E-Commerce

2 SKS | Semester 7 | UNIKOM

Lesson 2

Nizar Rabbi Radliya

nizar@email.unikom.ac.id

PHP : Variable, Constant, Data Type, Operator

Universitas Komputer Indonesia | 2017

PHP Variables

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)

Creating (Declaring) PHP Variables

```
<?php
$merk  = “Adidas Superstar”;
$price = 800000;
?>
```

Output Variables

```
<?php
$merk  = “Adidas Superstar”;
$price = 800000;

echo “Nama Produk = $merk”;
echo <br>;
echo “Harga Produk = Rp. ”. number_format($price, 0 , '' , ',' );
?>
```

PHP Constants

- Constants are like variables except that once they are defined they cannot be changed.
- A valid constant name starts with a letter or underscore (no $ sign before the constant name).

Syntax Constants

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

Parameters:

- name: Specifies the name of the constant
- value: Specifies the value of the constant
- case-insensitive: Specifies whether the constant name should be case-insensitive. Default is false

Creating (Declaring) PHP Constants

```
<?php
define ("PI", 3.14);
echo PI;
?>
```

Creating (Declaring) PHP Constants

```
<?php
define ("PI", 3.14, true);
echo pi;
?>
```

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

The PHP ***`var_dump()`*** *function returns the data type and value*

PHP Operators

Operators are used to perform operations on variables and values.

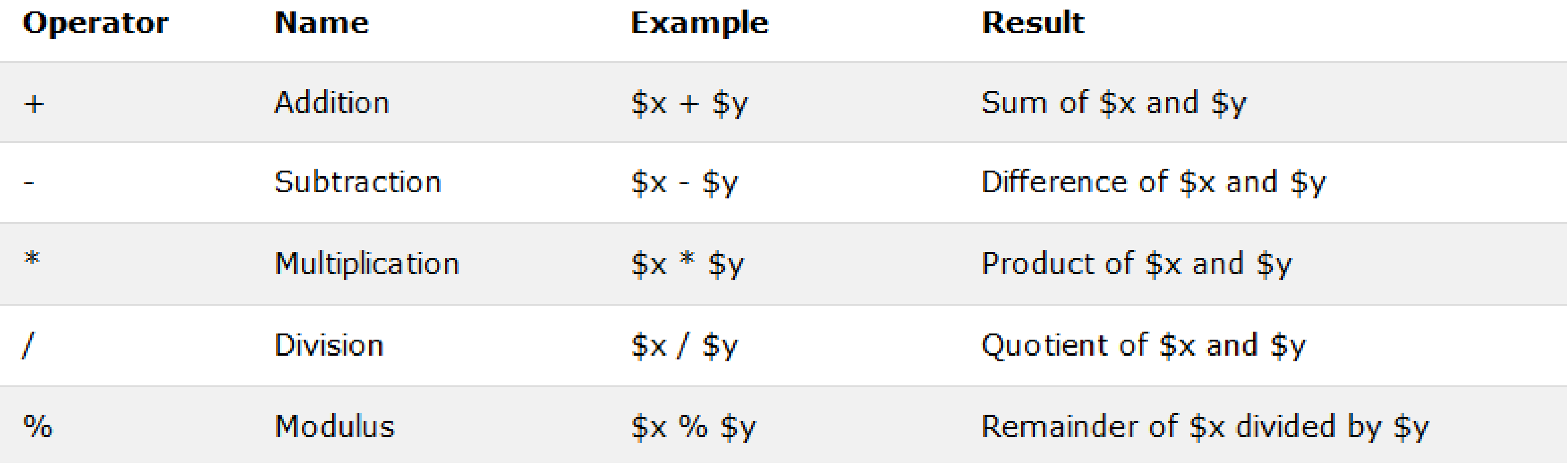
PHP divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Increment/Decrement operators
- Logical operators
- String operators
- Array operators

Learn more: https://www.w3schools.com/php/php_operators.asp

PHP Arithmetic Operators

The PHP arithmetic operators are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.



