Desain dan Pemrograman Web

9. PHP

M. Udin Harun Al Rasyid, S.Kom, Ph.D
http://lecturer.eepis-its.edu/~udinharun
udinharun@eepis-its.edu

Table of Contents

What is PHP

- PHP is a powerful tool for making dynamic and interactive Web pages.
- PHP stands for **P**HP: **H**ypertext **P**reprocessor
- PHP is a server-side scripting language, like ASP
- PHP scripts are executed on the server
- PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
- PHP is an open source software, free to download and use.

What is PHP file

- PHP files can contain text, HTML tags and scripts
- PHP files are returned to the browser as plain HTML
- PHP files have a file extension of ".php", ".php3", or ".phtml"

What is MySQL

- MySQL is a database server
- MySQL is ideal for both small and large applications
- MySQL supports standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use

PHP + MySQL

- PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform)
- PHP runs on different platforms (Windows, Linux, Unix, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP is FREE to download from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

Download PHP, MySQL, Apache Server

- Download PHP for free here: http://www.php.net/downloads.php
- Download MySQL for free here: http://www.mysql.com/downloads/
- Download Apache for free here: http://httpd.apache.org/download.cgi

PHP Syntax

- A PHP script always starts with **<?php** and ends with **?>**.
  A PHP script can be placed anywhere in the document.
- On servers with shorthand-support, you can start a PHP script with <? and end with ?>.
- For maximum compatibility, recommended to use the standard form (<?php) rather than the shorthand form.
- A PHP file must have a .php extension.
- A PHP file normally contains HTML tags, and some PHP scripting code.

-
  ```
  <html>
  <body>
  <?php
  echo "Hello World";
  ?>
  </body>
  </html>
  ```
- There are two basic statements to output text with PHP: **echo** and **print**.

PHP Variables

Rules for PHP variable names:

- Variables in PHP starts with a $ sign, followed by the name of the variable
- The variable name must begin with a letter or the underscore character
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- A variable name should not contain spaces
- Variable names are case sensitive (y and Y are two different variables)

Creating (Declaring) PHP Variables

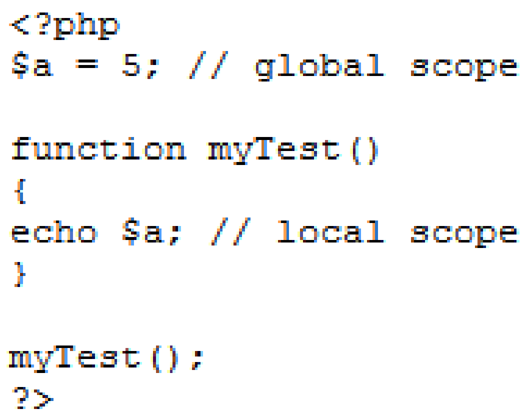
- $myCar="Volvo";
- After the execution of the statement above, the variable **myCar** will hold the value **Volvo**.

PHP Variable Scope

- PHP has four different variable scopes:
  - local
  - global
  - static
  - parameter

Local Scope

- A variable declared **within** a PHP function is local and can only be accessed within that function.

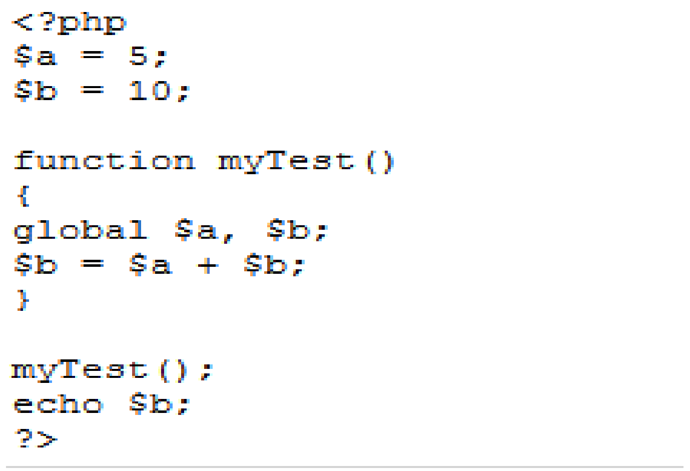


```
<?php
$a = 5; // global scope

function myTest()
{
echo $a; // local scope
}

myTest();
?>
```

- The script above will not produce any output because the echo statement refers to the local scope variable $a, which has not been assigned a value within this scope.

Global Scope

- Global scope refers to any variable that is defined outside of any function.
- Global variables can be accessed from any part of the script that is not inside a function.
- To access a global variable from within a function, use the **global** keyword.



```
<?php
$a = 5;
$b = 10;

function myTest()
{
global $a, $b;
$b = $a + $b;
}

myTest();
echo $b;
?>
```

- The script above will output 15.

