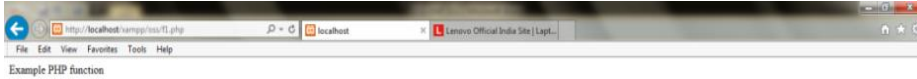
(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS



Example 2:

```
<?php
    Function add($a,$b)
    {
        $total = $a + $b;
        return $total; //Function return, No output is shown
    }
    echo "Total=",add(55,20);
?>
```

OUTPUT:



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

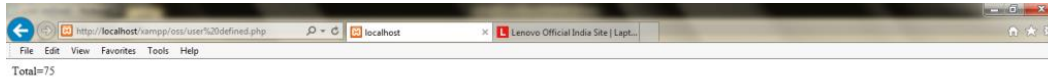
(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS



Few Rules to name Functions

1. A **function name** can only contain alphabets, numbers and underscores. No other special character is allowed.
2. The name should start with either an alphabet or an underscore. It should not start with a number.
3. And last but not least, function names are not case-sensitive.
4. The opening curly brace { after the function name marks the start of the function code, and the closing curly brace } marks the end of function code.

RETURNING VALUES

All functions in PHP return a value, even if you don't explicitly cause them to. Thus, the concept of "void" functions does not apply to PHP. You can specify the return value of your function by using the return keyword:

Example:

```
<?php  
function hello()
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

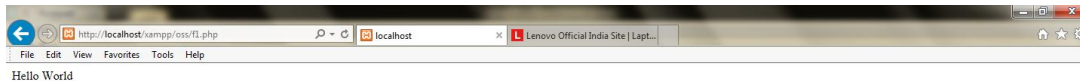
Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
{  
    return "Hello World";  
}  
$txt = hello();  
echo hello();  
?>
```

OUTPUT:



Note, however, that even if you don't return a value, PHP will still cause your function to return NULL.

PHP FUNCTION ARGUMENTS

Information can be passed to functions through arguments. The following example has a function with two arguments:

Example:



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

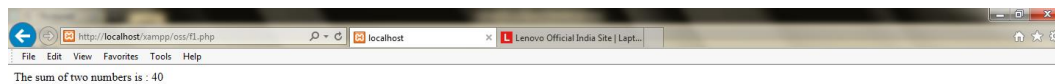
Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
<?php
function addFunction($number, $number)
{
    $number = $number + $number;
    echo "The sum of two numbers is : $number"; //Outputs as 30
}
addFunction(10, 20);
?>
```

OUTPUT:



PHP DEFAULT ARGUMENT VALUE

The following example shows how to use a default parameter. If we call the function setWeight() without arguments it takes the default value as an argument:

Example:



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

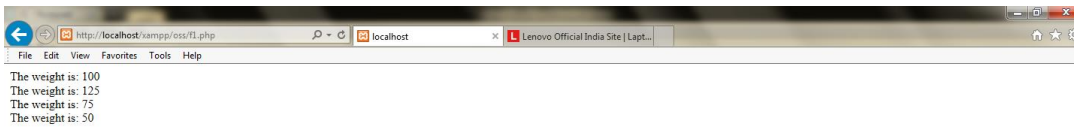
Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
<?php
function setWeight($minWeight=50)
{
    echo "The weight is: $minWeight <br>";
}
setWeight(100);
setWeight(125);
setWeight(75);
setWeight(); // will use the default value of 50
?>
```

OUTPUT:



Advantages of User-defined Functions

1. **Reusable Code:** As it's clear from the example above, you write a function once and can use it for a thousand times in your program.



- 2. Less Code Repetition:** In the example above we just had one line of code in the function, but what if we have 10 lines of code. So rather than repeating all those lines of code over and over again, we can just create a function for them and simply call the function.
- 3. Easy to Understand:** Using functions in your program, makes the code more readable and easy to understand.