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

Syntax Arithmetic Operators

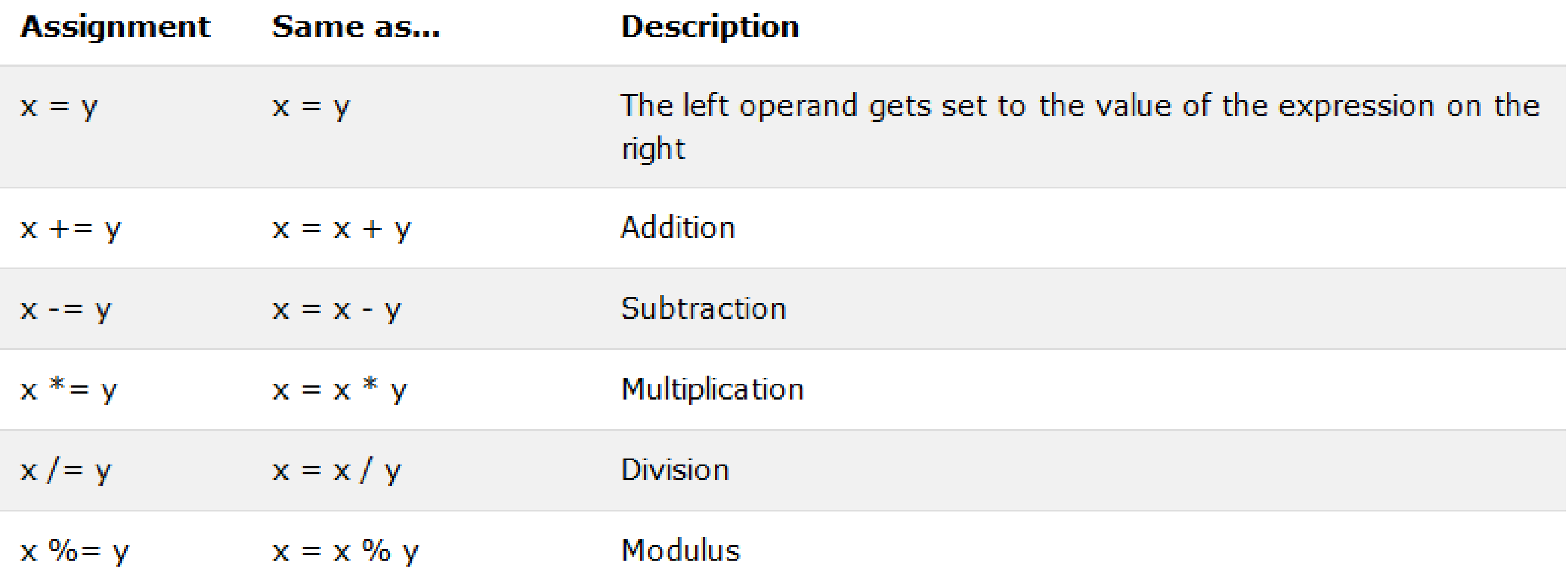
```
<?php
$harga_satuan       = 15000;
$jumlah_barang      = 2;

$harga_bayar        = $harga_satuan * $jumlah_barang;

echo “Harga Bayar : ”. $harga_bayar;
?>
```

PHP Assignment Operators

The PHP assignment operators are used with numeric values to write a value to a variable.
The basic assignment operator in PHP is "=". It means that the left operand gets set to the value of the assignment expression on the right.



