

⇒ Introduction to PHP :-

PHP - PHP is the most popular server side scripting language for creating dynamic web page.

PHP stands for Hypertext Pre-processor.

PHP is very popular and widely used open source server side scripting language to write dynamically generating web pages.

PHP is developed in 1994. Rasmus Lerdorf was the person who developed PHP. It was initially known as "Personal Home Page".

PHP scripts are executed on the server and the result is sent to the web browsers as plain HTML. PHP can be integrated with the no of popular databases, including MySQL, PostgreSQL, Oracle, Microsoft Server SQL, SyBase --- etc

At present current major version of PHP is seven.

→ What we can do with PHP?

There are lot more things we can do with PHP

- 1) We can generate pages & files dynamically.
- 2) We can create, open, read, write and close files on the server
- 3) We can collect data from a web form such as user information, email, phone number etc.
- 4) We can send emails to the user of your website
- 5) We can send and receive cookies to track visitors of your website
- 6) We can store, delete and Modify information in your database
- 7) We can restrict unauthorized access to our websites
- 8) We can encrypt data for safe transmission over internet

→ Advantages of PHP over other languages :-

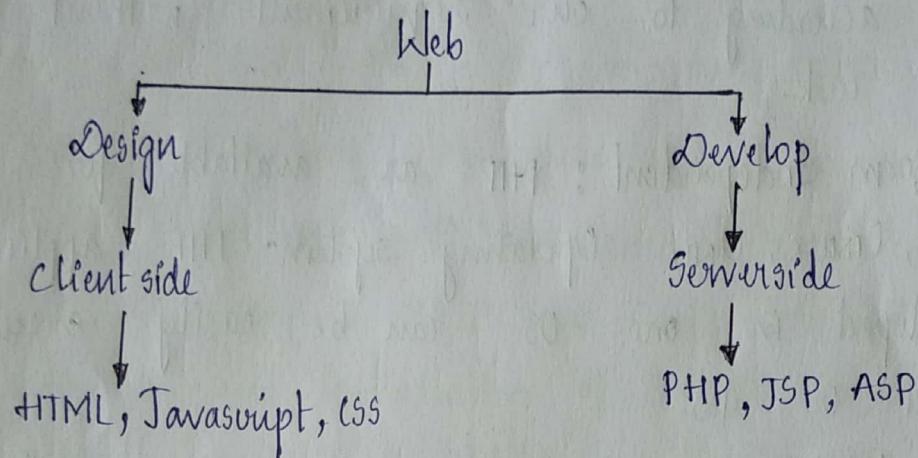
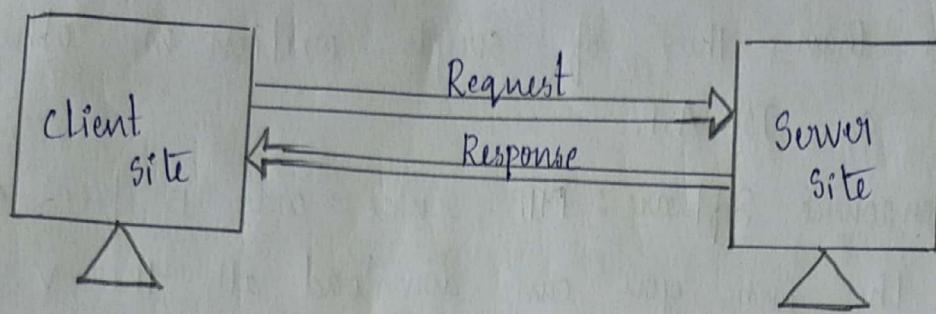
There are several advantages why one should use PHP. They are

- 1) Easy to learn
- 2) Open source
- 3) Portability
- 4) fast performance
- 5) Vast community.

Web Architecture :-

(2)

3



Web server :

It is a software used to run the web applications.

It handles the request from client processing request and sends response to client

Ex:- Apache tomcat, java ws , IIS (internet information Service),
Bua

Web Browser :

It is a software used to access websites from web server.

