Intro to PHP

D3 Software Engineering

School of Applied Science

Telkom University

2022

What is PHP

- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.
- Developed by Rasmus Lerdorf in 1995
- PHP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server
- PHP is free to download and use

What is a PHP File?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are 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

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Download it from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

PHP Installation – What Do I Need?

To start using PHP, you can:

- Find a web host with PHP and MySQL support
- Install a web server on your own PC, and then install PHP and MySQL
- or alternatively, you can use XAMMP package. Download from this link https://www.apachefriends.org/download.html

Basic PHP Syntax

- A PHP script can be placed anywhere in the document.
- A PHP script starts with **<?php** and ends with **?>**:
- PHP statements end with a semicolon (;).

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

```php
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

My first PHP page

Hello World!

Comments in PHP

- A comment in PHP code is a line that is not read/executed as part of the program. Its only purpose is to be read by someone who is looking at the code.
- PHP supports several ways of commenting:

```
<html>
<body>

<?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
*/

// You can also use comments to leave out parts of a code line
$x = 5 /* + 15 */ + 5;
echo $x;
?>

</body>
```

PHP Case Sensitivity (1)

- In PHP, all keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are NOT case-sensitive.

```
<!DOCTYPE html>
<html>
<body>

<?php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>

</body>
</html>
```

Hello World!
Hello World!
Hello World!

PHP Case Sensitivity (2)

- However; **all variable names are case-sensitive.**

```
<!DOCTYPE html>
<html>
<body>

<?php
$color = "red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLOR . "<br>";
?>

</body>
</html>
```

My car is red
My house is
My boat is

PHP 5 Variables - Creating (Declaring) PHP Variables

- In PHP, a variable starts with the $ sign, followed by the name of the variable
- Think of variables as containers for storing data.
- A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).
- Rules for PHP variables:
  - 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)

Example of variables

```php
<?php
    $i;
    $nama;
    $Umur;
    $_lokasi_memori;
    $ANGKA_MAKSIMUM;
?>
```

```php
<?php
    $nama = "Naufal";
    $umur = 19;
    $pesan = "Saya sedang belajar PHP";
?>
```

PHP is a Loosely Typed Language

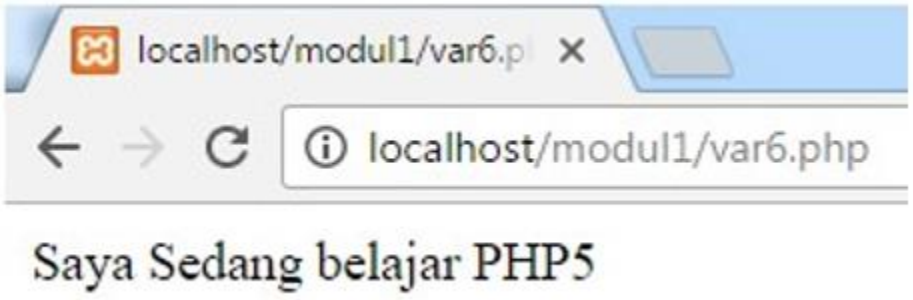
- In the previous example, notice that we did not have to tell PHP which data type the variable is.
- PHP automatically converts the variable to the correct data type, depending on its value.
- In other languages such as C, C++, and Java, the programmer must declare the name and type of the variable before using it.
-

PHP echo and print Statements

- In PHP there are two basic ways to get output: echo and print.
- The differences are small: echo has no return value while print has a return value of 1 so it can be used in expression. Echo is faster than print.

```php
<?php
    $a='Saya Sedang belajar PHP';
    $b=5;

    print $a;
    echo $b;
?>
```



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).
- Syntax

```
define(name, value, case-insensitive)
```

```
<!DOCTYPE html>
<html>
<body>

<?php
// case-sensitive constant name
define("GREETING", "Welcome to W3Schools.com!");
echo GREETING;
?>

</body>
</html>
```

PHP Data Types

- Variables can store data of different types, and different data types can do different things.
- PHP supports the following data types:
  - String
  - Integer
  - Float (floating point numbers - also called double)
  - Boolean
  - Array
  - Object
  - NULL
  - Resource

PHP Integer

- An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647.

```
<!DOCTYPE html>
<html>
<body>

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

</body>
</html>
```

int(5985)

PHP Float

- A float (floating point number) is a number with a decimal point or a number in exponential form.

```
<!DOCTYPE html>
<html>
<body>

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

</body>
</html>
```

float(10.365)

PHP Boolean

- A Boolean represents two possible states: TRUE or FALSE.

