TOPICS

1. About PHP

PHP is an acronym of Hypertext Preprocessor

PHP or Hypertext Preprocessor is a server side web scripting language that can be embedded in HTML.

All of the php scripts are executed in the web browser

PHP is an open source web development and it is also free so you can download php on its official page <http://php.net/downloads.php>.

PHP was developed in 1994 by Rasmus Lerdorf. PHP Is the successor of PHP/FI. Rasmus Lerdorf used php on his home page so that he can track of who are looking at his resume.

**Version History:**

PHP 1.0 – June 8, 1995

PHP 2.0 – November 1, 1997

PHP 3.0 – June 6 1998

PHP 4.0 – May 22, 200

PHP 4.1 – December 10, 2001

PHP 4.2 – April 22, 2002

PHP 4.3 – December 27, 2002

PHP 4.4 – July 11, 2005

PHP 5.0 – July 13, 2004

PHP 5.1 – November 24, 2005

PHP 5.2 – November 2, 2006

PHP5.3 – June 30, 2009

PHP 5.4 – March 1, 2012

PHP 5.5 – June 20, 2013

PHP 5.6 – August 28,2014

PHP 6.x – Not released

PHP 7.0 – December 3, 2015

PHP 7.1 – December 1, 2016

PHP 7.2 – November 30, 2017

**Installation of PHP**

There are three components that you need to have in order to use, create, or develop php web pages. These components also need to be installed in your computer system. This Three components are Web Server, Database, and PHP Parser.

* Web Server – PHP also works with IIS, the common web server that most of the developer’s use is the Apache server. If you don’t have the apache server installed in your system you can download this on <https://httpd.apache.org/download.cgi>.
* Database – PHP also works with databases there are many databases available online but the most frequent and common database that was being used in PHP was the Oracle MySql databases such as PhpMyAdmin. If you don’t have this installed in your system you can download this on <https://www.mysql.com/downloads/>
* Parser – To generate HTML results to be sent to the web server you need to have a parser to process your scripts. To install PHP parser in your computer, follow the guide below.

INSTALLATION OF PHP PARSER

To check or configure if you have properly installed PHP in your computer type this URL in your browser. http://127.0.0.1/info.php

If it displays something about related on installation of your php it means that you have properly installed and configured your php in your computer.

Configure Apache by following this guide <https://www.tutorialspoint.com/php/php_apache_configuration.htm>

Configure your PHP.INI

This is the final step in installing your PHP you will be guided by this manual.

<https://www.tutorialspoint.com/php/php_ini_configuration.htm>

If you are using Windows IIS, then you should also configure your IIS which you can refer to your IIS reference manual for more instructions and guides.

1. **PHP DATA TYPES AND PHP ARRAY**

PHP Data Types

In php you can store your data by using data types, this data types are also familiar In java. This variable can store data depending on their kind like numeric, Boolean, array, and more. This are the data types that php supports.

Integer

String

Boolean

Array

NULL

Object

Resource

**PHP integer**

An integer is non-decimal.

An integer is a whole number

An integer is not a fraction

Example of PHP integer

<!DOCTYPE html>

<html>

<body>

<?php

$integer = 1000;

var\_dump($integer);

?>

</body>

</html>

**PHP String**

This are characters or any texts that you prefer but strings should be inside “quotes. Either you can use Single ‘ ‘ or double quotes “ ”.

Example of PHP String

<!DOCTYPE html>

<html>

<body>

<?php

$hello = “Anya ngy Lubong?”;

$hi = ‘Anya ngy?’;

echo $hello ‘<br>’;

echo $hi;

?>

</body>

</html>

**PHP Boolean**

Same as java programming, Boolean only contains two states which are True or False

Example of Boolean

$t = true;

$f = false;

**PHP Array**

It stores one or more values in one variable.

Example of array

<!DOCTYPE html>

<html>

<body>

<?php

$members = array(“Trio”, “Young”, “Matabilis”, “Sea”);

var\_dump($members);

?>

</body>

</html>

Note: the var\_dump function is used to display the data type and its value.

**PHP Object**

Stores data and information for processing data.

To create an object in PHP, it is a must to declare the object explicitly.

First thing to do is to create a class.

Example of PHP Object

<!DOCTYPE html>

<html>

<body>

<?php

class Slippers {

function Slippers(){

$this -> brand = “Sandugo”;

}

}

//create a new object

$feet = new Slippers();

//show object properties

echo $feet->brand;

?>

</body>

</html>

PHP NULL value

Null means there are no values.

Example of null value

<!DOCTYPE html>

<html>

<body>

<?php

$hello = “Anya ngy Lubong?”;

$hello = null;

var\_dump($hello);

?>

</body>

</html>

PHP Resource

This data type is special in PHP because this stores references outside the php script.

PHP Array

PHP array is similar to what we know of to java programming. An array stores values with one or more values. Example if you want to store many objects like numbers in your program you can store them using an array instead of defining each numbers. Let us say you want to store 50 numbers in your program, so instead of defining the with variables and values you can store them on an array that has a length of 50.

