Basics of PHP

PHP File

- PHP files can contain text, HTML, CSS, JavaScript and PHP code
- PHP code is executed on the server and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

Basic PHP Syntax

- A PHP script can be placed anywhere in the document.
- A PHP script starts with <?php and ends with ?>
- Syntax:

```
<?php                    // PHP starting directive
    // PHP code
?>                       // PHP ending directive
```

- PHP statements end with a semicolon (;)
- keywords (e.g. if, else, while, echo, etc.), classes, functions and user-defined functions are not case-sensitive.
- All variable names are case-sensitive!
- Comments (// or # or /*……..*/)

PHP Variable

- Variables are "containers" for storing information.
- In PHP, a variable starts with the $ sign, followed by the name of the variable.

Example:

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

Q: How PHP is a Loosely Typed Language?

- PHP automatically associates a data type to the variable and depending on its value.

PHP Variables Scope

- In PHP, variables can be declared anywhere in the script.
- The scope of a variable is the part of the script where the variable can be referenced/used.
- PHP has three different variable scopes:
  - local
  - global
  - static

PHP Variables Scope

Local Scope:

- A variable declared within a function has a LOCAL SCOPE.
- It can only be accessed within a function.

Global Scope:

- A variable declared outside a function has a GLOBAL SCOPE.
- It can only be accessed outside a function.

PHP Variables Scope

1. The global Keyword:

- The global keyword is used to access a global variable within a function.
- Use the global keyword before the variables inside the function.

2. $GLOBALS:

- All global variables are stored in an array called $GLOBALS[index].
- The index holds the name of the variable.
- This array is also accessible within functions and can be used to update global variables directly.

PHP Variables Scope

Static Scope:

- When a function is executed, all of its variables are deleted. However, sometimes we want a local variable NOT to be deleted.
- Static keyword contained information of local variables from the last time the function was called.

Example

```
function test()
{
static $x=0;
echo $x;
$x++;
}
echo "First time:";
test();
echo "\n Second time:";
test();
echo "\n Third time:";
test();
?>
```

PHP Output Statements

echo and print

- Similarity: They are used to output data to the screen.
- Difference:
  - 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 while print can take one argument.
  - echo is faster than print.
- echo and print statements can be used with or without parentheses: eg. echo or echo()

PHP Data Types

- PHP supports the following data types:
  - String
  - Integer
  - Float (floating point numbers - also called double)
  - Boolean
  - Array
  - Object
  - NULL - If a variable is created without a value, it is automatically assigned a value of NULL.
- var_dump() function returns the data type and value

PHP Constants

- PHP constants are name or identifier that can't be changed during the execution of the script.
- PHP constants can be defined by 2 ways:
  - Using define() function
  - Using const keyword

PHP constant: define()

- Use the define() function to create a constant. It defines constant at run time.
- Syntax: define(name, value, case-insensitive)

PHP Constants

PHP constant: const keyword or Constant() function

- PHP introduced a keyword const to create a constant. The const keyword defines constants at compile time. It is a language construct, not a function. The constant defined using const keyword are case-sensitive.

Example:

```
<?php
const MESSAGE=" Welcome to GUNI";
echo MESSAGE;
?>
```

- There is another way to print the value of constants using constant() function instead of using the echo statement.
- Syntax: constant (name)

Example:

```
<?php
   define("MSG", " Welcome to GUNI ");
   echo MSG, "</br>";
   echo constant("MSG");
   //both are similar
?>
```

Operators

- Arithmetic Operators
- Assignment Operators
- Bitwise Operators
- Comparison Operators
- Incrementing/Decrementing Operators
- Logical Operators
- String Operators
- Array Operators
- Type Operators
- Execution Operators
- Error Control Operators

Dynamic Input

Ways to take input from User at run time:

- readline(‘Message’)
- fscanf(STDIN, "%d %d", $a, $b)
- fgets(STDIN)
- Command line arguments: $argv[1]
- Using Form and global variables $_GET, $_POST, $_REQUEST

Conditional Statements

- if statement - executes some code if one condition is true
- if...else statement - executes some code if a condition is true and another code if that condition is false
- if...elseif...else statement - executes different codes for more than two conditions
- switch statement - selects one of many blocks of code to be executed

Iterative Statements (Loops)

- while - loops through a block of code as long as the specified condition is true
- do...while - loops through a block of code once, and then repeats the loop as long as the specified condition is true
- for - loops through a block of code a specified number of times
- foreach - loops through a block of code for each element in an array

PHP Array

- It is used to hold multiple values of similar type in a single variable.
- There are 3 types of array in PHP:
  - Indexed Array
  - Associative Array
  - Multidimensional Array

PHP Array

PHP Indexed Array

- PHP index is represented by number which starts from 0.
- Used to store number, string and object in the PHP array.
- All PHP array elements are assigned to an index number by default.
- There are two ways to define indexed array:

```
$season=array("summer","winter", "monsoon");
        or
$season[0]="summer";
$season[1]="winter";
$season[2]="monsoon";
```

PHP Array

PHP Associative Array

- To associate name with each array elements in PHP using => symbol.
- There are two ways to define associative array:

```
$salary=array("Mr.A"=>"350000",  "Mr.B"=>"450000", "Mr.C"=>"200000");
```

or

```
$salary["Mr.A"]="350000";
$salary["Mr.B"]="450000";
$salary["Mr.C"]="200000";
```

PHP Array

PHP Multidimensional Array

- PHP multidimensional array is also known as array of arrays.
- It allows to store tabular data in an array.
- PHP multidimensional array can be represented in the form of matrix which is represented by row * column.

