

Index**UNIT-1 PHP Basics and Form Handling****07 to 72**

1.1 Different Kind of Programming Languages

1.2 What is PHP?

1.3 Apache Web Server and It's Flavours

1.4 PHP Syntax and Technics

1.5 PHP printing statement

1.6 PHP Comments

1.7 PHP Variable, Data type and Scope of Variable

1.8 Operators in PHP

1.9 Flow Control Statements

1.10 Array in PHP

1.11 PHP Function

1.12 PHP Form Handling and PHP Global

1.13 Include and require function to include a PHP file in other PHP file

1.14 PHP Date and Time function

1.15 PHP Maths functions

1.16 PHP String Functions

Practical

1. Write a program in PHP to display "Learning PHP" in bold format.
2. Write a program in PHP to demonstrate the use of comments, echo and print.
3. Create a program in PHP to demonstrate the use of If ... Else and switch statements.
4. Create an array named \$sub, assign five elements to it and display the elements assigned using for loop and for each statement.

5. Create an array named \$student that stores 5 elements bounded to different keys and access the same using the key element.
6. Write a program in PHP to demonstrate the use of multidimensional arrays.
7. Create two functions in PHP, parameterized and non-parameterized for implementing string concatenation operation.
8. Write a PHP program to display information of PHP in the browser.
9. Write a program in PHP to sort the array of given 5 numbers in ascending and descending order.
10. Write a program to count the total number of times a specific value appears in an array.

UNIT-2 Problem Solving by Searching

73 to 120

- 2.1 How to Create Database in MySQL
- 2.2 Different Data types in Database
- 2.3 How to create table in MySQL
- 2.4 Different PHP Functions related to MySQL Database
- 2.5 PHP CRUD Operations

Practical

1. Create a form containing two input fields (Name, Email ID) and a submit button. When the user clicks on submit button, the form data should be sent for processing to PHP file ,which should display the welcome message with the email_id on the PHP page. Form data should be sent by HTTP GET/POST method.
2. Write a PHP script for that creates a database named "DB-1" in MySQL.
3. Write a PHP script for creating a product table in the specified database with fields Pro_id, Pro_name, Pro_price, QOH. Also display an acknowledgement for the same.
4. Create a form containing four input fields(Pro_id, Pro_name, Pro_price, QOH) and Submit button. When the user clicks on the submit button an PHP script should be executed which inserts the record in the product table.

5. Create a form containing one input field(Pro_id) and a search button. When the user clicks on the Search button a PHP script should get executed and should display the details of the product for the Pro_id specified.
6. Create a form containing two input fields (Pro_id, QOH) and Update button. When the user clicks on the Update button the quantity of the Pro_id specified should get updated using a PHP script.
7. Create a form containing one input field(Pro_id) and a Delete button. When the user clicks on the Delete button a PHP script should get executed and should delete the record of the product for the Pro_id specified.

121 to 160

UNIT-3 AJAX and Validation

- 3.1 What is AJAX?
- 3.2 AJAX Request Object
- 3.3 XMLHttpRequest Methods
- 3.4 XMLHttpRequest Object Properties

Practical

1. Create a form containing one input field (Name). When the user enters his/her name and as any key is released , the form should display a welcome message for the user.Implement using AJAX.
2. Repeat the above question to demonstrate the use of keydown and keypress events.
3. Write a program for converting a string into uppercase using AJAX.
4. Create a form containing a combobox with some product names as items. Whenever a user selects a particular product from the combobox, it shuold be sent to the server asynchronously (i.e. without pressing submit button). Implement using AJAX.
5. Write a program to demostrate the example of sending items selected from radio and checkbox to server asynchronously.
6. Write a program to validate a blank field and also validate the length of the data entered(i.e. minimum lenght of 5).

7. Write a program to validate an Email ID using regular expression and by using DOM.
8. Write a program that checks a particular stuId already exists in the student(stuId,stu_name,mob,country) table or not. If stuId exists then display a message "User Already Exist. Try another stuId". If it does not exist then add the data in the student table.Implement using AJAX.

UNIT-4 AJAX, PHP, MSQl all combined and Project

161 to 177

Q&L of ICA

matrixableV bnu XAJA oT

4.1 Project Management

4.2 Modules of E-commerce Application

4.3 Logical Steps with screen shot for Admin Module

4.4 Logical Steps with screen shot for UserModule

Practical

1. Create a Shopping Cart Project in PHP Script.
2. Create a Shopping Cart Project in AJAX.

UNIT-1

PHP Basics and Form Handling

1. Write a program in PHP to display "Learning PHP" in bold format.
2. Write a program in PHP to demonstrate the use of comments, echo and print.
3. Create a program in PHP to demonstrate the use of If ... Else and switch statements.
4. Create an array named \$sub, assign five elements to it and display the elements assigned using for loop and for each statement.
5. Create an array named \$student that stores 5 elements bounded to different keys and access the same using the key element.
6. Write a program in PHP to demonstrate the use of multidimensional arrays.
7. Create two functions in PHP, parameterized and non-parameterized for implementing string concatenation operation.
8. Write a PHP program to display information of PHP in the browser.
9. Write a program in PHP to sort the array of given 5 numbers in ascending and descending order.
10. Write a program to count the total number of times a specific value appears in an array

Unit -1 PHP Basics and Form Handling

❖ Introduction:

As we know there are different programming languages are develop day by day according to requirement of application development and for problem solving. Here in this topic we are discussing about web application development programming language, there are different programming languages are support to develop web applications like asp.net, jsp etc. but here we are discuss about open source programming language which is widely used for web application development and it is PHP.

1.1 Different Kind of Programming Languages:

Now a days different types of programming languages are develop for general purpose or a special purpose, but these languages have their specialties, and each language has its own advantages and disadvantages. Programming languages can be classified into a few types; however, these languages support multiple programming styles. Programming languages are used to control the performance of the computer or machine. At present, computer programmer has many choices to choose the language, but there are many differences between programming languages.

1.1.1 Procedural Programming Language:

The procedural programming language is used to execute a sequence of statements which lead to a result. Typically, this type of programming language uses multiple variables, heavy loops and other elements, which separates them from functional programming languages. Functions of procedural language may control variables, other than function's value returns. For example, printing out information.

1.1.2 Functional Programming Language:

Functional programming (FP) is the process of building software by composing **pure functions**. In such programming language programmer will develop different sequence of functions rather than statements like procedural language. Each function takes in an input value and returns a consistent output value without altering or being affected by the program state. These functions complete a single operation and can be composed in sequence to complete complex operations. This programming code is highly modular, because functions can reused throughout the program and it will work with suitable parameters and call at any time where programmer will wont also return equal outputs.

1.1.3 Object Oriented Programming Language:

The focus of OOP programing is not on structure but on modelling a data. It is classify with real world entity and divided into class and objects. It is highly extensible

programming language. It is also provide data security. IT is reusable and accessible by their behaviour. OOP languages permit higher level of abstraction for solving real-life problems. This language follows bottom-up approach.

1.1.4 Scripting Language

A scripting language is a computer language that doesn't have to be compiled. One advantage of scripting languages is that the code can be quickly edited and re-executed. They are used in a variety of manners including on the client side of web applications, on the server side of web applications. A script, short for scripting language, is a programming language like any other, and may be similar in nature to other languages such as C, C++ or Java. The primary difference between a scripting language and other programming languages is that a program written in a scripting language is interpreted rather than compiled. In interpreted languages, the text we create that contains our commands is read by an interpreter that does the conversion to machine code itself, as it is running the script. The text here is still human-readable and does not have to be recompiled if a change is made. Such kind of scripting languages are popular now a days JavaScript, PHP, Python, ruby, Perl.

1.2 What is PHP?

PHP stands for (**Hypertext Preprocessor**), IT works with Hypertext Transfer Protocol. It is a powerful scripting language for developing a web applications. IT is used to develop dynamic and interactive webpages. IT is open source Language, why it is open source because we don't have to buy any license version of PHP also it is freely update all the latest version of PHP. We can directly download it and use it for developing an application. PHP is an HTML-embedded scripting language. We can use PHP syntax inside the HTML tag also we can write HTML tag inside the PHP program syntax. IT is also known as a Server-side Scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. So anyone who has basic programming knowledge can switch over on this language. PHP scripts are executed on the server. It is a interpreted language, means while you execute PHP program the syntax of PHP are interpreted by webserver in form of request on webserver and give response. PHP supports many databases MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.

1.2.1 Why PHP?

It takes information from clients or user and stores it or matches it with database. It redirects from one webpage to another based on action and also bring value from one page to another. It can set session and cookies for user authentication and user's unique identification. IT retrieve XAML data and Json data. It is also work with JavaScript for handle clients request on its own browser. IT support web-services and also call api.

1.2.2 Advantages of PHP:

Open Source: It is open source, means it is free of cost to download and ready to use from its official website.

Cross Platform: we can install PHP in any operating system like Windows, Linux etc. It remains same code if you transfer source code from any operating system.

Easy to Learn: PHP scripting language is easy to understand. Its all syntax is borrowed from basic C language so user can easily understand it and also well organised. All the inbuilt functions are also easy to understand.

Flexible Language: This scripting language more flexible than other scripting language. Once you finish your project and then you want to make changes in that project you can easily work on that project to rebuild. A developer does not have to write fresh codes or command functions, as changes to the existing codes and functions can be done and used.

Efficient Performance: Depending on how the web developer codes, PHP has the potential to turn in an efficient language. It is scalable when used for writing codes and can also be used for creating a large number of applications. It is the programming language of choice when a website has several webpages.

Embedding: PHP is easy to embed with HTML or other Scripting language like JavaScript, JQuery and Json. We can convert static web application in to dynamic application by embedding PHP with other Scripting language.

API Call: we can use PHP language to call any API in our web application like, Google map, plugins etc. also it is helpful to create API for mobile application development.

Database connectivity: PHP is easily connected with the database and make the connection securely with databases. It has a built-in module that is used to connect to the database easily. There are many web applications, which require strong programming language with a good database management system. PHP and its database connection solve the purpose for development of web applications. It reduces the time to connect to a database management system as well. Multiple databases can be integrated with PHP.

1.2.3 Disadvantages of PHP:

Security: Since it is open sourced, all people can see the source code. If there are bugs in the source code, it can be used by people to explore the weakness of it.

Not suitable of large applications: It will be difficult to use it for programming huge applications. Since the programming language is not highly modular, huge applications created out of the programming language will be difficult to maintain.

Weak Type: It has a weak type, which can cause incorrect data and knowledge to user.

Poor Error Handling Method: The framework has a bad error handling method. It is not a proper solution for the developers. Therefore, as a qualified PHP developer, you

will have to overcome it. Using more features of PHP framework and tools cause poor performance of online applications.

1.2.4 What is PHP File?

PHP file contains PHP program syntax HTML tags and other supportive scripting language. PHP files are interpreted by browser, after interpretation it will show the result in form of HTML design or Plain text. PHP programming files are save with extension “.php”, “.php3” or “.phtml”.

1.3 Apache Web Server and It's Flavours:

Apache is a HTTP server is free to install in any operating system, it is also an open source application. It delivers web content through the internet. Apache is just one component that is needed in a web application stack to deliver web content. One of the most common web application stacks involves LAMP, or Linux, Apache, MySQL, and PHP. Apache is the web server that processes requests and serves web assets and content via HTTP. Apache server has three different platforms are available based on Operating system support. IT works on any operating System just you have to download the Apache server's application in your system based on your Operating system. There are three Platforms are available. All three platforms are developed under Apache so it is also Open Source.

- Windows Apache MySQL PHP/Perl/Python (WAMP)
- Linux Apache MySQL PHP/Perl/Python (LAMP)
- Cross platform Apache MySQL PHP/Perl/Python (XAMPP)

We will discuss all one by one in detail

1.3.1 Windows Apache MySQL PHP/Perl/Python. (WAMP):

IT is work with windows operating system because it's first character ‘W’ means it is develop for windows operating system by Apache friends. It allows you to develop web application with windows web development environment. IT includes Apache 2 server configuration files with PHP 5 and MySQL Database. WAMP server is managed database with the help of phpMyadmin and SQLite manager. It will send request from client to server with the help of PHP scripting language. So when you want to start your PHP programing, first of all you want to install WAMP server in your device, no need to download PHP and MySQL individual in your device.

1.3.2 Linux Apache MySQL PHP/Perl/Python. (WAMP)

IT is work with Linux operating system because its first character ‘L’ means it is develop for Linux operating system by Apache friends. It allows you to develop web application with Linux web development environment. IT includes Apache 2 server configuration files with PHP 5 and MySQL Database. LAMP server is managed database with the help of phpMyadmin and SQLite manager. It will send request from client to server with the

help of PHP scripting language. So when you want to start your PHP programming, first of all you want to install LAMP server in your device, no need to download PHP and MySQL individual in your device.

1.3.3 Cross Platform Apache MySQL PHP/perl/Python. (XAMPP)

IT is work with any operating system because it's first character 'X' means it is developed for Cross Platform. Here the meaning of Cross Platform is any Operating System. It allows you to develop web application with any Operating System web development environment .IT includes Apache 2 server configuration files with PHP 5 and MySQL Database. XAMPP server is managed database with the help of phpMyadmin and SQLite manager. It will send request from client to server with the help of PHP scripting language. So when you want to start your PHP programming, first of all you want to install LAMP server in your device, no need to download PHP and MySQL individual in your device.

1.3.4 How to install and work with XAMPP server?

To start work with XAMPP you need to install it first. You can install it from any web application which gives you its exe file. It is easy to install different versions of XAMPP from its official website www.apachefriends.org. After downloading its latest version you need to install the file, after installing a file you will see the XAMPP control panel.