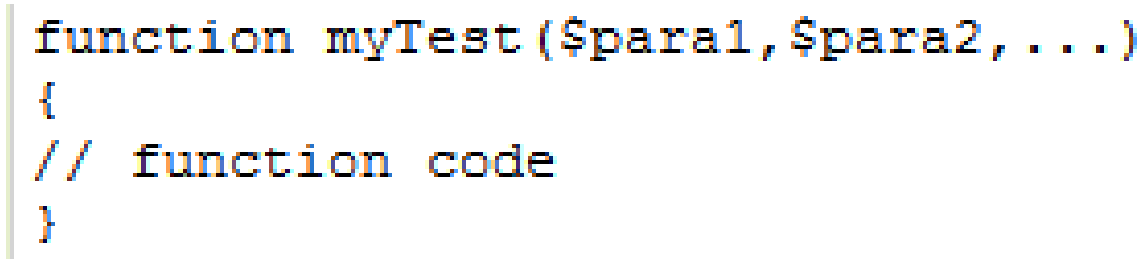
Static Scope

- When a function is completed, all of its variables are normally deleted. However, sometimes you want a local variable to not be deleted.
- To do this, use the **static** keyword when you first declare the variable.

  example: static $rememberMe;

Parameters

- A parameter is a local variable whose value is passed to the function by the calling code.
- Parameters are declared in a parameter list as part of the function declaration

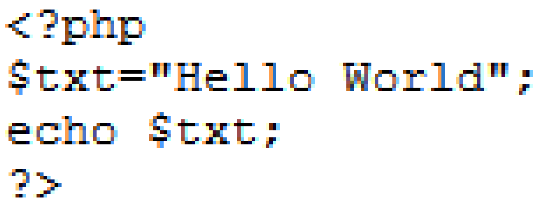


```
function myTest($para1,$para2,...)
{
// function code
}
```

- Parameters are also called arguments.

PHP String Variables

- A string variable is used to store and manipulate text.



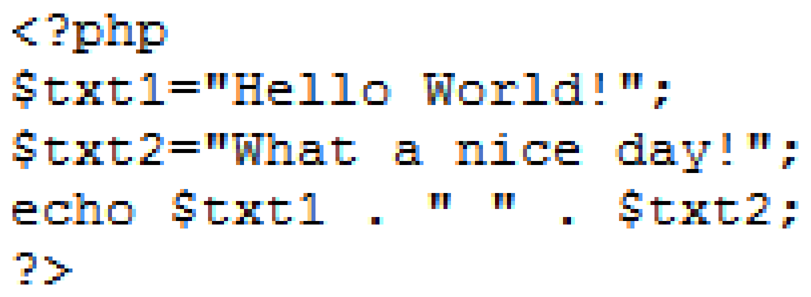
```
<?php
$txt="Hello World";
echo $txt;
?>
```

- The output of the code above will be:

  Hello World

The Concatenation Operator

- The concatenation operator (.) is used to put two string values together.
- To concatenate two string variables together, use the concatenation operator:

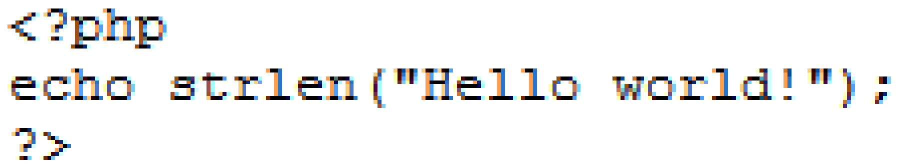


```
<?php
$txt1="Hello World!";
$txt2="What a nice day!";
echo $txt1 . " " . $txt2;
?>
```

- The output of the code above will be:
  Hello World! What a nice day!

The strlen() function

- The strlen() function is used to return the length of a string.



```
<?php
echo strlen("Hello world!");
?>
```

