1. Explain php programing beyond definition?

PHP (short for Hypertext Preprocessor) is the most widely used open source and general purpose server side scripting language used mainly in web development to create dynamic websites and applications. It was developed in 1994 by Rasmus Leadoff. A survey by W3Tech shows that almost 79% of the websites in their data are developed using PHP. It is not only used to build the web apps of many tech giants like Facebook but is also used to build many CMS (Content Management System) like WordPress, Drupal, Shopify, WooCommerce etc.

2. Why do we need to use php programming?

- It is easy to learn, with a simple syntax and easy-to-follow code documentation
- It is supported and routinely updated by a large and active developer community
- It is flexible and compatible with cloud services, hosting providers, and a variety of web technologies
- It is faster than server-side scripting languages like Python
- It is free and open source
- It has great database connectivity.

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

The latest PHP version we have today is PHP 8.2

PHP 8.2 is planned to release on November 24, 2022, with the most recent stable version being PHP 8.1.5. PHP 8.2 is built to renew and bring ease to development and fix bugs found in **the older PHP versions.**

Updated features for the latest 3 release

PHP 8 Improvements and New Features

- Constructor Property Promotion.
- Validation for Abstract Trait Methods.
- Incompatible Method Signatures.
- Arrays Starting With a Negative Index.
- Union Types 2.0.

- Consistent Type Errors for Internal Functions.

  PHP 8.0 is a major update of the PHP language. It contains many new features and optimizations including named arguments, union types, attributes, constructor property promotion, match expression, null safe operator, JIT, and improvements in the type system, error handling, and consistency.

  PHP 8.1 is a major **update** of the PHP language. It contains many **new features**, including enemas, readonly properties, first-class callable syntax, fibers

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

A release is the distribution of the final version or the newest version of a software 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?

- **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.

- **Interpreted**

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

- **Faster**

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

- **Open Source**

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

- **Platform Independent**

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

- **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.

- **Error Reporting**

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

- **Real-Time Access Monitoring**

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

- **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?

Case sensitive means that it matters if characters are in lower or uppercase

PHP classes are a mix between variables and functions, so they are partially case-sensitive. 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 a fatal error: cannot redeclare the function.

Example: "Computer" and "computer" are two different words because the "C" is uppercase in the first example and lowercase in the second example.

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

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
- Remind yourself of what you did - Most programmers have experienced coming back to their own work a year or two later and having to re-figure out what they did. Comments can remind you of what you were thinking when you wrote the code

Types of PHP comments

Single-line comments

Syntax for single-line comments: can be used for small block of code

Single-line comments start with **two forward slashes (*//*)**.

Example:// This is a comment

Echo ”hello world”;

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

<?php
// This is a single-line comment

# This is also a single-line comment
?>

</body>
</html>
```

Multiple-line comments: can be used to comment out large blocks of code. It begins with /* and ends with */. /*

Example: /* */

Echo “hello world”;

Syntax for multiple-line comments:

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

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

```
*/
?>

</body>
</html>
```

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

a. Echo() vs print():echo has no return value while print has a return value of 1 so it can be used in expressions. Echo can take multiple parameters (although such usage is rare) while print can take one argument. Echo is marginally faster than print.

Example:

```
<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>
```

Example:

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

b. Print() vs printf():print() outputs one or more strings.

Example:

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

printf():outputs a formatted string whereas

Example:

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

c. Printf() vs print_r():

The printf() function outputs a formatted string.

Example:

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

The **print_r()** function is used to print human-readable information about a variable. If the argument is an array, **print_r()** function prints its keys and elements (same for objects).

Example:

```php
<?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():

The **print_r()** function is used to print human-readable information about a variable. If the argument is an array, **print_r()** function prints its keys and elements (same for objects).

Example:

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

echo "<br>";

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

var_dump function usually used for debugging and prints the information (type and value) about a variable/array/object.