Ex:- Internet explorer (IE), Google chrome, netscape, --etc

- It is an open source allowing users to view and edit the script if needed.

→ PHP features :-

- i) Performance : The scripts written in PHP executes faster than those of scripts written in other languages such as JSP, ASP.
- ii) Opensource Software : PHP source code is free available on the web, you can download all the versions of PHP according to own requirements without paying any cost.
- iii) Platform Independent : PHP are available for windows, MAC, LINUX, Unix Operating systems. PHP application developed in one OS can be easily executed in other OS also.
- iv) Compartability :- PHP is compatible with almost all local servers used today like Apache, IIS etc
- v) Embedded :- PHP can be easily embeded within html tags and script

→ Key points of PHP :-

- 1) PHP file should be saved with .php extension
- 2) Every program in PHP must be start with <?php --- end with ?>
- 3) Every variable name must be prefix with '\$' symbol.
- 4) PHP is case sensitive in variable functions point of view but it is case insensitive in point of view and in
- 5) Every time line should terminate with ';' (semicolon)

PHP Installation :-

To install PHP, we have to install AMP (Apache server, MySQL, PHP) Softwarestack. It is available for all OS.

- There are no of opensource tools are available to work with PHP application.

- WAMP (Windows AMP) - It can run under windows OS.
- LAMP (LINUX AMP) - It runs under Linux OS
- SAMP (Solaris AMP) - It runs under Solaris OS
- XAMPP (X - any OS AMP and Perl) - It is for cross platform, and installation of XAMPP software. XAMPP control panel is available to start the Apache and MySQL servers.

→ Comments in PHP :-

By using comments we can stop execution of line or set of lines. PHP generally uses two types of comments. they are

- Single line comment (# or //)
It is used to stop execution of single line by using '#', '//' symbols
- Multiline comments (*---*)
It is used to stop execution of multiple lines by using "/*-----*/"

- So, PHP comments can be used to describe any code so that other developer can understand easily.

- It can also used to hide any code

→ Steps to create and execute PHP applications:-

- 1) Open the editor (notepad) and implement PHP script.
- 2) Save that file in the root directory with an extension ".php".
- 3) To see the output open web browser and send the request to the server with : file name.

Example php code :

```
<?php  
    echo "Welcome to PHP";  
?>
```

- Save in the root directory with "welcome.php" name and type URL in webbrowser like `http://localhost/welcome.php`

O/p :- Welcome to PHP

→ Declaration style tags in PHP :-

PHP supports different types of declaration style tags.

Universal Style tag :-

This tag supports all functionalities.
syntax : <?php

ii) Short open tag :-

This tag supports very few functions of PHP.
syntax : <?

iii) Script style tag :-

It is similar to java script declaration.
syntax : <script language="php">

> Php script
</script>

→ Output functions in PHP :-

There are three types of functions are

- i) print
- ii) echo
- iii) printf

i) Print :-

It is one of the output functions in PHP by using print. we can not print multiple statements.

Ex:- <?php
print(" Welcome");
?>

ii) Echo :-

By using echo we can display multiple
Ex:- <?php
echo "Welcome", " to ", " PHP";
?>

iii) printf :-

By using printf we can display output with the help
of format specifier.

Ex:- <?php
\$name = "MRECW";
\$class = 7;
printf("%s has %d classes", \$name, \$class);
?>

Apart from these three we also have

iv) var_dump :-

It is used to print or display variable value
along with variable data type

Ex:- <?php
\$name = "MRECW";
\$class = 7;
var_dump(\$name);
var_dump(\$class);
?>

O/p:-

MRECW String (7)
7 integer

variables in PHP :-

- 1) A variable in PHP is a name of memory location that holds data.
- 2) A variable is a temporary storage i.e used to store data temporarily.
- 3) "In PHP, a variable is declared using '\$' sign followed by variable name"

Syntax : \$variableName = value ;

Ex: <?php

```
$str = "hello";
$a = 100;
$y = 12.5;
echo "String is : ". $str . "<br>";
echo "Integer is : ". $a . "<br>";
echo "float is : ". $y . "<br>";
?>
```