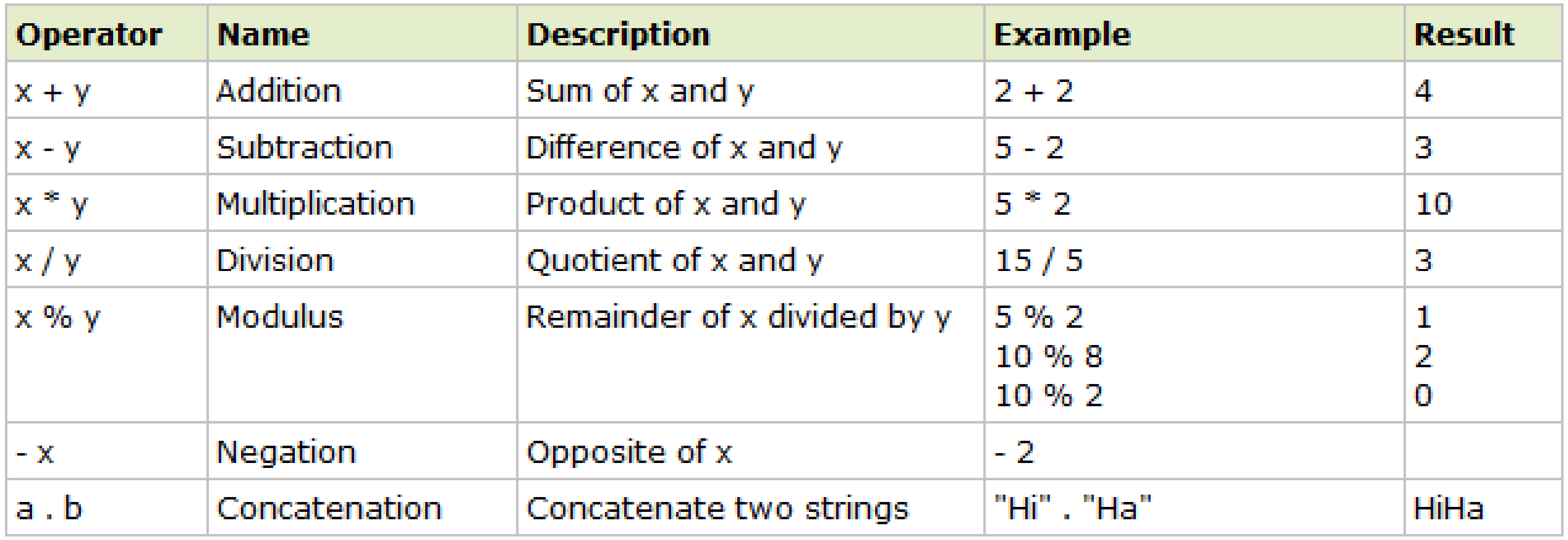
- The output of the code above will be:

  12.

PHP Operators

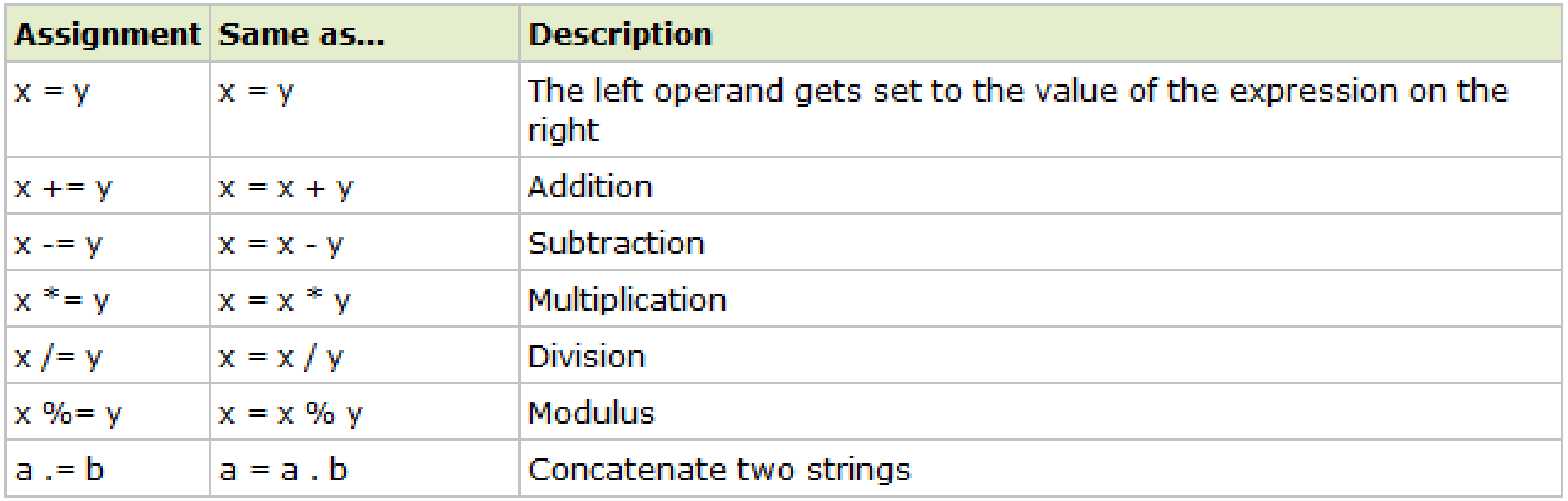
- The assignment operator **=** is used to assign values to variables in PHP.
- The arithmetic operator **+** is used to add values together.

