

**Name: NDIKUBWIMANA Maurice**

**Reg No: 21RP09654**

**Date 26/01/2023**

**Class: L6y2 ITA**

**Assignment**

**ITLBP601: Develop Backend using PHP**

**Q1. Explain php programing beyond definition**

- **PHP** is an acronym for **PHP Hypertext Preprocessor** which is a server side programming language that can generate dynamic page content, create, open, read, write, delete, and close files on the server and your database as, well can be used to control user-access and it can encrypt data.
- **PHP** can hold text, HTML, CSS, JavaScript, and PHP codes which makes it more powerful enough to be at the core of the biggest blogging system on the web and to run large social networks too.

**Q2. Why do we need to use php programming?**

- PHP runs on various platforms such as Windows, UNIX, Linux, Mac OS, and others.
- It's compatible with almost all servers like Apache, IIS, and more.
- It supports a variety of databases
- Additionally, PHP is free from its official web

- PHP runs efficiently on the server side

***Q3. What is the latest php version we have today and list the updated features for the latest 3 release?***

➤ PHP version 8.2

PHP version 8.2 features

- It allows Null, True and False as standalone types.
- Removal of libmysql support from mysqli
- PHP 8.2 allows constants in traits.

**Q4. What is different between new release vs stable release of a software product?**

A stable release is a version that has been tested as thoroughly as possible and is as reliable as we can make it. It is usually for developers that improve the current product that will be the next release.

While,

**A new release** is the distribution of the final version or the newest version of a software application.

***Q5. What are the main features of php programming?***

- PHP is a platform-independent which means that it runs on almost all platforms
- PHP is a server side script that executes on web browser

***Q6. With a help of examples explain why php is case sensitive?***

This means that the variables, keywords etc. can't run in both cases (upper and lower cases)

## Example

In this program;

```
<!DOCTYPE html>

<html>

<body>

<?php

$x = 100;

$y = 50;

var_dump($x > $y); // returns true because $x is greater
than $y

?>

</body>

</html>
```

The output will be;

**bool(true)**

but when we change the case of variables - **x and y are put in upper case**- it will not give the output

```
<!DOCTYPE html>

<html>

<body>

<?php
```

```
$X = 100;
```

```
$Y = 50;
```

```
var_dump($x > $y); // returns true because $x is greater than  
$y
```

```
?>
```

```
</body>
```

```
</html>
```

The output will be;

```
bool(false)
```

***Q7. What and why do we use comments while writing php codes, with a help of example explain different types of php comments?***

- **Single-line comment:** with single line comment, you are allowed to only have a short line, up to one line. This is symbolized by double slashes before comment line.

### **Example**

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
// This is a single-line comment
```

```
# This is also a single-line comment
```

```
?>
```

```
</body>
```

```
</html>
```

- **Multiple-line comments:** These are comments that span over multiple lines

### Example

```
<!DOCTYPE html>
<html>
<body>

<?php
/*
This is a multiple-lines comment block
that spans over multiple
lines
*/
?>

</body>
</html>
```

**Q8. Differentiate with real example the following php output functions:**

#### a. Echo() vs print():

- The main difference is that echo has no return value while print has a return value of 1.
- Additionally, echo can take multiple parameters while print can take one argument.
- Lastly, echo is marginally faster than print.

- **Example on echo command**

```
<?php
echo "<h2>Lesson 1!</h2>";
echo "Hello world!<br>";
echo "I'm Robert MUGABO!<br>";
echo "This ", "string ", "was ", "made ", "with multiple
parameters.";
?>
```

- ***Example on print command***

```
<?php
print "<h2>PHP print command!</h2>";
print "Hello world!<br>";
print "I'm about to print something!";
?>
```

## **b. Print() vs printf()**

- The printf( ) function builds a formatted string by inserting values into a template.
- Print() can be used with or without parentheses and always returns an integer value, which is 1. Lastly, Using print, we cannot pass multiple arguments.

### **Example**

