PHP Introduction

Table of Contents

1. About PHP

- PHP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is a server-side 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.
- PHP scripts are executed on the server.
- PHP is an acronym for "PHP: Hypertext Preprocessor".
- 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".
- 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.

2. How to run PHP

- PHP can be run on various platforms like Windows, Linux, Unix, Mac OS X, etc.
- PHP files can be run on a local server or a remote server.
- To run PHP on a local server, you need to install a web server (Apache),

PHP, and MySQL.
- You can use XAMPP, WAMP, MAMP, etc., to run PHP on a local server.

3. PHP Syntax

- A PHP script starts with `<?php` and ends with `?>`.
- PHP statements end with a semicolon (;).
- PHP files have extension ".php".
- PHP is case-sensitive.
- In PHP, keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are NOT case-sensitive.

4. PHP Comments

- Comments in PHP are the same as in JavaScript.
- Single-line comments start with `//`.
- Multi-line comments start with `/*` and end with `*/`.

5. PHP Variables

- 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).
- The value of a variable can change over time.
- PHP automatically converts the variable to the correct data type, depending on its value.
- PHP variables can store different types of data, such as strings, numbers, objects, arrays, etc.

```php
<?php
  $name = "John";
  $age = 25;
?>
```

6. PHP Data Types

- PHP supports the following data types:
  - String
  - Integer
  - Float (floating point numbers - also called double)
  - Boolean
  - Array

  - Object
  - NULL
  - Resource

```php
<?php
  $name = "John"; // String
  $age = 25; // Integer
  $height = 5.11; // Float
  $isMale = true; // Boolean
  $friends = array("Jane", "Doe", "Smith"); // Array
  $car = null; // NULL
  $object = new stdClass(); // Object
  $object->name = "John";
  $object->age = 25;
?>
```

7. PHP Strings

- A string is a sequence of characters, like "Hello World!".
- A string can be any text inside quotes. You can use single or double quotes.
- A string can be assigned to a variable.
- A string can be output with `echo` or `print` functions.

```php
<?php
  $name = "John";
  echo "Hello, $name!";
?>
```

8. PHP Constants

- A constant is an identifier (name) for a simple value.
- The value cannot be changed during the script.
- A valid constant name starts with a letter or underscore (no $ sign before the constant name).

```php
<?php
  define("SITE_URL", "https://www.example.com");
  echo SITE_URL;
?>
```

9. PHP Operators

- PHP operators are used to perform operations on variables and values.
- PHP divides the operators into the following groups:
  - Arithmetic operators

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

```php
  $y = 5;
  echo $x + $y; // 15
?>
```
– Assignment operators
```php
<?php
  $x = 10;
  $y = 5;
  $x += $y; // $x = $x + $y;
  echo $x; // 15
?>
```
– Comparison operators
```php
<?php
  $x = 10;
  $y = 5;
  var_dump($x == $y); // false
?>
```
– Increment/Decrement operators
```php
<?php
  $x = 10;
  echo ++$x; // 11
?>
```
– Logical operators
```php
<?php
  $x = 10;
  $y = 5;
  var_dump($x > 5 && $y < 10); // true
?>
```
– String operators
```php
<?php
  $x = "Hello";
  $y = "World!";
  echo $x . " " . $y; // Hello World!
?>
```
– Array operators
```php
<?php
  $x = array("a" => "red", "b" => "green");
  $y = array("c" => "blue", "d" => "yellow");
  var_dump($x + $y);
?>
```
– Conditional assignment operators
```php
<?php
// If $x is not set, the value of $y will be 5
  $x = 10;
  $y = $x ?: 5;
  echo $y; // 10
```

```php
    // Equivalent to
    $x = 10;
    $y = $x ?? 5;
    echo $y; // 10
  ?>
```

10. PHP If...Else...Elseif

- PHP `if` statement executes some code if one condition is true.
- PHP `else` statement executes some code if the condition is false.
- PHP `elseif` statement executes some code if the condition is false.

```php
<?php
  $age = 20;
  if ($age < 18) {
    echo "You are a minor.";
  } elseif ($age >= 18 && $age < 60) {
    echo "You are an adult.";
  } else {
    echo "You are a senior citizen.";
  }
?>
```

