PHP Programming

MOHAMMED NAJEM ALI
FULL STACK WEB DEVELOPER

PHP Introduction

- 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.
- 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"

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
- With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

PHP Installation

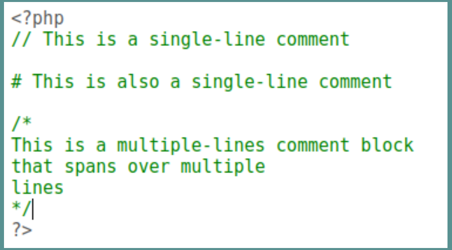
- install a web server (apache, nginx)
- install PHP
- install a database, such as MySQL
- You can install Xampp or WampServer

PHP Syntax

- A PHP script can be placed anywhere in the document.
- A PHP script starts with <?php and ends with ?>
- The default file extension for PHP files is ".php".
- A PHP file normally contains HTML tags, and some PHP scripting code.
- PHP statements end with a semicolon (;).
- In PHP, keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are not case-sensitive.
- all variable names are case-sensitive!

PHP Comments

- A 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.



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

# This is also a single-line comment

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

PHP Variables

- Variables are "containers" for storing information.
- In PHP, a variable starts with the $ sign, followed by the name of the variable.
- Unlike other programming languages, PHP has no command for declaring a variable. It is created the moment you first assign a value to it.
- A variable name cannot start with a number
- Variable names are case-sensitive ($age and $AGE are two different variables)
- In PHP 7, type declarations were added. This gives an option to specify the data type expected when declaring a function, and by enabling the strict requirement, it will throw a "Fatal Error" on a type mismatch.

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
- A variable declared **outside** a function has a GLOBAL SCOPE and can only be accessed outside a function.
- A variable declared **within** a function has a LOCAL SCOPE and can only be accessed within that function.
- The global keyword is used to access a global variable from within a function. To do this, use the global keyword before the variables (inside the function).

PHP Variables Scope

- PHP also stores all global variables in an array called $GLOBALS[*index*]. The *index* holds the name of the variable. This array is also accessible from within functions and can be used to update global variables directly.
- Normally, when a function is completed/executed, all of its variables are deleted. However, sometimes we want a local variable NOT to be deleted. We need it for a further job. To do this, use the static keyword when you first declare the variable.

PHP Constants

- Constants are like variables except that once they are defined they cannot be changed or undefined.
- 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).
- **Note:** Unlike variables, constants are automatically global across the entire script.
- To create a constant, use the define() function.
- you can create an Array constant using the define() function.

PHP echo and print Statements

- With PHP, there are two basic ways to get output: echo and 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.
- The echo statement can be used with or without parentheses: echo or echo().
- The print statement can be used with or without parentheses: print or print().

PHP Data Types

- PHP supports the following data types: String, Integer, Float (floating point numbers - also called double), Boolean, Array, Object, NULL, Resource.
- A string can be any text inside quotes. You can use single or double quotes.
- An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.
- The PHP var_dump() function returns the data type and value.
- A float (floating point number) is a number with a decimal point or a number in exponential form.
- A Boolean represents two possible states: TRUE or FALSE.
- An array stores multiple values in one single variable.

PHP Data Types

- Classes and objects are the two main aspects of object-oriented programming.
- A class is a template for objects, and an object is an instance of a class.
- When the individual objects are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.
- If you create a __construct() function, PHP will automatically call this function when you create an object from a class.
- Null is a special data type which can have only one value: NULL.
- A variable of data type NULL is a variable that has no value assigned to it.
- **Tip:** If a variable is created without a value, it is automatically assigned a value of NULL.

PHP Operators

- Operators are used to perform operations on variables and values.
- The PHP arithmetic operators are used with numeric values to perform common arithmetical operations (+, -, *, /, %, **).
- The PHP assignment operators are used with numeric values to write a value to a variable.
- The PHP comparison operators are used to compare two values (number or string) (==, ===, !=, <>, !==, >, <, >=, <=, <=>)
- The PHP increment/decrement operators are used to increment/decrement a variable's value. (++$x, $x++, --$x, $x--).
- The PHP logical operators are used to combine conditional statements. (and, or, xor, $$, ||, !)
- PHP has two operators that are specially designed for strings. (. , .=)

PHP if...else...elseif Statements

- Conditional statements are used to perform different actions based on different conditions.
- In PHP we have the following 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

PHP Loops

- Loops are used to execute the same block of code again and again, as long as a certain condition is true.
- In PHP, we have the following loop types:
  - 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 Break and Continue

- The break statement can be used to jump out of a loop.
- The continue statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.
- You can use break and continue in for or while loops .

