**Introduction**

PHP is one of the most widely used **server side scripting** language for web development. Popular websites like Facebook, Yahoo, Wikipedia etc.

PHP stands for Hypertext Pre-Processor. PHP is a scripting language used to develop static and dynamic webpages and web applications. Here are a few important things you must know about PHP:

* PHP is an Interpreted language, hence it doesn't need a compiler.
* To run and execute PHP code, we need a Web server on which PHP must be installed.
* PHP is a server side scripting language, which means that PHP is executed on the server and the result is sent to the browser in plain HTML.
* PHP is open source and free.

**Features of PHP**

* PHP is open source and free, hence you can freely download, install and start developing using it.
* PHP has a very simple and easy to understand syntax, hence the learning curve is smaller as compared to other scripting languages like JSP, ASP etc.
* PHP is cross platform; hence you can easily develop and move/deploy your PHP code/project to almost all the major operating systems like Windows, Linux, Mac OSX etc.
* All the popular web hosting services support PHP. Also the web hosting plans for PHP are generally the amongst the cheapest plans because of its popularity.
* Popular Content Management Systems like Joomla, Drupal etc are developed in PHP.
* With PHP, you can create static and dynamic webpages, perform file handling operations, send emails, access and modify browser cookies, and almost everything else that you might want to implement in your web project.
* PHP is fast as compared to other scripting languages like JSP and ASP.
* PHP has in-built support for MySQL, which is one of the most widely used Database management system.

**Uses of PHP**

* It can be used to create Web applications like Social Networks(Facebook, Digg), Blogs(Wordpress, Joomla), eCommerce websites(OpenCart, Magento etc.) etc.
* Common Line Scripting. You can write PHP scripts to perform different operations on any machine, all you need is a PHP parser for this.
* Create Facebook applications and easily integrate Facebook plugins in your website, using Facebook's PHP SDK. Check this link for more information.
* Sending Emails or building email applications because PHP provides with a robust email sending function.
* Wordpress is one of the most used blogging(CMS) platform in the World, and if you know PHP, you can try a hand in Wordpress plugin development.

**Requirements for PHP**

To run PHP scripts, we need the following services:

**PHP Parser**: To execute PHP scripts, PHP installation is required.

**Web Server**: Because PHP is mostly used to develop websites, hence most of its implementations comes bundled with Apache Web Server, which is required to host the application developed in PHP over HTTP.

**Database**: Any one database management system, which is generally MySQL, as PHP comes with a native support for MySQL

**XAMPP**

XAMPP stands for:

* X: Cross Platform, as it supports all the moder operating systems like Windows, Mac OSX, Linux etc.
* A: Apache Web Server
* M: MySQL database management system.
* P: PHP installation
* P: Perl scripting language

We can easily control, stop and restart, various services using the XAMPP Control Panel.

Upon successful installation, a folder with name xampp will be created in the C drive(by default). In the folder xampp there are many sub-folders like apache, cgi-bin, FileZillaFTP etc, but the most important sub-folders are:

* htdocs: This is the folder in which we will keep all our PHP files.
* mysql: This folder contains all the files for the MySQL database. By default the MySQL databse runs on port number 3306.
* php: This folder holds all the installation files for PHP. All the configurations for the current PHP installation is saved in php.ini file which is stored in this folder.

**IDEs and Editors:**

* Netbeans IDE
* Eclipse IDE
* PyStorm IDE
* Atom Editor
* Brackets Editor
* Notepad ++

**PHP Variables**

* A variable name will always start with a $ sign, followed by the variable name.
* A variable name should not start with a numeric value. It can either start with an alphabet or an underscore sign \_.
* A variable name can only contain alphabets, numbers and underscore symbol \_.
* Variable names in PHP are case-sensitive, which means $l is not same as $L.

**Echo vs print**

echo() function is used to print or output one or more strings. We have specifically mentioned string here because, the syntax of the echo function is:

echo(string)

Although you can use echo() function to output anything, as PHP parser will automaticallly convert it into string type.

echo doesn't need parenthesis, although you can use parenthesis if you want.

<?php

echo "I am open";

echo ("I am enclosed in parenthesis");

?>

<?php

$weird = "Stupid";

echo "I am $weird"; //I am Stupid

echo 'I am $weird'; // I am $weird

?>

**Note: when we use double quotes the value of the string variable gets printed, while if we use single quotes, the variable is printed as it is.**

**print**

The PHP print is exactly the same as echo, with same syntax and same usage.

**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.

**Data Types**

PHP Data types specify the different types of data that are supported in PHP language. There are total 8 data types supported in PHP, which are categorized into 3 main types. They are:

* Scalar Types: boolean, integer, float and string.
* Compound Types: array and object.
* Special Types: resource and NULL.

**PHP Boolean**

A boolean data type can have two possible values, either True or False.

$a = true;

$b = false;

NOTE: Here the values true and false are not enclosed within quotes, because these are not strings.

**PHP Integer**

An Integer data type is used to store any non-decimal numeric value within the range -2,147,483,648 to 2,147,483,647.

An integer value can be negative or positive, but it cannot have a decimal.

$x = -2671;

$y = 7007;

**PHP Float**

Float data type is used to store any decimal numeric value.