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

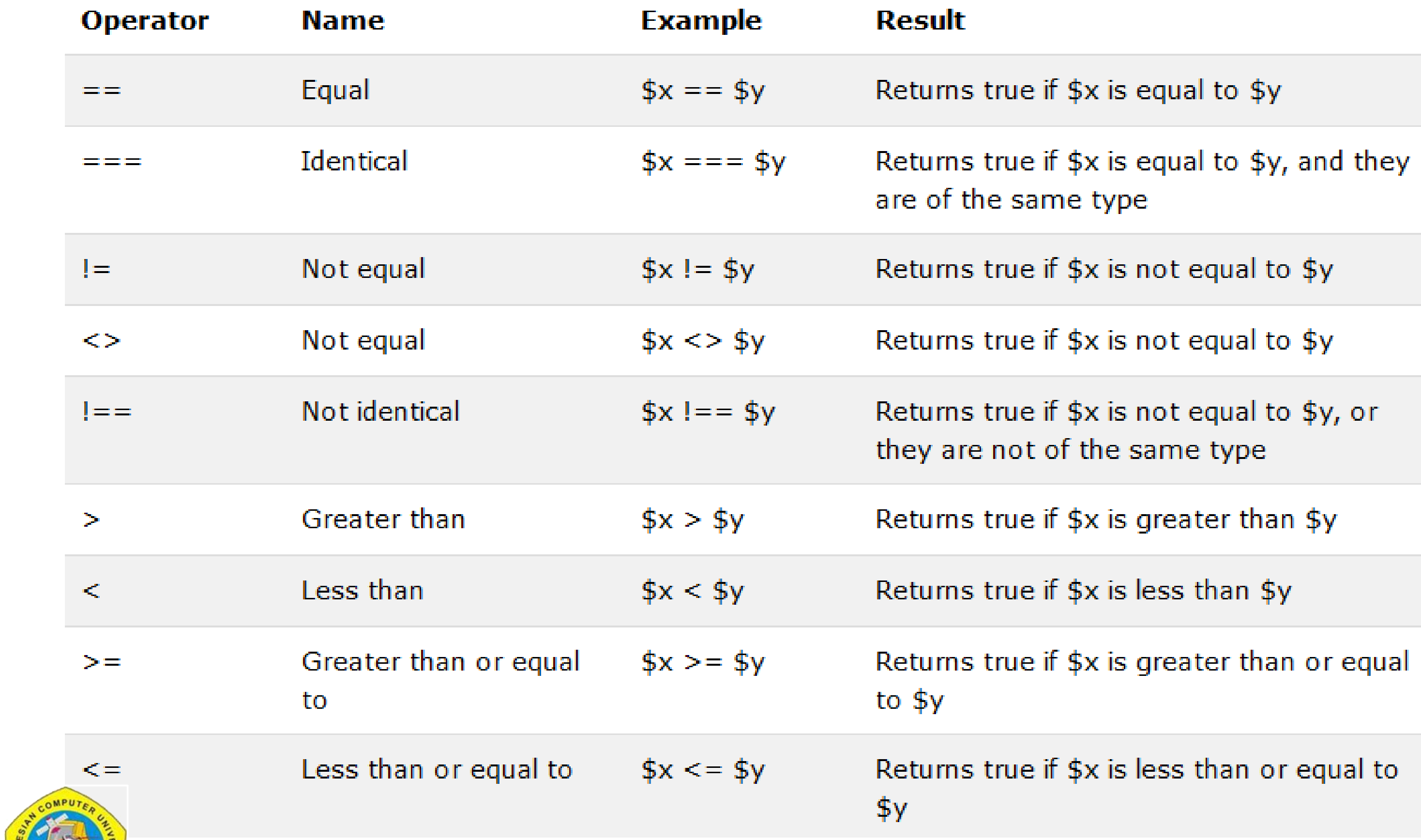
Syntax Assignment Operators

```
<?php
$stock = 20;
$stock += 100;

echo $stock;
?>
```

PHP Comparison Operators

The PHP comparison operators are used to compare two values (number or string).



Operator	Name	Example	Result
==	Equal	$x == $y	Returns true if $x is equal to $y
===	Identical	$x === $y	Returns true if $x is equal to $y, and they are of the same type
!=	Not equal	$x != $y	Returns true if $x is not equal to $y
<>	Not equal	$x <> $y	Returns true if $x is not equal to $y
!==	Not identical	$x !== $y	Returns true if $x is not equal to $y, or they are not of the same type
>	Greater than	$x > $y	Returns true if $x is greater than $y
<	Less than	$x < $y	Returns true if $x is less than $y
>=	Greater than or equal to	$x >= $y	Returns true if $x is greater than or equal to $y
<=	Less than or equal to	$x <= $y	Returns true if $x is less than or equal to $y

