

# PHP

## PHP Introduction

- "PHP: Hypertext Preprocessor"
- Open source scripting language

->A scripting language or script language is a programming language that is used to manipulate, customize, and automate the facilities of an existing system.

- PHP scripts are executed on the server
- PHP files have extension ".php"

## What Can PHP Do?

- 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

## Syntax

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

## Comments

```
<?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
    */

    $x = 5 /* + 15 */ + 5;

?>
```

## Variable

In PHP, a variable starts with the \$ sign, followed by the name of the variable:

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

## Variable scopes

PHP has three different variable scopes:

- local
- global
- static

## What is an Array?

- An array is a special variable, which can hold more than one value at a time.

Notation `=array();`

- **Indexed arrays** - Arrays with a numeric index
- **Associative arrays** - Arrays with named keys
- **Multidimensional arrays** - Arrays containing one or more arrays

1. `$cars = array("Volvo", "BMW", "Toyota");`

```
$cars[0] = "Volvo";
$cars[1] = "BMW";
$cars[2] = "Toyota";
```

2. `$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");`

```
$age['Peter'] = "35";
$age['Ben'] = "37";
$age['Joe'] = "43";
```

```
3. $cars = array (  
    array("Volvo",22,18),  
    array("BMW",15,13),  
    array("Saab",5,2),  
    array("Land Rover",17,15)  
);
```

## PHP while Loop

The `while` loop executes a block of code as long as the specified condition is true.

### Syntax

```
while (condition is true) {  
    code to be executed;  
}
```

### Example

```
<?php  
$x = 1;  
  
while($x <= 5) {  
    echo "The number is: $x <br>";  
    $x++;  
}  
?>
```

## PHP do...while Loop

The `do...while` loop will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.

### Syntax

```
do {  
    code to be executed;  
} while (condition is true);
```

## Example

```
<?php
$x = 1;

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

## Switch Case

Use the `switch` statement to **select one of many blocks of code to be executed**.

## Syntax

```
switch (n) {
    case label1:
        code to be executed if n=label1;
        break;
    case label2:
        code to be executed if n=label2;
        break;
    case label3:
        code to be executed if n=label3;
        break;
    ...
    default:
        code to be executed if n is different from all labels;
}
```

```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
```

```
}  
?>
```

## PHP include and require Statements

```
include 'filename';
```

or

```
require 'filename';
```

## PHP 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.

### Syntax

```
function functionName() {  
    code to be executed;  
}
```

### Example

```
<?php  
function writeMsg() {  
    echo "Hello world!";  
}  
  
writeMsg(); // call the function  
?>
```

## PHP Function Arguments

Information can be passed to functions through arguments. An argument is just like a variable.

Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

```
<?php
function familyName($fname) {
    echo "$fname Refsnes.<br>";
}

familyName("Jani");
familyName("Hege");
familyName("Stale");
familyName("Kai Jim");
familyName("Borge");
?>
```

The following example has a function with two arguments (\$fname and \$year):

```
<?php
function familyName($fname, $year) {
    echo "$fname Refsnes. Born in $year <br>";
}

familyName("Hege", "1975");
familyName("Stale", "1978");
familyName("Kai Jim", "1983");
?>
```