

PHP



PHP: Hypertext Preprocessor

Introduction to PHP:-



- ☐ *PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.*
- ☐ *PHP stands for “PHP: Hypertext Preprocessor”.*
- ☐ *Syntax based on Perl, Java , And C.*
- ☐ *Very good for creating dynamic content.*
- ☐ *PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.*
- ☐ *Powerful, But somewhat risky!*

What can PHP Do:-



- ☐ *PHP can generate dynamic content.*
- ☐ *PHP can create, open, read, write, delete and close files on the server.*
- ☐ *PHP can collect form data.*
- ☐ *PHP can encrypt data.*
- ☐ *Use Xampp as a server.*
- ☐ **Rasmus Lerdorf** unleashed the first version of PHP way back in 1994.

How To Run Program In PHP:-



- ☐ *URL Open.*
- ☐ *Server Must Be ON.*
- ☐ *URL Address:- 127.0.0.1/folder name/file name*
or
local host/folder name/filename

PHP Scripts:-



- ☐ *Typically file ends in “.php”- -this is set by the web server configuration.*
- ☐ *Separated in files with the <?php ?>.*
- ☐ *PHP commands can make up an entire files, or can be contained in html- -this is a choice....*
- ☐ *Program lines end in “ ; ” or we get an error.*
- ☐ *Server recognizes embedded script and executes.*
- ☐ *Result is passed to browser, Source isn't visible.*

Example of PHP Script:-



```
□<!DOCTYPE html>
<html>
<body>
<?php
echo "My first PHP
script!";
?>
</body>
</html>
```

Output:-

My first PHP script!

String Print:-



☐ “ ”
☐ ‘ ’
☐ ;
☐ “ ‘ ’ ”
☐ “ ” ‘ ’

Note:- If we want embed another language, like HTML in PHP then all HTML works as a string.

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 expressions. echo can take multiple parameters (although such usage is rare) while print can take one argument. echo is marginally faster than print.*

Example of echo statements:-

```
□ <!DOCTYPE html>
<html>
<body>
<?php
echo "<h2>PHP is
Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn
PHP!<br>";
echo "This ", "string ", "was
", "made ", "with multiple
parameters.";
?>
</body>
</html>
```

□ Output:-

PHP is Fun!

Hello world!

I'm about to learn PHP!

*This string was made with
multiple parameters.*

Example of print statements:-

```
□ <!DOCTYPE html>
  <html>
  <body>
  <?php
  print "<h2>PHP is
  Fun!</h2>";
  print "Hello world!<br>";
  print "I'm about to learn
  PHP!";
  ?>
  </body>
  </html>
```

□ Output:-

PHP is Fun!

Hello world!

I'm about to learn PHP!

PHP Variables:-



□ Rules for PHP variables:

- A variable starts with the \$ sign, followed by the name of the variable.*
- A variable name must start with a letter or the underscore character.*
- A variable name cannot start with a number.*
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _).*
- Variable names are case-sensitive (\$age and \$AGE are two different variables).*

PHP Data Types:-



- *Types of declaration of variable is known as Data Type.*
- *PHP supports the following data types:*
 - *String*
 - *Integer*
 - *Float (floating point numbers - also called double)*
 - *Boolean*
 - *Array*
 - *Object*
 - *NULL*
 - *Resource*

PHP String:-



❑ *<!DOCTYPE html>*

<html>

<body>

<?php

\$x = "Hello world!";

\$y = 'Hello world!';

echo \$x;

*echo "
";*

echo \$y;

?>

</body>

</html>

❑ *Output:-*

Hello world!

Hello world!

PHP Integer:-



- *An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.*

****Rules for integers:***

- *An integer must have at least one digit.*
- *An integer must not have a decimal point.*
- *An integer can be either positive or negative.*
- *Integers can be specified in three formats: decimal (10-based), hexadecimal (16-based - prefixed with 0x) or octal (8-based - prefixed with 0).*

Example of PHP Integer:-



```
<?DOCTYPE html>
<html>
<body>
<?php
$x = 5985;
var_dump($x);
?>
</body>
</html>
```

Output:-
int(5985)

Example PHP Float:-

□ *<!DOCTYPE html>*

<html>

<body>

<?php

\$x = 10.365;

var_dump(\$x);

?>

</body>

</html>

□ *Output:-*

float(10.365)

PHP Boolean:-



□ *A Boolean represents two possible states: TRUE or FALSE.*

□ *\$x = true;*
\$y = false;

Booleans are often used in conditional testing. You will learn more about conditional testing in a later chapter of this tutorial.

