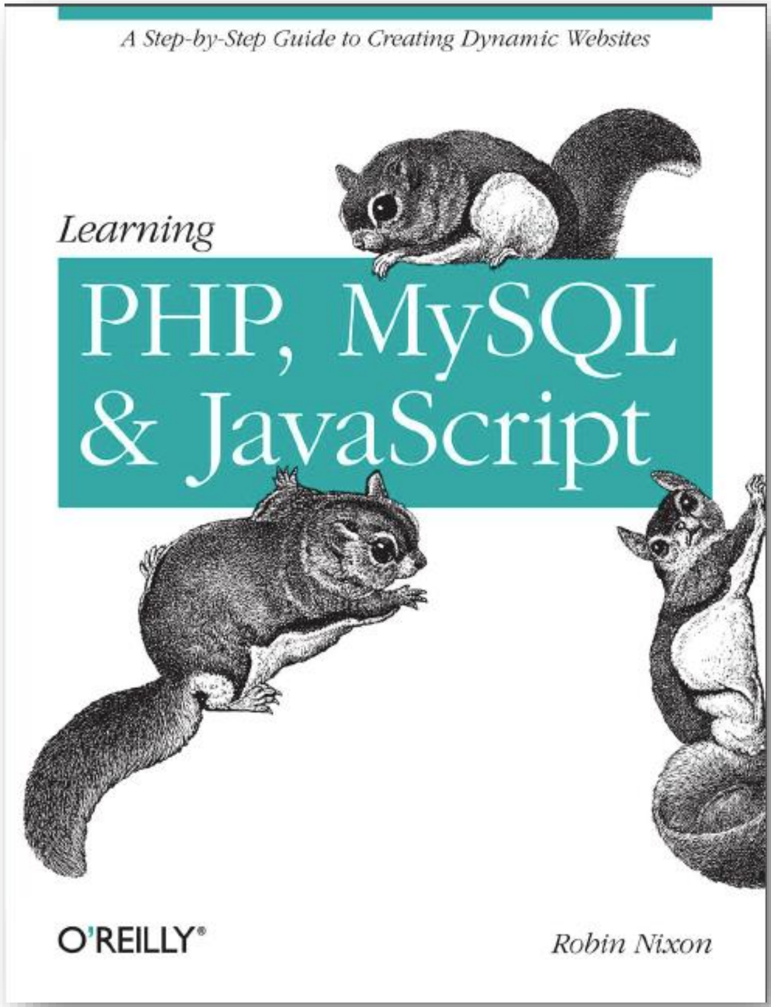
CENG420

# Chapter 5 PHP

Fall 2019 LIU/BIU Dr. Ghantous

CHAPTER 3 & 4 IN



## WHAT IS IT?

- **Server side scripting**
- **Server sends file to a scripting engine before sending it to the user**
- **Very powerful language that can take care of all your operations in a website:**
- **variables**
- **Functions**
- **Database entry and modification**
- **form handling**
- **maintaining sessions and cookies**
- **etc**
- **Similar to the C language syntax and to pearl language (Variables start with $ sign)**

## HOW TO USE IT?

PHP can be embedded in your page along with html and javascript.