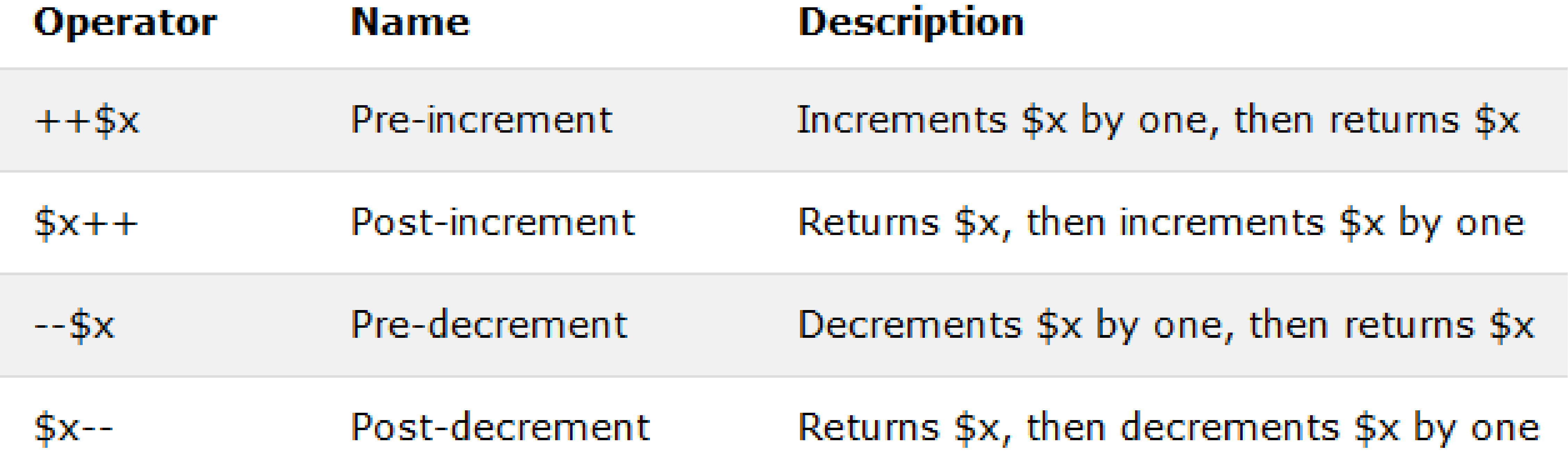
Syntax Comparison Operators

```
<?php
$username1 = “admin”;
$username2 = “admin”;

var_dump($username1 == $username2); // returns true because values are equal
?>
```

PHP Increment / Decrement Operators

The PHP increment operators are used to increment a variable's value.
The PHP decrement operators are used to decrement a variable's value.



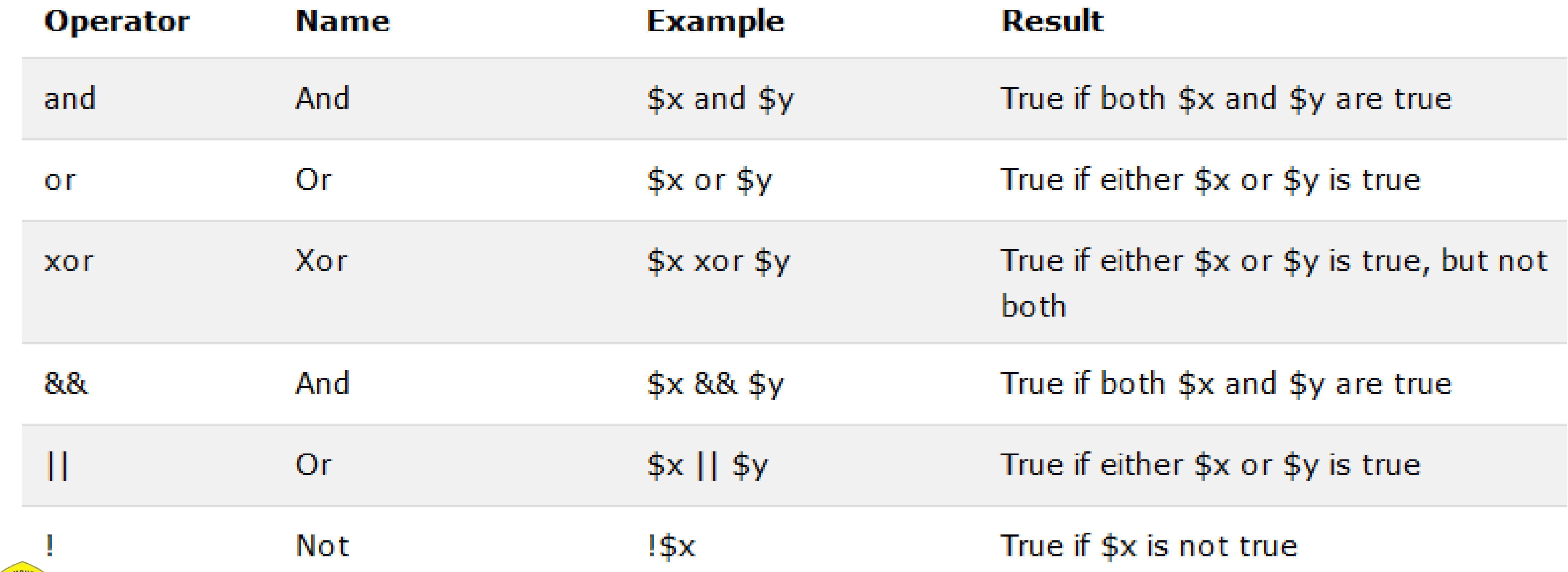
Operator	Name	Description
++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