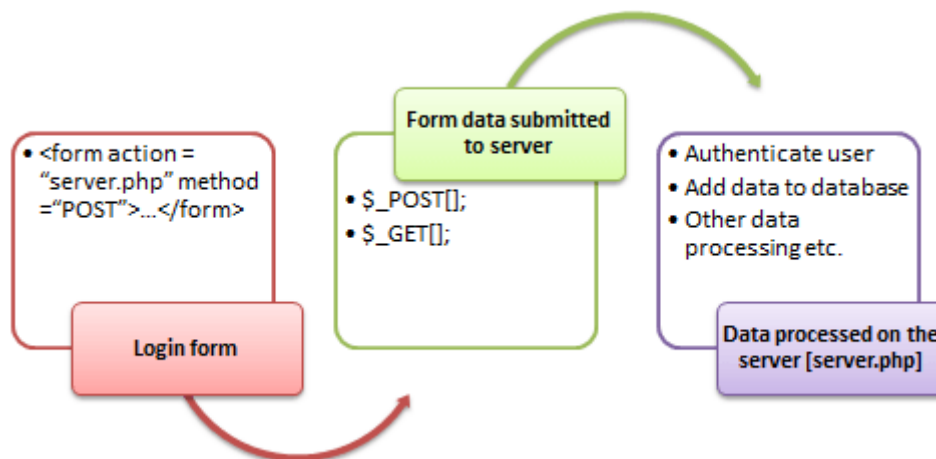
4. PHP BASIC FORM PROCESSING

PHP FORM

A form is an HTML tag that contains graphical user interface items such as input box, check boxes radio buttons etc. The form is defined using the `<form>...</form>` tags and GUI items are defined using form elements such as input.

When you login into a website or into your mail box, you are interacting with a form. Forms are used to get input from the user and submit it to the web server for processing.

The diagram below illustrates the form handling process.



When and why we are using forms?

- Forms come in handy when developing flexible and dynamic applications that accept user input.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

- Forms can be used to edit already existing data from the database

CREATE A FORM

We will use HTML tags to create a form. Below is the minimal list of things you need to create a form.

1. Opening and closing form tags `<form>...</form>`
2. Form submission type POST or GET
3. Submission URL that will process the submitted data
4. Input fields such as input boxes, text areas, buttons, checkboxes etc.

A Simple Registration Form

```
<html>
<head>
  <title>Registration Form</title>
</head>
<body>
  <h2>Registration Form</h2>
  <form action="registration_form.php" method="POST">
    First name:
    <input type="text" name="firstname"> <br> Last name:
    <input type="text" name="lastname">
    <input type="hidden" name="form_submitted" value="1" />
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

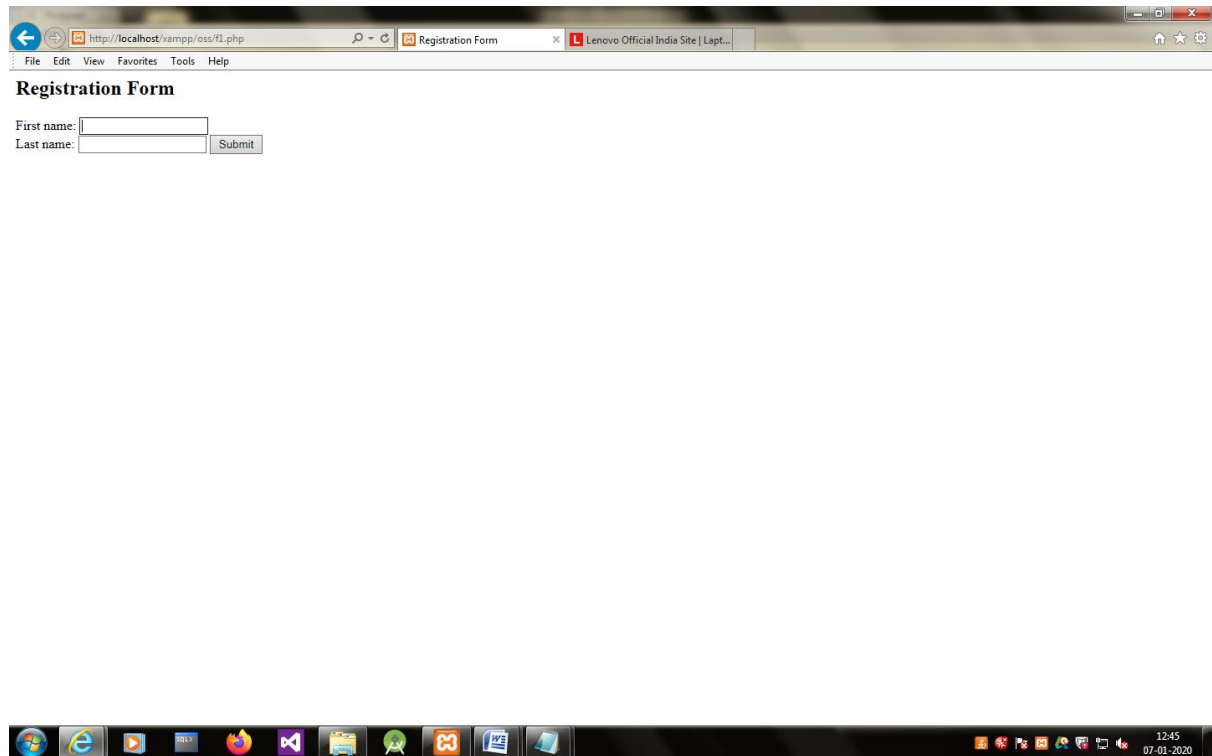
(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS



SUBMITTING THE FORM DATA TO THE SERVER

The action attribute of the form specifies the submission URL that processes the data. The method attribute specifies the submission type.

PHP POST method

- This is the built in PHP super global array variable that is used to get values submitted via HTTP POST method.
- The array variable can be accessed from any script in the program; it has a global scope.
- This method is ideal when you do not want to display the form post values in the URL.
- A good example of using post method is when submitting login details to the server.

Syntax:



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
<?php
    $_POST['variable_name'];
?>
```