There are 3 different types of arrays that we can use, which are Numeric Array, Associative Array, and Multidimensional Array. Numeric Array is an array that has an Numeric index and the values are displayed or can be accessed in a linear fashion. Associative Array is the opposite of that of the Numeric array, this array uses Strings as indexes instead of numeric likewise with its linear fashion it is stricter than that of numeric array. Lastly multidimensional array is from the word itself contains more than one array and its values can also be use with multiple indexes.

Note: The order of precedence in an array is displayed of their indexes. The first index is 0. For example, if you have 5 values stored in an array their corresponding indexes are as follows: 0,1,2,3,4.

**Example of Numeric Array**

<html>

<body>

<?php

$evenNumbers = array( 2, 4, 6, 8, 10);

foreach($evenNumbers as $value ) {

echo "Value is $value <br />";

}

?>

</body>

</html>

**The output of the code should be:**

Value is 2   
Value is 4   
Value is 6   
Value is 8   
Value is 10

*You can also create an Numeric array in a different array.*

<html>

<body>

<?php

$evenNumbers[0] = "two";

$evenNumbers[1] = "four";

$evenNumbers[2] = "six";

$evenNumbers[3] = "eight";

$evenNumbers[4] = "ten";

foreach( $evenNumbers as $value ) {

echo "Value is $value <br />";

}

?>

</body>

</html>

**The output of this code should be:**

Value is two   
Value is four   
Value is six   
Value is eight   
Value is ten

**Example of Associative Array:**

<html>

<body>

<?php

$grades = array("Erika" => 90, "Sonny" => 85, "Jessie" => 80, "Kenan" => 75);

echo "The grade of Erika in webtek is ". $grades['Erika']. "<br />";

echo "The grade of Sonny in webtek is ". $grades['Sonny']. "<br />";

echo "The grade of Jessie in webtek is ". $grades['Jessie']. "<br />";

echo "The grade of Kenan in webtek is ". $grades['Kenan']. "<br />";

?>

</body>

</html>

**The output should of the code should look like this:**

The grade of Erika in webtek is90  
The grade of Sonny in webtek is 85  
The grade of Jessie in webtek is 80  
The grade of Kenan in webtek is 75

*You can also create the Associative array in a different way:*

<html>

<body>

<?php

$grades = array("Erika" => 90, "Sonny" => 85, "Jessie" => 80, "Kenan" => 75);

$grades['Erika'] = "A+";

$grades['Sonny'] = "B+";

$grades['Jessie'] = "C+";

$grades['Kenan'] = "D+";

echo "The grade of Erika in webtek is ". $grades['Erika'] . "<br />";

echo "The grade of Sonny in webtek is ". $grades['Sonny']. "<br />";

echo "The grade of Jessie in webtek is ". $grades['Jessie']. "<br />";

echo "The grade of Kenan in webtek is ". $grades['Kenan']. "<br />";

?>

</body>

</html>

**The output should look like this:**

The grade of Erika in webtek is A+  
The grade of Sonny in webtek is B+  
The grade of Jessie in webtek is C+  
The grade of Kenan in webtek is D+

**Example of Multidimensional Array**

<html>

<body>

<?php

$grade = array(

"Erika" => array (

"Webtek" => 90,

"ProgApps" => 87,

"Database" => 85,

"SMA" => 80

),

"Sonny" => array (

"Webtek" => 93,

"ProgApps" => 87,

"Database" => 88,

"SMA" => 81

),

"Jessie" => array (

"Webtek" => 93,

"ProgApps" => 89,

"Database" => 78,

"SMA" => 75

),

"Kenan" => array (

"Webtek" => 88,

"ProgApps" => 78,

"Database" => 76,

"SMA" => 95

)

);

echo "Grade for Erika in Webtek : " ;

echo $grade['Erika']['Webtek'] . "<br />";

echo "Grade for Sonny in ProgApps : ";

echo $grade['Sonny']['ProgApps'] . "<br />";

echo "Grade for Jessie in Database : " ;

echo $grade['Jessie']['Database'] . "<br />";

echo "Grade for Kenan in SMA : " ;

echo $grade['Kenan']['SMA'] . "<br />";

?>

</body>

</html>

**The output of the code should look like this:**

Grade for Erika in Webtek : 90  
Grade for Sonny in ProgApps : 87  
Grade for Jessie in Database : 78  
Grade for Kenan in SMA : 95

1. **Control Structures**

PHP Conditional Statements

Sometimes we want to write our code with conditions wherein it will only execute if the condition have been met.

Here are some conditional statements:

If statement – if condition is true it will execute the code

If..else statement - if condition is true it will execute the code otherwise it will execute the code that lies on false.