PHP Arrays:-



□ *An array stores multiple values in one single variable.*

□ ***Example***

```
<?php  
$cars = array("Volvo","BMW","Toyota");  
var_dump($cars);  
?>
```

□ ***Output:-***

```
array(3) { [0]=> string(5) "Volvo" [1]=> string(3)  
"BMW" [2]=> string(6) "Toyota" }
```

PHP Objects:-



- ☐ *An object is a data type which stores data and information on how to process that data.*
- ☐ *In PHP, an object must be explicitly declared.*
- ☐ *First we must declare a class of object. For this, we use the class keyword. A class is a structure that can contain properties and methods.*

PHP Null Value:-



- ☐ *Null is a special data type which can have only one value: NULL.*
- ☐ *A variable of data type NULL is a variable that has no value assigned to it.*

Note: *If a variable is created without a value, it is automatically assigned a value of NULL.*

Variables can also be emptied by setting the value to NULL:

Example of PHP Null Value:-

```
☐ <!DOCTYPE html>
<html>
<body>
<?php
$x = "Hello world!";
$x = null;
var_dump($x);
?>
</body>
</html>
```

☐ Output:-

NULL

PHP Operator:-



- ☐ *Operators are used to perform operations on variables and values.*

PHP divides the operators in the following groups:

- ☐ *Arithmetic operators. (For normal operations)*
- ☐ *Assignment operators. (=)*
- ☐ *Relational operators. (< , > , < = , = >)*
- ☐ *Logical operators. (and , or)*

PHP Conditional Statements:-



In PHP we have the following conditional statements:

- ❑ ***if statement*** - executes some code if one condition is true.
- ❑ ***if...else statement*** - executes some code if a condition is true and another code if that condition is false.
- ❑ ***if...elseif....else statement*** - executes different codes for more than two conditions
- ❑ ***switch statement*** - selects one of many blocks of code to be executed.

PHP- The If Statement:-



☐ *The if statement executes some code if one condition Is true.*

☐ *Example*

```
<?php  
$t = date("H");  
if ($t < "20") {  
    echo "Have a good day!";  
}  
?>
```

Output:-

Have a good day!

PHP- The If ... Else Statement:-



- *It executes some code if a condition is true and another code if that condition is false.*

□ **Example:-**

```
<?php  
$t = date("H");
```

```
if ($t < "20") {  
    echo "Have a good day!";  
} else {  
    echo "Have a good night!";  
}  
?>
```

Output:-

Have a good day!

PHP- The If .. Elseif..else Statement:-



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

□ **Syntax**

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

Example If .. Elseif..else Statement:-



```
❑<?php
    $t = date("H");
    if ($t < "10") {
        echo "HII!";
    } elseif ($t < "20") {
        echo "Have a good day!";
    } else {
        echo "Have a good night!";
    }
?>
```

Output:-

Have a good day!

The PHP Switch Statement:-



□ *“Use the switch statement to select one of many blocks of code to be executed”.*

□ **Example:-**

```
?php
$color = "red";
switch ($color) {
    case "red":
        echo "Your color is red!";
        break;
    case "blue":
        echo "Your color is blue!";
        break;
    case "green":
        echo "Your color is green!";
        break;
    default:
        echo "Your color is neither red, blue, nor green!";
}
?>
```

Output:-

Your color is red!

How To Get Form Value:-



Super Global Variable

- *\$_GET* [' '];
- *\$_POST* [' '];
- *\$_REQUEST* [' '];

How To upload image on a server:-



PHP Loop



In PHP, we have the following looping statements:-

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

PHP for loop:-



- ☐ *The for loop is used when you know in advance how many times the script should run.*

Parameters:

- ☐ *init counter: Initialize the loop counter value.*
- ☐ *test counter: Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.*
- ☐ *increment counter: Increases the loop counter value.*

Example 1 of for loop:-

```
<!DOCTYPE html>
<html>
<body>
<?php
for ($x = 0; $x <= 10;
    $x++) {
    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: 4
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9
The number is: 10
```

Example 2 of for loop:-

```
<!DOCTYPE html>
<html>
<body>
<?php
for ($i = 1; $i <= 10;
    $i++)
{
    $c=$i*2;
    echo $c."<br>";
}
```

Output:-

2
4
6
8
10
12
14
16
18
20

PHP while loop:-



- *The while loop executes a block of code as long as the specified condition is true.*

Syntax

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

Example 1 of while loop:-



```
<!DOCTYPE html>
<html>
<body>
<?php
$a = 1;
while($a <= 5) {
    echo $a "<br>";
    $a++;
}
?>
</body>
</html>
```

Output:-

1

2

3

4

5

Example 2 of while loop:-



□ *DOCTYPE html>*

<html>

<body>

<?php

\$a = 1;

while(\$a <= 10) {

*\$c=\$a*2;*

*echo \$c "
";*

\$a++;

}

?>

</body>

</html>

□ *Output:-*

2

4

6

8

10

12

14

16

18

20

PHP foreach loop:-



□ *The foreach loop works only on arrays, and is used to loop through each key/value pair in an array.*

□ *Syntax:-*

foreach(array name of variable as desired variable)

{

Desired variable;

