

Assignment No.5

Lucky Page No.:
Date : 20/6/23

Q.2) What is PHP? Explain php echo and print statement.

→ PHP :-

- PHP is an acronym for "PHP: Hypertext Preprocessor".
- PHP is a widely-used, open source scripting language.
- PHP scripts are executed on the server.
- PHP is free to download and use.

PHP echo and print statements:-

- echo and print are more or less the same. They are both used to output data to the screen.
- The differences are small: echo has no return value while print has a return value of 1 so it can be used in expression. echo can take multiple parameters while print can take one argument. echo is marginally faster than print.

Ex- The PHP echo statement.

- The echo statement can be used with or without parentheses: echo or echo().

```
<!DOCTYPE html>
<html>
<body>
<?php
echo "<h2> PHP is fun </h2>";
echo "Hello World <br>";
echo "I'm about to learn PHP";
echo "This string was";
</body>
</html>
```

O/P.
PHP is fun
Hello World
I'm about to learn PHP
This string was

The PHP print statement :-

- The print statement can be used with or without parenthesis : print or print().

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
print "<h2> PHP is fun</h2>";
```

```
print "Hello World<br>";
```

```
print "I'am about to learn PHP";
```

```
?>
```

```
</body>
```

```
</html>
```

O/P.

PHP is fun

Hello world!

I am. about learn to php

- Q 2) In php which function is used for returning data type and value ? write any tree examples.
- The function used for returning data type and value in PHP is "var_dump()".

Examples :-

```
(1) <!DOCTYPE>
```

```
<html>
```

```
<body>
```

O/P.

string(7) "Santosh"

```
<?php
```

```
$name = "Joh Santosh";
```

```
var_dump($name);
```

```
?>
```

```
</body>
```

```
</html>
```

(2) <!DOCTYPE html>
<html>
<body>
<?php
\$rollno = 45;
var_dump(\$rollno);
?php>
</body>
</html>
Output :- int(45)

(3) <!DOCTYPE html>
<html>
<body>
<?php
\$is_student = true;
var_dump(\$is_student);
?php>
</body>
</html>
Output :- bool(true).

Q.3) Write a short note on PHP string.

→ PHP string Functions :-

i) strlen() - Return the length of a string.

ex- <!DOCTYPE html>
<html>
<body>

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

| O/P-
12.

<body>
</html>

ii) strrev() - Reverse a string

ex-<!DOCTYPE html>
<html>
<body>

<?php
echo strrev(strrev("Hello world!"));
?>

| O/P-
Hello world

</body>
</html>

iii) str_word_count() — count words in a string .

ex- <!DOCTYPE html>
<html>
<body>

<?php

echo str_word_count ("Hello world!");
 ?>

</body>
 </html>

OIP-
 2.

iv) strpos() - Search For a Text Within a String.

- It search specific text within a string. If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.
- The First position of character in a string is 0 (dot 1)

ex- <!DOCTYPE html>
 <html>
 <body>

<?php
 echo strpos ("Hello world", "world");
 ?>

</body>
 </html>

OIP-
 6

v) str_replace() - Replace Text with a string.

- The PHP str_replace() function replaces some characters with some other characters in a string.

ex- <!DOCTYPE html>
 <html>
 <body>

```
<?php
echo str_replace("world", "Dolly", "Hello world!");
?>

</body>
</html>
```

O/P.
Hello Dolly.

* PHP String Operators :-

operator	Name	Example	Result
.	concatenation	\$txt1.\$txt2 - Concatenation of \$txt1 and \$txt2.	
.=	concatenation assignment	\$txt1 .= \$txt2 - Appends \$txt1 to \$txt2	

Ex- <?php
 \$txt1 = "Hello";
 \$txt2 = "World!";
 echo \$txt1.\$txt2;
?>

O/P. Hello World!

<?php
 \$txt1 = "Hello";
 \$txt2 = "World!";
 \$txt1 .= \$txt2;
 echo \$txt1;
?>

O/P. Hello World!

Q.4) Explain case-sensitive and case-insensitive constant.

→ PHP Constants - A constant is an identifier (name) for a simple value. The value cannot be changed during the script.

- A valid constant name starts with a letter or underscore (no \$ sign before constant name)

- To create a constant, use the `define()` function.

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

parameters :

- name - Specific name of the constant.

- Value - Specific value of the constant.

- case-insensitive - Specific whether the constant name should be case-insensitive. Default is false.

- A case-sensitive - If constant is defined as

`MY_CONSTANT`, it must be used as `MY_CONSTANT` and not use `my-constant` or `My-constant`.

- A case-insensitive - If a constant is defined as

`MY_CONSTANT`, it can be used as `my-constant` or `My-constant`.

examples :- ① Case-sensitive -

`<!DOCTYPE html>`

`<html>`

`<body>`

`<?php`

`// case-sensitive constant name`

```

define("GREETING","Welcome to MCA!");
echo GREETING;
?>
</body>
</html> | O/P-

```

Welcome to MCA!

② case-insensitive-

```

def <!DOCTYPE html>
    <html>
        <body>
            <?php

```

//case-insensitive constant name

```

define("GREETING","Welcome to MCA!");
?> echo greeting ?>
    </body>
    </html> | O/P-

```

Welcome to MCA!

Q5) Write a PHP Program to print table of a number
7.