PHP Arrays

- An array is a special variable, which can hold more than one value at a time.
- In PHP, the array() function is used to create an array.
- In PHP, there are three types of arrays:
    - **Indexed arrays** - Arrays with a numeric index.
    - **Associative arrays** - Arrays with named keys.
    - **Multidimensional arrays** - Arrays containing one or more arrays.
- The count() function is used to return the length (the number of elements) of an array.

PHP Global Variables - Superglobals

- Some predefined variables in PHP are "superglobals", which 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.
- The PHP superglobal variables are:

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

PHP Form Handling

- The PHP superglobals $_GET and $_POST are used to collect form-data.
- Both GET and POST create an array (e.g. array(key1 => value1, key2 => value2, key3 => value3, ...)). This array holds key/value pairs, where keys are the names of the form controls and values are the input data from the user.
- $_GET is an array of variables passed to the current script via the URL parameters.
- $_POST is an array of variables passed to the current script via the HTTP POST method.

PHP Form Validation

- The htmlspecialchars() function converts special characters to HTML entities. This means that it will replace HTML characters like < and > with &lt; and &gt;. This prevents attackers from exploiting the code by injecting HTML or Javascript code (Cross-site Scripting attacks) in forms.
- **The preg_match() function searches a string for pattern, returning true if the pattern exists, and false otherwise.**
- The easiest and safest way to check whether an email address is well-formed is to use PHP's filter_var() function.

PHP Functions

- The real power of PHP comes from its functions.
- PHP has more than 1000 built-in functions, and in addition you can create your own custom functions.
- A user-defined function declaration starts with the word function.
- Information can be passed to functions through arguments. An argument is just like a variable.

PHP Strings

- The PHP strlen() function returns the length of a string.
- The PHP str_word_count() function counts the number of words in a string.
- The PHP strrev() function reverses a string.
- The PHP strpos() function searches for a specific text within a string. If a match is found, the function returns the character position of the first match. If no match is found, it will return FALSE.
- The PHP str_replace() function replaces some characters with some other characters in a string.

PHP Numbers

- PHP_INT_SIZE - The size of an integer in bytes
- PHP has the following functions to check if the type of a variable is integer:
  a. is_int()
  b. is_integer() - alias of is_int()
  c. is_long() - alias of is_int()
- PHP has the following functions to check if the type of a variable is float:
  a. is_float()
  b. is_double() - alias of is_float()
- NaN stands for Not a Number.
- The (int), (integer), or intval() function are often used to convert a value to an integer.

```php
<?php
// Cast float to int
$x = 23465.768;
$int_cast = (int)$x;
echo $int_cast;

echo "<br>";

// Cast string to int
$x = "23465.768";
$int_cast = (int)$x;
echo $int_cast;
?>
```

PHP Math

- The min() and max() functions can be used to find the lowest or highest value in a list of arguments
- The abs() function returns the absolute (positive) value of a number.
- The sqrt() function returns the square root of a number.
- The round() function rounds a floating-point number to its nearest integer.
- The rand() function generates a random number.
- To get more control over the random number, you can add the optional min and max parameters to specify the lowest integer and the highest integer to be returned..
- For example, if you want a random integer between 10 and 100 (inclusive), use rand(10, 100).

PHP Sorting Arrays

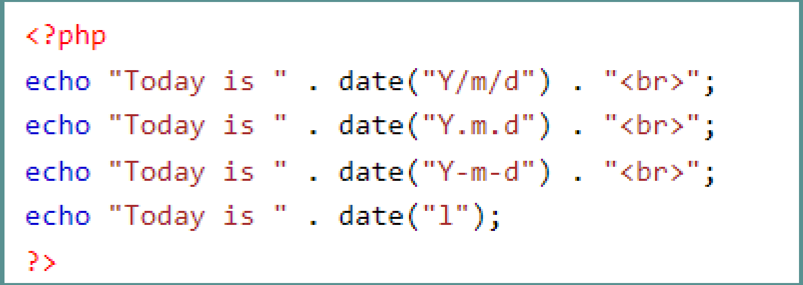
- sort() - sort arrays in ascending order
- rsort() - sort arrays in descending order
- asort() - sort associative arrays in ascending order, according to the value
- ksort() - sort associative arrays in ascending order, according to the key
- arsort() - sort associative arrays in descending order, according to the value
- krsort() - sort associative arrays in descending order, according to the key

PHP Date and Time

- The PHP date() function is used to format a date and/or a time.
- The PHP date() function formats a timestamp to a more readable date and time.
- date(format Required, timestamp Optional)
- The required *format* parameter of the date() function specifies how to format the date (or time).
- Here are some characters that are commonly used for dates:
  - d - Represents the day of the month (01 to 31)
  - m - Represents a month (01 to 12)
  - Y - Represents a year (in four digits)
  - l (lowercase 'L') - Represents the day of the week