Arithmetic Operators



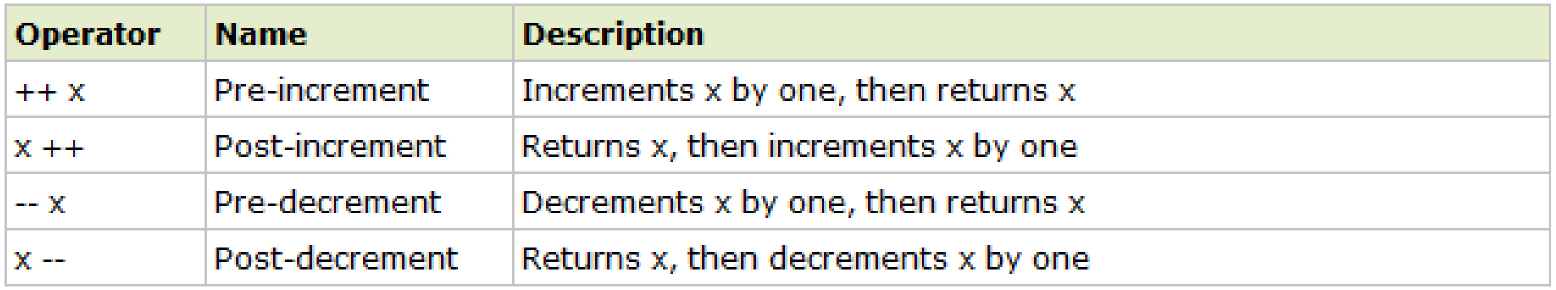
Operator	Name	Description	Example	Result
x + y	Addition	Sum of x and y	2 + 2	4
x - y	Subtraction	Difference of x and y	5 - 2	3
x * y	Multiplication	Product of x and y	5 * 2	10
x / y	Division	Quotient of x and y	15 / 5	3
x % y	Modulus	Remainder of x divided by y	5 % 2 10 % 8 10 % 2	1 2 0
- x	Negation	Opposite of x	- 2	
a . b	Concatenation	Concatenate two strings	"Hi" . "Ha"	HiHa

Assignment Operators



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
a .= b	a = a . b	Concatenate two strings