A float(floating point) value can also be either negative or positive.

$a = -2671.01;

$b = 7007.70;

**PHP String**

String data type in PHP and in general, is a sequence of characters(or anything, it can be numbers and special characters too) enclosed within quotes. You can use single or double quotes.

$str1 = "Hello";

$str2 = "What is your Roll No?";

$str3 = "4";

echo $str1;

echo "<br/>";

echo $str2;

echo "<br/>";

echo "Me: My Roll number is $str3";

Output:

Hello

What is your Roll No?

Me: My Roll number is 4

**PHP NULL**

NULL data type is a special data type which means nothing. It can only have one value, and that is NULL.

If you create any variable and do not assign any value to it, it will automatically have NULL stored in it.

Also, we can use NULL value to empty any variable

**Constants**

Constants are variables whose value cannot be changed. In other words, once you set a value for a constant, you cannot change it.

In PHP, there are two ways to define a constant:

* Using the define() method.

define(name, value, case-insensitive)

Parameters:

name: Name of the constant

value: Value of the constant

case-insensitive: Specifies whether the constant name is case sensitive or not. It's default value is false, which means, by default, the constant name is case sensitive.

Example: define(OMG, "Oh! my God.", true);

* Using the const keyword.

const OMG = "Oh! my God.";

Note: while using const keyword, the constant name is always case sensitive.

**Operators**

Operators are used to perform operations on PHP variables and simple values.

In PHP there are total 7 types of operators, they are:

* Arithmetic Operators
* Assignment Operators
* Comparison Operators
* Increment/Decrement Operators
* Logical Operators
* String Operators
* Array Operators

**Arithmetic Operators**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Operator | What does it do? | Example |
| Addition | + | It is used to perform normal addition. | $a + $b |
| Subtraction | - | It is used to perform normal subtraction. | $a - $b |
| Multiplication | \* | It is used to perform multiplication. | $a \* $b |
| Division | / | It is used to perform division. | $a / $b |
| Exponent | \*\* | It returns the first operand raised to the power the second operand. $a \*\* $b = $a$b | $a \*\* $b |
| Modulus(or, Remainder) | % | It returns the remainder of first operand divided by the second operand | $a % $b |

**PHP Assignment Operators**

Assignment operators are used to assign values to variables, either as it is or after performing some arithmetic operation on it. The most basic assignment operator is equal to=.

|  |  |
| --- | --- |
| Operator | Usage |
| = | $a = $b, will save the value of variable $b to the variable $a |
| +- | $a += $b is same as $a + $b |
| -= | $a -= $b is same as $a - $b |
| \*= | $a \*= $b is same as $a \* $b |
| /= | $a /= $b is same as $a / $b |
| %= | $a %= $b is same as $a % $b |

So basically, the assignment operator provides us with shorthand techniques to perform arithmetic operations.

**PHP Comparison Operators**

As the name suggest, these are used to compare two values.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Operator | What does it do? | Example |
| Equal | == | It returns true if left operand is equal to the right operand. | $a == $b |
| Identical | === | It returns true if left operand is equal to the right operand and they are of the same type. | $a === $b |
| Not Equal | != | It returns true if left operand is not equal to the right operand. | $a != $b |
| Not Identical | !== | It returns true if left operand is not equal to the right operand, and they are of different type as well. | $a !== $b |
| Greater than | > | It returns true if left operand is greater than the right operand. | $a > $b |
| Less than | < | It returns true if left operand is less than the right operand. | $a < $b |
| Greater than or equal to | >= | It returns true if left operand is greater than or equal to the right operand. | $a >= $b |
| Less than or equal to | <= | It returns true if left operand is less than or equal to the right operand. | $a <= $b |

**PHP Increment/Decrement Operators**

These operators are unary operators, i.e they require only one operand.

|  |  |
| --- | --- |
|  |  |
| Operator | Usage |
| ++$a | Pre Increment, It will first increment the operand by 1(add one to it) and then use it or return it. |
| $a++ | Post Increment, It will first return the operand and then increment the operand by 1. |
| --$b | Pre Decrement, It will first decrement the operand by 1(subtract one from it) and then use it or return it. |
| $b-- | Post Decrement, It will first return the operand and then decrement the operand by 1. |

These operators are very useful and handy when use loops or when we have simply increment any value by one in our program/script.

**PHP Logical Operators**

Logical operators are generally used when any action depends on two or more conditions.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Operator | What does it do? | Example |
| And | and or && | It returns true if both the operands(or expressions) returns true. | $a && $b |
| Or | or or || | It returns true if any one out of the two operands(or expressions) returns true, or both return true. | $a || $b |
| Xor | xor | It returns true if any one out of the two operands(or expressions) returns true, but not when both return true. | $a xor $b |
| Not | ! | This is a unary operator. It returns true, if the operand(or expression) returns false. | !$a |

**PHP String Operators**

String operators are used to perform operations on string. There are only two string operators, generally PHP built-in functions are used to perform various operations on strings.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Operator | What does it do? | Example |
| Concatenation | . (a dot) | It is used to concatenate(join together) two strings. | $a.$b |
| Concatenation Assignment | .= | It is used to append one string to another. | $a .= $b |