# PHP Basic

## What is PHP?

The full form of PHP is Hypertext Preprocessor. It is an open source scripting language widely used for web development. The code is executed on the server.

## Basic Syntax

We can place php script anywhere in the document and it starts with <?php and ends with ?>. we can show output on webpage with the help of echo (e.g. <?php Echo “This is a text.”; ?> ). In PHP, the keywords like if, elseif, echo, classes, functions, and user-defined functions are not case-sensitive whereas all variable names are case-sensitive.

## Constants and Variables

The constants cannot be changed or undefined once they are defined. We can use constants like  
const x = 100;  
or  
define (“x”, 100);  
Here define() is case-insensitive but const is case-sensitive.

A variable contains $ sign and the variable name (e.g. $name = “MAX” ;). Here, the $name variable holds the value called MAX. For text value, we use “” around the value. There are some established rules to maintain for PHP variables. Such as,

* It will start with $sign, followed by the name of the variable.
* The variable name should start with a letter or underscore character (\_).
* The variable name can not be started with number
* The variable name should contain alpha numeric characters and underscores
* It is case-sensitive.

PHP automatically associates a **data type (e.g. string, integer, float, Boolean, array, object, null, resource)** to the variable according to its value as it is a loosely typed language. But from PHP 7.0, we can declare data type. As a result, it gives us the scope of declaring the expected data type.

## Data Types and Casting

### Data Types

The variable’s value can be different types of data types. The data types supported by PHP are,

* **String**

$x = “Name”;

Here the data type of the value of x variable is string. Any text value with quotes (e.g. “” or ‘’) are considered as string data type.

* **Integer**

$y = 10;

An integer data type is what a non-decimal number is. So, here we can say the data type value of $y variable is an integer data type. For integer data type, the number must have to be at least one digit, must not have a decimal point, can be either positive or negative. We can use var\_dump() function to see the value and data type of the variables ( e.g. var\_dump($y); ).

* **Float**

$z = 3.5;

A float data type is what a decimal number is or a number in exponential form is.

* **Boolean**

**$active = true;**

True or False. These two are the possible states of Boolean data types. We can get the best out of Boolean in condition such as,

If ($active = true) {

echo $y + $z;

} else {

echo “$x”;

}

* **Array**

$brand = array(“Samsung”, “Huawei”, “Xiaomi”);

When we need to store multiple values in a variable we can use array like the above example. Here, the data type of the brand variable is an array data type.

* **Object**

Class Office {

public $employees;

public function \_\_construct($employees){

$this->employees = $employees;

}

public function totalEmployees(){

echo “This office has total” . $this->employees . “.”;

}

$myOffice = new Office(120);

var\_dump($myOffice);

Here the data type of myOffice variable is object data type as it contains an object (e.g. new Office(120); ). Object and classes are basically two main aspects of object-oriented programming. Generally, A class is template for objects and an object is an instance of a class.

* **Null**

$nothing = null;

Null is a specific data type and it has only one state which is null. So, when we say the variable nothing’s value is null, it means the data type of the nothing variable is null.

* **Resource**

Since it is an advance topic, we will discuss it later.

### Casting

The concept of casting says that sometimes we need to declare the variables' specific data type or sometimes we need to change the data type from one to another. In this case, casting can be done.

$x = 5;  
Here, the data type of the variable x is integer. So, if need to make its’ data type string, we can do like this,  
$x = (string) $x;

It will convert the data type of variable x into string like “5”.

Similarly, we can use casting like (integer), (bool), (array), (object), (unset) to do casting.

## String data type

$firstName = ‘Miraj’

$name = “ $firstName Hossain”

Here, $name will print Miraj Hossain. But if we wrote $name like   
$name = ‘$firstName Hossain’

It would print $firstName Hossain. This is actually the primary difference between single quote strings and double quote strings. The double quotes strings can parse while the single quote strings cannot.

### Heredoc and Nowdoc strings

Heredoc works like double quote and Nowdoc works like single quote. The main difference between Heredoc and Nowdoc is that Heredoc can parse while Nowdoc cannot. Both of these syntaxes are quite identical. Both use <<< and an identifier at the start point and end point and content inside of it. But in case of Nowdoc we need to write the identifier start point with single quote.

Heredoc Code example,  
$x = 1;

$y = 2;

$text = <<< Text

$x line

$y line

Text;

echo nl2br($text);

it will print like   
1 line  
2 line

Nowdoc Code example,

$x = 1;

$y = 2;

$text = <<< ‘Text’

$x line

$y line

Text;

echo nl2br($text);

it will print like   
$x line  
$y line

Heredoc and Nowdoc are comparatively readable when compress expressions are used. One thing to be remembered that the space inside of identifiers will be counted as strings.

## Array Data type

In PHP, array is actually an ordered map meaning it associates values to keys. We can use it for several cases as it can be treated like an array, list ()