**What is PHP?**

=>PHP is a server side scripting language.

=>that is used to develop Static websites or Dynamic websites or Web applications.

=>PHP stands for Hypertext Pre-processor, that earlier stood for Personal Home Pages.

=>PHP scripts can only be interpreted on a server that has PHP installed.

=>A PHP file contains PHP tags and ends with the extension “.php”.

## PHP Data Types

* Alphanumeric characters are classified as strings
* Whole numbers are classified integers
* Numbers with decimal points are classified as floating points.
* True or false values are classified as Boolean.

Integer – whole numbers e.g. -3, 0, 69. The maximum value of an integer is platform-dependent. On a 32 bit machine, it’s usually around 2 billion. 64 bit machines usually have larger values. The constant PHP\_INT\_MAX is used to determine the maximum value.

Floating point number – decimal numbers e.g. 3.14. they are also known as double or real numbers. The maximum value of a float is platform-dependent. Floating point numbers are larger than integers.

Character string – e.g. Hello World

Boolean – e.g. True or false.

## PHP Constant

**Define constant**– A constant is a variable whose value cannot be changed at runtime.

Suppose we are developing a program that uses the value of PI 3.14, we can use a constant to store its value.

## PHP Operators

### ***Arithmetic operators***

Arithmetic operators are used to perform arithmetic operations on numeric data. The concatenate operator works on strings values too. PHP supports the following operators.

| **Operator** | **Name** | **Description** | **Example** | **Output** |
| --- | --- | --- | --- | --- |
| + | Addition | Summation of x and y | 1 + 1; | 2 |
| – | Subtraction | Difference between x and y | 1 – 1; | 0 |
| \* | Multiplication | Multiplies x and y | 3 \* 7; | 21 |
| / | Division | Quotient of x and y | 45 / 5; | 9 |
| % | PHP Modulus | Gives remainder of dividing x and y | 10 % 3; | 1 |
| -n | Negation | Turns n into a negative number | -(-5); | 5 |
| x . y | Concatenation | Puts together x and y | “PHP” . ” ROCKS”;10 . 3; | PHP ROCKS103 |

### ***Assignment Operators***

Assignment operators are used to assign values to variables. They can also be used together with arithmetic operators.

| **Operator** | **Name** | **Description** | **Example** | **Output** |
| --- | --- | --- | --- | --- |
| x = ? | assignment | Assigns the value of x to ? | $x = 5; | 5 |
| x += ? | addition | Increments the value of x by ? | $x = 2;$x += 1; | 3 |
| X -= ? | subtraction | Subtracts ? from the value of x | $x = 3;$x -= 2; | 1 |
| X \*=? | multiplication | Multiplies the value of x ? times | $x = 0;$x \*=9; | 0 |
| X /=? | division | Quotient of x and ? | $x = 6;$x /=3; | 2 |
| X %=? | modulus | The reminder of dividing x by? | $x = 3;$x %= 2; | 1 |
| X .=? | concatenate | Puts together items | ” $x = ‘Pretty’;$x .= ‘ Cool!’;” | Pretty Cool! |

### ***Comparison operators***

Comparison operators are used to compare values and data types.

| **Operator** | **Name** | **Description** | **Example** | **Output** |
| --- | --- | --- | --- | --- |
| X == y | Equal | Compares x and y then returns true if they are equal | 1 == “1”; | True or 1 |
| X === y | identical | Compares both values and data types. | 1 === “1”; | False or 0. Since 1 is integer and “1” is string |
| X != y, x <> y | PHP Not equal | Compares values of x and y. returns true if the values are not equal | 2 != 1; | True or 1 |
| X > y | Greater than | Compares values of x and y. returns true if x is greater than y | 3 > 1; | True or 1 |
| X < y | Less than | Compares values of x and y. returns true if x is less than y | 2 < 1; | False or 0 |
| X >= y | Greater than or equal | Compares values of x and y. returns true if x is greater than or equal to y | 1 >=1 | True or 1 |
| X <= y | Less than or equal | Compares values of x and y. returns true if x is greater than or equal to y | 8 <= 6 | False or 0 |

### ***Logical operators***

When working with logical operators, any number greater than or less than zero (0) evaluates to true. Zero (0) evaluates to false.

| **Operator** | **Name** | **Description** | **Example** | **Output** |
| --- | --- | --- | --- | --- |
| X and y, x && y | And | Returns true if both x and y are equal | 1 and 4;True&& False; | True or 1False or 0 |
| X or y, x || y | Or | Returns true if either x or y is true | 6 or 9;0 || 0; | True or 1False or 0 |
| X xor y | Exclusive or, xor | Returns true if only x is true or only y is true | 1 xor 1;1 xor 0; | False or 0True or 1 |
| !x | Not | Returns true if x is false and false if x is true | !0; | True or 1 |

## PHP Comments

:-

[PHP Include, Require & Comments](https://www.guru99.com/images/2013/04/php_single_comment.jpg)single line comment

/\* any thinkkkkk \*/ multiline comment….

## What is a PHP Array?

A PHP array is a variable that stores more than one piece of related data in a single variable.

* [Numeric Arrays](https://www.guru99.com/arrays.html#1):-

$movie[0] = 'Shaolin Monk';

* [PHP Associative Array](https://www.guru99.com/arrays.html#2):-

$variable\_name['key\_name'] = value;

$persons = array("Mary" => "Female", "John" => "Male", "Mirriam" => "Female");

* [PHP Multi-dimensional arrays](https://www.guru99.com/arrays.html#3):-

$movies =array(

"comedy" => array("Pink Panther", "John English", "See no evil hear no evil"),

"action" => array("Die Hard", "Expendables"),

"epic" => array("The Lord of the rings"),

"Romance" => array("Romeo and Juliet")

);

* [PHP Array operators](https://www.guru99.com/arrays.html#4):-

|  |  |
| --- | --- |
| x + y | Union |
| X == y | Equal |
| X === y | Identical |
| X != y, x <> y | Not equal |
| X !== y | Non identical |

## PHP Array Functions

### Count function

The count function is used to count the number of elements that an php array contains. The code below shows the implementation.

<?php

$lecturers = array("Mr. Jones", "Mr. Banda", "Mrs. Smith");

echo count($lecturers);

?>

### is\_array function

The is\_array function is used to determine if a variable is an array or not. Let’s now look at an example that implements the is\_array functions.

<?php

$lecturers = array("Mr. Jones", "Mr. Banda", "Mrs. Smith");

echo is\_array($lecturers);

?>

### Sort

This function is used to sort arrays by the values.

If the values are alphanumeric, it sorts them in alphabetical order.

If the values are numeric, it sorts them in ascending order.

It removes the existing access keys and add new numeric keys.

The output of this function is a numeric array

<?php

$persons = array("Mary" => "Female", "John" => "Male", "Mirriam" => "Female");

sort($persons);

print\_r($persons);

?>

### ksort

This function is used to sort the array using the key. The following example illustrates its usage.

<?php

$persons = array("Mary" => "Female", "John" => "Male", "Mirriam" => "Female");

ksort($persons);

print\_r($persons);

?>

### asort

This function is used to sort the array using the values. The following example illustrates its usage.

<?php

$persons = array("Mary" => "Female", "John" => "Male", "Mirriam" => "Female");

asort($persons);

print\_r($persons);

?>