HERE,

“\$_POST[...]” is the PHP array

“variable_name” is the URL variable name.

Example:

```
<form action="registration_form.php" method="POST">
First name:<input type="text" name="firstname"><br>
Last name:<input type="text" name="lastname">
<br>
<input type="hidden" name="form_submitted" value="1"/>
<input type="submit" value="submit">
</form>
```

PHP GET method

- This is the built in PHP super global array variable that is used to get values submitted via HTTP GET method.
- The array variable can be accessed from any script in the program; it has a global scope.
- This method displays the form values in the URL.
- It's ideal for search engine forms as it allows the users to book mark the results.

Syntax:

```
<?php
    $_GET['variable_name'];
?>
```

HERE,

“\$_GET[...]” is the PHP array.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

“variable_name” is the URL variable name.

Example:

```
<form action="registration_form.php" method="GET">
First name:<input type="text" name="firstname"><br>
Last name:<input type="text" name="lastname">
<br>
<input type="hidden" name="form_submitted" value="1"/>
<input type="submit" value="submit">
</form>
```

HTML FORM ELEMENTS

1. The <input> Element

The most important form element is the <input> element.

The <input> element can be displayed in several ways, depending on the type attribute.

Example:

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

If the type attribute is omitted, the input field gets the default type: "text".

2. The <select> Element

The <select> element defines a **drop-down list**:

Example:

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

a) <option> :



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

The `<option>` element defines an option that can be selected. By default, the first item in the drop-down list is selected. To define a pre-selected option, add the `selected` attribute to the option:

Example:

```
<option value="fiat" selected>Fiat</option>
```

b) Visible Values:

Use the `size` attribute to specify the number of visible values:

Example:

```
<select name="cars" size="3">  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>
```

c) Allow Multiple Selections:

Use the `multiple` attribute to allow the user to select more than one value.

Example:

```
<select name="cars" size="4" multiple>  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="fiat">Fiat</option>  
  <option value="audi">Audi</option>  
</select>
```

3. The `<textarea>` Element

The `<textarea>` element defines a multi-line input field (a **text area**):



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

Example 1:

```
<textarea name="message" rows="10" cols="30">
```

The cat was playing in the garden.

```
</textarea>
```

The **rows** attribute specifies the visible number of lines in a text area. The **cols** attribute specifies the visible width of a text area. This is how the HTML code above will be displayed in a browser:



You can also define the size of the text area by using CSS:

Example 2:

```
<textarea name="message" style="width:200px; height:600px;">
```

The cat was playing in the garden.

```
</textarea>
```

4. The <button> Element

The **<button>** element defines a clickable **button**:

Example:

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```

This is how the HTML code above will be displayed in a browser:

Click Me!



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

PHP FILES AND FOLDER ACCESS

BASICS OF FILE HANDLING

File handling is needed for any application. For some tasks to be done file needs to be processed. File handling in PHP is similar as file handling is done by using any programming language like C. PHP File System allows us to create file, read file line by line, read file character by character, write file, append file, delete file and close file. PHP has many functions to work with normal files.

