**PHP APPLICATION THEOERY**

1. How to run php application in desktop :-
2. open xampp and start mysql and apace
3. open chrome and run <http://localhost/shubham_php/file_Name.php>
4. and press enter your php output displays
5. Destination of php application :- C:\xampp\htdocs\shubham\_php
6. Datatypes in php :-

No datyte in php. Every data stored in to $variable\_Name=value;

Ex.$no=10,

$no=10.2;

$no=’A’;

$no=”shubham pakhale”;

1. Output in php :-

By using ECHO keyword we get output on console window

1. **Var\_dump** function is used for finding datatype of entered element by user.
2. **Function FUNCTION\_NAME (PARAMETER)**

**{ ;}**

1. Loops in php:-
2. **FOR LOOP ;-**

for($i=0;$i<$no;$i++)

{}

1. **If loop:-**

If(condition)

{

echo “”;

}

1. **While loop :-**

$i=1;

While(condition)

{}

1. **Do while loop :-**

$i=0;

Do

{Logic;}

While(condition)

1. **Foreach loop :-**

$arr =array();

Foreach($arr as $i)

{echo $i}

Ex. $arr=array(10,20,30);

foreach($arr as $value)

{

echo $value;

}

1. `**every php program start with**  **<?php**

**And ends with-> ?>**

1. Array of heterogeneous elements in php are available any datatyped elements can be entered in php array.

Array in php - $arr= array(10,20,30); //way 1

$arr[0]=10;

$arr[1]=20;

$arr[2]=30;

// way 2

foreach($arr as $i)

{

Echo “$i, “;

}

1. Associative array :-

$array\_Name= array(“name1”=value,”name2”=value)

Foreach($array\_Name as $key => $value)

{

Echo “”.$key.””.$value;

}

Ex. $marks= array("shubham"=>87,"mayuri"=>81,"prerna"=>94);

$no=count($marks);

echo $no;

echo "<br>";

foreach($marks as $key=>$val)

{

echo "Name is ".$key." marks is :".$val;

echo "<br>";

}

1. SUPER GLOBAL VARIABLES IN PHP :-
2. **$GLOBALS** – ex.

$var3=10;

Loop{

$var3=$GLOBALS["var3"];

$var3=20;

echo $var3; //20

}

1. $
2. // case insensitive in php

ECHO "HELLO";

echo "hello";

Echo "hello";

1. Comments in php :-

#Shell Style comment

//C++ Style comment

/\*

C Style Comment

\*/

1. file inclusion in php :-
2. using include keyword :-

<?php

Include ‘filename.php’;

?>

1. using require keyword :-

ex. <?php

require("menu1.php");

?>

1. menu :-

ex.

1. CONSTANT IN PHP :-

define("MyConst","This is user defined constant <br>"); // constant string Myconst=”This is user defined constant; - in c++

echo(MyConst);

//Built In constants:

echo("We are in file: ".\_\_FILE\_\_."<br>"); //file location

echo("We are on line: ".\_\_LINE\_\_."<br>"); //line number

echo("We are in Function: ".\_\_FUNCTION\_\_."<br>"); //function name on which function it is defined

echo("We are in Class: ".\_\_CLASS\_\_."<br>"); //class name

?>

1. STRING IN PHP :-

EX. <?php

$string1="Hello World";

$string2="1234";

//String Concatenation Operator

echo $string1 . " " . $string2;//Hello World 1234

echo("<br>");

//Using the strlen() function

echo strlen("Hello world!");//12

echo("<br>");

//Using the strpos() function

echo strpos("Hello world!","world");//6

// count no of words in string

echo str\_word\_count("Hello world!"); // 2

// reverse the string

echo strrev("Hello world!"); //!dlrow olleH

// replace word with other word

echo str\_replace("world", "India", "Hello world!"); // Hello India!

echo("<br>");

?>

1. FUNCTION IN PHP :-

Ex. <?php

//Creating PHP Function

/\* Defining a PHP Function \*/

function fun()