```php
<?php
    $benar=true;
    $salah=false;

    echo "benar = $benar, salah = $salah";
    // hasil output: benar = 1, salah =
?>
```

PHP Array

- An array stores multiple values in one single variable.

```
<!DOCTYPE html>
<html>
<body>

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

</body>
</html>
```

array(3) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW" [2]=> string(6) "Toyota" }

PHP Object

- An object is a data type which stores data and information on how to process that data.
- In PHP, an object must be explicitly declared.
- First we must declare a class of object. For this, we use the class keyword. A class is a structure that can contain properties and methods:

```
<html>
<body>

<?php
class Car {
    function Car() {
        $this->model = "VW";
    }
}
// create an object
$herbie = new Car();

// show object properties
echo $herbie->model;
?>

</body>
```

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 (Introduced in PHP 5.6)

Example of arithmetic operator

```php
<?php
    $hasil1= -3;
    $hasil2=3+5;
    $hasil3=8-4.5;
    $hasil4=2*5;
    $hasil5=3+8/5-3;
    $hasil6=10 % 4;

    echo "\$hasil1:"; var_dump($hasil1); // $hasil1:int(-3)
    echo "<br \>";
    echo "\$hasil2:"; var_dump($hasil2); // $hasil2:int(8)
    echo "<br \>";
    echo "\$hasil3:"; var_dump($hasil3); // $hasil3:float(3.5)
    echo "<br \>";
    echo "\$hasil4:"; var_dump($hasil4); // $hasil4:int(10)
    echo "<br \>";
    echo "\$hasil5:"; var_dump($hasil5); // $hasil5:float(1.6)
    echo "<br \>";
    echo "\$hasil6:"; var_dump($hasil6); // $hasil6:int(2)
?>
```

PHP String Operators

Operator	Name	Example	Result
.	Concatenation	$txt1 . $txt2	Concatenation of $txt1 and $txt2
.=	Concatenation assignment	$txt1 .= $txt2	Appends $txt2 to $txt1

```
1   <!DOCTYPE html>
2   <html>
3   <body>
4
5   <?php
6   $txt1 = "Hello";
7   $txt2 = " world!";
8   echo $txt1 . $txt2;
9
10  echo "<br>";
11
12  $txt1 = "Hello";
13  $txt2 = " world!";
14  $txt1 .= $txt2;
15  echo $txt1;
16
17  ?>
18
19  </body>
20  </html>
```

Hello world!
Hello world!

PHP Logical Operators

Operator	Name	Example	Result
and	And	$x and $y	True if both $x and $y are true
or	Or	$x or $y	True if either $x or $y is true
xor	Xor	$x xor $y	True if either $x or $y is true, but not both
&&	And	$x && $y	True if both $x and $y are true
||	Or	$x || $y	True if either $x or $y is true
!	Not	!$x	True if $x is not true

```
1 - Hello World!
2 - Hello World!
```

```php
<!DOCTYPE html>
<html>
<body>

<?php
$x = 100;
$y = 50;

if ($x == 100 and $y == 50) {
    echo "1 - Hello World!";
}
echo "<br>";

if ($x == 100 or $y == 50) {
    echo "2 - Hello World!";
}
echo "<br>";

if ($x == 100 && $y == 30) {
    echo "3 - Hello World!";
}
echo "<br>";

?>

</body>
</html>
```

Comparison Operators

Contoh	Nama Operator	Hasil
$a == $b	Sama dengan	**TRUE** jika $a sama dengan $b (tanpa melihat tipe data)
$a === $b	Identik dengan	**TRUE** jika $a sama dengan $b, dan memiliki tipe data yang sama
$a != $b	Tidak sama dengan	**TRUE** jika $a tidak sama dengan $b (tanpa melihat tipe data)
$a <> $b	Tidak sama dengan	**TRUE** jika $a tidak sama dengan $b (tanpa melihat tipe data)
$a !== $b	Tidak identik dengan	**TRUE** jika $a tidak sama dengan $b, dan memiliki tipe data yang tidak sama
$a < $b	Kurang dari	**TRUE** jika $a kurang dari $b
$a > $b	Lebih dari	**TRUE** jika $a lebih dari $b
$a <= $b	Kurang dari atau sama dengan	**TRUE** jika $a kurang dari atau sama dengan $b
$a >= $b	Lebih dari atau sama dengan	**TRUE** jika $a lebih dari atau sama dengan $b