NOTE

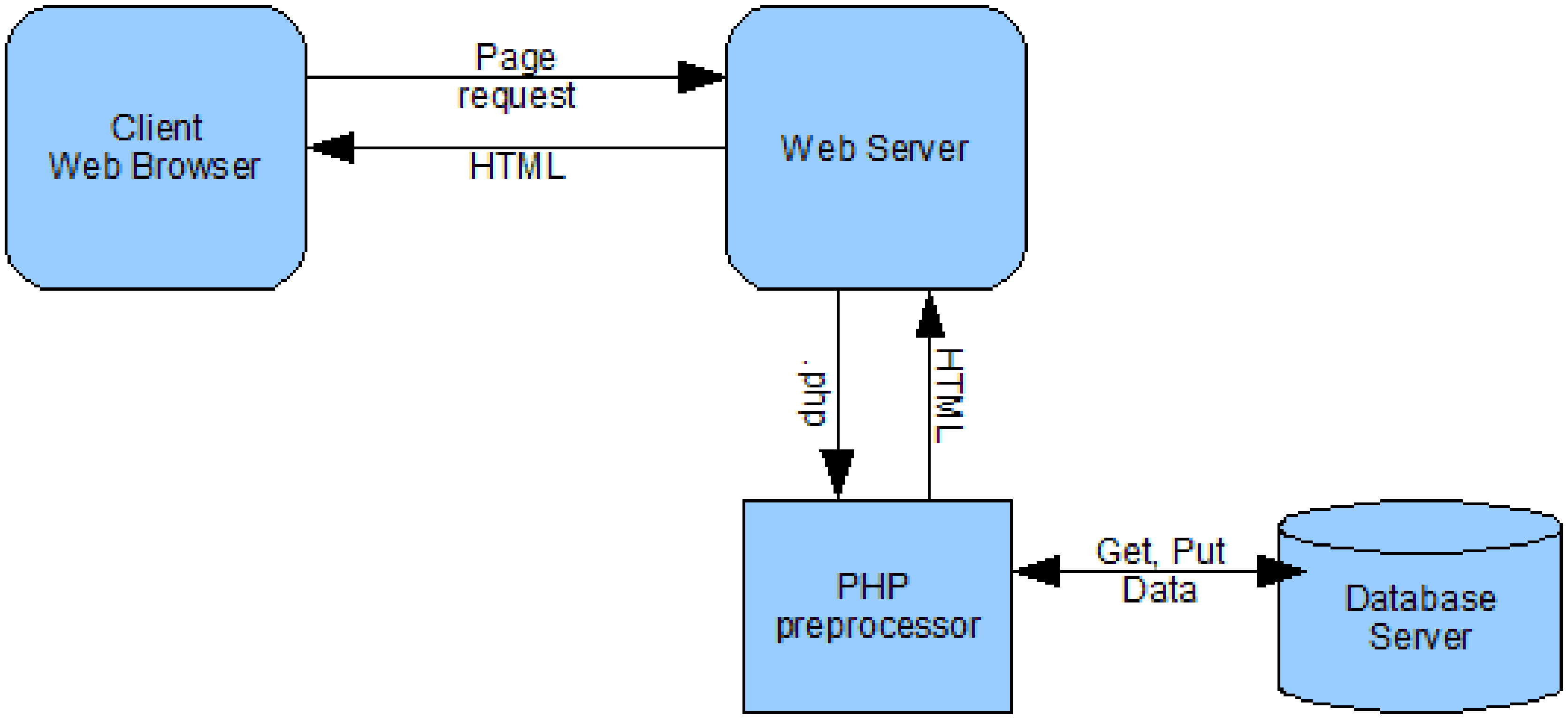
whenever your page contains any PHP, no matter how small the code is, the whole page is considered a php type and should be saved with a .php extension.

## WHERE ARE THEY STORED?

- PHP scripts are stored at the server.
- Whenever the user requests a page, the server executes the script, converts it into html (and JS/CSS if any) and then sends to the browser.
- The browser does't receive any single php code nor it does know how to execute it.

- In normal cases, we place all our website (a collection of html, css, js, and php files) into a folder (and subfolders) on the server.
- In this class, we will place them all in a local server. Mainly we will use XAMPP (or MAMP for mac users)