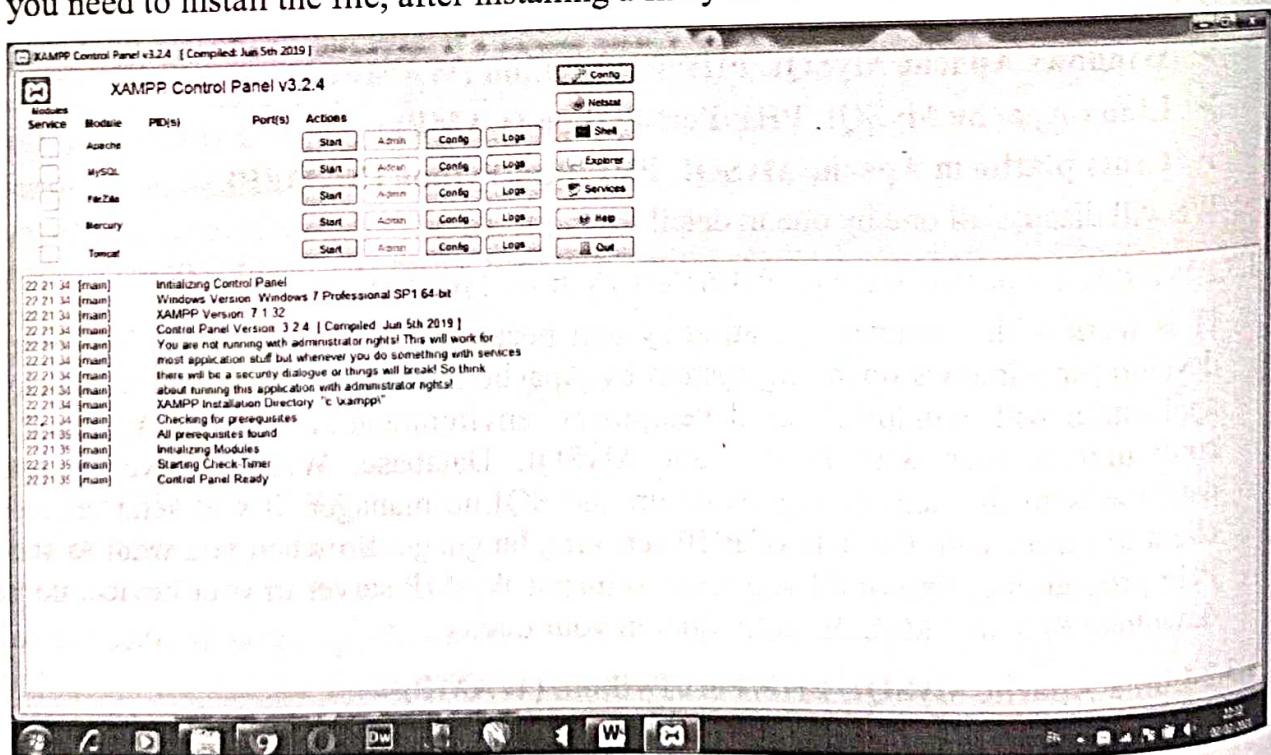


Fig 1.1 XAMP Control Panel.

You can see the different service list in control panel, for PHP programming we need to start apache server's service which will help you on your local machine to execute PHP Scripting pages. Also it will send request from client browser to server or database.

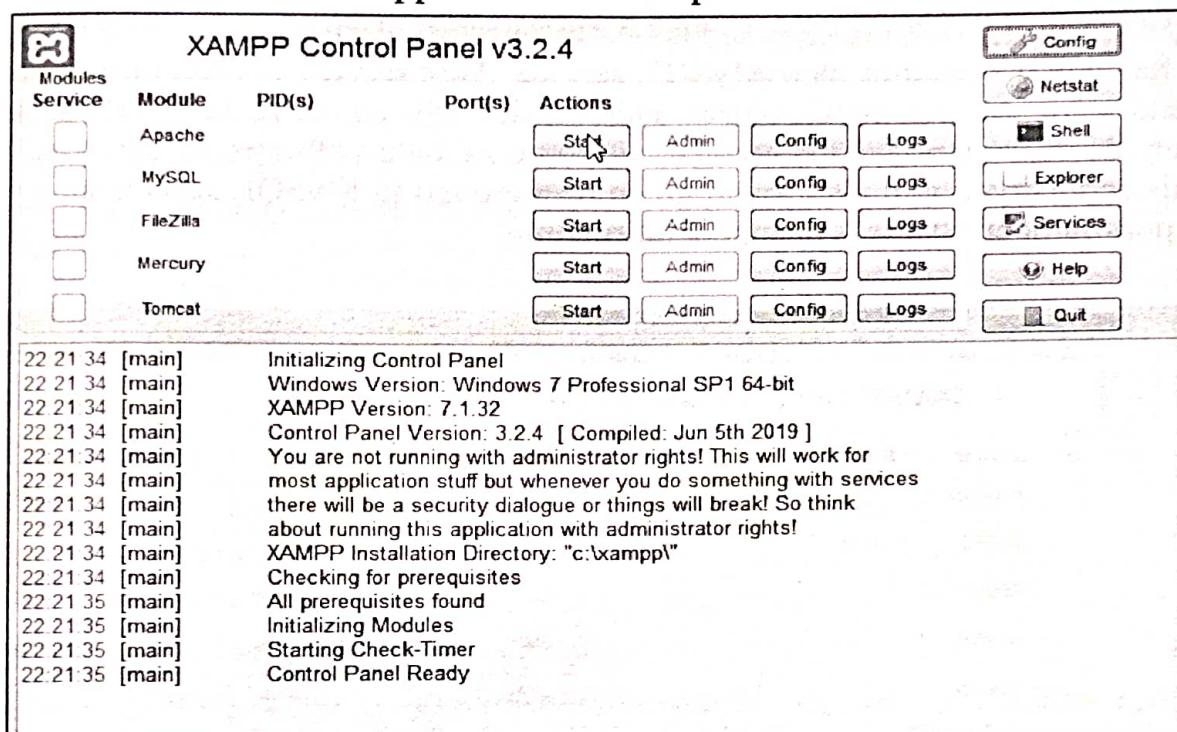


Fig 1.2 Apache and MySQL server Start.

After starting apache you will see the green background on apache service and also will display port number of Apache. It will change the status of apache with running.

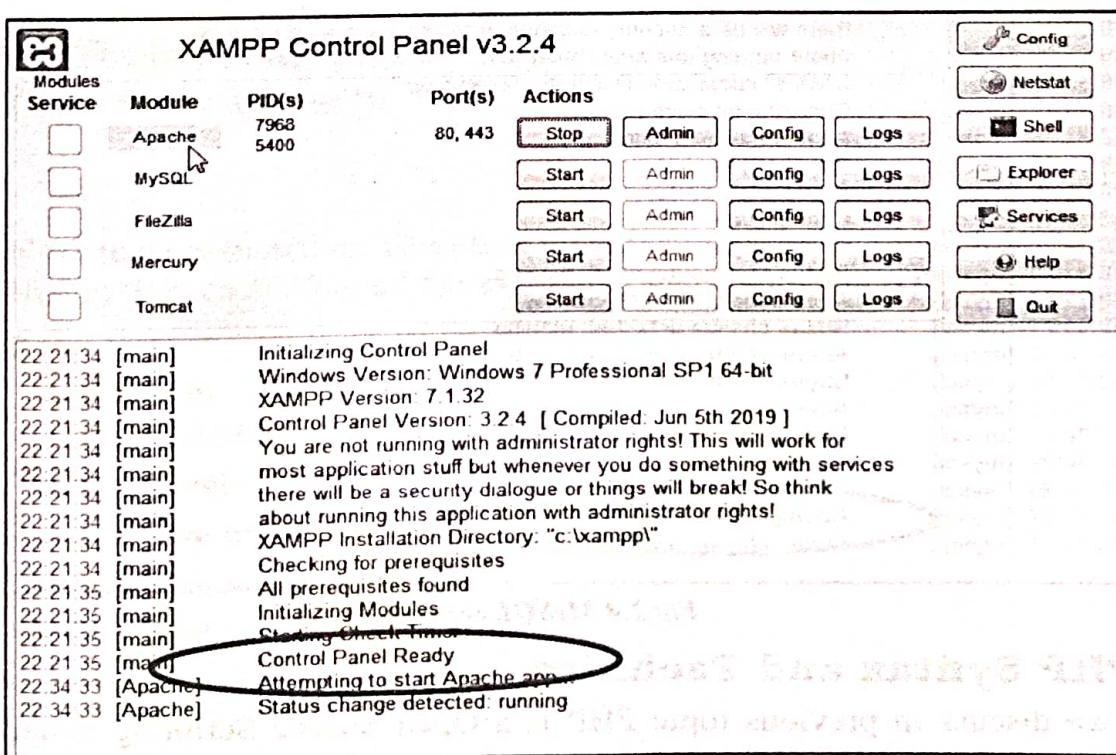


Fig 1.3 Apache server Started.

After this we need to start MySQL service. This service is used only when we want to send some request for Database. This service will communicate with database with the help of query string. This service is also coordinate with apache services. After starting this service you will also found green background in MySQL service also will show you a port number. Status is change with running.

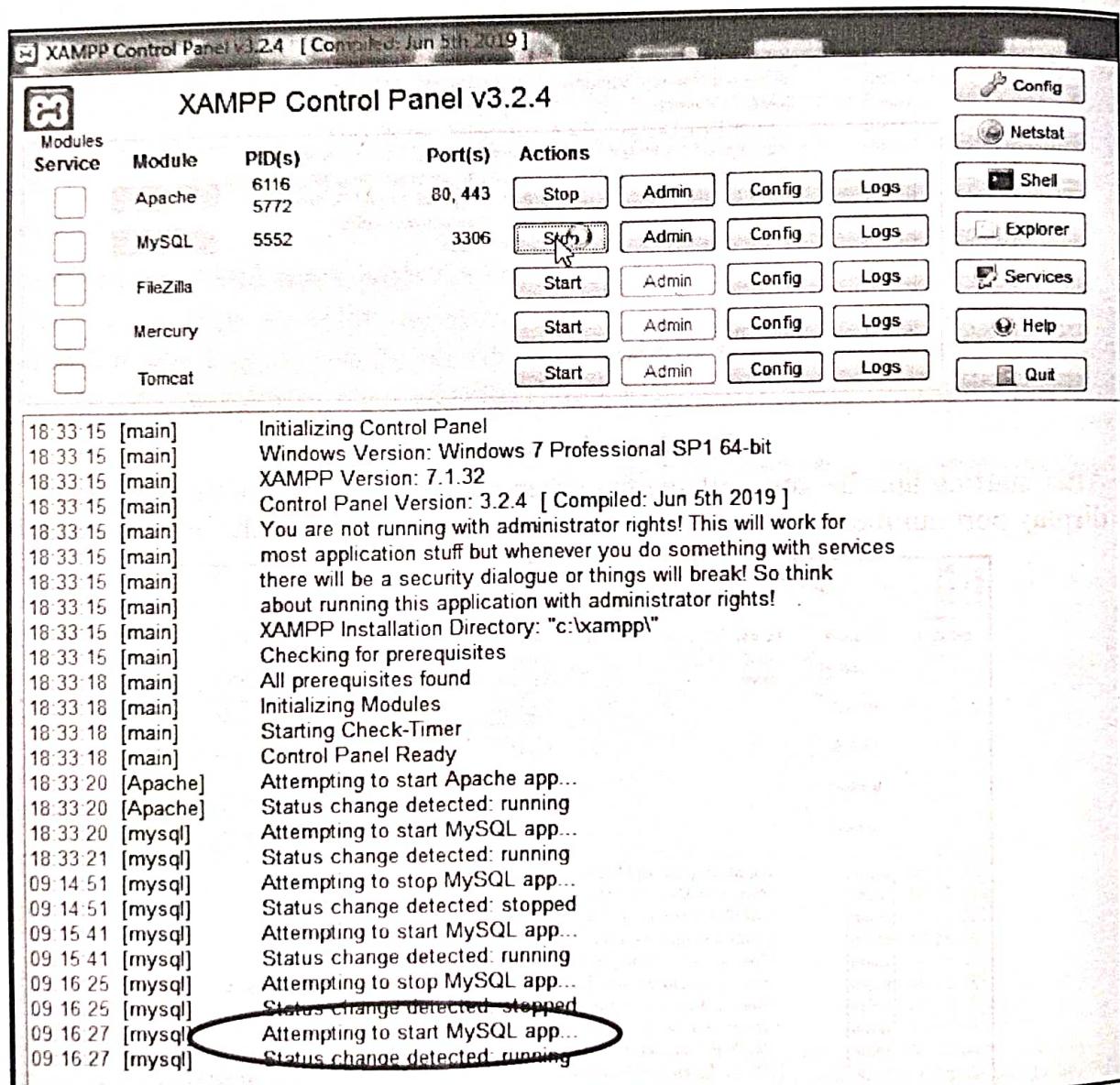


Fig 1.4 MySQL server Started.

1.4 PHP Syntax and Techniques.

As we discuss in previous topic PHP is a Open Source Scripting language, means this language is also start with mark-up tag. We need to write mark-up tag of PHP before

starting of writing syntax. While we start to write PHP program first will start PHP mark-up tag `<?php` also end with mark-up tag `?>`, inside this opening and closing tag we will write our PHP syntaxes. Make sure while you open tag at that time we write PHP but at the time of closing no need to write PHP only use question mark and greater than sign.

Basically PHP syntax is borrowed from ‘C’ programming language so it follows same rules of syntax.

PHP Syntax Rules.