Syntax Increment / Decrement Operators

```
<?php
$x = 10;
echo ++$x; //result 11
echo “<br/>”;
$x = 10;
echo $x++; //result 10
?>
```

PHP Logical Operators

The PHP logical operators are used to combine conditional statements.



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

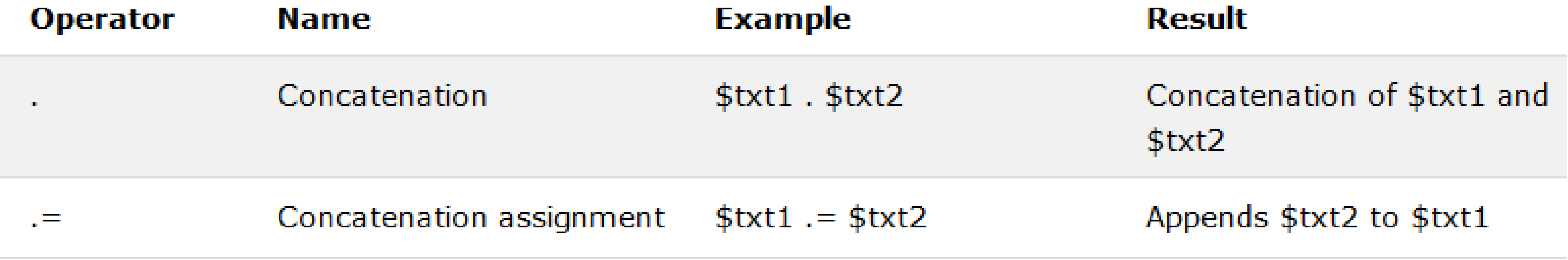
Syntax Logical Operators

```
<?php
$type  = “member”;
$poin  = 100;

if ($type == “member” and $poin == 100) {
    echo “Selamat Anda Mendapatkan Potongan Harga.”;
}
?>
```

PHP String Operators

PHP has two operators that are specially designed for strings.