# TO VISUALIZE THIS



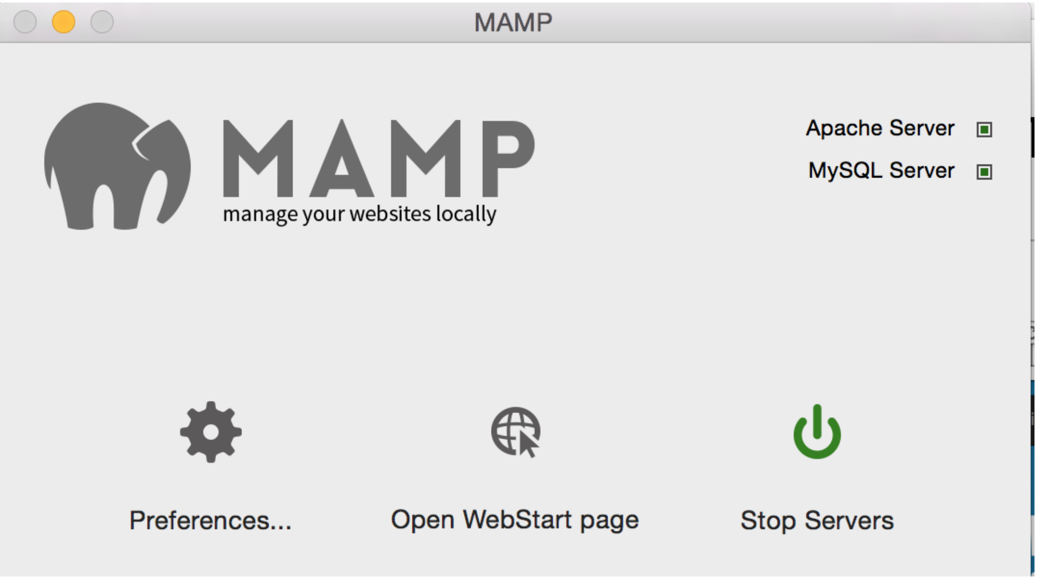
## XAMPP

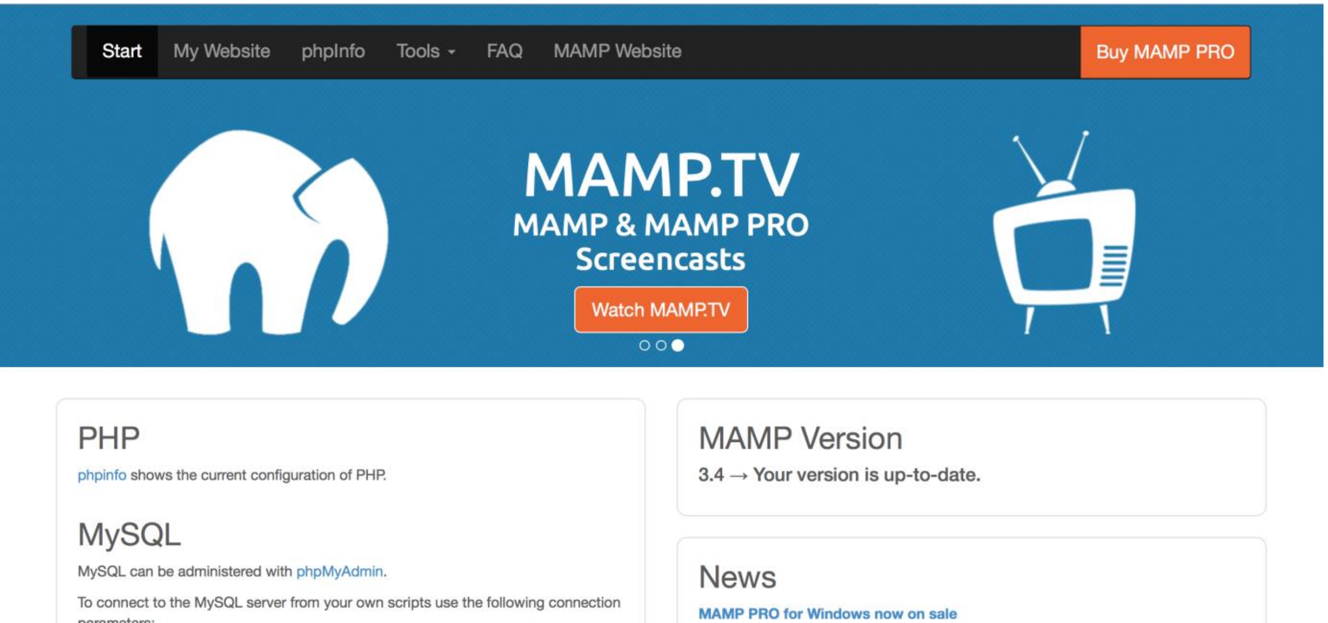
- **XAMPP** is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

- It will be also used to create our databases later and connect to them via PHP.