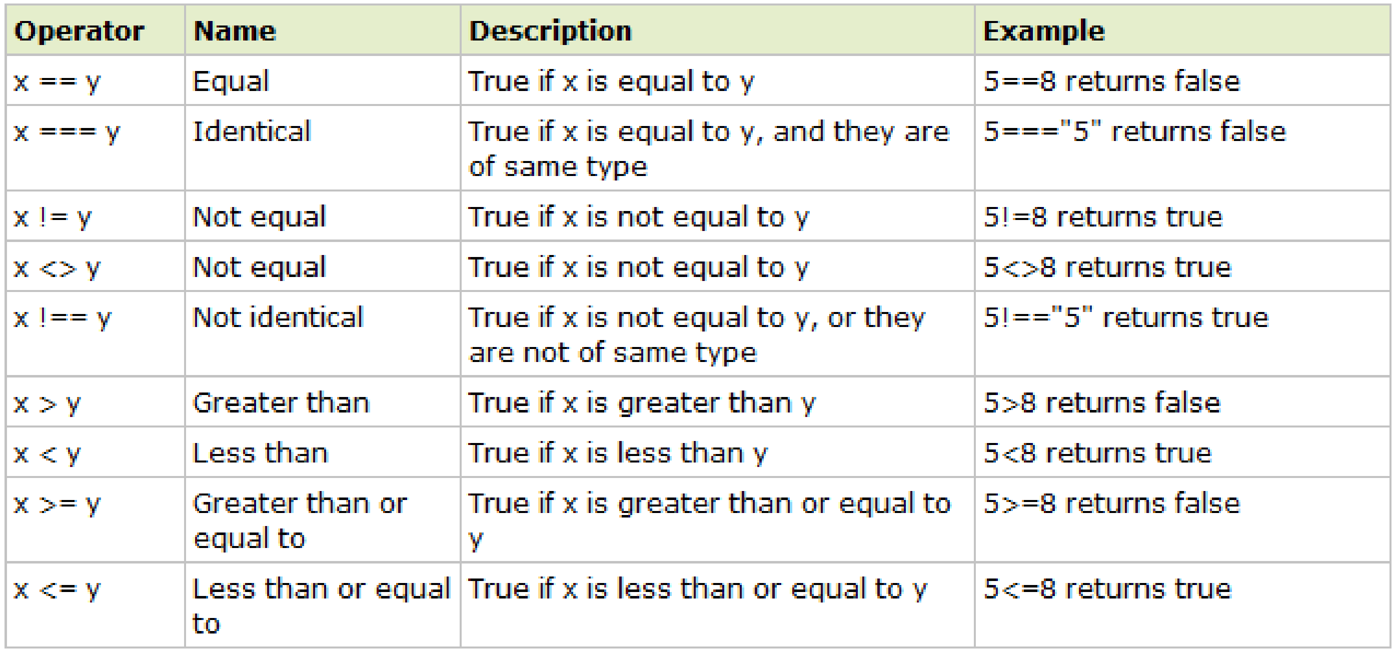
Incrementing/Decrementing Operators



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

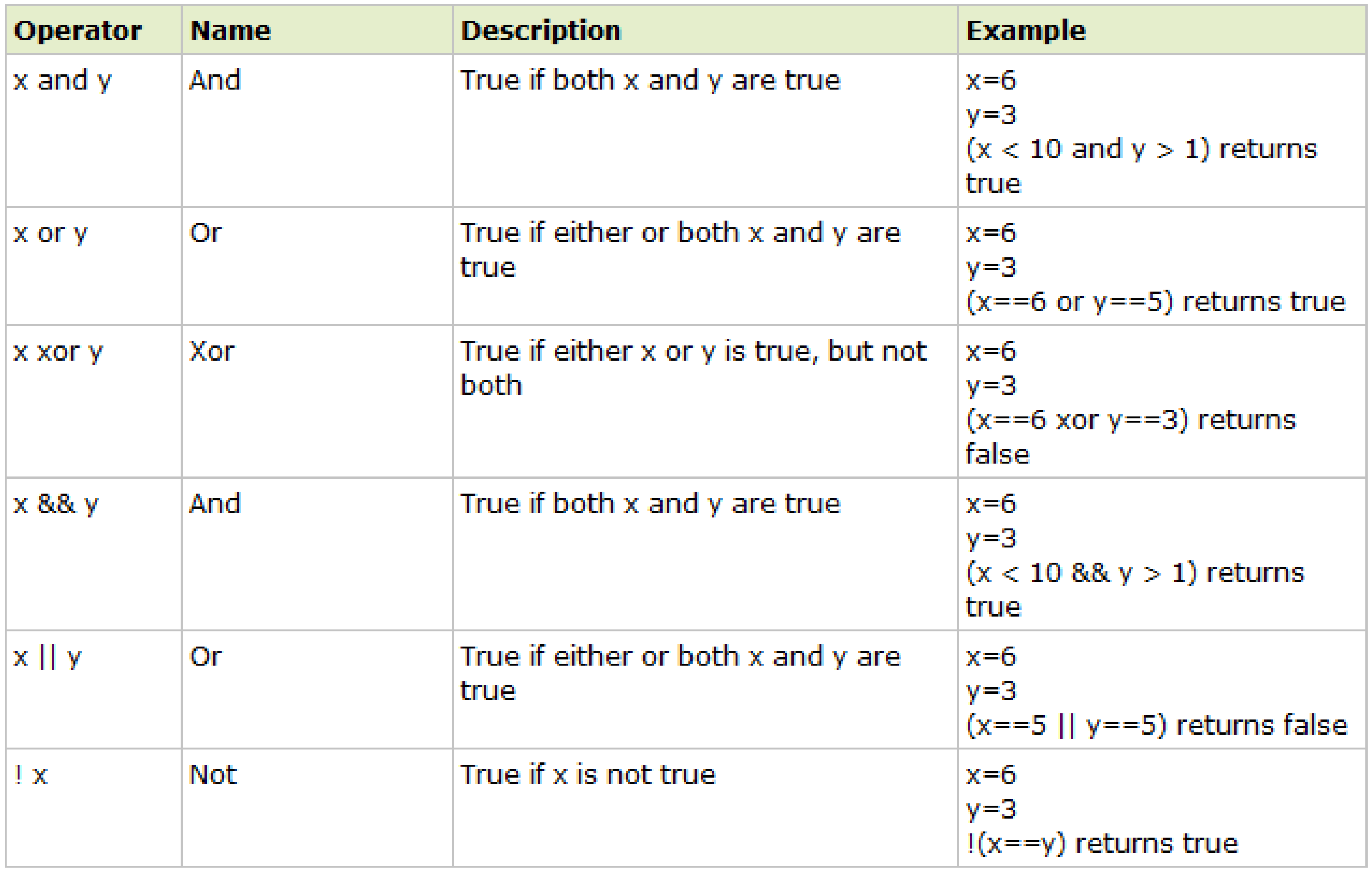
Comparison Operators

- Comparison operators allows you to compare two values



Operator	Name	Description	Example
x == y	Equal	True if x is equal to y	5==8 returns false
x === y	Identical	True if x is equal to y, and they are of same type	5==="5" returns false
x != y	Not equal	True if x is not equal to y	5!=8 returns true
x <> y	Not equal	True if x is not equal to y	5<>8 returns true
x !== y	Not identical	True if x is not equal to y, or they are not of same type	5!=="5" returns true
x > y	Greater than	True if x is greater than y	5>8 returns false
x < y	Less than	True if x is less than y	5<8 returns true
x >= y	Greater than or equal to	True if x is greater than or equal to y	5>=8 returns false
x <= y	Less than or equal to	True if x is less than or equal to y	5<=8 returns true

Logical Operators



Operator	Name	Description	Example

x and y	And	True if both x and y are true	x=6
			y=3
			(x < 10 and y > 1) returns true

x or y	Or	True if either or both x and y are true	x=6
			y=3
			(x==6 or y==5) returns true