Those functions are:

1) fopen():

PHP fopen() function is used to open a file. First parameter of fopen() contains name of the file which is to be opened and second parameter tells about mode in which file needs to be opened.

Example:

```
<?php
    $file = fopen("demo.txt",'w');
?>
```

Files can be opened in any of the following modes :

- **“w”** – Opens a file for write only. If file not exist then new file is created and if file already exists then contents of file is erased.
- **“r”** – File is opened for read only.
- **“a”** – File is opened for write only. File pointer points to end of file. Existing data in file is preserved.
- **“w+”** – Opens file for read and write. If file not exist then new file is created and if file already exists then contents of file is erased.
- **“r+”** – File is opened for read/write.
- **“a+”** – File is opened for write/read. File pointer points to end of file. Existing data in file is preserved. If file is not there then new file is created.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

- “x” – New file is created for write only.

2) fread() :

After file is opened using fopen() the contents of data are read using fread(). It takes two arguments. One is file pointer and another is file size in bytes.

Example:

```
<?php
    $filename = "demo.txt";
    $file = fopen( $filename, 'r' );
    $size = filesize( $filename );
    $filedata = fread( $file, $size );
?>
```

3) fwrite() :

New file can be created or text can be appended to an existing file using fwrite() function. Arguments for fwrite() function are file pointer and text that is to written to file. It can contain optional third argument where length of text to written is specified.

Example:

```
<?php
    $file = fopen("demo.txt", 'w');
    $text = "Hello world\n";

    fwrite($file, $text);
?>
```

4) fclose() :

File is closed using fclose() function. Its argument is file which needs to be closed.

Example:



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
<?php
    $file = fopen("demo.txt", 'r');
    //some code to be executed
    fclose($file);
?>
```

5) fgetc():

The fgetc() function returns a single character from an open file.

Example:

```
<?php
    $file = fopen("test.txt", "r");
    echo fgetc($file);
    fclose($file);
?>
```

6) fgets():

The fgets() function is used to read a single line from a file.

The example below outputs the first line of the "web1.txt" file:

Example:

```
<?php
    $myfile = fopen("web1.txt", "r") or die("Unable to open file!");
    echo fgets($myfile);
    fclose($myfile);
?>
```

7) copy():

The copy() function is used to copy a file.

Example:

Copy "source.txt" to "target.txt":



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
<?php
    echo copy("source.txt","target.txt");
?>
```

8) feof():

The `feof()` function checks if the "end-of-file" (EOF) has been reached. The `feof()` function is useful for looping through data of unknown length. The example below reads the "webdictionary.txt" file line by line, until end-of-file is reached:

Example:

```
<?php
    $myfile = fopen("web1.txt", "r") or die("Unable to open file!");
    // Output one line until end-of-file
    while(!feof($myfile))
    {
        echo fgets($myfile) . "<br>";
    }
    fclose($myfile);
?>
```

9) unlink():

There is **no delete()** function in PHP.

If you need to delete a file, look at the `unlink()` function.

Example:

```
<?php
    $file = "test.txt";
    if (!unlink($file))
    {
        echo ("Error deleting $file");
    }
    else
```




M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
{  
    echo ("Deleted $file");  
}  
?>
```

FOLDER ACCESS

Creating a Directory in PHP

A new directory can be created in PHP using the *mkdir()* function. This function takes a path to the directory to be created. To create a directory in the same directory as your PHP script simply provide the directory name. To create a new directory in a different directory specify the full path when calling *mkdir()*.

A second, optional argument allows the specification of permissions on the directory (controlling such issues as whether the directory is writable):

Example:

```
<?php  
    $result = mkdir ("/path/to/directory", "0777");  
?>
```

Deleting a Directory

Directories are deleted in PHP using the *rmdir()* function. *rmdir()* takes a single argument, the name of the directory to be deleted. The deletion will only be successful if the directory is empty. If the directory contains files or other sub-directories the deletion cannot be performed until those files and sub-directories are also deleted.

Example:

```
<?php  
    rmdir("college");  
    echo "Directory Deleted";  
?>
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

Renaming a Directory

To rename or move a directory , PHP using rename() function with the old and new path names as arguments.