- We can place PHP syntax anywhere in document.
- The default file extension for PHP files is ".php".
- PHP script starts with `<?php` and ends with `?>`.

`<?php`

//statement of php syntax;

`?>`

- PHP syntax are end with semicolon (`;`)
- All the core programming keywords (for example. If, else, for, while, echo etc.) , functions, user define function, classes are case-insensitive. Therefore the three echo statements in the following example are equal.

`<?php`

Echo ("hello student");

echo ("hello student");

ECHO("hello student");

`?>`

- Variables are case-sensitive. Therefore the three variables in the following example are different then each other all have different value and data type.

`<?php`

\$price=10;

\$Price=30.50;

\$PRICE=40;

echo("The price=\$price");

echo("The price=\$Price");

echo("The price=\$PRICE");

`?>`

- PHP whitespace is insensitive, means it will never matter how many number of space you write in syntax or how many tabs you write in one row syntax also not read carriage-return (end of line characters).

Therefore in bellow example it will not read space tab and new line. Between $2 + 2$ it will do addition.

```
<?php  
    $four = 2 + 2; // single spaces  
    $four = 2+ 2; // spaces and tabs  
    $four =  
        2+  
        2; // multiple lines  
?>
```

1.5 PHP printing statement:

PHP has different printing statements. We can use following printing statements in PHP **echo, print , printf() and print_r()**.

Note: we will discuss **print_r()** function latter at the time of array.

1.5.1 Echo:

- This function is used to print one or more statements.
- This function is faster than print function. Because it is not return any value.
- echo is not consider as a function so no need to write parentheses at the time of write echo statement.
- When you write any string message you need to write this message in " " double quotes.
- If you write to print any number or any variable than no need to write double quotes.

```
<?php  
    echo "hello World";  
    echo 123;  
    echo $no;  
?>
```

1.5.2 Print:

- This function is also same like echo.
- It is return value after function is call.
- It is little slower than echo.
- Print is not considering as a function so no need to write parentheses at the time of write print statement.

- When you write any string message you need to write this message in "" double quotes.
- If you write to print any number or any variable than no need to write double quotes.

```
<?php  
    print "hello Asia";  
    print 100;  
?>
```

1.5.3 Printf()

- This function is used for formatted output.
- IT is used same like c programming printf() function.
- It is return value after function is call.
- Printf() is a function so parentheses are compulsory at the time of write printf() statement.
- When you use printf() function with variable then you have to define data type symbol with % sign as an argument of printf() function.
- We can use more number of argument in printf().
- After the argument complete we use coma (,) and write variable list in order of printing.
- parameter values are as follow
 - %d integer
 - %f for floating point
 - %s for string
- %b binary value
- %e for scientific notation
- %o for octal
- %x for hexadecimal

```
<?php  
    $a=5;  
    $b="Diya";  
    printf ("hello Friends");  
    printf(" %s age is %d years.", $b, $a);  
?>
```

1.6 PHP Comments:

Like C programming language it has two types of comments. One is Single line comment and second is Multiline comment. IT also use same character sets for comment, for single line comment we use two **forward slash** (/) and also use **Hash (#)** sign. Any syntax are written with this sign are considering as single line comment it will not interpret by browser, it will ignore this syntax.

```
<?php
```

```
//echo ("single line comment using forward slash");
```

```
# echo "single line comment using Hash sing";
```

```
?>
```

For multiline comment it use forward slash and star /*) and End with same order star and forward slash (*/). In between any syntax is written are considering as multiline comment it will not interpret by browser, it will ignore this syntax.

```
<?php
```

```
/* echo "single line comment using forward slash";
```

```
echo "single line comment using Hash sing"; */
```

```
?>
```

1.7 PHP Variable, Data type and Scope of Variable:

PHP Variable has also same data type like other programming languages. Int, float, double, char, string etc. In PHP variable is declare with \$ sign, means while you write \$ sign before any name it will consider as a variable name in PHP.

Rules of Declaring Variable:

- Variable start with character or underscore sign.
- It cannot start with numbers.
- Variable name contains only alpha numeric and underscore sign.
- \$a1,\$A_1,\$_123 are the valid variable declaration.
- \$1a \$1_A are invalid variable declaration.
- In PHP no need to give any data type at the time of declaring variable. Here variable is work as a container.
- PHP variable data type interpreted at run time by browser observing its value. So no need to define int, float, char before variable name.
- For Example \$a=100, \$b="student", \$c=50.90.
- PHP variables are case-sensitive.
- \$No, \$No and \$NO are there different variables.
- In PHP we can use same variable more than one time in hole program. Or we can also use same variable name with different value.

```
<?php
```

```
$age=7;
```

```
$name="Diya";
```

```
$h="3.25";
```

```
$c='A';
```

```
$_wt=12.70;
```

```
echo $name;  
echo $h;  
print $age;  
echo $_wt;  
echo $c;  
printf("%s age is %d years",$name,$age);  
printf("%s height is %f feets",$name,$h);  
?>
```

There are three different scope to declare variable in PHP. Local, Global and Static.

1.7.1 Local Scope: Variable declare inside the function is consider as a local variable. That variable cannot access outside the function. If you declare variable which is same name outside the function, so local variable and outside function variable both are consider as a different variable.

```
<?php  
$no=90;  
function printno()  
{  
    $no=80;  
    echo "Local number=". $no;  
}  
printno();  
echo "Outside function number=". $no;  
?>
```

1.7.2 Global Scope: variable which is declared outside the function is consider as a global variable, it is access outside the function directly. If you want to access global variable inside function then you need to define **global** word before the variable name then it can access same global variable inside function.

```
<?php  
$no=100;  
function printno()  
{  
    global $no;  
    $no=120;  
    echo "global variable inside function=". $no;  
}
```

```

printno();
echo "global variable value=". $no;
?>

```

1.7.3 Static variable: It is the characteristic of PHP to delete the variable, ones it completes its execution and the memory is freed. But sometimes we need to store the variables even after the completion of function execution. To do this we use static keyword and the variables are then called as static variables.

```

<?php
function static_var()
{
    // static variable
    static $num = 5;
    $sum = 2;
    $sum++;
    $num++;
    echo $num, "\n";
    echo $sum, "\n";
}
// first function call
static_var();
// second function call
static_var();
?>

```

1.8 Operators in PHP:

Operators are used in programming and scripting language to do some arithmetic calculations also used for some logical functionality. In PHP operators are same as other programming language and also it used in same manner. Operators perform operation on variable and values. In any kind of operation there are two things are used operator and operand. For example $5+5$ in this statement 5 is consider as a operand and $+$ is consider as an operator. Operators are some special characters that are used from a different perspective for different meninges in program logic. Operators are divided into two different categories.

Unary prefix operators, which precede a single operand.

Binary operators, which take two operands and perform a variety of arithmetic and logical operations.

Types of Operator:

- Arithmetic
- Comparison
- Logical
- Assignment
- Conditional
- Increment/Decrement
- String

1.8.1 Arithmetic Operator:

Arithmetic Operator is used to do some arithmetic Operations in program like addition, subtraction, multiplication, division, modulation. This operator is belonging to binary operator where two operands and one operator are used.

Operator	Description	Example
+	Used for Addition/Sum of two number or two variables.	10+90; \$a+\$b;
-	Used for Subtraction of two number or two variables.	100-50; \$a-\$b;
*	Used for Multiplication of two number or two variables.	50*50; \$a*\$b;
/	Used for Division of two number or two variables.	50/2; \$a/\$b;
%	Used for get Reminder of two number or two variables.	50%4; \$a%\$b;

```

<?php
$a=40;
$b=60;
$sum=$a+$b;
$subtraction=$a-$b;
$multiply=$a*$b;
$division=$a/$b;
$modulo=$a%$b;
echo("Sum=$sum <br>");
echo("Subtraction=$subtraction <br>");
```

```

echo("Multiplication=$multiply <br>");  

echo("Division=$division <br>");  

echo("Modulation=$modulo <br>");  

?>

```

1.8.2 Comparison Operator:

This Operator is used to compare two operands and return logical value based on comparison. It return Boolean value based on Comparison (True/False). We can use any numerical or string operand to compare. Following are the Comparison Operators. It is also belonging to binary operator where two operands and one operator are used.

Operator	Description	Example
Double Equals (==)	Used for compare two operand. Returns true if both are equal. This operator ignore the data type of variable and value at the time of Comparison	\$a=3;\$b=3 \$a==\$b; true \$a=='3'; true \$a=="3" ;true
Not Equals (!=)	Used for compare two operands. Returns true if both are not equal.	\$a=3;\$b=4; \$a!=\$b; true \$a<>\$b; true
Not Equals (\neq)		
Equals(==)	Used for compare two operands. Returns true if both are equal. This operator compares data type of variable and value at the time of Comparison, if both are same then it will return true.	\$a=5;\$b=5; \$a==\$b; true \$a=="5";False
Greater than (>)	This operator is used to compare two operands. It will return true if left operand vale is greater than right operand value;	\$a=20; \$b=10; \$a>\$b; true.
Less than (<)	This operator is used to compare two operands. It will return true if left operand vale is less than right operand value;	\$a=5;\$b=20; \$a<\$b; true.
greater or Equals (\geq)	This Operator is used to compare greater and equals both at same time. It will return true if left operand value is either greater than or equals to the right operand value	\$a=10; \$b=4; \$a>=\$b; true; \$a>=10; true;
Less or Equals (\leq)	This Operator is used to compare less and equals both at same time. It will return true if left operand value is either less than or equals to the right operand value.	\$a=10; \$b=5; \$b<=\$a; true; \$a<=10; true;

```

<?php
    $a=20;
    $b=10;
    if($a>=$b)
    {
        echo("A is greater or Equal to B");
    }
    else if($a<=$b)
    {
        echo("B is Greater or Equal to A");
    }
    else if($a===$b)
    {
        echo("A and B are same and their Data types are also same");
    }
    else if($a===$b)
    {
        echo("A and B are same but their Data types are not same");
    }
    else
    {
        echo("A and B are not same");
    }
?>

```

1.8.3 Logical Operator:

Logical Operator is used to check logical value. They return Boolean value. It is used to compare two different Logic or Conditions.

Operator	Description	Example
Logical And (&)	This operator returns true if both expressions are fulfil it's condition.	\$a>\$b && \$a>\$c;
Logical Or ()	This operator returns true if either first or second one of the expression is fulfil it's condition.	\$a>\$b \$a>\$c;
Logical Not (!)	This operator is reversing the Boolean result of the Operand or Condition.	!(\$a>\$b) return true !(\$a<\$b) return false

```

<?php
    $a=''; $b='';
    if($a>100 || $a<=200 && $a!=0)
    {
        echo ("A is between 100 to 200");
    }
    if($b>=50 && $b<=100)
    {
        echo ("$b is between 50 to 100");
    }
    if(!$a && !$b)
    {
        echo ("A and B is Empty");
    }
?

```

1.8.4 Assignment Operator:

This operator is used to assign value to a variable. It will always assign new value in left side of operator variable. We can also use this operator to assign some computed Expression result value to the variable.

Operator	Description	Example
Equalsto (=)	This operator is used to assign direct value to the left side operand. It is also used to assign some Computed Expression result to the left side operand.	a=3;b=3 a=b+10;
Plus and Equals to (+=)	This operator is first of all Add value of left side and right side operand and then assign result to left side operand. It is equivalent of $a=a+b;$	a=3;b=4; a+=b; now $a=7$
Minus and Equals to (-=)	This operator is first of all Minus value of left side and right side operand and then assign result to left side operand. It is equivalent of $a=a-b;$	a=5;b=5; a-=b; now $a=0;$
Minus and Equals to (*=)	This operator is first of all Multiply value of left side and right side operand and then assign result to left side operand. It is equivalent of $a=a*b;$	a=5;b=5; a*=b; now $a=25;$
Minus and Equals to (/=)	This operator is first of all Divide value of left side and right side operand and then assign result to left side operand. It is equivalent of $a=a/b;$	a=5;b=5; a/=b; now $a=1;$

Minus and Equals to (%)	This operator is first of all Modulo value of left side and right side operand and then assign result to left side operand. It is equivalent of $a=a*b;$	a=5;b=5; a%b; now a=0;
-------------------------	--	------------------------------

<?php

```
$a=20; $b=10; $c=0;
```

```
$c+=$a;
```

```
echo("sum=". $c."<br>");
```

```
$c-=$b;
```

```
echo("Subtraction=". $c."<br>");
```

```
$b*=$a;
```

```
echo("Multiplication=". $b."<br>");
```

```
$b/=$c;
```

```
echo("Division=". $b."<br>");
```

```
$b%=$c;
```

```
echo("Modulo=". $b."<br>");
```

?>

1.8.5 Conditional or Ternary Operator:

Ternary operator starts with conditional expression followed by ? operator. Second part z(after ? and before : operator) will be executed if condition turns out to be true. If condition becomes false then third part (after :) will be executed.

<?php

```
$a = 10;$b = 5;
```

```
$c = $a > $b? $a : $b;
```

```
echo("Value of C=$c");
```

?>

1.8.6 Increment or Decrement Operator:

These operators are also known as unary operator, means it has only single operand. This operator will increment or decrement the value of operand by 1, if you use ++ sign then it will add 1 in actual value, if you used -- then it will decrease the value by 1 in actual value. We can add this operator before or after the operand.

