

(1) Define `$FILES`

→ `$FILES` is a PHP ~~super~~-global variable. It contains names, sizes and mime types of files uploaded in the current HTTP request.

(2) write any two variable handling function with syntax and example.

→ (1) `print_r()` : Prints the information about a variable in a human-readable way.

syntax: `print_r();`

example: `print_r($fruits);`

(2) `isSet()` : Checks whether a variable is set (declared and not NULL)

syntax: `isSet(variable);`

example: `isSet($num);`

(3) explain foreach loop with syntax and example

→

→ The foreach loop is used to traverse the array elements. It works only on array and object.

→ It will issue an error if you try to use it with the variable of different datatype.

→ The foreach loop works on elements basis rather than index.

→ It provides an easiest way to iterate the elements of an array.

→ In Foreach loop, we don't need to increment the value.

Syntax:

```
foreach ($array as $value) {  
    // code to be executed  
}
```

Example:

```
<?php  
$season = array("Summer", "Winter",  
                "Autumn", "Rainy");
```

```
foreach($season as $element) {  
    echo "$element";  
    echo "<br>";  
}  
}
```

Output:

Summer
Winter
Autumn
Rainy

(3) Give the difference between echo and print statement.

↳ print

→ print statement does not support multiple arguments.

→ print statement is slow as compared to the echo statement.

→ print statement returns an integer value that is set to 1.

→ print statement behaves like a function.

→ print statement is less preferable.

echo

→ echo statements supports multiple arguments.

→ echo statement is faster than a print statement.

→ echo statement does not return any value.

→ echo statement does not behave like a function.

→ echo statement is more preferred.

(5) What is array? Explain types of array in brief.

↳

Array: Array in PHP is a type of data structure that allows us to store multiple elements of similar datatype under a single variable.

→ The arrays are helpful to create a list of elements of similar types, which can be accessed using their index or key.

- An array is created using array() function in PHP.
- There are basically three types of arrays in PHP:
 - 1) Indexed or Numeric Arrays
 - 2) Associative Arrays
 - 3) Multidimensional Arrays

1) Indexed or Numeric Arrays:

- These type of arrays can be used to store any type of elements, but an index is always a number.

Syntax:

```
$array_name = array (list of elements);
```

Lx.

```
$fruits = array ("Apple", "Banana",  
                 "Grapes", "Mango");
```

(2) Associative Arrays:

- These types of arrays are similar to the indexed array but instead of linear storing, every value can be assigned with a user-defined key or string value.

Syntax:

```
$array_name = [  
    Key => value  
]
```

ex.

~~\$melme-one~~ = [

"Zeck" \Rightarrow "Zack"

"Anthony" \Rightarrow "Any"

"Rcmi" \Rightarrow "Rcmii".

3

(3) Multi-dimensional arrays:
→ multi-dimensional arrays are such arrays that store ~~other~~ instead of a single element.

Syntax! ~~\$category-meme = category C~~

Key \Rightarrow anony (

Key \Rightarrow value,

J;

example:

~~\$ favorites & = array C~~

"Dave Punk" \Rightarrow crooky (

"mob" => "56789\538",

"email" \Rightarrow "duy@gmail.com"

三

"John" \Rightarrow crazy c

"mob" → "9876543201"

"emilei1" => "John@Gmail.com"

3

3

(G) Define scripting languages & explain Server-side & client side scripting.

- The scripting language is basically a language where instructions are written for a runtime environment.
- They don't require the compilation step and are either interpreted. It brings new functions to applications and glue complex system together.

1. Client-side scripting:

- Web-browsers execute client-side scripting. It is used when browsers have all code.
- Source code is used to transfer from webserver to user's computer over the internet and run directly on browsers.
- It is also used for validations and functionality for user events.
- It allows for more interactivity. It usually performs several actions without going to the server.
- It cannot be basically used to connect to databases on a web server.

2. Server-side Scripting :

- Web servers are used to execute server-side scripting. They are basically used to create dynamic pages. It can also access the file system residing at the webserver.
- A server-side environment that runs on a scripting language is a web server.
- Scripts can be written in any of a number of server-side scripting languages available.
- It is used to receive and generate content for dynamic pages.
- When you need to store and retrieve information a database will be used to contain data.

(7) Difference between GET & POST methods.
Explaining POST method in details.

GET

POST

→ In GET method we can not send large amounts of data either limited data of some number of characters is sent because the request parameter is appended into the URL.

→ In POST method large amount of data can be sent because the request parameters is appended into the body.

→ GET request is comparatively better than POST so it is used more than the POST request.

→ GET requests are only used to request data.

→ GET request is comparatively less secure because the data is exposed in URL bar.

