

Introduction to PHP



SoftUni Team
Technical Trainers



SoftUni



Software University

<https://softuni.bg>

sli.do

#WordPress

Table of Content

1. PHP Introduction
2. PHP Syntax and Basics
3. PHP Variables, Arrays, Operators
4. PHP Loops
5. PHP Functions
6. Practice



Some Assumptions

- Programming is fairly new to you
- But you have some experience
- PHP is new to you
- You know what a variable, condition statement, and loop are



A Few Notes Before We Start



- The course is touching the surface
- There will always be something else to show
- We'll try to have less "watch me how I code" part and more interaction
- We'll have less code in terms of lines, but hopefully we'll have a better understanding what code does



PHP Introduction

What Is PHP?

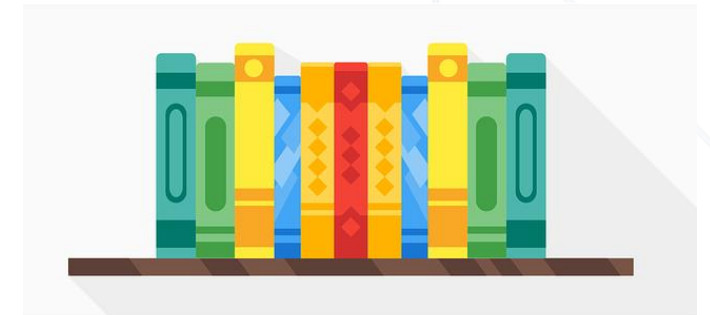
- **PHP** (recursive acronym for **PHP: Hypertext Preprocessor**) is a widely-used **open source** general-purpose scripting language that is especially suited for web development and can be embedded into **HTML**

Resource: www.php.net



PHP Documentation

- Official Documentation
 - <https://www.php.net/docs.php>
- https://www.w3schools.com/php/php_intro.asp
- Google is your friend
- ChatGPT* as well, if you know how to use it



PHP Opening and Closing Tags

```
<?php  
// PHP code goes here  
?>
```

```
<?php // some code here ?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>Heading</h1>
```

```
<?php
```

```
echo "Hello World!";
```

```
?>
```

```
</body>
```

```
</html>
```

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

- 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)

PHP Variables - Example

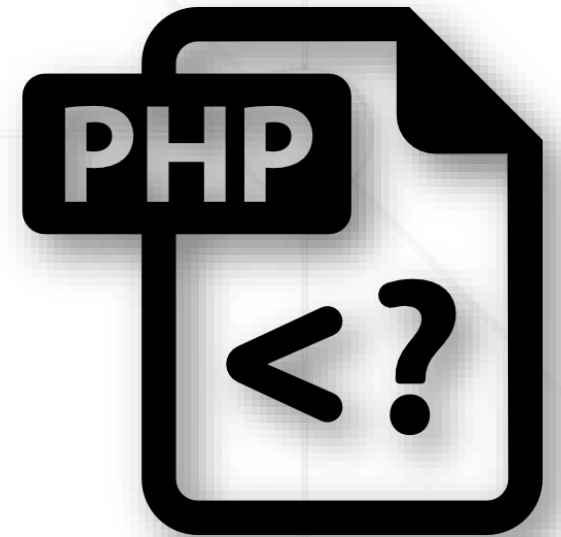
PHP variable starts with \$

```
$string = 'example';  
echo $string;
```

```
$number = 123;  
echo $number;
```

```
$another_number = 123.05;  
echo $another_number;
```

- 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
$txt1 = "Learn PHP";
$txt2 = "softuni.bg";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>
```



PHP Data Types

- **String**
- **Integer and Float**
- **Boolean**
- **Array**
- **Object**
- **NULL**
- **Resource*** - The special resource type is not an actual data type. It is the storing of a reference to functions and resources external to PHP



PHP Data Types - String

```
<?php
$x = "Hello world!";
$y = 'Hello world!';

echo $x;
echo "<br>";
echo $y;
?>
```

PHP Data Types - Integer and Float

```
<?php  
$x = 5985;  
var_dump($x);  
?>
```

```
<?php  
$x = 10.365;  
var_dump($x);  
?>
```

PHP Data Types - Boolean

```
<?php  
$x = true;  
$y = false;  
?>
```

PHP Data Types - Array

```
<?php  
$cars = array("Volvo", "BMW", "Toyota");  
var_dump($cars);  
?>
```

- 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)

```
$array = array( 1, 2, 3 );  
$array = [1, 2, 3];
```

```
$fruits = array("Apple", "Banana", "Orange");  
$fruits = array('Apple', 'Banana', 'Orange');
```