O/p :-
String is : hello
Integer is : 100
float is : 12.5

Ex: Write a PHP program to find sum of two numbers

```
<?php
$a = 5;
$b = 4;
$c = $a + $b;
echo "sum of ". $a . " & ". $b . " = ". $c;
?>
```

O/p: sum of 5 & 4 = 9

types of variables :

there are two types of variables in PHP. They

- i) Local variables
- ii) Global variables

i) Local variables :-

A variable declared within a function is called as Local variables and can only be accessed within the function.

ii) Global variables :-

A variable declared outside of a function is called as global variable and it can accessed outside (or) within function.

→ PHP \$ and \$\$ variables :-

- \$var (single dollar) is a normal variable with the name var that stores any value like string, integer, float etc
- \$\$var (double dollar) is a reference variable that stores the value of the \$var inside it

Ex: <?php

```
$a = "abc";  
$$a = 200;  
echo "a = ". $a . "<br>";  
echo $$a . "<br>";  
echo "abc = ". $abc;
```

?>

O/p: a = abc

\$abc

abc = 200

Ex:- <?php
 \$a = "UP";
 \$a = "Lucknow";
 echo \$a."
";
 echo \$a."
";
 echo "Capital of ". \$a ." is ". \$a;
 ?>

O/p:- UP
 \$UP

Capital of UP is Lucknow.

⇒ Datatypes in PHP :-

PHP datatypes are used to hold different types of data (or) values.

- different data types can be different things
- PHP supports following datatypes

- i) String
- ii) float
- iii) Array
- iv) Integer
- v) Boolean
- vi) Object
- vii) Null

i) String :

We can represent a sequence of characters in ' ', (or)
 ee " .

Ex:- <?php
 \$n = "Hello world";
 echo \$n;
 ?>

i) float:
It is also called as double. It takes decimal
(or) number is exponential number (form)

Ex:- <?php

\$n = 10.5;

echo \$n;

?>

iii) Array:
It is a collection of similar datatype elements. An array stores multiple values in one single variable

Ex:- <?php

\$cars = array ("BMW", "volvo", "skoda", "Audi");

echo \$cars;

?>

O/p: [0] BMW
[1] volvo
[2] Skoda
[3] Audi

iv) Integers:

The range of integers in PHP is between
-2,147,483,648 to +2,147,483,647

- 0-9 digits
- no ',' or blank space is given

Ex: <?php

\$n = 5985;

echo \$n;

?>

Boolean :-

A boolean represents two possible states true or false.
In php if the value is true then it prints '1' and if false no value

Ex:- <?php

```
$x=true;
echo $x;
```

?>

o/p:- 1

<?php

```
$x=false;
echo $x;
```

?>

o/p:- null

vii) Object :-

An object is a datatype which stores data and information on how to process the data

Ex:- <?php

```
class car
```

{

```
function car() {
```

```
    $this->model = "vw";
```

}

```
}
```

```
$hello = new car();
echo $hello->model;
```

?>

viii) Null :-

Null is a special datatype which can have only one value : NULL

- A variable of datatype NULL, is a variable that has no values assigned to it

Note : If a variable is created without a value
is automatically assigned a value of NULL

to get the
?php

Ex:- <?php

```
$x = "Hello world";  
$x = NULL;  
var_dump($x);  
?>
```

⇒ String :-

1. The commonly used functions to manipulate strings
get the length of string - strlen()

```
<?php  
echo strlen("Hello world");  
?>
```

O/p :- 11

2. Count the number of words in a string -
str_word_count()

```
<?php  
echo str_word_count("Hello world");  
?>
```

O/p :- 2

3. Reversing a string - strrev()

```
<?php  
echo strrev("Hello");  
?>
```

O/p :- olleh

to get the position of a word - strpos()

<?php

echo strpos("Hello world", "world");

?>

O/p:- 6

5. To replace a text within a string - str-replace()
 this function replaces some characters with some other
 characters in string

<?php

echo str_replace("world", "Mrew", "Hello world");

?>

O/p:- Hello Mrew

⇒ Arrays :-

Arrays can be represented in three ways

- Indexed
- Associative
- Multidimensional

An Array stores multiple values in one single variable.
 An array can hold many values under a single name,
 and we can access the values by referring an index
 number.

i) Indexed : Arrays with a numeric index

- Index always starts with '0'

- We can access the array with index values.

```
<?php
```

```
$cars = array ("BMW", "Skoda", "Audi");
```

```
echo "I like ". $cars[0] . ", ". $cars[1] . " and " . $cars[2];
```

```
?>
```

Op:- I like BMW, Skoda and Audi.