}

Example of foreach loop:-



```
❑ <!DOCTYPE html>
<html>
<body>
<?php
$colors
= array("red", "green", "blue", "yellow");
foreach ($colors as $value) {
    echo "$value <br>";
}
?>
</body>
</html>
```

Output:-

red
green
blue
yellow

PHP String:-



- ☐ *strupper*
- ☐ *strtolower*
- ☐ *strrev*
- ☐ *strlen*
- ☐ *substr*

Example of strings:-



```
☐ <?php
```

```
$a="rcew";
```

```
echo strupper($c);
```

```
echo strtolower($c);
```

```
echo strrev($c);
```

```
echo strlen($c);
```

```
?>
```

☐ *Output:-*

RCEW

rcew

wecr

4

Example of substring:-



```
<?php  
$a=7062033739;  
echo“xxxxxxx”.substr($a,7,3);  
?>
```

Output:-

xxxxxxx739

PHP function:-



- ❑ *include- It is used to include the file, if file is not found then warning will be show and script continue.*
- ❑ *require- It is used to include the file, But file is not found in error will be show with warning and screen stop.*
- ❑ *include_once-*

.require_once-

- *Implode- Implode is used to convert array Into string.*
- *Explode- Explode is used to convert into array.*

PHP Array



□ *An array is a special variable, which can hold more than one value at a time.*

□ *Create an Array in PHP-*

In PHP, the array() function is used to create an array:

array();

PHP Array function:-



- ❑ *max:- find the largest value.*
- ❑ *min:-find the smallest value.*
- ❑ *array-sum:-means sum between two arrays.*
- ❑ *array-product:- means multiple.*
- ❑ *array-merge:- merge between two arrays.*
- ❑ *sort:-accending order.*
- ❑ *r-sort:-deccending order.*
- ❑ *array-pop:- delete last value.*
- ❑ *array push:-add value in last.*

- *array-shift:- delete first value.*
- *array-un shift:- add value in first*
- *print-r:- print the index and name value first.*

Example of max:-

```
☐ <?php  
$arr=array(4,9,80,95,  
300,1,600);  
echo max($arr);  
?>
```

☐ *Output:-*
600

Example of min:-

```
☐ <?php  
$arr=array(4,9,80,95,  
300,1,600);  
echo min($arr);  
?>
```

☐ Output:-
1

Example of sum:-



```
<?php
$arr=array(4,9,80,
95,300,1,600);
echo array
sum($arr);
?>
```

Output:-
1089

Example of product:-



```
☐ <?php
```

```
$arr=array(4,9,80,  
95,300,1,600);
```

```
echo
```

```
array_product($arr  
r);
```

```
?>
```

```
☐ Output:-
```

```
492480000000
```

Example of merge:-



```
□ <?php
$arrr=array(4,9,80,95,300
,1,600);
$arr=array(7,9,34,86,986,)
;
$ar=array_merge($arrr,$
arr);
foreach($ar as $d)
{
    echo $d."<br>";
}
?>
```

□ Output:-

```
4
9
80
95
300
1
600
7
9
34
86
986
```

Example of sort:-



```
<?php
$arrr=array(4,9,80,95,
300,1,600);
sort($arrr);
foreach($arrr as $n)
{
    echo $n."<br>";
}
?>
```

Output:-

```
1
4
9
80
95
300
600
```

Example of rsort:-

```
<?php
$arrr=array(4,9,80,95,
300,1,600);
rsort($arrr);
foreach($arrr as $d)
{
    echo $d."<br>";
}
?>
```

Output:-

```
600
300
95
80
9
4
1
```

Example of array_pop:-

```
<?php
$arrr=array(4,9,80,95,
300,1,600);
array_pop($arrr);
foreach($arrr as $d)
{
    echo $d."<br>";
}
?>
```

Output:-

```
4
9
80
95
300
1
```

Example of array_push:-

```
<?php
$arrr=array(4,9,80,95,
300,1,600);
array_push($arrr,111,1
21);
foreach($arrr as $d)
{
    echo $d."<br>";
}
?>
```

Output:-

```
4
9
80
95
300
1
600
111
121
```


Example of array_shift:-

```
<?php
$arrr=array(4,9,80,95,
300,1,600);
array_shift($arrr);
foreach($arrr as $d)
{
    echo $d."<br>";
}
?>
```

Output:-


```
9
80
95
300
1
600
```

Example of array_unshift:-

```
<?php
$arrr=array(4,9,80,95,
300,1,600);
array_unshift($arrr,00
,131);
foreach($arrr as $d)
{
    echo $d."<br>";
}
?>
```

Output:-

```
0
131
4
9
80
95
300
1
600
```



```
<?php
$arrr=array(4,9,80,95,
300,1,600);
print_r($arrr);
foreach($arrr as $d)
{
    echo $d."<br>";
}
?>
```

Output:-

```
Array ( [0] => 4 [1] => 9 [2]
=> 80 [3] => 95 [4] => 300
[5] => 1 [6] => 600 ) 4
9
80
95
300
1
600
```