```
if ( $a == $b ) {}  
  
if ( $a == $b ) {  
    // some Logic here  
} else {  
    // another Logic  
}  
  
if ( $a == $b ) {  
    // some Logic here  
} elseif ( $c == $d ) {  
    // another Logic  
} else {  
    // in the last else  
}
```




PHP Operators

- **Operators** are used to perform operations on variables and values
- **PHP divides** the operators in the following groups:
 - **Arithmetic operators**
 - **Assignment operators**
 - **Comparison operators**
 - **Increment / Decrement operators**
 - **Logical operators**
 - **String operators**
 - **Array operators**
 - **Conditional assignment operators**



PHP Arithmetic Operators

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power

PHP Assignment Operators

Assignment	Same as...	Description
<code>x = y</code>	<code>x = y</code>	The left operand gets set to the value of the expression on the right
<code>x += y</code>	<code>x = x + y</code>	Addition
<code>x -= y</code>	<code>x = x - y</code>	Subtraction
<code>x *= y</code>	<code>x = x * y</code>	Multiplication
<code>x /= y</code>	<code>x = x / y</code>	Division
<code>x %= y</code>	<code>x = x % y</code>	Modulus

PHP Arithmetic Operators

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power

PHP Comparison Operators

Operator	Name	Example	Result
<code>==</code>	Equal	<code>\$x == \$y</code>	Returns true if \$x is equal to \$y
<code>===</code>	Identical	<code>\$x === \$y</code>	Returns true if \$x is equal to \$y, and they are of the same type
<code>!=</code>	Not equal	<code>\$x != \$y</code>	Returns true if \$x is not equal to \$y
<code><></code>	Not equal	<code>\$x <> \$y</code>	Returns true if \$x is not equal to \$y
<code>!==</code>	Not identical	<code>\$x !== \$y</code>	Returns true if \$x is not equal to \$y, or they are not of the same type
<code>></code>	Greater than	<code>\$x > \$y</code>	Returns true if \$x is greater than \$y
<code><</code>	Less than	<code>\$x < \$y</code>	Returns true if \$x is less than \$y
<code>>=</code>	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if \$x is greater than or equal to \$y
<code><=</code>	Less than or equal to	<code>\$x <= \$y</code>	Returns true if \$x is less than or equal to \$y
<code><=></code>	Spaceship	<code>\$x <=> \$y</code>	Returns an integer less than, equal to, or greater than zero, depending on if \$x is less than, equal to, or greater than \$y. Introduced in PHP 7.



PHP Loops

- Often when you write code, you want the **same block of code** to run **over and over again** a certain number of times
- Instead of adding several almost **equal code-lines** in a script, we can use 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 Loops - while

```
<?php
$x = 1;

while($x <= 5) {
    echo "The number is: $x <br>";
    $x++;
}
?>
```

PHP Loops - do while

```
<?php
$x = 1;

do {
    echo "The number is: $x
    <br>";
    $x++;
} while ($x <= 5);
?>
```

```
<?php  
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}  
?>
```

```
<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
    echo "$value <br>";
}
?>
```

```
<?php
for ($x = 0; $x < 10; $x++) {
    if ($x == 4) {
        break;
    }
    echo "The number is: $x
    <br>";
}
?>
```

```
<?php
for ($x = 0; $x < 10; $x++) {
    if ($x == 4) {
        continue;
    }
    echo "The number is: $x <br>";
}
?>
```

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

- **PHP Built-in** Functions
 - **PHP** has more than **1000** built-in **functions**, and in addition you can create your own custom functions
- **PHP User** Defined Functions
 - A function is **a block of statements** that can be used repeatedly in a program
 - A function will **not execute automatically** when a page loads
 - A function will **be executed by a call** to the function

PHP Functions - Example

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

write_msg(); // call the function
?>

<?php
function write_msg( $name ) {
    echo "Hello world, $name!";
}

write_msg( 'SoftUni' ); // call the function
?>
```

PHP Superglobals



- A superglobal is a predefined variable in PHP that is always accessible, regardless of scope, and can be used globally in any script
- <https://www.php.net/manual/en/language.variables.superglobals.php>

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

PHP \$_GET and \$_POST

- \$_GET and \$_POST are the most common used *
- Or at least they'll be the most used in our course

- `$_GET` is a superglobal array in PHP used to collect form data sent via the URL query string
- Data is visible in the URL, making it useful for retrieving values that don't require security (e.g., search queries)
- How it Works:
 - Form data is appended to the URL after **the ?** symbol
 - Access the data with **`$_GET['parameter_name']`**

PHP \$_GET Example

```
// URL: http://example.com?name=Alice
```

```
<?php
if ( isset( $_GET['name'] ) ) {
    echo 'Hello, ' . $_GET['name'];
} else {
    echo 'Hello, Guest!';
}
?>
```

Output:

- If the URL contains ?name=Alice, it will output: Hello, Alice!
- If no name is provided, it will output: Hello, Guest!