PHP Date and Time

- Other characters, like"/", ".", or "-" can also be inserted between the characters to add additional formatting.



```php
<?php
echo "Today is " . date("Y/m/d") . "<br>";
echo "Today is " . date("Y.m.d") . "<br>";
echo "Today is " . date("Y-m-d") . "<br>";
echo "Today is " . date("l");
?>
```

PHP OOP

- OOP stands for Object-Oriented Programming.
- Procedural programming is about writing procedures or functions that perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions.
- OOP is faster and easier to execute
- OOP provides a clear structure for the programs
- OOP makes it possible to create full reusable applications with less code and shorter development time

PHP OOP

- A class is a template for objects, and an object is an instance of a class.
- When the individual objects are created, they inherit all the properties and behaviors from the class, but each object will have different values for the properties.
- The $this keyword refers to the current object, and is only available inside methods.
- You can use the instanceof keyword to check if an object belongs to a specific class

PHP OOP - Constructor

- A constructor allows you to initialize an object's properties upon creation of the object.
- If you create a __construct() function, PHP will automatically call this function when you create an object from a class.
- A destructor is called when the object is destructed or the script is stopped or exited.
- If you create a __destruct() function, PHP will automatically call this function at the end of the script.

PHP OOP - Access Modifiers

- Properties and methods can have access modifiers which control where they can be accessed.
- There are three access modifiers:
  - public - the property or method can be accessed from everywhere. This is default
  - protected - the property or method can be accessed within the class and by classes derived from that class
  - private - the property or method can ONLY be accessed within the class

PHP OOP - Inheritance

- Inheritance in OOP = When a class derives from another class.
- The child class will inherit all the public and protected properties and methods from the parent class. In addition, it can have its own properties and methods.
- An inherited class is defined by using the extends keyword.
- Inherited methods can be overridden by redefining the methods (use the same name) in the child class.
- The final keyword can be used to prevent class inheritance or to prevent method overriding.

PHP OOP - Class Constants

- Constants cannot be changed once it is declared.
- Class constants can be useful if you need to define some constant data within a class.
- A class constant is declared inside a class with the const keyword.
- Class constants are case-sensitive. However, it is recommended to name the constants in all uppercase letters.
- We can access a constant from outside the class by using the class name followed by the scope resolution operator (::) followed by the constant name.
- Or, we can access a constant from inside the class by using the self keyword followed by the scope resolution operator (::) followed by the constant name

PHP OOP - Abstract Classes

- Abstract classes and methods are when the parent class has a named method, but need its child class(es) to fill out the tasks.
- An abstract class is a class that contains at least one abstract method. An abstract method is a method that is declared, but not implemented in the code.
- An abstract class or method is defined with the abstract keyword.
- The child class method must be defined with the same name and it redeclares the parent abstract method.
- The child class method must be defined with the same or a less restricted access modifier.
- The number of required arguments must be the same. However, the child class may have optional arguments in addition

PHP OOP - Interfaces

- Interfaces allow you to specify what methods a class should implement.
- Interfaces make it easy to use a variety of different classes in the same way. When one or more classes use the same interface, it is referred to as "polymorphism".
- Interfaces are declared with the interface keyword.

PHP OOP - Interfaces

- Interface are similar to abstract classes. The difference between interfaces and abstract classes are:
    - Interfaces cannot have properties, while abstract classes can
    - All interface methods must be public, while abstract class methods is public or protected
    - All methods in an interface are abstract, so they cannot be implemented in code and the abstract keyword is not necessary
    - Classes can implement an interface while inheriting from another class at the same time.
- A class that implements an interface must implement **all** of the interface's methods.

PHP OOP - Traits

- Traits are used to declare methods that can be used in multiple classes.
- Traits can have methods and abstract methods that can be used in multiple classes, and the methods can have any access modifier (public, private, or protected).
- Traits are declared with the trait keyword.
- To use a trait in a class, use the use keyword.

PHP OOP - Static Methods

- Static methods can be called directly - without creating an instance of the class first.
- Static methods are declared with the static keyword.
- To access a static method use the class name, double colon (::), and the method name.
- A static method can be accessed from a method in the same class using the self keyword and double colon (::)
- Static methods can also be called from methods in other classes. To do this, the static method should be public.

PHP OOP - Namespaces

- Namespaces are qualifiers that solve two different problems:
    - They allow for better organization by grouping classes that work together to perform a task
    - They allow the same name to be used for more than one class
- Namespaces are declared at the beginning of a file using the namespace keyword.
- For further organization, it is possible to have nested namespaces.