- To count the number of elements in array we can use count() [Gives the length of array]

```
<?php
```

```
$cars = array ("BMW", "Audi", "Skoda");
```

```
echo count ($cars);
```

```
?>
```

ii) Associative :-

Arrays uses named keys that you assign to them.

```
<?php
```

```
$age = array ("Peter" => 35, "Ben" => 37, "Tom" => 43);
```

```
echo "Peter is ". $age[Peter]. " years old";
```

```
?>
```

iii) Multidimensional :-

An array contains one or more arrays.

```
<?php
```

```
$cars = array (array ("BMW", 22, 18),
```

```
array ("Skoda", 5, 2),
```

```
array ("Audi", 17, 15));
```

```
) ;
```

```

echo $cars[0][0]. "in stock". $cars[0][1]. "sold". $cars[0][2].
    "<br>";
echo $cars[1][0]. "in stock". $cars[1][1]. "sold:". $cars[1][2].
    "<br>";

```

?7

→ Operators :-

Operators are used to perform operations on variables and values.

PHP divides operators into following groups

i) Arithmetic operators :-

there are +, -, /, %, *

Ex:- <?php

\$a=10;

\$b=5;

echo \$a+\$b;

echo \$a-\$b;

echo \$a*\$b;

echo \$a/\$b;

echo \$a%\$b;

?>

ii) Assignment operators :-

Assignment operators are used with numeric values to write a value to the variable. The basic assignment operator in PHP is '='. It means left operand gets set to the value of assignment expression on right.

- Assignment operators are $=, +=, -=, *=, /=, \cdot=$

Ex:- <?php

\$n=10;

echo \$n;

?>

<?php

\$x=10;

\$x +=5;

echo \$x;

?>

- iii) Comparison operators :-

Comparison operators are used to compare values (numbers or string)

- Comparison operators are $==, !=, >, >=, <, <=$

Ex:-

<?php

\$x=10;

\$y=10;

var_dump (\$x==\$y);

?>

O/P:- bool(true)

- iv) Increment / Decrement operators :-

Increment operators are used to increment a variable value

- Decrement operators are used to decrement a variable value.

| Operators | Name | Description |
|-----------|----------------|---|
| $++\$x$ | pre-increment | increment $\$x$ by 1, then return $\$x$. |
| $\$x++$ | post-increment | return $\$x$, then increment $\$x$ by 1 |
| $--\$x$ | pre-decrement | decrement $\$x$ by 1, then return $\$x$. |
| $\$x--$ | post-decrement | return $\$x$, then decrement $\$x$ by 1 |

Ex:- <?php
 $\$x=10;$ <?php <?php
 echo $++\$x;$ $\$x=10;$ $\$x=10;$
 ?> echo $\$x++;$?>
 o/p: 11 o/p: 10 o/p: 9

v) String Operators :-

PHP has 2 operators that are specially designed for strings.

- - used to combine 2 strings i.e as concatenation.
- = - concatenation assignment

Ex :-

<?php

\$tx1 = "James";

\$tx2 = " Bond";

echo \$tx1.\$tx2;

?>

o/p:- James Bond

vi) Logical operators :-

The PHP logical operators used to combine conditional statements.

| Operator | Example | Result |
|----------|------------|--|
| && (AND) | \$x && \$y | True, if \$x and \$y are true |
| (OR) | \$x \$y | True, if either of \$x or \$y are true |
| ! (not) | !\$x | True, if \$x is false |

Ex :- <?php

\$x=100;

\$y=90;

if (\$x==100 && \$y==50)

echo "true";

?>

o/p:- true

<?php

\$x=90;

if (\$x!=90) {

echo "Hi";

?>

o/p:- Hi

Expressions :-

Expressions are the most important working blocks of PHP. The most basic forms of expressions are constants and variable. A very common type of expressions are comparision expressions. These expressions are evaluated to their false or true values. These expressions are most commonly used inside conditional executions such as if-statements.