- `$_POST` is a superglobal array in PHP used to collect form data sent via an HTTP POST request
- Data is not visible in the URL, making it more secure for sensitive information (e.g., passwords, login forms)
- How it Works:
 - Data is sent through the HTTP request body, rather than the URL.
 - Access the data with `$_POST['parameter_name']`

PHP \$_GET Example

```
<!-- HTML Form -->
<form method="post" action="">
    <input type="text" name="username" placeholder="Enter your name">
    <input type="submit" value="Submit">
</form>

<!-- PHP to handle POST request -->
<?php
if ( $_SERVER['REQUEST_METHOD'] === 'POST' ) {
    $name = sanitize_text_field( $_POST['username'] );
    echo 'Welcome, ' . $name;
}
?>
```

Output:

- After submitting the form, the page will display: Welcome, [entered name]

- **Classes**
- **OOP**
- **`__construct()`**
- **namespaces ***

```
<?php
class Fruit {
    // Properties
    public $name;
    public $color;

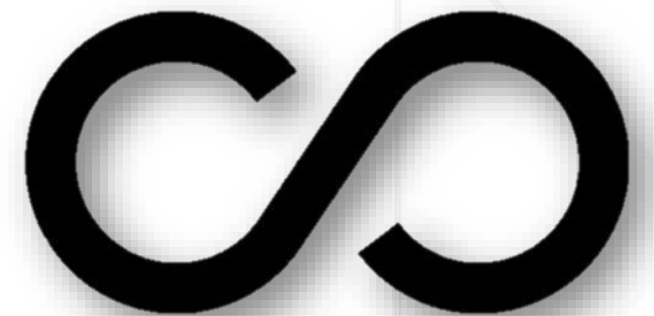
    // Methods
    function set_name($name) {
        $this->name = $name;
    }
    function get_name() {
        return $this->name;
    }
}
```



Practice

Let's Explore Some Basic PHP

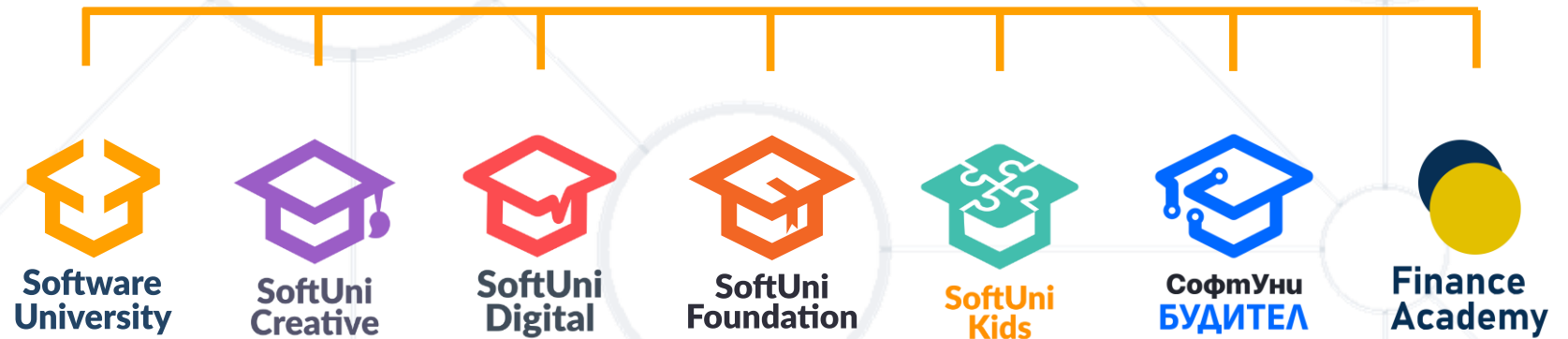
- Form handling
- **String** operations
- **Array** operations
- **Math** operations
- **Sessions**
- **Cookies**
- Working with files
- Many many more



- We know more about **PHP**
- We have **reviewed** the basics
- We know what **variable** is
- We know what **loop** is
- We can use **PHP basics** for our **WP project**



Questions?



SoftUni Diamond Partners



**SUPER
HOSTING
.BG**



VIVACOM

THE CROWN IS YOURS

INDEAVR
Serving the high achievers



- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg, about.softuni.bg
- Software University Foundation
 - softuni.foundation
- Software University @ Facebook
 - facebook.com/SoftwareUniversity



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg/>
- © Software University – <https://softuni.bg>