```php
<?php
    echo "1. 12 < 14 = "; var_dump(12<14); // bool(true)
    echo "<br />";
    echo "2. 14 < 14 = "; var_dump(14<14); // bool(false)
    echo "<br />";
    echo "3. 14 <= 14 = "; var_dump(14<=14); // bool(true)
    echo "<br />";
    echo "4. 10 <> '10' = "; var_dump(10<>'10'); // bool(false)
    echo "<br />";
    echo "5. 10 == '10' = "; var_dump(10=='10'); // bool(true)
    echo "<br />";
    echo "6. 10 === '10' = "; var_dump(10==='10'); // bool(false)
    echo "<br />";
    echo "7. '150' == '1.5e2' = "; var_dump('150'=='1.5e2'); // bool(true)
    echo "<br />";
    echo "8. 'duniailkom' == 0 = "; var_dump('duniailkom'==0); // bool(true)
    echo "<br />";
?>
```

PHP Increment / Decrement Operators

Operator	Name	Description
++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

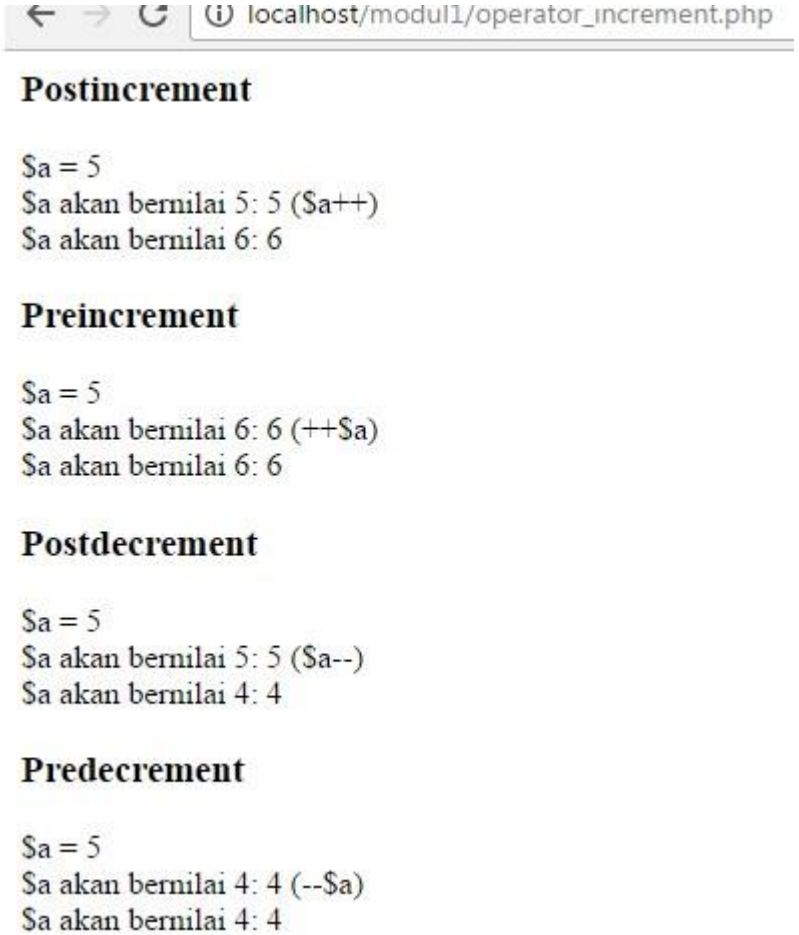
Increment and Decrement Example

```php
<?php
    echo "<h3>Postincrement</h3>";
    $a = 5;
    echo "\$a = $a <br />";
    echo "\$a akan bernilai 5: " . $a++ . " (\$a++)<br />";
    echo "\$a akan bernilai 6: " . $a . "<br />";

    echo "<h3>Preincrement</h3>";
    $a = 5;
    echo "\$a = $a <br />";
    echo "\$a akan bernilai 6: " . ++$a . " (++\$a)<br />";
    echo "\$a akan bernilai 6: " . $a . "<br />";

    echo "<h3>Postdecrement</h3>";
    $a = 5;
    echo "\$a = $a <br />";
    echo "\$a akan bernilai 5: " . $a-- . " (\$a--)<br />";
    echo "\$a akan bernilai 4: " . $a . "<br />";

    echo "<h3>Predecrement</h3>";
    $a = 5;
    echo "\$a = $a <br />";
    echo "\$a akan bernilai 4: " . --$a . " (--\$a)<br />";
    echo "\$a akan bernilai 4: " . $a . "<br />";
?>
```



Reference

- https://www.w3schools.com/php/
- Modul Praktikum Pemrograman Web