

MODULE NAME: DEVELOP BACK END USING PHP

DATE: 26/01/2023

REGNO: 21RP01077

CLASS: ITB

## **ASSISSNMENT2**

1. Explain PHP programing beyond definition?

PHP is a widely-used, open source scripting language. PHP scripts are executed on the server. PHP is free to download and use.

Is an open source scripting language and is mainly used by developers to create web pages and websites.

2. Why do we need to use php programming?

### **Because:**

- PHP allows web developers to create dynamic content and interact with databases
- PHP is known for its simplicity, speed, and flexibility
- PHP makes website pages load faster as compared to many other web development technologies.
- PHP is often the first programming language many web developers are exposed to.
- it is easy to learn PHP and its prominence on the web

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

### **The latest version of PHP IS:**

#### **PHP8.2**

- 🚦 PHP 8.2 is a **major update of the PHP language**. It contains many new features, including read-only classes, null, false, and true as stand-alone types, deprecated properties, performance improvement and more.

PHP 8.2 brings type-system improvements for expressive and fine-grained type-safety, read only classes, sensitive parameter redaction support, new random extension, and many new features along with several changes to streamline and modernize PHP language.

PHP 8.2 is released on **December 8, 2022**.

## PHP 8.1

- Released on November 25, 2021, PHP 8.1 is finally here, packed with several exciting features. In this article, we'll cover in detail what's new in PHP 8.1. From its new features and performance improvements to significant changes and deprecations, we'll go through them all in-depth.

## PHP 7.4

- PHP 7.4 is the latest stable version of PHP. It was released on November 28, 2019 and it's the last version before PHP 8. It brings lots of new features, syntax additions and fixes. In this post you'll find a list with everything that's new and changed to help you prepare for the upgrade. Let's start though with a few highlights, included in PHP 7.4.

### 4. What is different between new release vs stable release of a software product?

**A new release:** is the distribution of the final version or the newest version of a software application. A software release may be public or private and generally signifies the unveiling of a new or upgraded version of the application.

**A stable release:** is a version that has been tested as thoroughly as possible and is as reliable as we can make it. It does not have all the new features of a beta release and it does not have the latest fixes for problems.

### 5. What are the main features of php programming?

#### 1. Simple

It is very simple and easy to use, compare to other scripting language it is very simple and easy, this is widely used all over the world.

#### 2. Interpreted

It is an interpreted language, i.e. there is no need for compilation.

#### 3. Faster

It is faster than other scripting language e.g. asp and jsp.

#### 4. Open Source

Open source means you no need to pay for use php, you can free download and use.

## 5. Platform Independent

PHP code will be run on every platform, Linux, Unix, Mac OS X, Windows.

## 6. Case Sensitive

PHP is case sensitive scripting language at time of variable declaration. In PHP, all keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are NOT case-sensitive.

## 7. Error Reporting

PHP have some predefined error reporting constants to generate a warning or error notice.

## 8. Real-Time Access Monitoring

PHP provides access logging by creating the summary of recent accesses for the user.

## 9. Loosely Typed Language

PHP supports variable usage without declaring its data type. It will be taken at the time of the execution based on the type of data it has on its value

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

- ✚ PHP is a unique programming language in terms of case sensitivity
- ✚ PHP classes are a mix between variables and functions, so they are partially case-sensitive.

**Example:**

```
// ☒ you can create two variables like this:
$num = 30;
$NUM = 10;

echo $num; // 30
echo "\n".$NUM; // 10

// ☒ but you can't have two functions like this:
function greetings(){
    echo "Hello World!";
}

// ☒error: Cannot redeclare GREETINGS()
function GREETINGS(){
```

```
echo "Hello World!";  
}
```

As you can see in the example above, the variables \$num and \$NUM can have different values.

But when you declare two functions with the same name, PHP produces an error: cannot redeclare the function. This happens even though the function names use different letter cases.

7. What and why do we use comments while writing php codes, With a help of example explain different types of php comments.

- ✚ PHP comments are usually meant to help programmers understand and interpret the PHP code
- ✚ A comment in PHP code is a line that is not executed as a part of the program. Its only purpose is to be read by someone who is looking at the code. Comments can be used to: Let others understand your code.



**Example:**

PHP supports both **one-line and multi-line comments**. A one-line comment starts with the # or // . A multi-line comment starts with /\* and end with \*/ .