```

→ <!DOCTYPE html>
    <html>
        <body>

```

```

            <?php
                $num=7;
                for ($i=1; $i<=10; $i++) {
                    echo $num."x".$i."=". $num*$i. "<br>";
                }
?>
</body></html>

```

Output :-

$7 \times 1 = 7$
 $7 \times 2 = 14$
 $7 \times 3 = 21$
 $7 \times 4 = 28$
 $7 \times 5 = 35$
 $7 \times 6 = 42$
 $7 \times 7 = 49$
 $7 \times 8 = 56$
 $7 \times 9 = 63$
 $7 \times 10 = 70$

Q6) Write a PHP program to check Armstrong number (371), (342).

→ <!DOCTYPE html>
 <html>
 <body>

```
<?php
$num = 371;
$total = 0;
$x = $num;
```

while (\$x != 0)

{

\$rem = \$x % 10;

\$total = \$total + \$rem * \$rem * \$rem;

\$x = \$x / 10;

}

if (\$num == \$total)

{

echo "Yes, it is an Armstrong number";

```
else
{
echo "No, it is not Armstrong number";
}
?> </body> | O/P.
</html> | Yes, it is an Armstrong number
```

(ii) 342

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
$num = 342;
```

```
$total = 0;
```

```
$x = $num;
```

```
while ($x != 0) {
```

```
    $rem = $x % 10
```

```
    $total = $total + $rem * $rem * $rem;
```

```
    $x = $x / 10;
```

```
}
```

```
if ($num == $total)
```

```
{
```

```
    echo "Yes, it is an Armstrong number";
```

```
}
```

```
else
```

```
{
```

```
    echo "No, it is not Armstrong number";
```

```
}
```

```
?> </body> </html>
```

| O/P.

| No, It is not Armstrong number.

Q.7)

Write a PHP programs to print star triangle.

→ <!DOCTYPE html>

<html>

<body>

<?php

\$row=6;

for (\$i=\$row; \$i>=1; \$i--) {

 for (\$j=1; \$j<\$i; \$j++) {

 echo "*";

}

 echo "
";

}

?>

</body>

</html>

Output:-

* * * * *

* * * *

* * *

* *

*

Q.8) Describe terms break and continue in PHP.

→ * PHP Break &

You have already - It was used to "jump out of a switch statement".

- The break statement can also be used to jump out of a loop.
- This example jumps out of the loop when x is equal to 4;

```
<!DOCTYPE html>
<html>
<body>
<?php
for ($x=0; $x<10; $x++) {
    if ($x==4) {
        break;
    }
}
```

echo "The number is : \$x
";

}

?>

```
</body>
</html>
```

Output: The number is : 0

The number is : 1

The number is : 2

The number is : 3

* PHP continue

- The continue statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.
- The example / skip value of 4:

```
<!DOCTYPE html>
<html>
<body>
<?php
for ($x=0; $x<10; $x++) {
if ($x==4) {
    continue;
}
echo "The number is: $x<br>";
}
```

```
}
```

```
?>
```

```
</body>
</html>
```

Output:

The number is: 0

The number is: 1

The number is: 2

The number is: 3

The number is: 5

The number is: 6

The number is: 7

The number is: 8

The number is: 9

Q.9) Write note on php user defined functions with example.

- Besides the built-in PHP functions, it is possible to create your own functions.
- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute automatically when a page loads.
A function will be executed by a call to the function.

- Create user defined function -
- A user-defined function declaration starts with the word function;

Syntax:

```
function functionName() {
    code to be executed;
}
```

Note - A function name must start with a letter or an underscore. Function names names are not case-sensitive.

example:

```
<!DOCTYPE html>
<html>
<body>
<?php
function writeMsg() {
    echo "Hello world!";
}
```

writeMsg(); ? >

</body>

</html>

c:\> Hello world!

Q.10) Explain in brief php file handling.

→ PHP File system allows us to create file, read file line by line, read file character by character , write file , append file, delete file

- PHP open File - fopen()

- The PHP fopen function is used to open a file.

- Syntax :

```
resource fopen(string $filename, string $mode)
```

- example:

```
$file=fopen("sp.txt","r");
```

```
$file=fopen("sp.txt","w");
```

```
fclose($file);
```

- PHP Read File - fread()

- The PHP fread() function is used to read the content of the file. It accepts two arguments: resources and file size.

- Syntax:

```
string fread(resource $handle, int $length)
```

- Create txt file in c:/xampp/htdocs

- example:

```
<?php
```

```
$filename = "sp.txt";
```

```
$handle=fopen($filename,"r");
```

```
$contents = fread($handle, filesize($filename));
```

```
echo $contents;  
fclose($handle);
```

- PHP Write File - fwrite()

- The PHP fwrite() function is used to write content of the string into file.

- syntax:

```
int fwrite(resource $handle, string $string  
[, int $length])
```

- example:

```
<?php  
$fp = fopen ("sp.txt", "w");  
fwrite ($fp, 'Hello');  
fwrite ($fp, 'World');  
fclose ($fp);
```

```
echo "File written successfully";
```

```
?>
```

- PHP Delete File - unlink()

- The PHP unlink() function is used to delete file.

- syntax:

```
bool unlink (string $filename [, resource  
[$context]])
```

- example:

```
<?php  
unlink ("sp.txt");  
echo "File deleted successfully";  
?>
```