- Once you install xampp, open it and launch the apache and mysql servers.
- If they fail to launch, it might be a port problem. Apache and vmware virtual macine use the same port. You might need to override and change one port.

# XAMPP/MAMP RUNNING STATE





PHP

phpinfo shows the current configuration of PHP.

MySQL

MySQL can be administered with phpMyAdmin.

To connect to the MySQL server from your own scripts use the following connection

MAMP Version

3.4 → Your version is up-to-date.

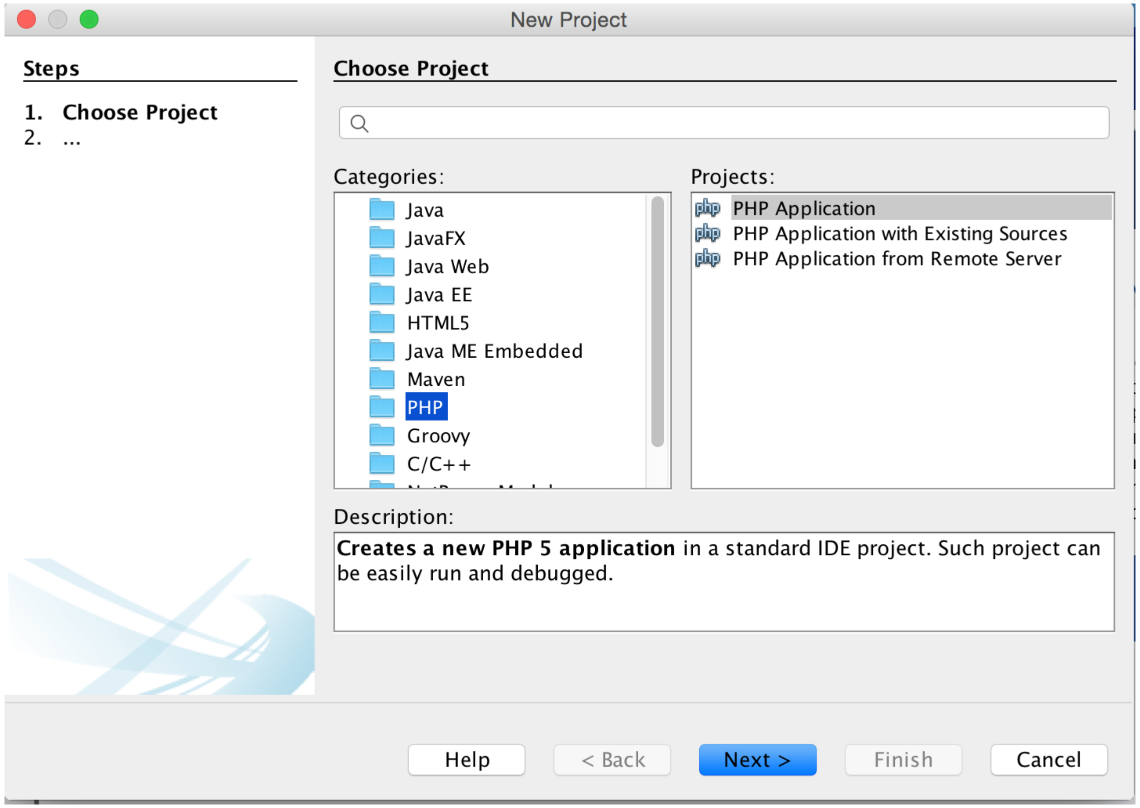
News

MAMP PRO for Windows now on sale

## Creating the project in NetBeans

# IN NETBEANS

- Create a new project of type "PHP".
- Normally, if you have xampp installed, the default folder for saving the files will be xampp/htdocs/*projectname*



New Project

**Steps**

1. **Choose Project**
2. ...