→ Constants :-

Constants are like variables except that once they are defined they can not be changed or undefined.

Syntax :- define(name, value, case-insensitive)

Ex:- <?php

```
define("greeting", "Welcome to MRECW");
echo greeting;
```

?>

Output:- Welcome to MRECW.

⇒ Control Structures :-

Any PHP script is built of a series of statements. A statement can be an assignment, a function call, a loop, a conditional statement or even a statement that does nothing.

i) Conditional statement :-

Conditional statement are used to perform different actions based on different conditions. In PHP we follow different conditional statement.

- a) if statement
- b) if-else statement
- c) if-else if-else statement
- d) switch statement

a) if statement :-

The if statement is used to execute some code only if a specified condition is true.

Syntax :- if (condition) {

code to be executed;

 if statement condition is true)

Ex:- <?php

 \$dt = date("U");

 if (\$dt < "20") {

 echo "Have a nice day";

}

?>

if-else statement :-

Use the if-else statement to execute some code if condition is true, and another code if condition is false.

Syntax :-

```
if (condition) {
```

code to be executed if

condition is true;

```
} else {
```

code to be executed if

condition is false;

```
}
```

Ex:-

```
<?php
```

```
$t = date("H");
```

```
if ($t < "20") {
```

```
echo "have a nice day";
```

```
} else {
```

```
echo "have nice evening";
```

```
}
```

```
?>
```

c) if-else if-else statement :-

Use the if-else if-else statement to specify a new condition to test, if the first condition is false.

with statement
by

Syntax :- if (conditions) {
 code to be executed if
 condition 1 is true;
} else if (condition 2) {
 code to be executed if condition 1 is false
 and condition 2 is true;
} else {
 code to be executed if both condition 1, 2
 are false;
}

Ex:- <?php
\$t = date("H");
if (\$t < "10") {
 echo "have a nice day";
} else if (\$t < "20") {
 echo "have a nice evening";
} else {
 echo "have a nice sleep";
}

?>

Switch statement :-

Use the switch statement to select one of many blocks of code to be executed.

Syntax :- switch(*n*) {

case label1 : code to be executed if *n*=label1;
break;

case label2 : code to be executed if *n*=label2;
break;

⋮

default : code to be executed if *n* is different
from all labels;

}

Ex:- <?php

\$fcolor="red";

switch (\$fcolor) {

case "red": echo "your favorite color is red!";
break;

case "green": echo "your favorite color is green!";
break;

default : echo "your favorite color is neither
red nor green";

}

?>

ii) Loop :-

In PHP we have following loop statements

- a) while
- b) do while
- c) for
- d) for each

a) while loop :-

Syntax: while (condition) {

code to be executed;

}

Ex:- <?php

\$x=1;

while (\$x<=10) {

echo "the number is \$x
";

\$x++;

}

?>

b) do - while loop :-

Syntax: do {

code to be executed;

} while (condition);

Ex:- <?php

\$x=1;

do {

echo "the number is \$x
";

\$x++;

} while (\$x<=10);

?>

loop :-
syntax: for (

for loop :-

Syntax : for (initialization; condition; increment/decrement) {

code to be executed;

}

Ex:- <?php

for (\$n=1; \$n<=10; \$n++) {

echo "the number is \$n
";

}

?>

d) for-each loop :-

The for each loop works only on arrays, and is used to through each key (or) value pair in the array.

Syntax :- foreach (\$array as \$value) {

code to be executed;

}

For every loop iteration the value of the current array element is assigned to \$value and the array pointer is moved by 1 untill it reaches to last array elements.