Example:

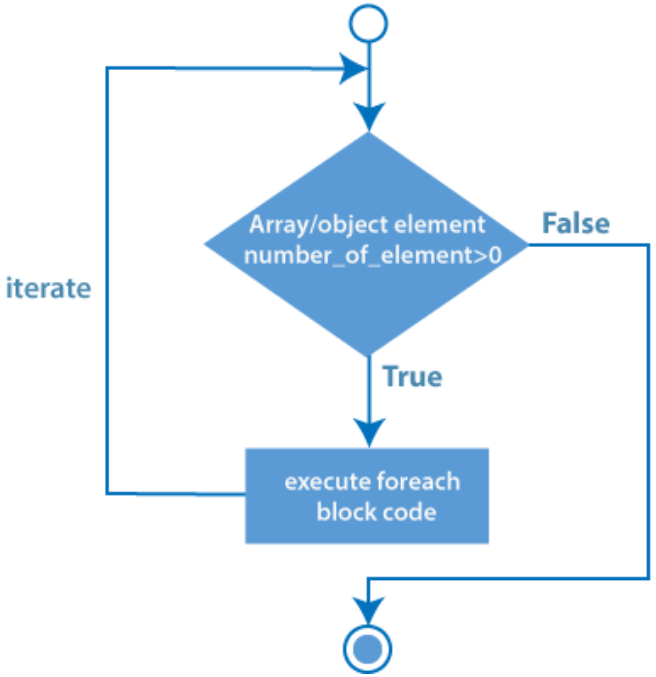
```
$emp = array
 (
 array(1, "emp1",400000),
 array(2, "emp2",500000),
 array(3, "emp3",300000)
 );
```

Foreach loop

- 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 variables 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 $value) {
    //code to be executed
}
        or
foreach ($array as $key => $element) {
    //code to be executed
}
```



For Vs Foreach loop

BASIS COMPARISON	FOR	FOREACH
Implemented over	Variable	Numerical and associative arrays
Working	At the end of the given condition	At the end of the array count
Syntax	for(expr1; expr2; expr3) {//If expr2 is true, do this}	foreach ($array as $value) {//Do Something} or foreach ($array as $key => $value) {//Do Something}
Scope	It works with all types of data	It works only on array and object

PHP Array built-In Functions

- array()
- count()
- sort()
- array_push()
- array_reverse()
- array_search()
- array_unique()
- array_walk()
- in_array()
- unset()

String & in-built functions

- Single quote
- Double quote
- heredoc syntax: <<<

- **Functions:** strlen, strcmp, strrev, strtolower, trim, explode, implode, join, split, crypt, htmlentities, echo, print……

User defined function

- Syntax:

```
function functionname()
{
//code to be executed
}
```

- PHP supports Call by Value (default), Call by Reference, Default argument values and Variable-length argument list.

User defined function

Types of functions:

- Call by Value (default)
- Call by Reference
- Default argument values
- Variable-length argument list
- Recursion

PHP Global Variables - Superglobals

Superglobals: Predefined variables in PHP
Always accessible in all scopes.
It can be accessed from any function, class or file.

The PHP superglobal variables are:

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

PHP Global Variables - Superglobals

PHP $GLOBALS

- $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).
- PHP stores all global variables in an array called $GLOBALS[index]. The index holds the name of the variable.
- Example:

```
<?php
$x = 10;
$y = 20;

function addition() {
  $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

addition();
echo $z;
?>
```

PHP Global Variables - Superglobals

PHP $_SERVER

- $_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.
- Example:

```
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER[''REQUEST_METHOD''];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

PHP Global Variables - Superglobals

PHP $_GET

- PHP $_GET is a PHP super global variable which is used to collect form data after submitting an HTML form with method="get".
- $_GET can collect data sent via URL.

Example: Assume we have an HTML page that contains a hyperlink with parameters:

index.html

```
<html>
<body>
 <a href="test.php?subject=WT&technology=PHP">Test $GET</a>
 </body>
</html>
```

Test.php

```
<html>
<body>
 <?php
echo "Study " . $_GET['subject'] . " at " . $_GET[‘technology'];
?>
</body>
</html>
```

PHP Global Variables - Superglobals

PHP $_POST

- PHP $_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.

- Example:

```
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
 $name = $_POST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;  }  }
?>
```

PHP Global Variables - Superglobals

PHP $_REQUEST

- PHP $_REQUEST is a PHP super global variable which is used to collect data after submitting an HTML form.
- The $_REQUEST function is used to get the form information sent with POST method or GET method.

PHP Global Variables - Superglobals

PHP $_ENV:

- $_ENV is another superglobal associative array in PHP.
- It stores environment variables available to current script now been deprecated.
- Environment variables are imported into global namespace. Most of these variables are provided by the shell under which PHP parser is running. Hence, list of environment variables may be different on different platforms.
- This array also includes CGI variables in case whether PHP is running as a server module or CGI processor.
- PHP library has getenv()function to retrieve list of all environment variables or value of a specific environment variable

PHP Global Variables - Superglobals

PHP $_ENV:

- Getenv() function: To retrieve list of all environment variables or value of a specific environment variable

- Example:

```
<?php
$arr=getenv();
foreach ($arr as $key=>$val)
echo "$key=>$val";
?>
```