- If you use the ++ operator as prefix like: ++variable. The value of variable is incremented by 1 then, it returns the value.
- If you use the ++ operator as postfix like: variable ++. The original value of variable is returned first then, variable is incremented by 1.

```
<?php
```

```
    $a=5;  
    echo $a++;  
  
    $c=$a;  
    echo $c;  
    --$c;  
    $d=$c;  
    echo($d);
```

```
?>
```

1.8.7 Concatenation operator:

In PHP for merge two string , to merge string with number or to merge string with variable we use ‘.’ operator. This operator always merge string data with other data types.

```
<?php
```

```
    echo ("hello". "Student");  
    echo ("Your Registration no=".123);  
    $a=10; $b="Diya Patel";  
    echo($b." your age=". $a);
```

```
?>
```

1.9 Flow Control Statements:

Flow controls is one of the powerful tools in all programming language and it is also used in PHP .Flow control statements are used to build intelligent and logic in your web page .It is used to constructing a script for decisions and repeating a task until specified criteria are not met. It is always return Boolean value while it fulfils the criteria. PHP use two types of Flow Control Statements.

1. Conditional Flow Control Statements.
2. Looping Flow Control Statements.

1.9.1 Conditional Flow Control Statements:

This types of flow control use with decision. In our daily life we have to make decisions. Like which dress to wear, which task we have to give first priority and many more. Such kind of decision statements also used in programming language as well as scripting language. Decision making statement always return Boolean value either true or false depend on that program flow will execute. In PHP we use 4 different types of conditional statements:

1. if
2. if.... Else
3. nested if
4. Switch

1) if Statement:

The if statement is a fundamental conditional statement in programming language. This statement is verified a condition or set of conditions. After verification it defines certain statements or block of statements will be executed or not. If certain condition is true then block of statement inside that condition is execute otherwise not execute.

Syntax:

If (condition)

{

 Body of if block...;

}

<?php

 \$age=65;

 if(\$age>60)

 {

 echo("Age is greater than 60");

 }

?>

2) if ...else Statement:

It is a similar like If statement. But in If statement only one block of If condition is execute. If condition return false then it is execute second block of statement where we use else block. This will execute while if block return false, means else block return true. It is always used in group. It will execute else block bases on true and false of if condition.

Syntax:

if (conditional statements)

{

 Body of if block

}

else

{

 Body of else block

}

<?php

 \$A=5;

```
?> if($A%2==0)
{
    echo("A is Even no");
}
else
{
    echo("A is Odd no");
}
```

3) Nested if Statement:

PHP allows us to add if statement inside another if or else statements. It is called nested if statement. We can use more than one if statements inside another if statements. It will check conditions one by one in order of nesting. Nested if is also used to check dependent conditions.

Syntax:

```
if (condition)
{
    if (condition)
    {
    }
}
else
{
    if (condition)
}
```

```
<?php
$A=12;
$B=5;

if ($A!=$B)
{
    echo("A and B are not same<br>");
    if($A>$B)
    {
        echo("A is greater than B");
    }
    else
    {
```

```

        echo("B is greater than A");
    }
}
else
{
    echo("A and B are same");
}
?>

```

4) Switch Statement:

In previous topic we learn to check multiple conditional statements, but it quite difficult to develop a nesting structure of multiple conditions. There is another option we have to check multiple conditional statements and it is easy to develop. Switch statement compare variable with multiple case. Once the case is match that block of statements will execute. In program we need to define one variable which use as an argument of switch() that variable value is match with all the cases under switch statement. Each case is design with different number or characters. It is used as matching of variable value. If Case is not match then default block is execute.

Syntax :

Switch(expression)

{

case value1:

 statements;

 break;

case value2:

 statements;

 break;

default:

 statements;

}

<?php

 \$sw='B';

 switch(\$sw)

{

 case 'A':

 echo("Select Block A");

 break;

 case 'B':

 echo ("Select Block B");

 break;

 case 'C':

 echo ("Select Block C");

```

echo ("Select Block c");
default:
echo ("case not match in default block");
}
?>

```

1.9.2 Looping Flow Control Statements:

In PHP we can use different four types of looping statements. This four types are derived in two different categories Exit Control loop and Entry control loop. Exit control loop means such looping statement will check the condition of loop at the time of exit from loop body. If condition is true then it will come again in loop body. IN this category the body of loop will execute at least one time. Entry control loops are such looping statement will check the condition of loop at the time of entry in to the loop body. if condition is true then it will enter into the body of loop otherwise it will not go inside the body and stop the looping.

Exit control loop is Do while loop in this loop the body of loop will execute first then it will check while condition if it is true then it will come again in to the body otherwise it will come out from loop and execute next statement of program.

Exit Control Loop

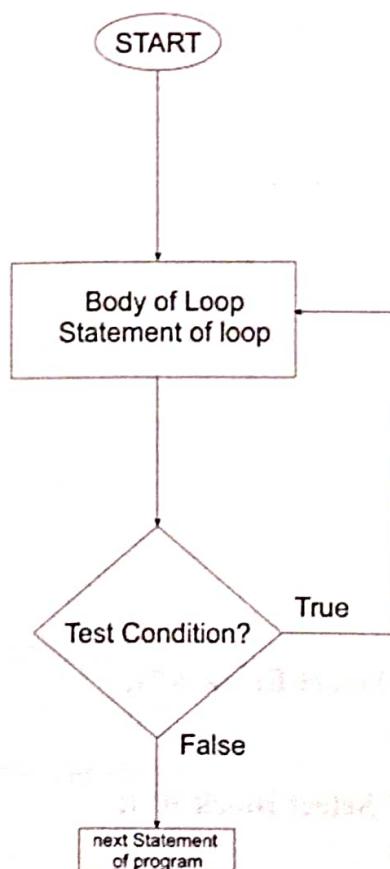


Fig 1.5 Exit Control Loop Structure.

Entry control loops are while loop, for loop and foreach loop. In these loops the body of loop will not execute first, it will check condition if it is true than it will enter in to the body otherwise it will come out from loop and execute next statement of program. Every time it will check condition first then execute body of loop.

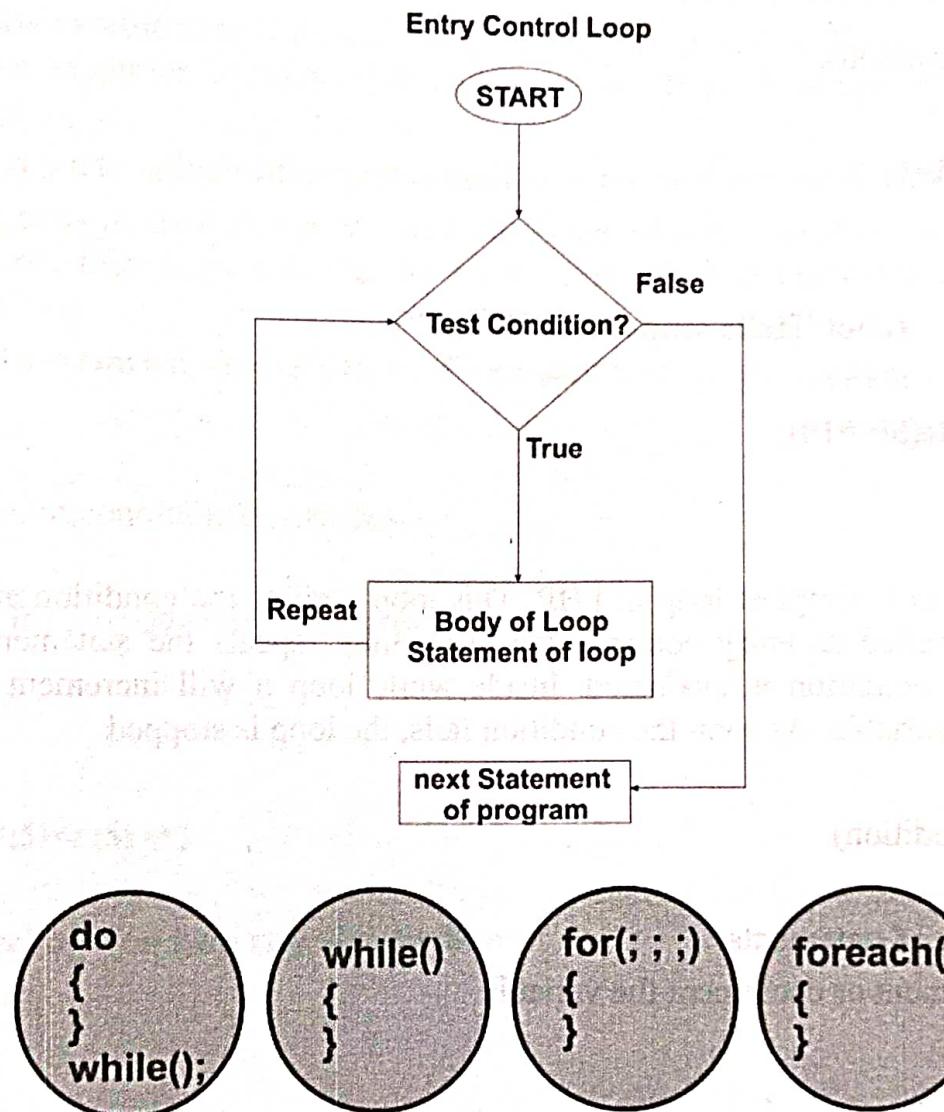


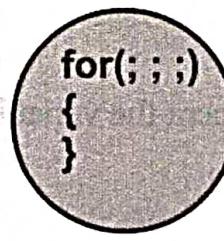
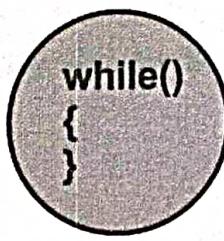
Fig 1.6 Entry Control Loop Structure.

- **Do While:**

Do while loop is evaluate the condition at the end of each loop iteration. In do while loop it will execute block of code at least one time. Then the condition is evaluated, if the condition is true, the statement is repeated as long as the specified condition evaluated to is true. Before while condition inside do it will increment or decrement the value of variable until the condition become true.

Syntax:

do



```
{ Block of code execute repeat
    Increment or decrement;
}
while(condition);
<?php
    $i=1;
    do
    {
        echo("Hello student no $i<br>");
        $i++;
    }while($i<=10);
<?
```

- **While:**

While loop is simplest loop in PHP. This loop verifies the condition at the time of entry so it is called as entry control loop. This loop repeats the statements as long as the specified condition is evaluated. Inside while loop it will increment or decrement the value of variable. As soon as the condition fails, the loop is stopped.

Syntax:

```
while(condition)
```

```
{
```

Block of statements;

Increment or decrement the variable

```
}
```

```
<?php
```

```
    $i=1;
```

```
    while($i<=5)
```

```
{
```

```
        echo("PHP while loop $i <br>");
```

```
        $i++;
```

```
}
```

```
?>
```

- **For:**

For loop is entry control loop means it will execute the block of statement after checking the conditional statement. In for loop statement divide in three different parts. First part is initialization, second is conditional part and third is increment or decrement part.

First two parts execute then if condition return true then loop transfer control inside body of loop. After execution of body third part is execute. Each three parts are separated by semicolon (;).

Initialization part: -initialization part is used to initialize the variable with value.

Condition part:- in condition part it will check the initialize variable met the condition, if it return true then control is transfer to the body of loop otherwise it will stop the execution of loop.

Increment/Decrement: -in this part it will increment or decrement the value of initialize variable.

Syntax:

```
for(initialization;condition;increment)
```

```
{
```

Body of for loop execute statements.

```
}
```

```
}
```

```
<?php
```

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

```
{
```

```
echo("Foo loop run $i time<br>");
```

```
}
```

```
?>
```

- **Foreach:**

This loop is used with only array data type in PHP. It is easy to traverse in array using this loop. It will select each element of array instead of their index or key and access the value. In foreach loop we do not need to increment the value. There are two argument pass in foreach, first one is an array variable which we want to access and second is user define variable which will access and receive the value of all the array elements.

There are two syntaxes are used.

Syntax :

```
foreach(array variable as user define variable)
```

```
{
```

```

}
foreach(array variable as user define key=> value)
{
}
<?php
$Sa=array(1,2,3,4,5);
foreach($a as $ans)
{
    echo($ans."<br>");
}
$product=array("name"=>"mouse","color"=>"black","price"=>150);
foreach($product as $p=>$v) //access key and value both.
{
    echo("[ ".$p." ]=>".$v."<br>");
}
foreach($product as $p=>$v)
{
    echo($p."<br>"); //print only key part of array
}
foreach($product as $v) //access only value part of array
{
    echo($v."<br>");
```

?>

1.10 Array in PHP:

As we know in real life data are come in multiple formats. These all data are store under common variable. In PHP array variable is used to store such kind of data in single variable. PHP array is different than other programming language like c, c++, java etc.in PHP array is declare without data type, means like PHP variable no need to define any data type with array variable also. As we know array is store only similar data type values but in PHP we can store any data type value in array, because there is no fix data type at the time of declaration of array variable. We can declare array in two different ways. we can use short syntax like variable name then initialize values of array in square bracket.

Syntax:

```
$arr=[1,3,5,7,9];
```

```
$a=['a','b',11,22,"abcd"];
```

We can use array function to declare array in PHP. In this function we will pass the values of array. It is the most popular method to declare an array in php.

Syntax:

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

```
$std=array("miten","miten@gmail.com",30,'M',"Ahmedabad");
```

There are three types of array in PHP

- Index Array
- Associative Array
- Multidimensional Array

1.10.1 Index Array:

This array is use numeric index to store a value. It store value in linearly. the first value of index array is place on 0 index then second value is on 1 index and so on up to n umber index will define automatic based on number of values in array.

We can access index array value by using index of array element.

For example:

```
<?php  
$arr=array(11,22,33,44,55,66);  
Echo $arr[3] ;  
?>
```

In above example array is define with \$arr and there are 6 elements in array first element on 0 index and so on. I have access 3rd index array element so it will print 44 on screen.