Ex:- <?php

\$color = array ("white", "black", "blue");

foreach (\$color as \$value) {

echo "I like \$value
";

}

?>

⇒ Functions :-

A function is a block of statement that can be used repeatedly in a program. A function will not be executed immediately when a page is loaded or loads.

A function will be executed by a call to the function.

→ Create a function :-

A function declaration starts with the word "function".

Syntax :-

```
function fun-name()
{
    code to be executed;
}
```

A function name can start with a letter (or) - (Underscore) but not with number.

• function names are not case sensitive.

Ex:-

```
<?php
function write-msg() {
    echo "Hi";
}
write-msg();
?>
```

O/p:- Hi

can types of functions :-

1) function with arguments :

Information can be passed to functions through arguments . An argument is like a variable . Arguments are specified after the function name inside the parenthesis . you can add many arguments as you want , just separate them with ","

Ex :- <?php

```
function addfun($num1,$num2) {
    $sum=$num1+$num2;
    echo " sum of $num1 & $num2 : ". $sum;
}
addfun(10,20);
```

Ex:- <?php

```
function familyname($fname,$year) {
    echo $fname. " born in '$year' ";
}
familyname ("abc",1995);
familyname ("pqr",1986);
```

2) Default argument value :-

If you call the function without takes the default value as argument

Ex:- <?php

```
function setheight($minheight=50) {  
    echo "the height is : ". $minheight .<br>;  
}  
setheight(350);  
setheight();  
?
```

O/p:- The height is : 350

The height is : 50

3) functions returning value :-

To let a function returning a value, use the return statement.

Ex:- <?php

```
function sum($x,$y) {  
    $z = $x + $y;  
    return $z;  
}  
echo "5+10=".sum(5,10);  
?
```

O/p : 5+10=15

Reading Data from web form controls like textboxes, radiobuttons, lists etc :-

the PHP super global \$-get and \$-POST are used to collect form data

→ \$-get vs \$-post :-

i) \$-get :-

\$-get is an array of variables passed to the current script via URL parameters.

- Information sent from the form with \$-get method is visible to everyone (All variable names and values are displayed in URL)
- Using \$-get method we can send limited data only.
- \$-get should never be used for sending passwords or other sensitive information.

ii) \$-post :-

\$-post is an array of variables passed to the current script via http post method.

- Information send from a form with POST method is invisible to others and has no limits on the amount of information to send.
- Developers prefer \$-post for sending form data.

NOTE:- Both \$-get and \$-post are treated as super globals which means that they are always accessible, regardless of scope and we can access them from any function.

~~ndling file
Configure the
Ensure that
your
(a) In
e~~

Ex :- <html>
<head>
 <title> Form </title>
</head>
<body>
 <form action="welcome.php" method="POST">
 Name : <input type="text" name="name">

 Email : <input type="text" name="email">

 <input type="submit" />
 </form>
</body>
</html>

O/p :-

Name : xyz
Email : pqr@gmail.com
Submit

When User fills the above form and clicks submit button, the form data is sent for processing to a .php file name (welcome.php). The form data is sent with the http POST method.

Welcome.php :-

<html>
<body>
 Welcome <?php echo \$POST["name"]; ?>

 Your email address is <?php echo \$POST["email"]?
</body>
</html>

Handling file uploads :-

Configure the "php.ini" file :-

- Ensure that PHP is configured to allow file uploads
In your "PHP.ini" file, search for the file-uploads directive, and set it to on.
- Create html form

Ex:- <html>

<body>
<form action="upload.php" method="POST"
enctype="multipart/form-data">

Select image to upload :

<input type="file" name="filetoupload"
id="filetoupload">

<input type="submit" value="uploadimage"
name="submit">

</form>

</body>

</html>

O/P :- Select image to upload :

- The form also needs the following attributes enctype="multipart/form-data". It specifies which content-type to use while submitting.

NOTE :- The type="file" attribute of <input> to show the input field as a file-select control, with browse button next to the input control.