→ POST request is comparatively less better than GET method, so it is used less than the GET request.

→ POST request can be used to create and modify data.

→ POST request is comparatively more secure because the data is not exposed in the URL bar.

POST:

→ The HTTP POST method is mainly used at the client (Browser) side to send data to a specified server in order to create or rewrite a particular resources/data.

→ This data sent to the server is stored in the request body of the HTTP request.

→ POST method eventually leads to the creation of a new resource or updating an existing one.

→ Due to this dynamic use, it is one of the most used HTTP methods.

Explain any five array function with syntax and example.

(1) count () :-

→ PHP count() function counts all elements present in an array.

Syntax:-

~~\$noOf~~

`$noElements = count($array);`

Example:-

`$season = array("summer", "winter",
"monsoon");`

`$countSeason = count($season);`

(2) ~~sort~~ sort () :-

→ PHP sort() function sorts all the elements in an array.

Syntax:
~~Example~~

`sort($array);`

Example:-

`$numbers = array(5, 3, 2, 4, 1);`

`sort($numbers);`

(3) array_reverse () :-

→ PHP array_reverse() function returns an array containing elements in reversed order.

Syntax:

\$reversed = array_reverse(\$array);

Example:

\$numbers = array(5, 4, 3, 2, 1);

\$reversed = array_reverse(\$numbers);

(4) array () :

→ PHP array() function creates and returns an array. It allows you to create indexed, associative and multi-dimensional array.

Syntax:

\$array_name = array([mixed \$...]);

Example:

\$season = array("summer", "winter", "monsoon");

(5) array_keys :-

→ PHP array_keys function returns all the key of an array.

Syntax:

\$keys = array_keys(\$array);

example:

\$a = array("volvo" => "xc90",
"BMW" => "x5",
"Toyota" => "Highlander");

print_r(array_keys(\$a));

(3) Explain any five variable function in details.

4

(1) gettype():

→ PHP gettype() function returns the type of a variable.

Syntax:

gettype(variable);

example:

\$a = 3;

echo gettype(\$a);

Output:

integer

(2) empty():

→ PHP empty() function checks whether a variable is empty.

Syntax:

empty(variable);

example:-

`$cl = 0;`

`echo empty($cl);`

output: true

(3) `isSet()` :-

→ PHP `isSet()` function checks whether a variable is set.

Syntax:

`isSet(variable, ...);`

example:-

`$cl = 0;`

`echo isSet($cl);`

output: true

(4) `print_r()`:

→ PHP `print_r()` function prints the info. about a variable in a human readable way.

Syntax:

`print_r(variable; return);`

example:-

`$cl = array("red", "green", "blue");`

`print_r($cl);`

Output:-

Array (

[0] => red

[1] => green

[2] => blue

(5) var_dump():

→ PHP var_dump() function dumps info about one or more variables.

Syntax:

var_dump (var1, var2, ...);

Example:

\$a = 32;

echo var_dump(\$a);

Output:

int(32)

(10) Explain file uploading in PHP with example

→ A PHP script can be used with a HTML form to allow users to upload files to the server.

→ Initially files are uploaded into a temporary directory and then relocated to a target destination by a PHP script.

→ The process of uploading a file follows these steps :-

→ The user opens the page containing a HTML form featuring a text field, a browser button and a submit button.

→ The user clicks the browser button and selects a file to upload from the local pc.

→ The full path to the selected file appears in

the text filed then the user clicks the submit button.

- The selected file is sent to the temporary directory on the Server
- The PHP script that was specified as the form handler in the form's `action` attribute checks that the file has arrived and then copies the file into and intended directory.
- The PHP script confirms the success to the user

Example

<?php

```

if(isset($_FILES['image'])) {
    $errors = array();
    $filename = $_FILES['image']['name'];
    $file_size = $_FILES['image']['size'];
    $file_tmp = $_FILES['image']['tmp_name'];
    $file_type = $_FILES['image']['type'];
    $file_ext = strtolower(end(explode('.',$filename)));
    $extensions = array("jpeg","jpg","png");

    if(in_array($file_ext,$extensions) == false) {
        $errors[] = "extension not allowed";
    }

    if($file_size > 2097152) {
        $errors[] = "File size must be exactly 2MB";
    }
}

```

```
if (empty($errors) == true) {  
    move_uploaded_file($_FILES['file']['tmp_name'], "images/" . $file_name);  
    echo "Success";  
}  
else {  
    print_r($errors);  
}  
?  
<html>  
  <body>  
    <form action="" method="post"  
          enctype="multipart/form-data">  
      <input type="file" name="image" />  
      <input type="submit" />  
    </form>  
  </body>  
</html>
```