1.10.2 Associative Array:

In PHP this array is flexible to store value. This array is use key name as an index of array and with key it store value. It is a pair of key and value. It is define with => equals to and greater than sign. Left side of this sign we define key and right side of this sign we define value. For example we store carname and price in array then we define array as follow.

```
$carlist=array("ertiga"=>900000,"i20"=>600000,"scorpio"=>1000000);
```

if we want to access this car then we access it with it's key name, it will give value of that key name. for example we print this array.

```
echo $carlist["ertiga"];
```

`echo $carlist["i20"];`

`echo $carlist["scorpio"];`

when we access array in above method then it will print the value. `echo $carlist["ertiga"];` will print 900000.

For example:

```
<?php
$student=array("name"=>"Suhit patel","age"=>32,"height"=>5.4);
echo $student["name"];
echo $student["age"];
echo $student["height"];
?>
```

1.10.3 Multi-dimensional Array:

This array is also using associative array. In this array we develop other nested array. This array allows us to store relative data in common group. Sometimes when we want to store more than one key with different values we use multi-dimensional array. In multi-dimensional array dimensions are used for indices. If you use two dimensional array then there are two indices are use. In PHP multi-dimensional array can work with index as well as with key, in bellow example Columns name, Salary and designation are key and ram, 2000, Manager are on index 0 or in row 0.

Syntax:

```
$arr=array("key1"=>array("value1","value2","value3"),
           "key2"=>array("v1","v2","v3"));
```

Row/Colum	Colom 1	Colom 2	Colom 3
Row 0	Name	Salary	Designation
Row 1	Ram	20000	Manager
Row 2	Mohan	15000	Sr.Developer
Row 3	Shiva	30000	Team Leader

`<?php`

`$emp=array("Name"=>array("ram","Mohan","Shiva"),`

`"Salary"=>array(10000,20000,30000),`

`"Designation"=>array("Manager","Sr.Developer","Team Leader"));`

`echo $emp["Name"][1];`

`echo $emp["salary"][1];`

```
echo $emp["Designation"][1];
?>
```

In above example we have make two dimensional array where name and salary are the colom name and ram,20000 is the row 1 record. When we access name and salary we define \$emp["Name"][1] where Name is Colom name, and 1 is an index of record.

1.10.4 Important inbuilt functions in Array.

- **print_r()** : This function is used to print hole array with its index and value if your array is associative array then it will print hole array with key and value.we will pass an array variable as an argument of print_r() function.we can use this function for code tracing, if we fetch record from data base then we want to confirm that data are fetch in order, then we print hole array using this function.

```
<?php
```

```
$arr=array(1,2,3,4,5,6);
```

```
print_r($arr);
```

```
?>
```

- **count()**: This function is used to count the total no of elements in array.it is also used to get the length of array. In count() function we pass array variable as an argument.

```
<?php
```

```
$a=array(1,2,3,4,5,6,7);
```

```
$len=count($a);
```

```
echo "<br>".$len;
```

```
for($j=0;$j<count($a);$j++)
```

```
{
```

```
echo $a[$j];
```

```
}
```

```
?>
```

- **sizeof()**: This function is used to count the total no of elements in array.it is also used to get the length of array. It is an alias of count() function. In sizeof() function we pass array variable as an argument.

```
<?php
```

```
$a=array(1,2,3,4,5,6,7);
```

```
$len(sizeof($a));
```

```
echo "<br>".$len;
```

```
for($j=0;$j<sizeof($a);$j++)
```

```
{
```

```
echo $a[$j];
}
```

```
?>
```

- **sort()**: This function is used to sort an index array in ascending order. If your value is alphabets in array then it will sort array in ascending, all elements in a to z format. In sort() function we pass array variable as an argument.

```
<?php
$a=array("Mobile","Ac","Television","Computer");
sort($a);
print_r($a);
?>
```

- **rsort()**: This function is used to sort an index array in descending order. If your value is alphabets in array then it will sort array in descending, all elements in z to a format. In rsort() function we pass array variable as an argument.

```
<?php
$a=array("Mobile","Ac","Television","Computer");
rsort($a);
print_r($a);
?>
```

- **asort()**: This function is used to sort an associative array in ascending order. It will observe only value part and make sorting, it will ignore key at the time of sorting. If your value is alphabets in array then it will sort array in ascending, all elements in a to z format. In asort() function we pass array variable as an argument. If your array is mixing of all data type value then it will first sort string or character then float or double or integer value in ascending order.

```
<?php
$emp=array("name"=>"Ram","designation"=>"Accountant","city"=>"Mumbai");
asort($emp);
print_r($emp);

$std=array("fullname"=>"Satisf","Age"=>22,"height"=>5.10,"Gender"=>'M');
asort($std);

foreach($std as $x=>$x_value)
```

```

{
echo "Key=". $x . ", Value=". $x_value;
echo "<br>";
}
?>

```

- **arsort()**: This function is used to sort an associative array in descending order. It will observe only value part and make sorting, it will ignore key at the time of sorting. If your value is alphabets in array then it will sort array in descending, all elements in z to a format. In arsort() function we pass array variable as an argument. If your array is mixing of all data type value then it will first sort string or character then float or double or integer value in ascending order.

```

<?php
$emp=array("name"=>"Ram","designation"=>"Accountant","city"=>"Mumbai")
;
arsort($emp);
print_r($emp);

$std=array("fullname"=>"Satish","Age"=>22,"height"=>5.10,"Gender"=>'M');
arsort($std);

foreach($std as $x=>$x_value)
{
echo "Key=". $x . ", Value=". $x_value;
echo "<br>";
}
?>

```

- **ksort()**: This function is used to sort an associative array in ascending order. It will observe only key part and make sorting, it will ignore value at the time of sorting. If your value is alphabets in array then it will sort array in ascending, all elements in a to z format. In ksort() function we pass array variable as an argument. If your array is mixing of all data type value then it will first sort string or character then float or double or integer value in ascending order.

```

<?php
$emp=array("name"=>"Ram","designation"=>"Accountant","city"=>"Mumbai")
;

```

```
ksort($emp);
print_r($emp);

$std=array("fullname"=>"Satisfi", "Age"=>22, "height"=>5.10, "Gender"=>'M');
ksort($std);

foreach($std as $x=>$x_value)
{
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

- **krsort()**: This function is used to sort an associative array in descending order. It will observe only key part and make sorting; it will ignore value at the time of sorting. If your key is alphabets in array then it will sort array in descending, all elements in z to a format. In krsort() function we pass array variable as an argument. If your array is mixing of all data type value then it will first sort string or character then float or double or integer value in ascending order.

```
<?php
$emp=array("name"=>"Ram", "designation"=>"Accountant", "city"=>"Mumbai")
;
krsort($emp);
print_r($emp);

$std=array("fullname"=>"Satisfi", "Age"=>22, "height"=>5.10, "Gender"=>'M');
krsort($std);

foreach($std as $x=>$x_value)
{
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

- **array_unique()**: This function will removes duplicate value from array and show only unique or single value. It will return whole array so we can use this function with other

variable which will accept the return array. we will pass array variable as an argument inside the function.

```
<?php  
$emp=array("name"=>"Ram","designation"=>"Accountant","name"=>"Ram");  
$k=array_unique($emp);  
print_r($k);  
?>
```

- **array_reverse()**: This function will convert an array in to reverse order. It will return whole array so we can use this function with other variable which will accept the return array. We will pass array variable as an argument inside the function.

```
<?php  
$emp=array("name"=>"Ram","designation"=>"Accountant","city"=>"Baroda");  
$k=array_reverse($emp);  
print_r($k);  
?>
```

- **Array_search()**: This function will search the value inside the array, if it found on any index or any key then it will print that index and key. We will pass two arguments in function first one is a search value and second is an array variable in which we want to search. If you have enter same value on different indices or keys then it will print only first index or key in which it match the word.

```
<?php  
$arr=array("a"=>"ram","b"=>"shiva","c"=>"ram");  
echo array_search("red",$arr);  
?>
```

- **Array_replace()**: This function will replace the array value from one array to another array. We have to pass two arguments in this function first one is an array variable which will replace its value and second is also an array variable whose value will replace in first one.

```
<?php  
$a=array("mobile","Television");  
$b=array("Computer","Leptop");  
Print_r(array_replace($a,$b));  
?>
```

- **Current()**: This function will print current element value of array. It will set the pointer on index inside the array and identify the value of current pointer.

- **next()**: This function moves the pointer to the next element of array. It will move pointer based on current pointer place.
- **previous()**: This function moves the pointer to the previous element of array. It will move pointer based on current pointer place.
- **end()**: This function moves the pointer to the last element of array.

```
<?php  
$a=array(1,2,3,4,5);  
echo "<br>";print_r($a);  
  
echo "<br>".current($a)."current<br>";  
echo next($a)."next<br>";  
echo current($a)."current<br>";  
echo prev($a)."prev<br>";  
echo current($a)."<br>";  
echo end($a);  
?>
```

1.11 PHP Function:

In PHP we can also use two different types of functions one is User define function and other is inbuilt function. Function is a logical block of code. Where we add some logical syntax and we use that code repeatedly in our program by calling a function. Functions allow us to group blocks of related code that perform a specific task together. Functions can either return values when called or can simply perform an operation without returning any value. This saves us time of redeveloping a code.

1.11.1 User Define Function:

User define functions are design for some routing task. Like adding a data in database, adding some specific types of search, authenticating user data etc. In user define function we have to follow some syntax to declare and call a function. In such function it is declare or design with key word function, after function key word user can give it's own name of function, after that it will make body of function. In function declaration no need to define any data type it is same like variable, function will interpret return type at run time.

Syntax of function

Function functionname()

{

//body of function.

{

There are four different types of functions:

- Without argument without return type.
- With argument without return type.
- Without argument with return type
- With argument with return type.

Before we create our first user defined function, let's look at the rules that we must follow when creating our own functions.

- Function names must start with a letter or an underscore but not a number.
- The function name must be unique.
- The function name must not contain spaces.
- It is considered a good practice to use descriptive function names.
- Functions can optionally accept parameters and return values too.

- **Without argument without return type function:**

In such function we declare a function without any variable as an argument inside the round braces. Also will not enter any value at the time of calling a function.it will not return any answer or value.it will use that value inside the function only.

```
<?php
```

```
//define a function that displays hello function
```

```
function hello_msg()
```

```
{
```

```
    echo "Hello student this is first function.";
```

```
}
```

```
hello_msg();
```

```
?>
```

- **With argument without return type function:**

In this function while we declare a function we will pass some variable inside the round braces as an argument. While function is call we enter some value that value will pass to the function as an argument. It will not return any answer or value. it will use that value inside the function only.

```
<?php
```

```
//define a function that calculate sum
```

```
function sum($no1,$no2)
```

```
{  
    $ans=$no1+$no2;  
    echo $ans;  
}  
sum(10,20);  
?>
```

- **Without argument with return type:**

In such function we declare a function without any variable as an argument inside the round braces. Also will not enter any value at the time of calling a function. It will return any answer or value. The value which is return by function will receive or catch at time of calling a function.

```
<?php  
// function along with three parameters  
function make_product()  
{  
    $product = 5*5*5;  
    return $product; //returning the product  
}  
  
// storing the returned value  
$ans = make_product();  
echo "The product is $ans ";  
?>
```

- **With argument with return type:**

In this function while we declare a function we will pass some variable inside the round braces as an argument. While function is call we enter some value that value will pass to the function as an argument. It will return any answer or value. The value which is return by function will receive or catch at time of calling a function.

Program_1.php

```
<!doctype html>  
<html>  
<head>
```

```
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<form action="product_calc.php" method="post">
    <label for="no1">No1</label><input type="number" name="no1" min=1 required><br>
    <label for="no2">No2</label><input type="number" name="no2" min=1 required><br>
    <label for="no3">No3</label><input type="number" name="no3" min=1 required><br>
    <input type="submit" name="submit" value="Submit">
    <input type="reset" name="reset" value="reset">
</form>
</body>
</html>
product_calc.php
<?php
    $n1=$_POST["no1"];
    $n2=$_POST["no2"];
    $n3=$_POST["no3"];
    // function along with three parameters
    function make_product($no1, $no2, $no3)
    {
        $product = $no1 * $no2 * $no3;
        return $product; //returning the product
    }
    // storing the returned value
    $ans = make_product($n1,$n2,$n3);
    echo "The product is $ans";
?>
```

1.12 PHP Form Handling and PHP Global.

In PHP we can handle HTML form data and use that data in any kind of calculation, used a data in database and any logical syntax. For that we use PHP superglobals `$_GET`, `$_POST` and `$_REQUEST`. At the time of declare phpglobals we will use `$_` sign and Capital letters. These phpglobals will work with form methods get and post. While we access any form element in other PHP page at that time phpglobals will access this element by it's name only, so it is compulsory to declare name in all the form controls. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the `<form>` tag.

`$_GET`: To retrieve data from get request, we need to use `$_GET` with name of form control, we will define name inside the square brackets, Get request is the default form request. The data passed through get request is visible on the URL browser so it is not secured. You can send limited amount of data through get request.