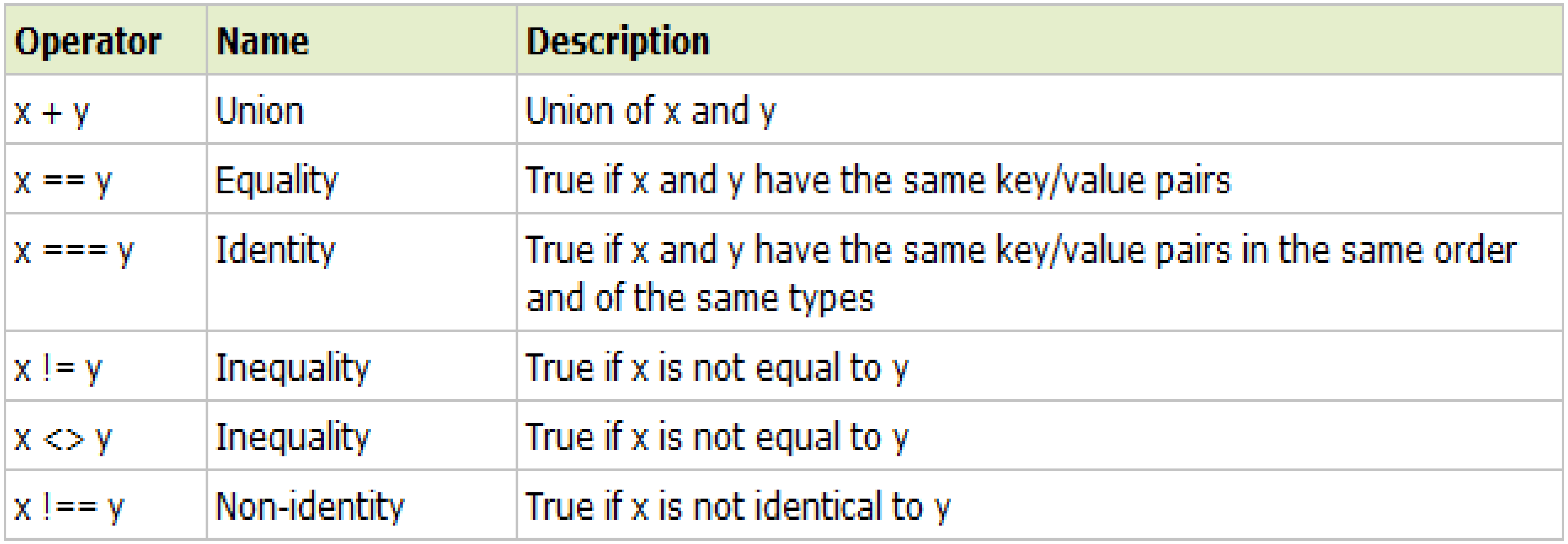
x xor y	Xor	True if either x or y is true, but not both	x=6
			y=3
			(x==6 xor y==3) returns false

x && y	And	True if both x and y are true	x=6
			y=3
			(x < 10 && y > 1) returns true

x || y	Or	True if either or both x and y are true	x=6
			y=3
			(x==5 || y==5) returns false

! x	Not	True if x is not true	x=6
			y=3
			!(x==y) returns true

Array Operators



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

Conditional Statements

In PHP we have the following conditional statements:

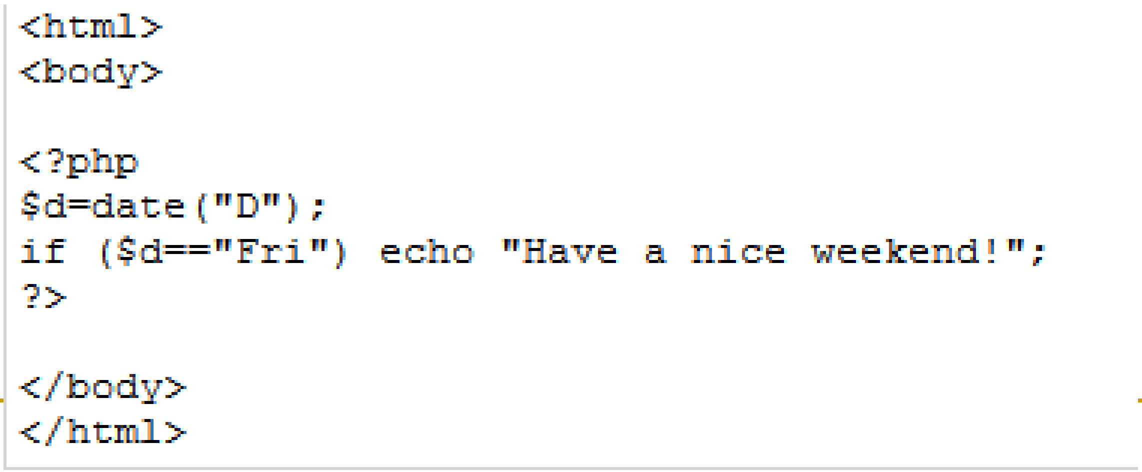
- **if statement** - use this statement to execute some code only if a specified condition is true
- **if...else statement** - use this statement to execute some code if a condition is true and another code if the condition is false
- **if...elseif....else statement** - use this statement to select one of several blocks of code to be executed
- **switch statement** - use this statement to select one of many blocks of code to be executed