11. PHP Switch

- PHP `switch` statement is used to select one of many blocks of code to be executed.
- The `switch` statement is similar to a series of `if` statements on the same expression.
- The value of the expression is compared with the values of each `case`.
- If there is a match, the block of code associated with that `case` is executed.

```php
<?php
  $favColor = "red";
  switch ($favColor) {
    case "red":
      echo "Your favorite color is red!";
      break;
    case "blue":
      echo "Your favorite color is blue!";
      break;
    case "green":
      echo "Your favorite color is green!";
      break;
    default:
      echo "Your favorite color is neither red, blue, nor green!";
```

```
  }
?>
```

12. PHP Loops

- PHP supports the following loop types:
  - `for` - loops through a block of code a specified number of times.
  - `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.
  - `foreach` - loops through a block of code for each element in an array.

```php
<?php
  // for loop
  for ($i = 0; $i < 5; $i++) {
    echo $i;
  }

  // while loop
  $i = 0;
  while ($i < 5) {
    echo $i;
    $i++;
  }

  // do...while loop
  $i = 0;
  do {
    echo $i;
    $i++;
  } while ($i < 5);

  // foreach loop
  $colors = array("red", "green", "blue", "yellow");
  foreach ($colors as $color) {
    echo $color;
  }
?>
```

13. PHP Functions

- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute immediately when a page loads.
- A function will be executed by a call to the function.

- PHP functions are case-insensitive.

```php
<?php
  function greet() {
    echo "Hello!";
  }

  greet();
?>

<?php
  function sum($x, $y) {
    return $x + $y;
  }

  echo sum(5, 10);
?>
```

14. PHP Arrays

- An array stores multiple values in a single variable.
- An array can store many values under a single name.
- An array can store different data types.
- An array can store key-value pairs.

```php
<?php
  // Indexed arrays
  $colors = array("red", "green", "blue");
  echo $colors[0]; // red

  // Associative arrays
  $age = array("John" => 25, "Doe" => 30);
  echo $age["John"]; // 25

  // Associative arrays
  $colors = array("red", "green", "blue");
  foreach ($colors as $color) {
    echo $color;
  }

  $age = array("John" => 25, "Doe" => 30);

  foreach ($age as $key => $value) {
    echo "Key=" . $key . ", Value=" . $value;
  }

  $age["Smith"] = 35;
```

```php
  unset($age["Doe"]);

  // Array functions
  $colors = array("red", "green", "blue");
  echo count($colors); // 3
  echo sort($colors); // green, red, blue
  echo rsort($colors); // blue, red, green
  echo array_push($colors, "yellow"); // 4
  echo array_pop($colors); // yellow
  echo array_shift($colors); // blue
  echo array_unshift($colors, "blue"); // 3

  $age = array("John" => 25, "Doe" => 30);
  echo array_key_exists("John", $age); // true
  echo array_values($age); // 25, 30
  echo array_keys($age); // John, Doe
  echo array_merge($colors, $age); // blue, red, green, John, Doe
  echo array_search("green", $colors); // 1
?>
```

15. PHP Classes and Objects

- PHP is an object-oriented programming language.
- A class is a blueprint for objects.
- An object is an instance of a class.
- A class can have properties and methods.
- Properties are variables that hold data.
- Methods are functions that perform actions.

```php
<?php
  class Car {
    public $color;
    public $model;

    public function __construct($color, $model) {
      $this->color = $color;
      $this->model = $model;
    }

    public function display() {
      echo "The car is " . $this->color . " and the model is " . $this->model;
    }
  }

  $car = new Car("red", "BMW");
  $car->display();
```

```
?>
```

Conclusion

- PHP is a powerful server-side scripting language.
- PHP is widely used for creating dynamic and interactive web pages.
- PHP is easy to learn and use.
- PHP is open-source and free.
- PHP is platform-independent.
- PHP is compatible with almost all servers used today (Apache, IIS, etc.).
- PHP supports a wide range of databases.
- PHP is secure and reliable.