Syntax

`$_GET["contrlname"];`

Program_1.php

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<form action="print_name.php" method="post">
    <label for="Name">Name</label><input type="text" name="name" required><br>
    <input type="submit" name="submit" value="Submit">
    <input type="reset" name="reset" value="reset">
</form>
</body>
</html>
print_name.php
<?php
Echo $_GET["name"];// this will print the name which is submit by form.
?>
```

\$_POST: To retrieve data from post request, we need to use **\$_POST** with name of form control, we will define name inside the square brackets, Post request is widely used to submit form that have large amount of data such as file upload, image upload, login form, registration form etc. The data passed through post request is not visible on the URL browser so it is secured. You can send large amount of data through post request.

Syntax

\$_POST[“controlname”];

login_form.php

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<form action="login_check.php" method="post">
    <label for="username">Username</label><input type="text"
name="username" required><br>
    <label for="password">Password</label><input type="password"
name="password" required><br>
```

```
    <input type="submit" name="submit" value="Submit">
```

```
    <input type="reset" name="reset" value="reset">
```

```
</form>
```

```
</body>
```

```
</html>
```

login_check.php

```
<?php
// this will print the name which is submit by form.
echo "you have entered following username and password.";
echo $_POST["username"];
echo $_POST["password"];
?>
```

\$_REQUEST: To retrieve data from get and post both method **\$_REQUEST** is used, we need to use **\$_REQUEST** with name of form control, we will define name inside the square brackets; It is also used with cookies and session data.

Syntax

\$_REQUEST["controlname"];

login_form.php

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<form action="login_check.php" method="post">
    <label for="username">Username</label>
    <input type="text" name="username" required><br>
    <label for="password">Password</label>
    <input type="password" name="password" required><br>
    <input type="submit" name="submit" value="Submit">
    <input type="reset" name="reset" value="reset">
</form>
</body>
</html>
```

login_check.php

```
<?php
echo "you have entered following username and password.<br>";
echo $_REQUEST["username"];
echo $_REQUEST["password"];
?>
```

- **Session a super global in PHP:**

Session is the period or time for which user will interact with your web application. Session is secure to send a value from one page to another page. In web application every client who visits our application will have separate session. Session is used to identify user/client uniquely in application. Session is also used to store value in database. a value which is stored in session variable will not store in computer, before initializing session

variable we must have to start session. Session is started with function `session_start()`, also whenever we want to access session variable at that time we must start session. After complete use of session variable we can also destroy or unset session variable. Session variable is set with super global `$_SESSION[" "]`. Inside the square brackets we can add name of session variable.

program1.php

```
<?php
Session_start();
$_SESSION["username"]="user_123@gmail.com";
$_SESSION["mobile"]="9427520202";
echo "session variable for username and mobile is set.";
?>
```

Program2.php

```
<?php
session_start();
if(isset($_SESSION["username"])&& isset($_SESSION["mobile"]))
{
    $user=$_SESSION["username"];
    $mobile=$_SESSION["mobile"];
    echo "your username is $user ";
    echo "your mobile no is $mobile";
}
else
{
    echo "Your session variables are deleted...";
}
?>
```

We can also modify session variable for that we initialize new value to the previously define session variable. For example in above example `$_SESSION["mobile"]` has mobile no 9427520202 now I want to change the value or modify value then I will initialize new mobile no to the session variable. `$_SESSION["mobile"]="9909980745";` now it will print this number.

To remove a global session variable or unset the session variable then we can use following two functions.

session_unset(): this function free all the session variables. Session variable still exists. Only value of variable is deleted.

session_destroy(): This function is different than unset. It destroys all of the data associated with the current session. Means it will destroy session variable with value there will no more exists session variable.

```
<?php
```

```
    session_start();
    if(isset($_SESSION["username"])&& isset($_SESSION["mobile"]))
    {
        echo "session is set";
    }
    else
    {
        echo "session is destroy";
    }
    session_unset(); // we can also use session_delete();
?>
```

- **Cookies a super global in PHP:**

Cookies is store in text file with size of 4kb under the client machine. It is used to keeping track of information such as username to retrieve the user who visited on website. A cookies can read by domain or server only. Cookies are usually set in an HTTP header but JavaScript can also set a cookie directly on a browser.

Setting Cookie In PHP: To set a cookie in PHP, the **setcookie()** function is used. The **setcookie()** function needs to be called prior to any output generated by the script otherwise the cookie will not be set.

Syntax :

```
setcookie(name, value, expire, path, domain, security);
```

Name: It is used to set the name of the cookie.

Value: It is used to set the value of the cookie.

Expire: It is used to set the expiry timestamp of the cookie after which the cookie can't be accessed. Suppose I want to set cookie for two days then I will set $\text{time}() + 2 * 24 * 60 * 60$

Path: It is used to specify the path on the server for which the cookie will be available. It is always “/”

Domain: It is used to specify the domain for which the cookie is available.

Security: It is used to indicate that the cookie should be sent only if a secure HTTPS connection exists. If it is set to 1 then cookie should be sent by HTTPS, otherwise 0 means it is sent by HTTP.

If you want to set more number of cookies from single page then you need to write setcookie() function that many times.

```
<?php
    $username="Maulik Patel";
    setcookie("username",$Username,time()+2*24*60*60,"/","",0 );
?>
```

• Retrieve cookie value:

To retrieve cookie value we can use `$_COOKIE` PHP globals. It will print the value of cookie name, inside the `$_COOKIE["name"]` is passed.

```
<?php
if(isset($_COOKIE["username"]))
{
    echo "Cookie is set with value=".$_COOKIE["username"];
}
else
{
    echo "Cookie is not set";
}
?>
```

• Deleting Cookie with PHP:

To delete a cookie you should call setcookie() with the name argument only and other arguments or empty strings but however this time, the expiration date is required to be set in the past.

```
<?php
    setcookie("username","",time()-60,"/","",0);
?>
```

Difference between Cookies and Session

Cookies	Session
Cookies are stored on client machine. Store user information	Session is stored on server side. Store user information.
Cookies are independent	Session are dependent on Cookie.

Cookies are expired based on time set at the time of setting a cookie value.	Session will expire while user will close the application or close browser window.
You do not need to start cookie as it is store in local machine.	You need to start session first using session_start() function before setting a session variable.
Maximum cookie size is 4kb	Session is store as many size as much data user store.
There is no specific function to close or destroy cookie	For destroy or close session we use specific function session_close() or session_destroy().

1.13 Include and require function to include a PHP file in other PHP file.

As we know PHP allows us to create various functions and various elements that are used multiple times in multiple pages. Scripting the same function in multiple pages is a task of great effort and would consume time. This can be avoided if we follow and use the concept of file inclusion which helps us to include various files including text or codes into a single program which saves the effort of writing the full function or code multiple times.

Include(): This function is used to add PHP file in another PHP file. This function works like c programming #include pre-processor. We will pass PHP file with absolute path of folder as an argument of function. This function will copy all the text and contents of file called within the PHP program.

Syntax : include("PHP file with absolute path.");

Example

welcome.php

<?php

 echo "Welcome Users on our Website";

<

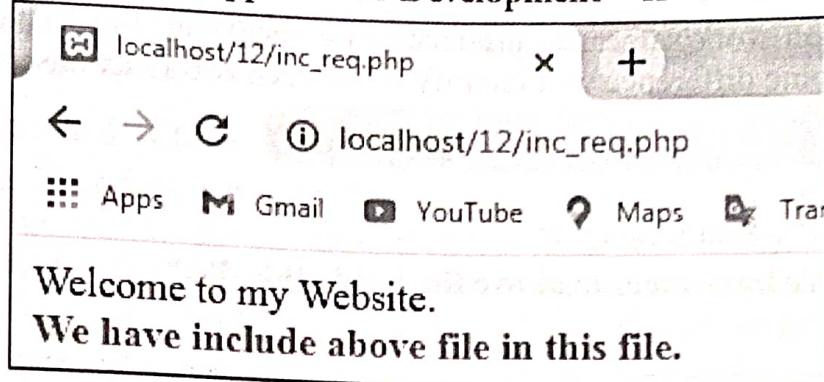
index.php

<?php

 Include("welcome.php");

 Echo "We have included above file inside this file.;"

<



Figs 1.7 include function.

require(): require function is also work as a same include.it is also copy all the text and contents of file called within the PHP program. We will pass PHP file with absolute path of folder, as an argument of function.

Example

welcome.php

```
<?php
    echo "Welcome Users on our Website";
```

<?

index.php

```
<?php
    require("welcome.php");
    Echo "We have included above file inside this file.";
```

<?

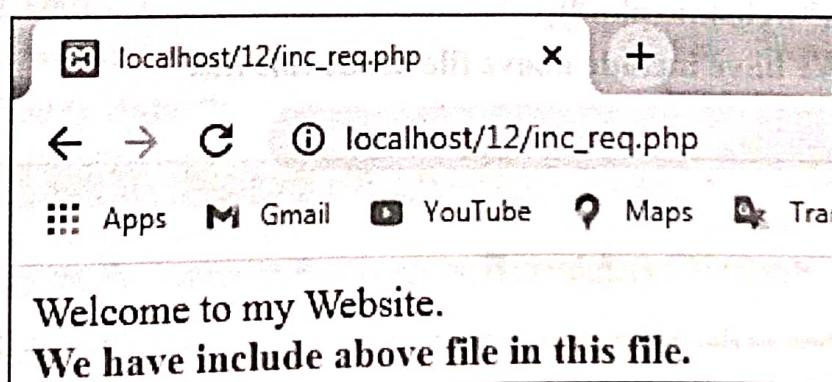


Fig 1.8 require function.

• Difference Between include() and require()

There is a single difference between the include() and require() function which we will see following this example.

Both function work same and produce same result but if any chance of fatal error then there will some difference will identify in between two functions.

Example:

```
<?php  
    Include("welcome.php");  
    echo "We have include above file inside this file."  
?>
```

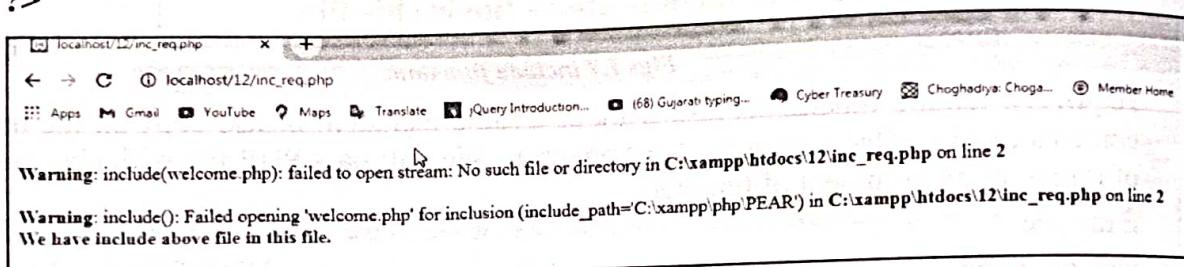


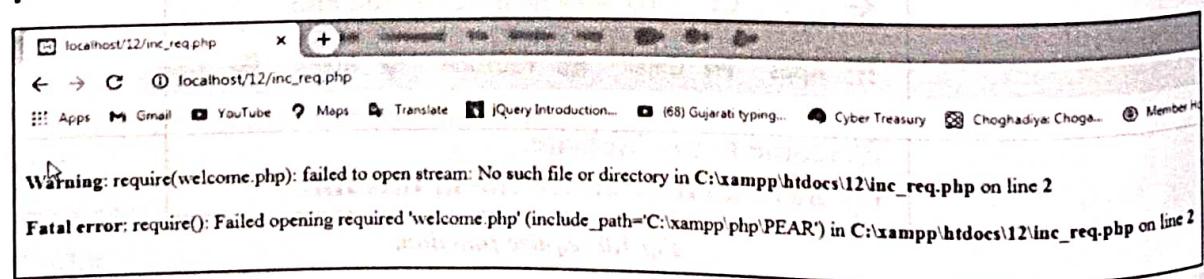
Fig 1.9 include function with warning message.

we don't have welcome.php file inside my PHP program's folder then I have included this file in my PHP page so it will get above output with warning messages with missing file. After that warning message I show my message "we have include above file in this file.", means include function will not stop execution of PHP program, it will continue if it is not finding file.

In case of require()

Example:

```
<?php  
    require("welcome.php");  
    echo "We have include above file inside this file."  
?>
```



Figs 1.10 require function with warning message.

we don't have welcome.php file inside my PHP program's folder then I have included this file in my PHP page so it will get above output with warning messages with missing file. After that warning message I don't show my message "we have include above file

in this file.”, means require function will stop execution of rest of PHP program, it will not continue if it is not founding file.

1.14 PHP Date and Time function:

In PHP Date and Time function is used to get current system date and time, it will get system date from server where this script will work. The *format* parameter in the date() function is required which specifies the format of returned date and time. This functions work with different arguments based on the argument we get different formats of date and time.

- **Date()**: this function will return current server date. There are two arguments are passed one is format parameters and second is timestamp, timestamp is optional.

Following are the format parameters used in date function.

d- Represent day of the month in double digit number (01 to 31).

D - Represent day of the week in text as an abbreviation (Mon to Sun).

m- Represent month in double digit number (01 to 12).

M- Represent month in text format (Jan to Dec).

y- Represent year in double digit number (01 or 21).

Y- Represent Year in four digit format (2001 or 2021).

We can arrange format parameters in single or double quotation marks. we can also use this parameter for getting only day, month or year, as as single argument of function.

Syntax date(“format parameters with special char - / : or space”);

<?php

```
echo "Today Date: ";
echo date('d-m-y');
echo "<br>Today Date: ";
echo date('D-M-Y');
```

?>

Similar you can use current time format parameters in date() function.

h- Represent hour in 12-hours format(01 to 12).

H- Represent hour in 24-hours format(00 to 23).

i - Represent minutes with leading zeros (00 to 59).

s - Represent seconds with leading zeros (00 to 59).

a - Represent lowercase ante meridiem and post meridiem (am or pm).

A - Represent uppercase Ante meridiem and Post meridiem (AM or PM).

<?php

```
echo "Today Date: ";
echo date('h-i-s a');
echo "<br>Today Date: ";
echo date('H-i-s A');
?>
```

- **time()**: This function is used to get current timestamp. Using this timestamp we can get current date and time in date function.