Example:

```
<?php
    rename("images","pictures");
    rename("/test/file1.txt","/home/docs/my_file.txt");
?>
```

5.2.2.4 Returning a parent Directory:

The dirname() function return the path of the parent directory.

Example:

```
<?php
    echo dirname("c:/testweb/home.php") . "<br />";
    echo dirname("c:/testweb/home.php", 2) . "<br />";
    echo dirname("/testweb/home.php");
?>
```

COOKIES

A cookie is a small file with the maximum size of 4KB that the web server stores on the client computer.

Once a cookie has been set, all page requests that follow return the cookie name and value. A cookie can only be read from the domain that it has been issued from. For example, a cookie set using the domain www.god.com cannot be read from the domain career.gog.com.

Types of Cookies:

1) Persistent cookies:

Persistent cookies are stored in a text file on the clients computer. A persistent cookie is a data file capable of providing websites with user preferences, settings and



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

information for future visits. Persistent cookies provide convenient and rapid access to familiar objects, which enhances the user experience (UX). A **persistent cookie** is also known as a **stored or permanent cookie**.

2) Non-Persistent cookies:

Non-Persistent cookies are stored in RAM on the client and are destroyed when the browser is closed.

Most of the websites on the internet display elements from other domains such as **advertising**. The domains serving these elements can also set their own cookies. These are known as **third party cookies**.

A cookie created by a user can only be visible to them. Other users cannot see its value.

Most web browsers have options for disabling cookies, third party cookies or both.

If this is the case then PHP responds by passing the cookie token in the URL.

The diagram shown below illustrates how cookies work.



Here,

- 1) A user requests for a page that stores cookies
- 2) The server sets the cookie on the user's computer



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

3) Other page requests from the user will return the cookie name and value

Uses of cookie

Cookies are often used to perform following tasks:

- **Session management:** Cookies are widely used to manage user sessions. For example, when you use an online shopping cart, you keep adding items in the cart and finally when you checkout, all of those items are added to the list of items you have purchased. This can be achieved using cookies.
- **User identification:** Once a user visits a webpage, using cookies, that user can be remembered. And later on, depending upon the search/visit pattern of the user, content which the user likely to be visited are served. A good example of this is 'Retargetting'. A concept used in online marketing, where depending upon the user's choice of content, advertisements of the relevant product, which the user may buy, are served.
- **Tracking / Analytics:** Cookies are used to track the user. Which, in turn, is used to analyze and serve various kind of data of great value, like location, technologies (e.g. browser, OS) form where the user visited, how long (s)he stayed on various pages etc.

5.3.3 CREATING COOKIES:

In PHP we can create/set a cookie using the `setcookie()` function.

Syntax:

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

The first argument which defines the name of the cookie is mandatory, rest all are optional arguments.

Parameters:

`setcookie()` has several parameters. Following table discusses those.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

Parameter	Description	Which type of data
name	Name of the cookie.	String
value	Value of the cookie, stored in clients computer.	String
expire	Unix timestamp, i.e. number of seconds since January 1st, 1970 (called as Unix Epoch).	Integer
path	Server path in which the cookie will be available.	String
domain	To which domain the cookie is available.	String
secure	If set true, the cookie is available over a secure connection only.	Boolean
httponly	If set true, the cookie is available over HTTP protocol only. Scripting languages like JavaScript won't be able to access the cookie.	Boolean

setcookie() returns boolean.

Example:

```
<?php
    $cookie_value = "tutorials";
    setcookie("san", $cookie_value, time()+3600, "/home/your_username/",
    "example.com", 1, 1);
    if (isset($_COOKIE['cookie']))
    echo $_COOKIE["san"];
?>
```

Output:

Tutorials

5.3.4 RETRIEVE A COOKIE VALUE

PHP `$_COOKIE` variable is used to retrieve a cookie value.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

Example 1:

```
<?php
    echo $_COOKIE["username"];
?>
```

isset() function is used, to detect cookie is set or not.

Example 2:

```
<?php
    if (isset($_COOKIE["username"]))
        echo "Welcome " . $_COOKIE["username"] . "!\n";
    else
        echo "Welcome guest!\n";
?>
```

You can do the following to view the full *\$_COOKIE* array.

Example 3:

```
<pre>
    <?php print_r($_COOKIE); ?>
</pre>
```

5.3.5 DELETE A COOKIE

To delete cookies, you just need to set a cookie on past date.