If...elseif…else statement – if conditions is true it will execute the codes for multiple condtions

Switch statement – selects one of code to execute

**If statement example**

<!DOCTYPE html>

<html>

<body>

<?php

$x = 1;

$y = 2;

if ($x < $y) {

echo "x is less than y";

}

?>

</body>

</html>

**If…Else Statement**

<!DOCTYPE html>

<html>

<body>

<?php

$x = 1;

$y = 2;

if ($x < $y) {

echo "x is less than y";

}else{

echo "y is greater than x";

}

?>

</body>

</html>

**If…ElseIf…Else Statement**

<!DOCTYPE html>

<html>

<body>

<?php

$x = 1;

$y = 2;

if ($x < $y) {

echo "x is less than y";

}elseif($x > $y){

echo "y is greater than x";

}else{

echo "x is equal to y";

}

?>

</body>

</html>

**Switch statement**

<!DOCTYPE html>

<html>

<body>

<?php

$favsubject = "webtek";

switch ($grade) {

case "webtek":

echo "Your favorite subject is webtek!";

break;

case "ProgApps":

echo "Your favorite subject is ProgApps!";

break;

case "Database":

echo "Your favorite subject is Database!";

break;

default:

echo "Your favorite subject is neither webtek, ProgApps, nor Database!";

}

?>

</body>

</html>

**LOOPS**

Is it possible to create a code that will run the same code again and again? Instead of populating your source code with same code, here loops come in handy.

Different loop statements:

while – like the if statement that executes the code if it is true

do…while – it will first execute the code once and when it is confirmed it is true it will now run the code repeatedly

for – it will execute the codes depending on the number of times that was specified

for…each – only works with arrays

**while loop example**

<!DOCTYPE html>

<html>

<body>

<?php

$number = 1;

while($number <= 3) {

echo "The number is: $number <br>";

$number++;

}

?>

</body>

</html>

**Do while loop example**

<!DOCTYPE html>

<html>

<body>

<?php

$number = 1;

do{

echo "The number is: $number <br>";

$number++;

} while($number <= 3);

?>

</body>

</html>

**For loop example**

<!DOCTYPE html>

<html>

<body>

<?php

for ($i = 0; $i <= 5; $i++) {

echo "Number: $i ";

}

?>

</body>

</html>

**For..each loop example**

<!DOCTYPE html>

<html>

<body>

<?php

$subjects = array("webtek", "progapps", "database", "networks");

foreach ($subjects as $val) {

echo "$val <br>";

}

?>

</body>

</html>

1. **User Defined Functions**

Aside from built-in functions in php, we can create our own function. The function in php is similar to other languages like that of the javascript functions. This functions are codes that will process codes when called and it will return a value. You can always have the option to use either built-in functions or create your own functions.

**Create a function**

<!DOCTYPE html>

<html>

<body>

<?php

function greetMe() {

echo "Good Morning!";

}

greetMe();

?>

</body>

</html>

**Function Arguments**

You can also use arguments in a function. Arguments are the same of that variables.

**Example**

<!DOCTYPE html>

<html>

<body>

<?php

function sureName($sname) {

echo "$sname Co<br>";

}

sureName("John");

sureName("Yuan");

sureName("Estrel");

sureName("Levi");

sureName("Pete");

?>

</body>

</html>

1. **Connecting to Database**

In PHP, you can connect your php script to a database. The most common and popular database that was being used is the MySQL database.

Firstly, you must configure your MySqli if it is properly installed to your computer if not download and install MySQLi in  <http://php.net/manual/en/mysqli.installation.php>

You can also use PDO which can be found and installed in this site <http://php.net/manual/en/pdo.installation.php>

**Example MySQL**

<?php

$db\_host = 'localhost';

$db\_name = login;

$db\_user = 'root';

$db\_pass = '';

$mysqli = new mysqli($db\_host, $db\_user, $db\_pass, $db\_name);

//error handler

if($mysqli->connect\_error){

printf("Connection failed" %s\n, $mysql->connect\_error);

exit();

}

**Example PDO**

<?php

$servername = "localhost";

$user = "root";

$pass = "";

try {

$con = new PDO("mysql:host=$servername;dbname=myDB", $user, $pass);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

echo "Successfully connected";

}

catch(PDOException $er)

{

echo "Connection failed: " . $er->getMessage();

}

?>

1. **PHP and HTTP request**

In PHP, to send information details to the web server there are 2 ways for the user to send information details to the server this method is the Get and the POST method.

What is the Get method?

The get method is an http method request that is an array of variables and this variable are passed on to the script through url parameters.

Save this code as our base code in our examples.

//This code is not yet fixed

<html>  
<body>  
 Welcome to WebTek Ninja!  
 Welcome <?php echo $\_POST["name"]; ?><br>  
 Your ID number is: <?php echo $\_POST["idnumber"]; ?>  
  
</body>  
</html>

**Example of get superglobal**

//This code is not yet fixed

<!DOCTYPE HTML>

<html>

<body>

<form action="welcome\_get.php" method="get">

Name: <input type="text" name="name"><br>

ID number: <input type="text" name="idnumber"><br>

<input type="submit">

</form>

</body>

</html>

What is the POST method?

The post method is also an http method that and is also an array of variables.

**Example of POST method**

//This code is not yet fixed

<!DOCTYPE HTML>

<html>

<body>

<form action="welcome.php" method="post">

Name: <input type="text" name="name"><br>

ID Number: <input type="text" name="idnumber"><br>

<input type="submit">

</form>

</body>

</html>

REFERENCE:

<https://www.tutorialspoint.com/php/php_arrays.htm>