```
<?php
    $time=time();
    echo $time."<br>";
    echo date('D-M-Y h:i:s',$time);
?>
```

- **mktim()**: This function is used to make timestamp based on date and time parameters. We will pass hour, minute, second, day, month and year by comma separated in this function as an argument. If you are not passing any parameter inside this then it will make timestamp based on current system date and time.

Syntax: mktim(*hour, minute, second, month, day, year*);

```
<?php
    $time=mktim(20,51,40,28,03,2021);
    echo $time;
    echo date('D-M-Y h:i:s',$time);
    $time2=mktim()
    echo $time2;
    echo date('D-M-Y h:i:s',$time2);
?>
```

- **date_create()**: This function will return a new Date time object. It will use to convert date format.
- **date_format()**: This function returns a date formatted according to the specified format. In date format we require two arguments one argument is from date_create() function and second is format parameters in which format you want to convert date. date format parameters are listed as above in date() function.

```
<?php
    $date_con=date_create("2021-01-02");
    echo date_format($date_con,"d/m/Y");
```

>

Above program will print date in format of dd-mm-yyyy it is converted from yyyy-mm-dd.

- **date_Diff()**: This is an inbuilt function in PHP which is used to calculate the difference between two dates. This function returns a Date Interval object on the success and returns FALSE on failure. There are two different dates are pass as an argument of **date_diff()**. There should be compulsory both date format are same either dd-mm-yyyy or yyyy-mm-dd.

<?php

```
$dc1=date_create("2021-01-02");
$dc2=date_create("2024-01-02");
$interval=date_diff($dc1,$dc2);
echo $interval->format('%R%a days');
?>
```

1.15 PHP Maths functions.

- **round()**: This function rounds a floating-point number. In round function if the value is 5 or above the 5 then will round the value next highest value. If you enter value **round(0.78)** then it will round this value to 1 if you pass **round(0.7823,2)** then it will round the value with 0.78 two digit round.
- **ceil()**: This function will return the round a fraction value near to the next highest value. If you have value 3.4 then it will round value to 4 because next highest value is 4.
- **floor()**: This function will return the round a fraction value near to the next lowest value. If you have value 3.4 then it will round value to 3 because next lowest value is 3.

<?php

```
echo round(0.98);//will prnt 1
echo round(4.96754,2);//will print4.97
echo ceil(3.4); // will print 4
echo ceil(9.99); //will print 10
echo ceil(-4.2);// will print -3
echo floor(4.3);// will print 4
echo floor(5.89);//will print 5
echo floor(-5.60);//will print-6
?>
```

- **sqrt()** : You can use the sqrt() function to find the square root of a positive number. This function returns a special value NAN for negative numbers. Here's an example:

```
<?php  
echo sqrt(9); // Outputs: 3  
echo sqrt(25); // Outputs: 5  
echo round(sqrt(10),2); // Outputs: 3.16  
echo sqrt(-16); // Outputs: NAN  
?>
```

- **rand()** : This function will generate random value between the range of min and max. We pass two arguments inside the rand() function. Both are optional, user can pass min value or max value as an argument if user can not pass any value then default min value is 0 and max value is getrandmax() function value.

```
<?php  
echo rand();  
echo rand(10,1000);  
//it will generate random value between 10 to 1000 numbers.  
?>
```

- **decbin()** and **bindec()**: The decbin() function is used to convert a decimal number into binary number, also its counterpart the bindec() function converts a number from binary to decimal. we have to pass decimal number or binary number as an argument.

```
<?php  
echo decbin(123)."  
"; // Outputs: 1111011  
echo decbin(12)."  
"; // Outputs: 1100  
echo decbin(100)."  
"; // Outputs: 1100100  
echo bindec(100100)."  
"; // Outputs: 36  
echo bindec(101010)."  
"; // Outputs: 42  
echo bindec(1000111)."  
"; // Outputs: 71  
?>
```

- **dechex()** and **hexdec()**: The dechex() function is used to convert a decimal number into Hexadecimal number, also its counterpart the hexdec() function converts a number from Hexadecimal to decimal. we have to pass decimal number or binary number as an argument. Same for decimal to octal or octal to decimal conversion we use these functions decoct() and octdec().

```
<?php  
echo dechex(35)."  
"; //output 23
```

```
echo dechex(289)."<br>";//output 121  
echo hexdec(186)."<br>";//output 390  
echo hexdec(654)."<br>";//output 1620
```

?>

- **max()**: This function will get the maximum number from array as well as list of numbers.
- **min()**: This function will get minimum number from array as well as list of numbers.

```
<?php
```

```
echo max(12,22,4,1,57,21);
```

```
echo max(array(3,7,1,2,9,4,6));
```

```
echo min(12,22,4,1,57,21);
```

```
echo min(array(3,7,1,2,9,4,6));
```

?>

1.16 PHP String Functions.

- **join()**: This function is used to join array elements in single string. We pass two arguments, first one is a separator which is used to join string and second one is an array variable. It is an alias of implode() function.

Syntax: **join(separator, array variable)**;

```
<?php  
$arr=array("hello","have","a","nice","day");  
echo join(",",$arr);
```

?>

- **implode()**: This function is used to join array elements in single string. We pass two arguments, first one is a separator which is used to join string and second one is an array variable.

Syntax: **implode(separator, array variable)**;

```
<?php  
$arr=array("one","two","three","four");  
echo implode(":",$arr);  
?>
```

- **explode()**: This function will create an array of string. It will separate the string and create array of string. It will use two arguments one is separator and second is string variable or string. It will separate the string with special character which is mention in

string. For example in your string there is a space between all the words then you can use space as a separator in argument.

```
<?php
```

```
    $str="Hello student good morning and have a nice day.;"
```

```
    $arr=explode(" ",$str);
```

```
    print_r($arr);// this will print array of string with index value.
```

```
    Echo $arr[0];
```

```
?>
```

- **ltrim()**: This function will strips whitespace from the beginning or left side of string.
- **rtrim()**: This function will strips whitespace from the end or right side of string.

```
<?php
```

```
    $str=" this will trim from beginning ";
```

```
    $ltrim=ltrim($str);
```

```
    $rtrim=rtrim($str);
```

```
    echo "<pre><mark>$ltrim</mark></pre>";
```

```
    echo "<pre><mark>$rtrim</mark></pre>";
```

```
?>
```

- **md5()**: This function will convert string in to Message Digest Algorithm 5 of string using the RSA Data Security. There are 2 different arguments are pass first one is string and second one is raw format, in raw format default value is false means it will convert in to 32 character hexadecimal number and if it is true then it will convert in to 16 character. Raw format is optional.

```
<?php
```

```
    $str="admin@123";
```

```
    echo md5($str); // e6e061838856bf47e1de730719fb2609
```

```
?>
```

- **parse_str()**: This function will parse a string in to variable. This function is useful when wan to store string value in to different variables. We need to pass two arguments first one is string and second is variable name in which it is convert. It will convert it into associative array variable.

```
<?php
```

```
    $str="name=Hima&age=3";
```

```
    Parse_str($str,$result);
```

```
> Echo $result["name"]; // it will print name Hima
```

```
> Echo $result["age"]; // it will print age 3
```

?>

- **str_replace()**: This function will replace all the occurrence of word in string which matching a replace word. There are three arguments are pass first one is find word second one is replace word and third is string in which we match a word. str_ireplace() function also perform same task but it is used with case-insensitive version of str_replace().

<?php

```
$str="Hello Student,Hello Parents and Hello Faculty member";
echo str_replace("Hello","Goo Morning",$str);
```

?>

- **str_split()**: This function will split sting in to array of chunks. There are two arguments are pass first one is string and second is element of array, default element is 1 it is optional.

<?php

```
$str="Have a nice Day! ";
$split=str_split($str);
print_r($split);// will print string in to individual array element.
echo $split[0];// will print H
echo $split[1];// will print a
```

?>

- **strcmp()**: This function will compare two string with case-sensitive. There are two different string are pass as an argument to compare sting. If both strings are same then it will return 0 otherwise it will return length of character which are not match. For case-insensitive searches, use thestrcasecmp() function.

<?php

```
$str1="Hypertext";
$str2="Hypertext";
echo strcmp($str1,$str2);
if(strcmp($str1,$str2)==0)
{
    echo "Both Strings are Equal";
}
```

?>

- **strpos()**: The strpos() function finds the position of the first occurrence of a string inside another string. This function is case-sensitive. For case-insensitive searches, use the stripos() function. There are two arguments are pass first one is actual string and second is a word or portion of string which we want to compare.

```
<?php  
    $str1="Hypertext";  
    $str2="Hypertext";  
    echo strcmp($str1,$str2);  
    if(strcmp($str1,$str2)==0)  
    {  
        echo "Both Strings are Equal";  
    }  
?>
```

- **strlen()**: This function will return the length of string. We have to pass string as an argument of this function.

```
<?php  
    $str="Hypertext Pre-processor (PHP)";  
    $len=strlen($str);  
    Echo "Total Length of String=$len";  
?>
```

- **substr()**: This function extracts a part of a string. There are three arguments are pass. First argument is string from which we create sub string. Second is start, specifies the position in the string from where the extraction begins. Third is length, specifies how many characters to extract.

```
<?php  
    $substr1=substr($str,9);  
    echo nl2br("\n $substr1");// it will start 9th letter to end  
    $substr2=substr($str,5,8);// it will start 5th letter to size of 8  
    echo nl2br("\n $substr2");  
    $substr3=substr($str,-4);// it will start from back side of string.  
    echo nl2br("\n $substr3");  
    $substr4=substr($str,-5,5);// it will star from back side.  
    echo nl2br("\n $substr4");  
?>
```

UNIT- 1 Practical

1. Write a program in PHP to display "Learning PHP" in bold format.

```
<?php  
echo "hello World";  
echo " Learning PHP ";  
echo 123;  
?>
```

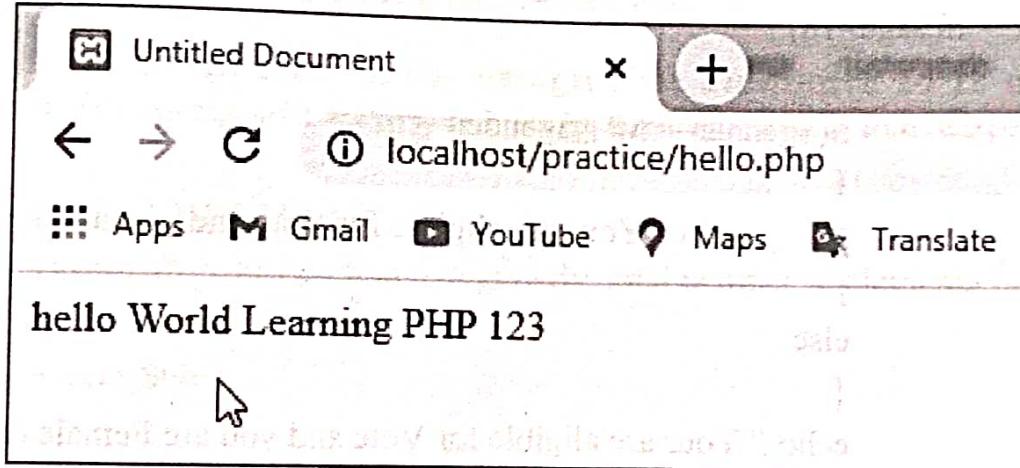


Fig 1.11 echo function to print message on browser window.

2. Write a program in PHP to demonstrate the use of comments, echo and print.

```
<?php  
echo("below line is in comment");  
// echo("this line is in single line comment");  
print "hello Asia";  
print 100;  
/* $age=35;  
print "Payal Patel age is ". $age ;  
print("this is a multiline comment in program");*/  
?>
```

3. Create a program in PHP to demonstrate the use of If ... Else and switch statements.

```
<!doctype html>
```

```
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>
<body>
<?php
    $age="15";
    $gender="f";
    if($age>18)
    {
        if($gender=='M' || $gender=='m')
        {
            echo "Your are eligible for Vote and you are male candidate";
        }
        else
        {
            echo ("Your are eligible for Vote and you are Female candidate");
        }
    }
    else
    {
        print("You are not eligible for Vote....");
    }
?>
</body>
</html>
```

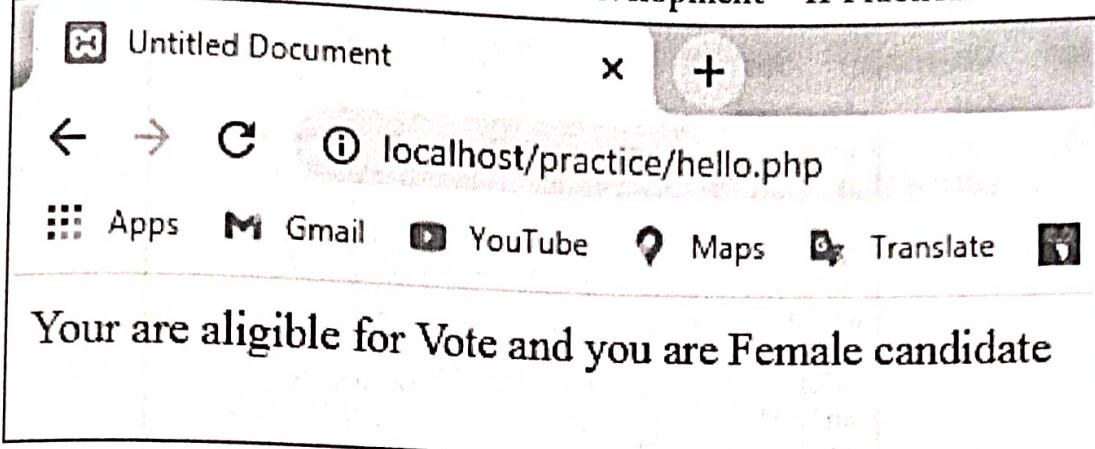


Fig 1.12 If else condition example.

4. Create an array named \$sub, assign five elements to it and display the elements assigned using for loop and for each statement.