```
<?php
$lang = "PHP";
$ret = print $lang." is a web development language.";
print "</br>";
print "Value return by print statement: ".$ret;
?>
```

## **c. Printf() vs print\_r()**

- The `printf( )` function builds a formatted string by inserting values into a template.
- The `print_r( )` function is useful for debugging—it prints the contents of arrays, objects, and other things, in a more-or-less human-readable form.

#### ***d. Print\_r vs var\_dump()***

- **The `print_r( )`** construct intelligently displays what is passed to it, rather than casting everything to a string, as `echo` and `print( )` do. Strings and numbers are simply printed.
- The **`var_dump()`** function is used to dump information about a variable that displays structured information such as the type and value of the given variable. `var_dump( )` is preferable for debugging. The `var_dump( )` function displays any PHP value in a human-readable format:

- ***Example on using var\_dump()***

```
<?php
```

```
var_dump(var_dump(45, 62.1, TRUE,  
                "sravan", array(1, 2, 3, 4,5,6))  
);
```

```
?>
```

- ***Code to display all data type variables using print\_r() function.***

```
<?php
```

```
// String variable  
$a = "Welcome to GeeksforGeeks";
```

```
// Integer variable  
$b = 450;
```

```
// Array variable
$arr = array('0' => "Computer",
            '1' => "science",
            '2' => "portal");

// Printing the variables
print_r($a);
echo"\n<br>";
print_r($b);
echo"\n<br>";
print_r($arr);
?>
```

***Q9. List and Describe different datatype we have in php by categorizing them in scalar, compound and special datatypes.***

**Scalar:** These are types that hold only single value.

- **Boolean:** These are php data types that hold only two values: **TRUE (1)** or **FALSE (0)**. It is often used with conditional statements.

**Integer:**

- An integer can be either positive or negative.
- An integer must not contain decimal point.
- Integer can be decimal (base 10), octal (base 8), or hexadecimal (base 16).
- The range of an integer must be lie between 2,147,483,648 and 2,147,483,647 i.e.,  $-2^{31}$  to  $2^{31}$ .
- **Float:** A floating-point number is a number with a decimal point. It can hold numbers with a fractional or decimal point, including a negative or positive sign.



- **String:** A string is a non-numeric data type. It holds letters or any alphabets, numbers, and even special characters.

**Compound:** These can hold multiple values.

- **Array:** An array is a compound data type. It can store multiple values of same data type in a single variable.
- **Object:** Objects are the instances of user-defined classes that can store both values and functions. They must be explicitly declared.

### **Special datatypes**

- **Resource:** These are used to store some function calls or references to external PHP resources. **For example** - a database call. It is an external resource.
- **NULL:** Null is a special data type that has only one value: **NULL**. There is a convention of writing it in capital letters as it is case sensitive.

***Q10. What is php variable, list the variable naming rules you have to obey while defining a variable in php?***

**PHP** variables are **characters that stores value or information such as text or integers in your code**. Variables in PHP are usually represented by a dollar sign (\$) followed by the name of the variable.

### **Variable naming rules you have to obey while defining a variable in php**

- A variable starts with the \$ sign, followed by the name of the variable

A variable name must start with a letter or the underscore character

- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_ )
- Variable names are case-sensitive (**\$age** and **\$AGE** are two different variables)

***Q11. List and explain at least 10 super global variables?***

- **\$GLOBALS:** is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).
- **\$\_SERVER:** is a PHP super global variable which holds information about headers, paths, and script locations.
- **\$\_REQUEST:** is a PHP super global variable which is used to collect data after submitting an HTML form.
- **\$\_POST:** is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". \$\_POST is also widely used to pass variables.
- **\$\_GET:** is a PHP super global variable which is used to collect form data after submitting an HTML form with method="get".
- **\$\_FILES:** Is an associative array containing items uploaded via HTTP POST method.
- **\$\_ENV:** An associative array of variables passed to the current script via the **environment** method.
- **\$\_COOKIE:** An associative array of variables passed to the current script via HTTP Cookies.
- **\$\_SESSION:** Is an associative array containing session variables available to the current script.

## ***References:***

1.  
<https://www.php.net/manual/en/reserved.variables>
2.  
[https://www.w3schools.com/php/php\\_superglobals](https://www.w3schools.com/php/php_superglobals)
3. <https://www.javatpoint.com/php-echo-and-print-statements>
4. <https://www.techtarget.com> > definition > [release](#)
5.  
<https://www.oreilly.com/library/view/programming-php>