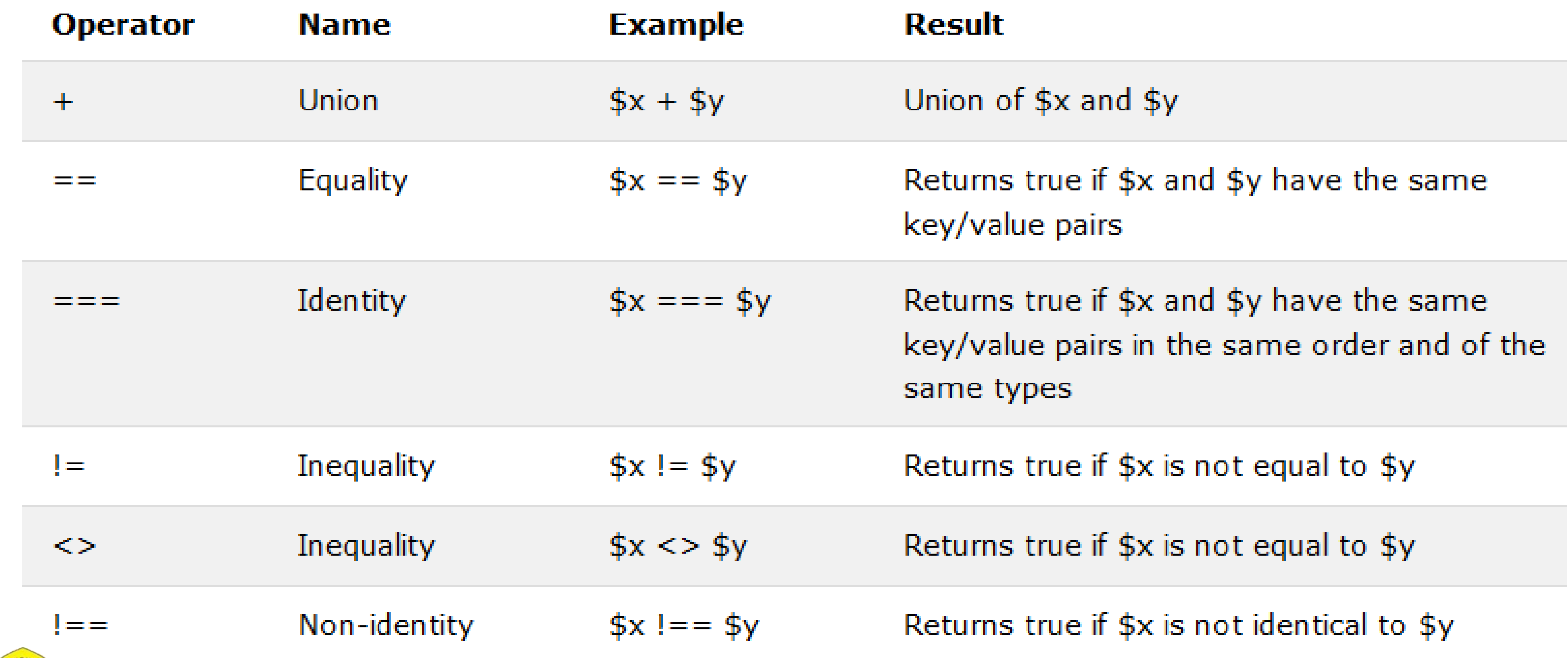
Operator	Name	Example	Result
.	Concatenation	$txt1 . $txt2	Concatenation of $txt1 and $txt2
.=	Concatenation assignment	$txt1 .= $txt2	Appends $txt2 to $txt1

Syntax String Operators

```
<?php
$first_name  = "Budi";
$last_name   = " Raharjo";
$first_name  .= $last_name;
echo $first_name; // result Budi Raharjo
?>
```

PHP Array Operators

The PHP array operators are used to compare arrays.



Operator	Name	Example	Result
+	Union	$x + $y	Union of $x and $y
==	Equality	$x == $y	Returns true if $x and $y have the same key/value pairs
===	Identity	$x === $y	Returns true if $x and $y have the same key/value pairs in the same order and of the same types
!=	Inequality	$x != $y	Returns true if $x is not equal to $y
<>	Inequality	$x <> $y	Returns true if $x is not equal to $y
!==	Non-identity	$x !== $y	Returns true if $x is not identical to $y

Syntax Array Operators

```
<?php
$x = array("a" => "red", "b" => "green");
$y = array("c" => "blue", "d" => "yellow");

print_r($x + $y); // union of $x and $y
?>
```

Case 1

Buatlah program php yang didalamnya terdapat beberapa variable untuk menyimpan data konsumen, sebagai berikut:

Nomor Konsumen	= 101
Nama Konsumen	= Rudi Saputra
Nomor HP	= 082190807066
Email	= rudis@email.com
Saldo	= Rp. 350.000

Semua variable harus ditampilkan pada browser.

Case 2

Buatlah program php untuk menghitung biaya transaksi sebagai berikut;

Daftar barang yang dibeli:

2 buku Pemrograman Web dengan harga 50.000

3 buku Akuntansi dengan harga 70.000

Keterangan: terdapat potongan harga 10% (nilai potongan tidak akan berubah) dari total harga bayar.

Case 1

Buatlah program php yang didalamnya terdapat beberapa variable untuk menyimpan data pengiriman, sebagai berikut:

Nomor Pengiriman	= 312
Jasa Pengiriman	= JNE
Tujuan	= Cimahi
Kode Pos	= 40534
Biaya	= Rp. 7.000

Semua variable harus ditampilkan pada browser.

Case 2

Buatlah program php untuk menghitung biaya pengiriman sebagai berikut;

Daftar pengiriman:

1 unit Laptop Asus (4kg)

1 unit Printer (1kg)

dari Bandung ke Tasikmalaya, ongkos kirim 9.000/kg

Keterangan: terdapat potongan ongkos 10% (nilai potongan tidak akan berubah) dari total ongkos kirim.

NEXT:

PHP Conditional