```
<?php
$sub=array(1,2,3,4,5,55,66,77,88);
$cnt=count($sub); //count function will give the total number of elements in array
echo $cnt;
for ($i=0;$i<$cnt;$i++)
{
    echo "<br>arr[\".$i.\"]=". $sub[$i];
}
foreach($sub as $val)
{
    echo "<br>". $val;
}

$m=array("banana",11,333.333,'A');
print_r($m); // this function will print an array with key and value also with index and value.
for($j=0;$j<count($m);$j++)
{
    echo "<br>arr[\".$j.\"]=". $m[$j];
}
?>
```

The screenshot shows a browser window with the URL `localhost/practice/hello.php`. The page content displays the output of a PHP script. It includes an array definition, its elements, and then the result of the `print_r` function.

```

9
arr[0]=1
arr[1]=2
arr[2]=3
arr[3]=4
arr[4]=5
arr[5]=55
arr[6]=66
arr[7]=77
arr[8]=88
1
2
3
4
5
55
66
77
88
Array ( [0] => banan [1] => 11 [2] => 333.333 [3] => A )
arr[0]=banan
arr[1]=11
arr[2]=333.333
arr[3]=A

```

Fig 1.13 Array function and different types of Array.

- 5. Create an array named \$student, that stores 5 element bounded to a different keys and access the same using the key element.**

```

<?php
$student=array("Name"=>"Hima
Patel","Age"=>5,"Height"=>3.4,"City"=>"Ahmedabad");
//print_r($student);
echo $student["Name"]."<br>";
echo $student["Age"]."<br>";
echo $student["Height"]."<br>";
echo $student["City"]."<br>";
foreach($student as $key=>$val)
{
    echo "<br>[".$key."]=".$val;
}
?>

```

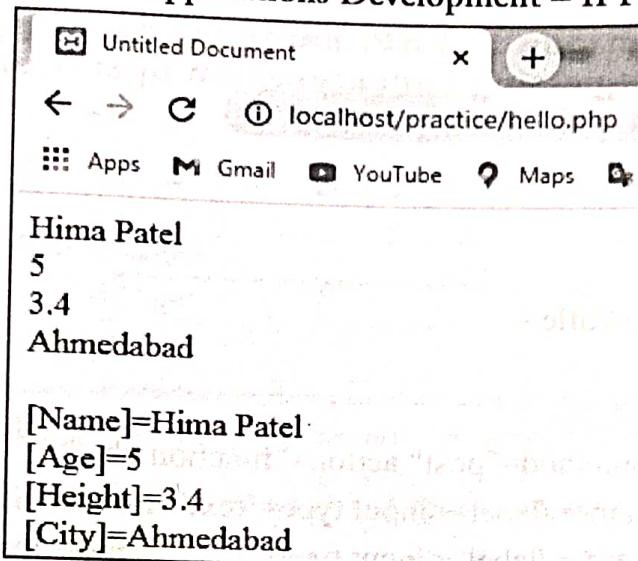


Fig 1.14 foreach Loop Structure.

6. Write a program in PHP to demonstrate the use of multidimensional arrays.

```

<?php
$shop=array("mobile"=>array("samsung"=>"m12","mi"=>"note9 pro","apple"=>"i12"),
            "leaptop"=>array("dell"=>"insp12","hp"=>"hp2200","lenovo"=>"Z350")
);
echo $shop["mobile"]["samsung"]."<br>";
echo $shop["mobile"]["mi"]."<br>";
echo $shop["mobile"]["apple"]."<br>";
echo $shop["leaptop"]["lenovo"]."<br>";
?>

```

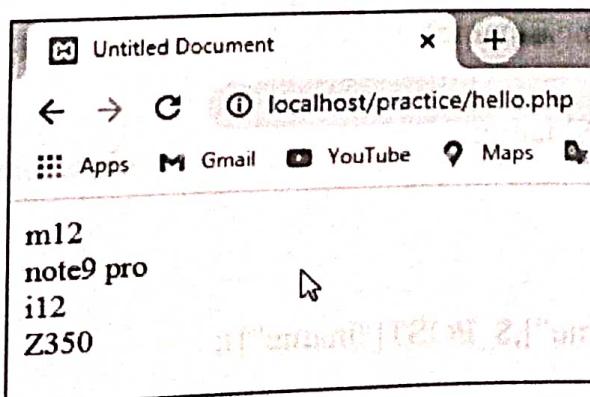


Fig 1.15 multi-dimensional array.

7. Create two functions in PHP, parameterized and non-parameterized for implementing string concatenation operation.

post_data.php

```
<!doctype HTML>
<html>
<head>
    <title>Post data </title>
</head>
<body>
<form name="add" method="post" action="function.php">
    <label for="n1">Fname</label><input type="text" name="fname" value="" /><br>
    <label for="n2">Lname</label><input type="text" name="lname" value="" /><br>
    <input type="submit" name="submit" value="submit" />
</form>
</body>
```

function.php

```
<?php
function string_cat()
{
    $fname=$_POST["fname"];
    $lname=$_POST["lname"];
    $fullname=$fname.$lname."<br>";
    echo $fullname;
}

//function call with argument.
function str_cat($fname,$lname)
{
    $flname=$fname.$lname;
    echo $flname;
}

string_cat();
str_cat($_POST["fname"],$_POST["lname"]);
?>
```

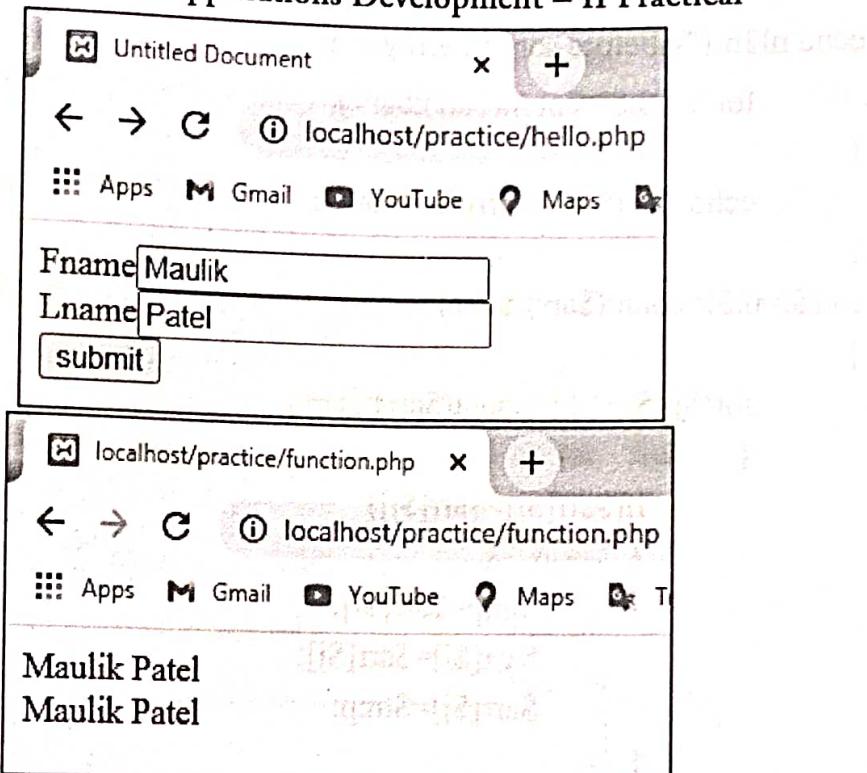


Fig 1.16 Function Example.

8. Write a PHP program to display information of PHP in the browser.

```
<?php
    echo "php version=".phpversion();
    Phpinfo();
    phpinfo(INFO_MODULES);
?>
```

9. Write a program in PHP to sort the array of given 5 numbers in ascending and descending order.

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
</head>

<body>
<?php
    $arr=array(11,4,23,3,12,59,2,34);
```

```
echo nl2br("\nbefore sorting array\n");
for($i=0;$i<count($arr);$i++)
{
    echo "arr[$i]=". $arr[$i]. "<br>";
}
for($i=0;$i<count($arr);$i++)
{
    for($j=$i+1;$j<count($arr);$j++)
    {
        if($arr[$i]>$arr[$j])
        {
            $tmp=$arr[$i];
            $arr[$i]=$arr[$j];
            $arr[$j]=$tmp;
        }
    }
}
echo nl2br("\nafter sorting array in Assending\n");

for($i=0;$i<count($arr);$i++)
{
    echo $arr[$i]. "<br>";
}
for($i=0;$i<count($arr);$i++)
{
    for($j=$i+1;$j<count($arr);$j++)
    {
        if($arr[$i]<$arr[$j])
        {
            $tmp=$arr[$i];
            $arr[$i]=$arr[$j];
            $arr[$j]=$tmp;
        }
    }
}
```

```

echo "<br>after sorting array in Descending<br>";
for($i=0;$i<count($arr);$i++)
{
    echo $arr[$i]."<br>";
}
?>
</body>
</html>

```

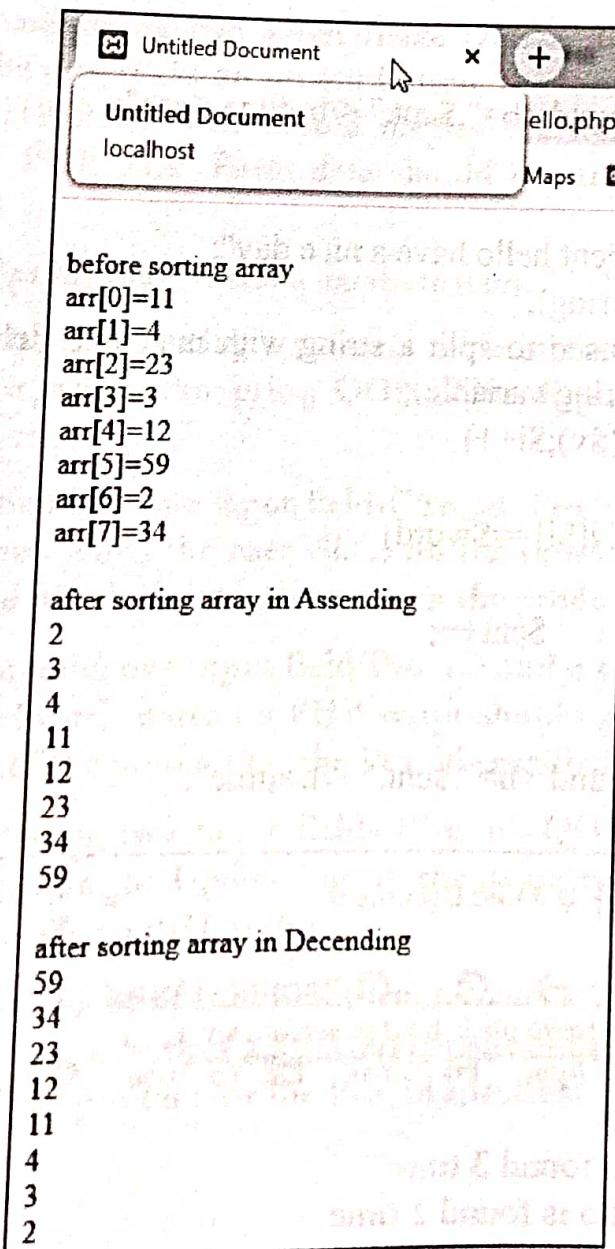


Fig 1.17 Sorting of Array.

10. Write a program to count the total number of times a specific value appears in an array.

```
<?php
$cnt=0;
$match=1;
$arr=array(1,3,22,1,4,35,21,1,5,33);
for($i=0;$i<count($arr);$i++)
{
    if($arr[$i]==$match)
    {
        $cnt++;
    }
}
echo $match." is found <b>".$cnt."</b>time <br>";
$cnt=0;
$word="hello";
$string="hello student hello have a nice day";
$v=explode(" ",$string);
//explode function is used to split a string with matching delimiter with string variable.
//explode(delimiter, string variable);
for($i=0;$i<count($v);$i++)
{
    if($v[$i]==$word)
    {
        $cnt++;
    }
}
echo $word." is found <b>".$cnt."</b>time";
?>
```

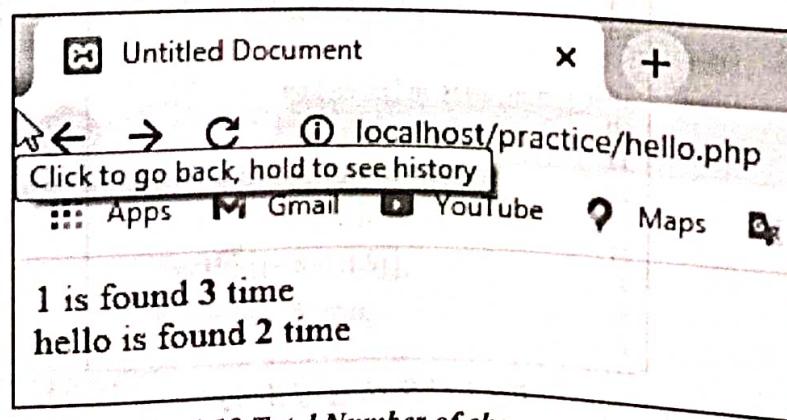


Fig 1.18 Total Number of character found.