**Multi-line comments**

```
<?php  
/* example of multi-line comment */  
?>
```

**One-line comment**

```
<?php  
// example of one line comment  
?>
```

8. Differentiate with real example the following php output functions:

a. Echo() vs print() :

✓ **Echo()**

```
<?php  
echo "<h2>php!</h2>";  
echo "hello!<br>";  
echo "am happy!<br>";
```

```
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
?>
```

✓ Print()

```
<?php
print "<h2>php!</h2>";
print "Hello world!<br>";
print "this is good!";
?>
```

b. Print() vs printf()

✓ Print()

```
<?php
print "<h2>php!</h2>";
print "Hello world!<br>";
print "this is good!";
?>
```

✓ Printf()

```
<?php
$number = 9;
$str = "Esther";
printf("There are %u million bicycles in
%s.", $number, $str);
?>
```

c. Printf() vs print\_r()

✓ Printf()

```
<?php
$number = 9;
$str = "Esther";
printf("There are %u million bicycles in
%s.", $number, $str);
?>
```

Print\_r()

```
<?php
$a = array("red", "green", "blue");
print_r($a);
```

```

echo "<br>";

$b = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
print_r($b);
?>

```

d. Print\_r vs var\_dump()

✓ **Print\_r()**

```

<?php
$a = array("red", "green", "blue");
print_r($a);

echo "<br>";

$b = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
print_r($b);
?>

```

✓ dump()

```

<?php
$a = 32;
echo var_dump($a) . "<br>";

```

```

$b = "Hello world!";
echo var_dump($b) . "<br>";

```

```

// Dump two variables

```

```

?>

```

9. List and Describe different data type we have in php by categorizing them in scalar, compound and special datatypes.

**Data type we have in php**

- Integer
- String
- Float
- Array
- Object
- Null
- Resource
- Boolean

### Scalar Types:

- **Boolean:** A Boolean represents two possible states: TRUE or FALSE. \$x = true; \$y = false; Booleans are often used in conditional testing.
- **Integer:** An integer data type is a non-decimal number between -2147483648 and 2147483647 in 32 bit systems, and between -9223372036854775808 and 9223372036854775807 in 64 bit systems. A value greater (or lower) than this, will be stored as float, because it exceeds the limit of an integer.
- **Float:** A float is a number with a decimal point or a number in exponential form. 2.0, 256.4, 10.358, 7.64E+5, 5.56E-5 are all floats.
- **String:** In PHP, a string data type is a non-numeric sequence of characters. Any character in the ASCII set can be a part of a string.

### Compound Types:

- **Array:** In PHP, there are three types of arrays: Indexed arrays - Arrays with a numeric index. Associative arrays - Arrays with named keys. Multidimensional arrays - Arrays containing one or more arrays.
- **Object:** An Object data type is a method for encoding key-value pairs. A single field value may contain one or more sets of key-value pairs. An Object data type is identified as a set of nested objects in the following format: ? { "key" : "value" }.

### Special Types:

- **Resource:** A **resource** is a special variable, holding a reference to an external **resource**. **Resources** are created and used by special functions.
- **Null:** Null is a special data type which can have only one value: NULL. A variable of data type NULL is a variable that has no value assigned to it.

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

**A Php variable:** can have a short name (like x and y) or a more descriptive name (age, carname, total\_volume). Rules for PHP variables: 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.

**The variables naming rules:**

- ✓ Variable names are case-sensitive.
- ✓ Subsequent characters may be letters, digits, dollar signs, or underscore characters.
- ✓ If the name you choose consists of only one word, spell that word in all lowercase letters.
- ✓ After the first initial letter, variable names can also contain letters and numbers.
- ✓ Uppercase characters are distinct from lowercase characters.
- ✓ No commas, space allowed in variable name.
- ✓ No other special symbols other than underscore is allowed.

11. List and explain at least 10 super global variables?

The PHP super global variables are:

- \$GLOBALS
- \$\_SERVER
- \$\_REQUEST
- \$\_POST
- \$\_GET
- \$\_FILES
- \$\_ENV
- \$\_COOKIE
- \$\_SESSION

🚦 **\$\_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 a global constant or predefined variable in PHP that can be used to associate array items that are uploaded through the HTTP POST method.
  
- ✚ **\$\_ENV**: is another superglobal associative array in PHP. It stores environment variables available to current script.
  
- ✚ **\$\_COOKIE**: is used to retrieve a cookie value. It typically an associative array that contains a list of all the cookies values sent by the browser in the current request, keyed by cookie name.
- ✚ **\$\_SESSION**: is an associative array that contains all session variables. It is used to set and get session variable values.

#### References:

[https://www.w3schools.com/c/exercise.php?filename=exercise\\_variab](https://www.w3schools.com/c/exercise.php?filename=exercise_variab)

[https://www.tutorialspoint.com/php/php\\_loop\\_types.htm](https://www.tutorialspoint.com/php/php_loop_types.htm)

<https://www.php.net/archive/2023.php#2023-01-05-3>