{

echo "Hello world";

echo("<br>");

}

/\* Calling a PHP Function \*/

fun();

//PHP Functions with Parameters

function add($num1, $num2)

{

$sum = $num1 + $num2;

echo "Sum of the two numbers is : $sum";

echo("<br>");

}

add(10, 20);

//Passing Arguments by Reference

function addFive($num)

{

$num += 5;

}

function addSix(&$num)

{

$num += 6;

}

$orignum = 10;

addFive( $orignum );

echo "Original Value is $orignum<br />";

addSix( $orignum );

echo "Original Value is $orignum<br />";

//PHP Functions returning value

function addFunction1($num1, $num2)

{

$sum = $num1 + $num2;

return $sum;

}

$return\_value = addFunction1(10, 20);

echo "Returned value from the function : $return\_value";

echo("<br>");

//Setting Default Values for Function Parameters

function gun($param = 10)

{

print $param;

}

gun(20);

gun();

echo("<br>");

//Dynamic Function Calls

function demofun()

{

echo "Hello<br />";

}

$function\_holder = "demofun";

$function\_holder();

//Anonymous Functions

$AnonymousAdd = create\_function('$a,$b','return ($a+$b);');

echo $AnonymousAdd(10,20);

echo("<br>");

?>

1. There are two ways the browser client can send information to the web server.

The GET Method

The POST Method

Before the browser sends the information, it encodes it using a scheme called URL

encoding.

In this scheme, name/value pairs are joined with equal signs and different pairs

are separated by the ampersand.

The GET Method

The GET method sends the encoded user information appended to the page request.

The page and the encoded information are separated by the ? character.

The POST Method

The POST method transfers information via HTTP headers.

The information is encoded as described in case of GET method and put into a

header called QUERY\_STRING.

The $\_REQUEST variable

The PHP $\_REQUEST variable contains the contents of both $\_GET, $\_POST,

and $\_COOKIE.

1. CLASS IN PHP :-
2. COSTRUCTOR –

Ex. Class shubham

{

Public function\_\_construct()

{

Echo “Inside constructor”;

}

1. DESTRUCTOR :-

Ex.class shubham

{

Public function\_\_destruct()

{

Echo “Inside destructor of shubham “;

}

}

1. SESSION IN PHP :-

Ex. <?php

session\_start();

if(isset($\_SESSION['counter']))

{

$\_SESSION['counter'] +=1;

}

else

{

$\_SESSION['counter']=1;

}

echo $\_SESSION['counter'];

?>

**\*\*\***

echo $\_SESSION['counter']; //it counts number of times user entered in the same website..this stores counter value at server side..

\*\*\*

#unset($\_SESSION ['counter']); //it destroy session and return values 1 everytime after it goes on another website..

1. ABSTRACT IN PHP :-

Abstract concept is same as pure virtual function in c++

Ex. Class base

{

Public abstract function fun();

}

Class derived extends base

{

Public function fun()

{

Echo “shubham “;

}  
}

1. Static in php :-
2. Static vatiable –

Ex. <?php

static $i=10;

$j=10;

$i++; //11 12 1 14..

$j++; //10 10 10 …

?>

1. Static character and static behaviour :-

Ex. <?php

Class demo

{

Public static $i=10;

Public static function fun()

{

}

}

Demo :: fun();

Echo Demo :: $i;

?>

1. Dynami memory allocation php :-

We have to create object of class using new keyword.. there is no need to deallocate the memory because automatic garbage collector is present in php as well as in java.

Class Demo

{}

$obj =new Demo();

$ret=0;

$ret=$obj->function\_name(parameter);

Echo $ret;

1. FINAL KEYWORD IN PHP :-

Final class amd final function in php

1. Final function in php :- if we make function as final then we can not override it in derived class .
2. Final class : if class is final then we can not extend that class in derived class but we can create object of that class.