NOTE:- We will need to create new directories called "upload" in directory where upload.php file resides uploaded file will save there.

ii) Create the upload.php script :-

```
<?php  
$target_dir = "upload";  
$target_file = $target_dir . basename($_FILES["file to upload"]  
["name"]);  
$uploadOk = 1;  
$imageFileType = Pathinfo($target_file, PATHINFO_EXTENSION);  
// check if image file is actually image or fake  
if (isset($_POST["submit"])) {  
    $check = getimagesize($_FILES["file to upload"]["temp-name"]);  
    if ($check != false) {  
        echo "file is an image". $check["mime"];  
        $uploadOk = 1;  
    } else {  
        echo "file is not an image";  
        $uploadOk = 0;  
    }  
}  
?>
```

- \$target_dir="uploads" specifies the directory where the file is going to be placed.
- \$target_file= specifies the path of file to be uploaded
- \$imageFileType holds the file extension of file
- \$uploadOk=1 is not used yet.

~~Explain~~ ~~call~~ connect to a Database (MySQL as Reference) :-

PHP connect to MySQL :

MySQLi (i stands for 'improved')

→ code for connection to server -

```

<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    // create connection
    $conn = mysqli_connect($servername, $username, $password);
    // check connection
    if (!$conn) {
        die("connection failed = ".mysqli_connect_error());
    } else {
        echo "connected successfully";
    }
    // create database
    $sql = "CREATE DATABASE MVDB";
    if (mysqli_query($conn, $sql)) {
        echo "Database created successfully";
    } else {
        echo "Error creating Database". mysqli_error($conn);
    }
    mysqli_close($conn);
?>

```

→ Create a table:-

```
<?php  
$servername = "localhost";  
$username = "username";  
$password = "password";  
$dbname = "MVDB";  
$conn = mysqli_connect($servername, $username, $password, $dbname);  
  
if (!$conn) {  
    die ("connection failed".mysqli_connect_error());  
}  
  
$sql = "CREATE TABLE myguests (id INT(6) UNSIGNED  
AUTO_INCREMENT PRIMARYKEY, firstname VARCHAR(30)  
NOT NULL, lastname VARCHAR(30) NOT NULL, email  
VARCHAR(50), reg_date TIMESTAMP");  
if (mysqli_query($conn, $sql)) {  
    echo "Table MyGuests created successfully";  
} else {  
    echo "Error creating table :".mysqli_error($conn);  
}  
mysqli_close($conn);  
?>
```

- MySQL is the most popular database system used with PHP
- MySQL is database system used on the web.
- MySQL is database system that runs on server.
- MySQL is ideal for both small and large applications
- MySQL compiles on no of platform.

database are useful for storing information categorical
A company may have database with following tables

- Employees
- Customers
- products
- Orders.

→ Executing simple query :

A query is a question request. we can query a database for specific information and have a record set return.

Ex: `SELECT lastname from Employees;`

→ Insert data :

```
$sql = "INSERT INTO Myguests(firstname, lastname, email)
VALUES ('John', 'Doe', 'john@gmail.com');"
```

// select column from table name:

```
$sql = "SELECT id, firstname, lastname FROM Myguests"
```

⇒ Handling results :-

```
$sql = "UPDATE Myguests SET lastname = 'Doe' WHERE id=2";
```

```
$sql = "DELETE FROM Myguests WHERE id=3";
```

```
$sql = "INSERT Myguests(firstname, lastname, email) VALUES
('CSE', 'A', 'CSE@email.com');"
```

⇒ Handling Sessions and Cookies :-

→ Cookies :-

Cookies are stored in User browser, until deleted by user (or) set as per the timer.

- It will not be destroyed even if you close the browser
- A cookie is often used to identify as user
- A cookie is a small file that server embeded on users computer. Each time the same computer requests a page with a browser, it will send the cookies too.
- With the PHP we can create and retrive cookie values.
- A cookie is created with the setCookie()

Ex :- setCookie(name,value,path, domain, secure, httponly)

PHP create a cookie :-

```
<?php  
$cookie_name = "user";  
$cookie_value = "John";  
setCookie($cookie_name, $cookie_value, time() + (60 * 30), "/");  
?  
<html>  
  <body>  
    if (isset($_COOKIE[$cookie_name])) {  
      echo " cookie named ". $cookie_name . " is set";  
    } else {  
      echo " cookie : ". $cookie_name . " is not set";  
    }  
  </body>  
</html>
```

Session :-

- sessions are stored in server.
- User can't disable the session.
- It will be destroyed if you close the browser.
- Sessions are more secure than cookies as it is stored in the server.

