Q1. What is PHP? What are the rules for naming a PHP variable?

Ans. • PHP is a server scripting language, and a powerful tool for making dynamic and interactive web pages.

• PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

→ Rules for naming a PHP variable:-

1. Valid characters:

- Variables can comprise letters (a-z, A-Z), numbers (0-9), and underscores (_).
- They must commence with a letter or underscore, not a number.

2. Case sensitivity:

- PHP variable names are case-sensitive.
- For instance, $variable Name and $VariableName are considered distinct.

3. Avoid Reserved keywords:

- Refrain from using PHP-reserved keywords such as if, else, for, etc, as variable names, as they serve specific purposes in the language.

4. Descriptive and Meaningful:

- Opt for descriptive and meaningful names that convey the purpose or content of the variable.
- Utilize ~~camel case~~ camelCase or snake-case conventions for multi-word variable names, maintaining consistency within your code.

Q2. How can PHP and HTML interact?

Ans. • PHP and HTML interact in a way that allows PHP to generate dynamic HTML content.

• PHP is a server-side language, which means PHP scripts run on the web server before any output is sent to the user's web browser.

• Here's a general flow of how PHP and HTML interact:

1. A user makes a request for a PHP page from their browser.

2. The request is sent to the server hosting the website.

3. The server sees that the requested file ~~a~~ is a PHP file and processes any PHP code ~~is~~ in the file.

• This PHP code can perform various tasks, such as retrieving data from a database or processing ~~a~~ form ~~to~~ data.

4. The PHP interpreter processes the PHP code and outputs HTML.

→ An example of a PHP script embedded in HTML:

```
<!DOCTYPE html>
<html>
<body>
<h1> my first PHP page </h1>
<? php
   echo "Hello World!";
?>

</body>
</html>
```

Q3. What are the different types of array in PHP?

Ans. An array is a special variable that can hold many values under a single name, and you can access the values by referring to an index number or name.

- In PHP, there are three types of arrays:-

1) Indexed arrays
2) Associative arrays
3) multidimensional arrays

1) PHP indexed arrays

- In indexed arrays each item has an index number.
- By default, the first item has index 0, the second item has item 1, etc.

eg:

```
$cars = array ('Volvo', 'Bmw', 'Toyota');
var_dump ($cars);
```

2) PHP associative arrays

- Associative arrays are arrays that use named keys that you assign to them.
- eg:

```
$car = array ('brand' => 'Ford', 'model' => 'Mustang', 'year' =>
                                                         1964);

var_dump ($car);
```

3) PHP multidimensional arrays

- A multidimensional array is an array containing one or more arrays.
- eg:

```
$cars = array (array ('Volvo', 22, 18),
               array ('Bmw', 15, 13),
               array ('Saab', 5, 2)
        );
```

Q4. Discuss the data types in PHP.

Ans. PHP supports the following data types:

1. PHP String

- A string is a sequence of characters like "Hello World".
- eg:

```
$x = "Hello World!";
var_dump($x);
```

2. PHP integer

- An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.
- eg:

```
$x = 5985;
var_dump($x);
```

3. PHP Float

- A float (floating point number) is a number with a decimal point or a number in exponential form.
- eg:

```
$x = 10.365;
var_dump($x);
```

4. PHP Boolean

- A boolean represents two possible states: TRUE or ~~Fa~~ FALSE.
- eg:

```
$x = true;
var_dump($x);
```

- Booleans are often used in conditional testing.

Q5. Discuss String manipulations in PHP with examples.

Ans • PHP provides a variety of built in functions to manipulate strings efficiently.

• Below are some common string operations in PHP, along with examples.

1. Concatenation (. operator)

• PHP uses the . operator to concatenate strings.

• eg:

```
<?php
 $str1 = "Hello";
 $str2 = "World!";
 $concatenated = $str1. $str2;
 echo $concatenated;
?>
```

2. String length (strlen())

• The strlen() function returns the length of a string.

• eg:

```
 <?php
  $stri = "PHP String manipulation";
  echo strlen($stri);
 ?>
```

3. Word count (str_word_count())

• The str_word_count() function counts the number of words in a string.

• eg:

```
<?php
 $str = "Hello; welcome to PHP!";
 echo str_word_count ($str);
?>
```