The if Statement

- Use the if statement to execute some code only if a specified condition is true.
- Syntax:

  if (*condition*) *code to be executed if condition is true;*

- will output "Have a nice weekend!" if the current day is Friday



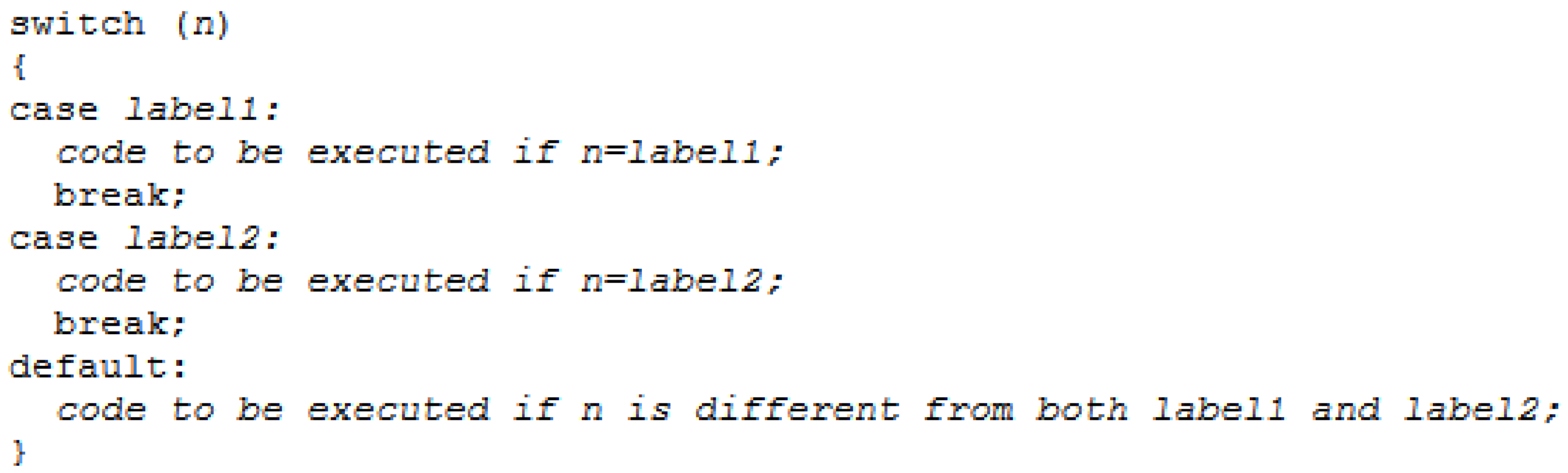
```
<html>
<body>

<?php
$d=date("D");
if ($d=="Fri") echo "Have a nice weekend!";
?>

</body>
</html>
```

- Use **break** to prevent the code from running into the next case automatically. The default statement is used if no match is found.

```
<html>
<body>

<?php
$x=1;
switch ($x)
{
case 1:
   echo "Number 1";
   break;
case 2:
   echo "Number 2";
   break;
case 3:
   echo "Number 3";
   break;
default:
   echo "No number between 1 and 3";
}
?>

</body>
</html>
```

PHP Switch Statement

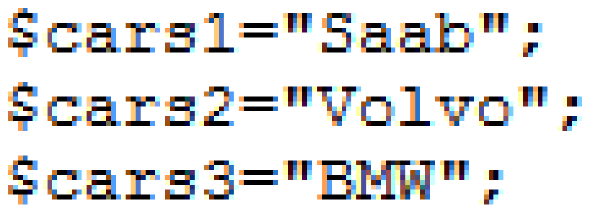
- Conditional statements are used to perform different actions based on different conditions.
- Use the switch statement to select one of many blocks of code to be executed.



```
switch (n)
{
case label1:
   code to be executed if n=label1;
   break;
case label2:
   code to be executed if n=label2;
   break;
default:
   code to be executed if n is different from both label1 and label2;
}
```

PHP Arrays

- An array is a special variable, which can store multiple values in one single variable.



```
$cars1="Saab";
$cars2="Volvo";
$cars3="BMW";
```

- There are three kind of arrays:
  - **Numeric array** - An array with a numeric index
  - **Associative array** - An array where each ID key is associated with a value
  - **Multidimensional array** - An array containing one or more arrays

Numeric Arrays

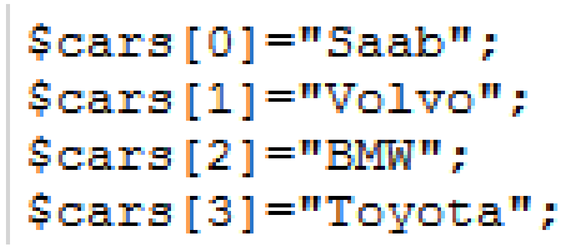
- A numeric array stores each array element with a numeric index.
- There are two methods to create a numeric array.
  - the index are automatically assigned (the index starts at 0).