Session creation in PHP :-

- when you work with an application, we open it and do some changes and then we close it.
- Session variables hold information about one single user and are available to all pages in one application.

NOTE :- If you need permanent storage, you may want to store the data in database.

- A session is started with the session_start()
- Session variables are set with PHP global variable \$-session.

```

Ex: <?php
    //session start
    session_start();
?>
<html>
    <body>
        <?php
            // set session variable
            $SESSION["fav color"] = "green";
            $SESSION["fav-animal"] = "dog";
            echo "session variables set";
        ?>
    </body>
</html>

```

⇒ File handling in PHP :-

file handling is an important part of any web application, you often need to open and process a file for different tasks.

- PHP has several functions for files i.e open, close, read, write, appending, deleting ---etc.
- We have text field called "web.txt" stored on the server that look like

Web.txt

| | | |
|------|--------------|-----------------|
| HTML | - Hypertext | markup language |
| PHP | - Hypertext | preprocessor |
| XML | - Extensible | Markup language |

Ex:- <html>
 <body>
 <?php
 echo readfile ("filename.txt");
 ?>
 </body>
 </html>

i) Create a file:

```
<?php  
$myfile = "Web.txt";  
$handle = fopen ($myfile,'w');  
?>
```

open a file :

```
<?php
    $myfile = "web.txt";
    $handle = fopen ($myfile, 'w');
?>
```

iii) Read a file :

```
<?php
    $myfile = "file.txt";
    $handle = fopen ($myfile, 'r');
    $data = fread ($handle, filesize . $myfile);
?>
```

iv) Write to a file :

```
<?php
    $myfile = "file.txt";
    $handle = fopen ($myfile, 'w');
    $data = "This is data";
    fwrite ($handle, $data);
?>
```

v) Append to a file :

```
<?php
    $myfile = "file.txt";
    $handle = fopen ($myfile, 'a');
    $data = "New data line 1";
    fwrite ($handle, $data);
    $new_data = " new data line 2";
    fwrite ($handle, $new_data);
?>
```

vi) close a file :

<? php

```
$myfile = "file.txt";
$handle = fopen($myfile, 'w');
fclose($handle);
```

?>

vii) file deletion :-

- unlink() is used to delete file

Ex:- <?php

```
$myfile = "file.txt";
unlink($myfile);
```

?>

⇒ List directories :-

The directory function allow you to retrieve information about directories and their contents.

there are several directory functions in PHP . they are

i) Chdir() - changes current directory

Ex:- <?php

```
//get current directory
echo getcwd(). "<br>";
// changing directory
chdir("images");
echo getcwd();
```

?>

Output:- xyz
images.

chroot() - change root directory

Ex :- <?php

```
chroot ('path/to/chroot');");

```

```
echo getcwd();
```

?>

iii) opendir() - open directory handle

iv) closedir() - close the directory handle

Ex :- <?php

```
$dir = "\images\";
```

// open directory and read its contents.

```
if (is_dir($dir)) {
```

```
if ($dh = opendir($dir)) {
```

```
while (($file = readdir($dh)) != false) {
```

```
echo "filename: ".$file."  
";
```

```
}
```

}

?>

v) getcwd() - returns the current working directory.

Ex :- <?php

```
echo getcwd();
```

?>

O/p:- xyz

vi) scandir() - List files, directories inside some directory i.e whatever directory we give

Ex:- <?php

```
$dir = "/images/";
$a = scandir($dir);
print_r($a);
```

?>

O/P:-

Array (

[0] => .

[1] => ..

[2] => cat.gif

[3] => dog.gif)