Example:

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

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

$c = 32.5;
```

```php
echo var_dump($c) . "<br>";

$d = array("red", "green", "blue");
echo var_dump($d) . "<br>";

$e = array(32, "Hello world!", 32.5, array("red", "green", "blue"));
echo var_dump($e) . "<br>";

// Dump two variables
echo var_dump($a, $b) . "<br>";
?>
```

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

The data type specifies the amount of memory that allocates to the variable associated with it. The data type determines the operations that you can perform on it.

PHP **Data Types: Scalar Types**

In simple words, a variable is called scalar type if it holds singular value only. There are 4 scalar data types in **PHP**.

1. Boolean: Booleans are like a switch which has only two possible values either 1 (true) or 0 (false).
2. Integer: Hold only whole numbers including positive and negative numbers, i.e numbers without fractional part or decimal point. They can be decimal (base 10), octal (base 8) or hexadecimal (base 16). The default base is decimal (base 10).

3. Float: Floating point numbers (also known as “floats”, “doubles”, or “real numbers”) are decimal or fractional numbers
4. String: In a string, letters or any alphabets, even numbers are included. These are written within double quotes during declaration.

PHP Data Types: Compound Types

In contrast to Scalar data types, a variable is called compound if it holds multiples values within. There are 2 compound data types in **PHP**.

1. Array: An array is a variable that can hold more than one value at a time. It is useful to aggregate a series of related items together
2. Object: An object is a data type which stores not only data but also information on how to process that data.

PHP **Data Types: Special Types**

There are 2 special data types in **PHP**.

1. Resource: A resource is a special variable, holding a reference to an external resource.
2. NULL: The special NULL value is used to represent empty variables in PHP. A variable of type NULL is a variable without any data. NULL is the only possible value of type null.

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

PHP variable: Variables are "containers" for storing information.

Variable is memory space used to store data during program execution

Naming rules to obey while defining 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)

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

Super global variables: are predefined *global variables*.

Means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

10 super global variables

- **`$GLOBALS`**: Is the super global variable that stores all user-defined global variables. The global variable names act as keys to their values.
- **`$_SERVER:`** contains data about headers, scripts, and paths. The keys to the values in this array are predefined.
- **`$_REQUEST`**: stores data input in the form of HTTP POST, GET and Cookies. The keys to this array are defined in the HTTP requests.
- **`$_POST`**: stores data input in the form of POST requests. The keys to this array are defined in the HTTP POST request.
- **`$_GET`**: has data input in the form of GET requests. The keys to this array are defined in the HTTP GET request.
- **`$_FILES`**: is a two-dimensional associative array that contains a list of files that were uploaded to the script using the POST method. The keys to this array are the names of the fields uploading the files and the data being accessed. For example, `$_FILES [file Uploaded][name]` accesses the name of the file being uploaded from the *file Uploaded* field.
- **`$_COOKIES`**: keeps data input via HTTP Cookies. The keys to this array are defined when the cookies are set.
- **`$_SESSION:`** holds session variables. Session variables can be accessed on multiple pages. This array's keys are defined by the users when they define session variables.
- **`$_ENV`**: contains information about the environment that PHP is running in. The keys to the values in this array are predefined.

https://www.google.com/search?q=1.+Explain+php+programing+beyond+definition%3F&rlz=1C1KNTJ_enRW1033RW1033&oq=1.+Explain+php+programing+beyond+definition%3F&aqs=chrome..69i57j33i160l3.1078j0j7&sourceid=chrome&ie=UTF-8
https://www.upwork.com/resources/why-use-php

https://www.cloudways.com/blog/php-8-2/#:~:text=PHP%208.2%20is%20planned%20to,stable%20version%20being%20PHP%208.1.
https://www.google.com/search?q=What+is+different+between+new+release+vs+stable+release+of+a+software+product&rlz=1C1KNTJ_enRW1033RW1033&oq=What+is+different+between+new+release+vs+stable+release+of+a+software+product&aqs=chrome.0.69i59j69i60l2.1144j0j4&sourceid=chrome&ie=UTF-8

https://www.google.com/search?q=why+php+is+case+sensitive&rlz=1C1KNTJ_enRW1033RW1033&oq=why+php+is+case+&aqs=chrome.0.0i512j69i57j0i390l4.8337j0j7&sourceid=chrome&ie=UTF-8

https://www.sitesbay.com/php/php-features-of-php

https://www.w3schools.com/php/phptryit.asp?filename=tryphp_comments
https://www.w3schools.com/php/func_var_var_dump.asp#:~:text=Definition%20and%20Usage,of%20the%20variable(s).

http://www.trytoprogram.com/php/php-data-types/

https://www.w3schools.com/php/php_variables.asp

https://www.educative.io/answers/what-are-superglobals-in-php