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

  - assign the index manually.

    ```
    $cars[0]="Saab";
    $cars[1]="Volvo";
    $cars[2]="BMW";
    $cars[3]="Toyota";
    ```

- Example: access the variable values by referring to the array name and index.



```
$cars[0]="Saab";
$cars[1]="Volvo";
$cars[2]="BMW";
$cars[3]="Toyota";
```

- Output:

  Saab and Volvo are Swedish cars.

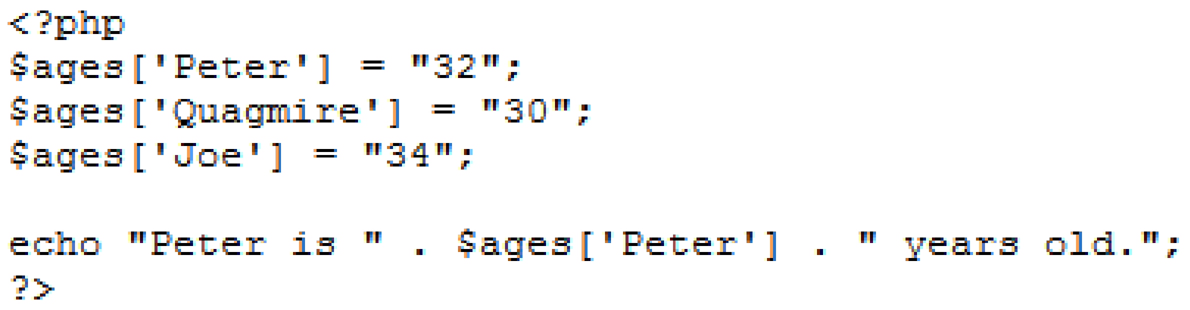
Associative Arrays

- An associative array, each ID key is associated with a value.

- Example 1: an array to assign ages to the different persons.
  $ages = array("Peter"=>32, "Quagmire"=>30, "Joe"=>34);

- Example 2: shows a different way of creating the array.
  $ages['Peter'] = "32";
  $ages['Quagmire'] = "30";
  $ages['Joe'] = "34";

- The ID keys can be used in a script.

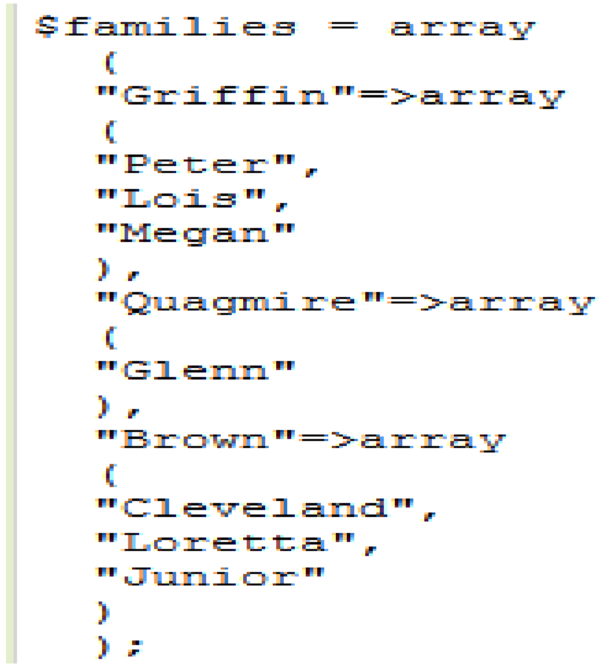


```
<?php
$ages['Peter'] = "32";
$ages['Quagmire'] = "30";
$ages['Joe'] = "34";

echo "Peter is " . $ages['Peter'] . " years old.";
?>
```

- Output: Peter is 32 years old.

Multidimensional Arrays

- In a multidimensional array, each element in the main array can also be an array. And each element in the sub-array can be an array, and so on.
- Create a multidimensional array, with automatically assigned ID keys.



```
$families = array
  (
  "Griffin"=>array
  (
  "Peter",
  "Lois",
  "Megan"
  ),
  "Quagmire"=>array
  (
  "Glenn"
  ),
  "Brown"=>array
  (
  "Cleveland",
  "Loretta",
  "Junior"
  )
  );
```

- The array above is similar to

```
Array
(
[Griffin] => Array
   (
   [0] => Peter
   [1] => Lois
   [2] => Megan
   )
[Quagmire] => Array
   (
   [0] => Glenn
   )
[Brown] => Array
   (
   [0] => Cleveland
   [1] => Loretta
   [2] => Junior
   )
)
```

- Example: Lets try displaying a single value from the array above.

```
echo "Is " . $families['Griffin'][2] .
" a part of the Griffin family?";
```

- Output: Is Megan a part of the Griffin family?

Finish