The following PHP code will delete, username cookie.

Example:

```
<?php
    setcookie("username", "Jhon", time()-3600);
?>
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

5.4 SESSION

A session is a way to store information (in variables) to be used across multiple pages. Unlike a cookie, the information is not stored on the user's computer. A session is a global variable stored on the server.

Each session is assigned a unique id which is used to retrieve stored values.

Whenever a session is created, a cookie containing the unique session id is stored on the user's computer and returned with every request to the server. If the client browser does not support cookies, the unique php session id is displayed in the URL

Sessions have the capacity to store relatively large data compared to cookies.

The session values are automatically deleted when the browser is closed. If you want to store the values permanently, then you should store them in the database.

Just like the \$_COOKIE array variable, session variables are stored in the \$_SESSION array variable. Just like cookies, the session must be started before any HTML tags.

Why and when to use Sessions?

1. You want to store important information such as the user id more securely on the server where malicious users cannot temper with them.
2. You want to pass values from one page to another.
3. You want the alternative to cookies on browsers that do not support cookies.
4. You want to store global variables in an efficient and more secure way compared to passing them in the URL
5. You are developing an application such as a shopping cart that has to temporary store information with a capacity larger than 4KB.
6. PHP session is used to store information on the server. The data will be available to all pages in one application.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

5.4.1 START A PHP SESSION

A session is started with the `session_start()` function.

Session variables are set with the PHP global variable: `$_SESSION`.

Now, let's create a new page called "demo_session1.php". In this page, we start a new PHP session and set some session variables:

Example:

```
<?php
    // Start the session
    session_start();
    // Set session variables
    $_SESSION["favcolor"] = "green";
    $_SESSION["favanimal"] = "cat";
    echo "Session variables are set.";
?>
```

5.4.2 STORING A SESSION VARIABLE

`$_SESSION[]` is an associative array, which is used to store information in a session. These variables can be accessed during the lifetime of a session.

Example:

```
<?php
    session_start();
    // store session data
    $_SESSION["username"] = "priya";
    $_SESSION["email"] = "priya@example.com";
    //retrieve session data
    echo $_SESSION["username"];
    echo "<br>";
    echo $_SESSION["email"];
```




M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

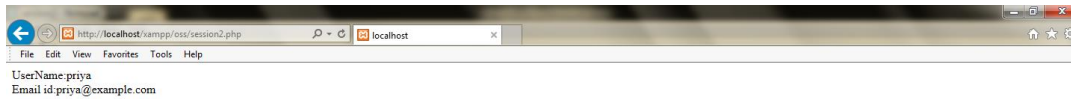
Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

?>

OUTPUT:



5.4.3 DESTROYING A SESSION

A complete PHP session can be terminated by using **session_destroy()** function, which does not require any arguments.

Example:

```
<?php
    session_start();
    session_destroy();
?>
```

If you wish to delete a **single session variable**, then you can use **unset()** function.

```
<?php
    session_start();
    if(isset($_SESSION[username]))
```



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

```
unset($_SESSION[username]);
```

```
?>
```

SAMPLE PROGRAM:

The code below shows how to create and retrieve values from sessions.

```
<?php
```

```
    session_start(); //start the PHP_session function
```

```
    if(isset($_SESSION['page_count']))
```

```
    {
```

```
        $_SESSION['page_count'] += 1;
```

```
    }
```

```
    else
```

```
    {
```

```
        $_SESSION['page_count'] = 1;
```

```
    }
```

```
    echo 'You are visitor number ' . $_SESSION['page_count'];
```

```
?>
```

Output:



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS



COOKIES VS SESSIONS

Both cookies and sessions are used for storing persistent data. But there are differences for sure.

Sessions	Cookies
Sessions are stored on server side.	Cookies are on the client side.
Sessions are closed when the user closes his browser.	For cookies, you can set time that when it will be expired.



M.M.E.S. WOMEN'S ARTS AND SCIENCE COLLEGE

(Affiliated to Thiruvalluvar University)

HAKEEM NAGAR - MELVISHARAM- 632 509

Walaja Taluk, Arcot Block Phone:04172 266167, 266463



E-NOTES / CS

<p>Sessions are safe than cookies. Because, since stored on client's computer, there are ways to modify or manipulate cookies.</p>	<p>Cookies are less safe.</p>
--	-------------------------------