PHP

PHP allows a you to embed some programs into the HTML code. These programs can give functionality of web pages in the browser window. PHP has similarity to JavaScript but in loading a web page they have difference. JavaScript is a client-side language, and client-side languages downloaded the web page from the server and it read and executed. While PHP is one of some server-side languages, before the web page appears to the browser, these languages will run first by the web server.

Example of a PHP code:

**<img src=”PHP/Codes/1.png”>**

<?php marks as the start of a PHP script and ?> marks its end.

The program will be interpreted by the web server, the web server will read the program between the two delimiters (<? php and ?>) and converts to HTML code before it appears to the requesting browser.

The program appears to be: Sunday, April 1st 2012.

PHP Interpreter

Local PHP Interpreter- It is a PHP engine that is already installed on computer

Remote PHP Interpreters- It can be installed on a remote host or in a virtual environment set up in a Vagrant instance.

Basic Syntax and Statements

PHP script is consisting of a commands and statements. Each PHP statements are always ends with semicolon.

Example: **<img src=”PHP/Codes/2.png”>**

An echo statement, is used to insert the text in the HTML code at the exact position where the PHP script was.

**<img src=”PHP/Codes/3.png”>**

Quotes are used to mark the beginning and the end of strings while parentheses are used in two purposes, first purpose is to indicate that the date is a function and second purpose is to mark the beginning and end of arguments in order to tell what the function will do.

PHP Variables, Operators and Comments

All variable names in PHP begin with a dollar sign, a single variable contain any data type, number or string. If the variable has an existing value -a number, then the variable was assigned a new value which is a string, it now contains a string value.

The operators of PHP may be used to assign values to variables, other operators can be used to perform mathematical operations on values, there are operators for comparing values and many more. For assigning values to variables, *assignment operator* is being use which is the equal sign ‘=’. There are also’+=’(Addition), ‘ -= ’(Subtraction), ‘\*=’(Multiplication), ‘/=’(Division), ‘%=’(Modulus) used with numeric values.

Example: **<img src=”PHP/Codes/4.png”>**

To perform mathematical operations, *arithmetic operators* are being used like ‘+’ addition operator for adding, ‘ – ‘ subtraction operator for subtracting, ‘ \* ’multiplication operator for multiplying and ‘ / ’ division operator for dividing.

There is also string operators for string values which are *concatenation* ( . ) and *concatenation* *assignment* (.=).Concatenation sticks the string value together and concatenation assignment assigning one variable sticks to another variable.

Example:

**<img src=”PHP/Codes/5.png”>**

Concatenation can be also applied in echo statement:

**<img src=”PHP/Codes/6.png”>**

Comments are used to indicate what is the code for. It is an explanatory text for every code. Comments begin with ‘//’ -for single comments and for block comments, it begins with ‘/\*’ and end with ‘\*/’. Comments between two delimiters are ignored by PHP interpreters.

There are Logical operators, Comparison operators, Array operators and

Increment and Decrement operators.

**<img src=”PHP/Codes/7.png”>**

**<img src=”PHP/Codes/8.png”>**

**<img src=”PHP/Codes/9.png”>**

**<img src=”PHP/Codes/10.png”>**

PHP Arrays

Array isa variable that contains multiple values. To create array in PHP is to use array command.

Example: **<img src=”PHP/Codes/11.png”>**

Knowing the index of each value, you can retrieve the value stored in an array. Each element has its index. The string value ‘one’ is first element and its index is zero.

**<img src=”PHP/Codes/12.png”>**

An existing value of an array can be change by indicating the index.

**<img src=”PHP/Codes/13.png”>**

And also, can add a new element by adding an index;

**<img src=”PHP/Codes/14.png”>**

Associative arrays are arrays that uses named keys which is being assigned.

EXAMPLE: **<img src=”PHP/Codes/15.png”>**

Constants

Constant is a name identifier for simple value, constant is case-sensitive and it must be in uppercase. Constants does not begin with dollar sign. Once it is already defined it cannot be redefine unlike the variable. In indicating constants, it can be done using define() function or using const keyword.

In defining constants using define() function, it will be defined to an arbitrary expression. While using const keyword in defining constants, it will define as a scalar expression.

Examples: Defining Constants using define() function

**<img src=”PHP/Codes/16.png”>**

Defining Constants using const keyword

**<img src=”PHP/Codes/17.png”>**

Functions

User -defined functions

The declaration starts with ‘function’. In naming a function, it begins with letter or underscore but not number. Creating a function, an opening curly brace will begin the function code and closing curly brace to end the function.

Example: **<img src=”PHP/Codes/18.png”>**

Function Arguments

The information in function can be passed by arguments. Arguments are just like variables and it also starts with dollar sign. Arguments are inside of the parentheses in the function, which is after the function name. The arguments can be multiple, the arguments must be comma separated.

Example:

**<img src=”PHP/Codes/19.png”>**

Returning Values

For the function to return a value, use return statement:

**<img src=”PHP/Codes/20.png”>**

Variable functions

Variable functions are which the variable has parentheses attach to the variable name. Variable functions won’t work in echo, print, unset(), isse(), empty().

**<img src=”PHP/Codes/21.png”>**

Class and Objects

In creating class, class keyword must be included followed by a class name and pair of curly braces. Inside the pair of curly braces are the definitions of properties, methods and constants which belongs to the class. A class may contain its own properties -variables, methods -functions, and constants.

Example:

**<img src=”PHP/Codes/22.png”>**

Php is one of the languages that is Object Oriented. Which means objects can be created that contains variables and functions. In creating an object in the class, the new keyword must be use.

**<img src=”PHP/Codes/23.png”>**

Control Structures

If…else…elseif

-if statement will execute if the condition is true

-else statement will execute if the condition is false.

-elseif statement is another option and will execute if the condition is true.

Syntax:

**<img src=”PHP/Codes/24.png”>**

Example: **<img src=”PHP/Codes/25.png”>**

switch

-switch statement is similar to if statement, but in switch statement it has many conditions with different actions.

Syntax:

**<pre>**

switch(n){

case label1: //each case ends with colon

code to execute, if n matches label1;

break; //end the execution

case label2:

code to execute, if n matches label2;

break;

case label3:

code to execute, if n matches label3;

break;

default:

code to execute if does not pair one of the labels;

}

**</pre>**

Example: **<img src=”PHP/Codes/26.png”>**

for loops

-for loop is used if you want to run the script many times.

Syntax:

**<pre>**

for (initialize the loop counter; test counter; increment counter) {

code to be executed;

}

**</pre>**

Example: **<img src=”PHP/Codes/27.png”>**

while loops

-as long as the specified condition is true, the block of codes will execute.

Syntax:

**<pre>**

while (condition is true){

code to execute;

}

**</pre>**

Example: <img src=”PHP/Codes/28.png”>

Include and Require statements

-Include and require statements are used for inserting or including files -PHP files, HTML files, txt files.

Syntax:

include ‘filename’; OR require ‘filename’;

Example: **<img src=”PHP/Codes/28.png”>**

The difference between the two statements is that, using the include statement, if the file cannot be found the script will continue to execute, while using the require statement, if the file cannot be found it will not execute right after the require statement.

Global variables-Superglobals

PHP superglobal variables are:

$\_SERVER

-it is an array that contains information like headers, paths, and scripts locations.

Syntax: $\_SERVER[elements]

Example: echo $\_SERVER[‘SERVER\_NAME’]; //this returns the name of the host server

$\_REQUEST

-it is an associative array contains the contents of $\_GET, $\_POST and $\_COOKIE.

Example: $myname=$\_REQUEST [‘fname’];

$\_POST

-It is used to collect the data from form and also used to pass variables.

$\_GET

-Can be used to collect the data from form and also collect data sent in URL.

$\_FILES

-It contains all the uploaded file information.

$\_COOKIE

-An associative array of variables passed to the script through HTTP cookies.

$\_SESSION

-An associative array that contains session variables included in the current script.

PHP with MySQL

Connect to the MySQL using mysqli()

Example: $conn = mysqli($servername,$username, $password);

INSERT data

Inserting data have some syntax rule to be followed:

* The SQL query must be quoted in PHP
* String values that are inside the SQL query must be quoted
* No quotations for numeric values
* NULL word must not be quoted

To insert data, INSERT INTO table\_name(column1, column2) VALUES (value\_one,value\_two,value\_three);

SELECT data

-to select data, SELECT column\_name1,column\_name\_2 FROM table\_name

Example: $sql=”SELECT id, fname,lname FROM Customers”;

DELETE data

-to delete record from table, DELETE FROM table\_name WHERE specific\_column=specific\_value

Example: $sql=”DELETE FROM Customers WHERE id=4”;

UPDATE data

-to update an existing data from the records, UPDATE table\_name SET column1=value1, column2=value2,... WHERE specific\_column=specific\_value

Example: $sql = "UPDATE Customers SET fname='Joel' WHERE id=2";

References:

PHP 5 Tutorial. (n.d.). Retrieved May 2, 2018, from <https://www.w3schools.com/php>

PHP manual. (n.d.). Retrieved May 2, 2018, from <http://php.net/manual>

Yank, K. (2012). PHP & MySQL: Novice to ninja (5th ed.). Collingwood, VIC, Australia: SitePoint Pty. Ltd.