**Choose Project**

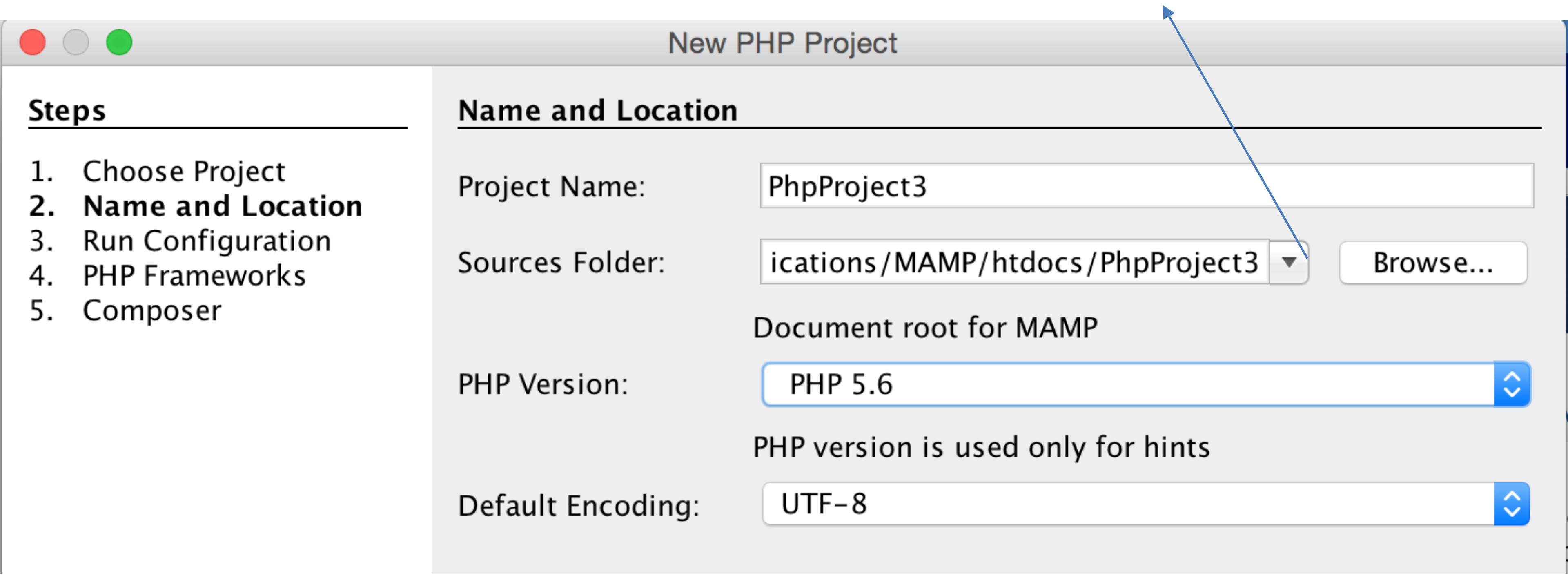
Categories:
- Java
- JavaFX
- Java Web
- Java EE
- HTML5
- Java ME Embedded
- Maven
- PHP
- Groovy
- C/C++

Projects:
- PHP Application
- PHP Application with Existing Sources
- PHP Application from Remote Server

Description:

**Creates a new PHP 5 application** in a standard IDE project. Such project can be easily run and debugged.

## THIS IS A MAC VERSION.  ON WINDOWS, WILL BE XAMPP/HTDOCS…



**Steps**

1. Choose Project
2. **Name and Location**
3. Run Configuration
4. PHP Frameworks
5. Composer

**Name and Location**

Project Name: PhpProject3

Sources Folder: ications/MAMP/htdocs/PhpProject3

Document root for MAMP

PHP Version: PHP 5.6

PHP version is used only for hints

Default Encoding: UTF-8

## N.B.

If you can’t find php type for new projects, you might need to install the plugin from netbeans.

Go to tools-> plugins->available plugins and download the php.

PART 1

## Creating PHP code blocks

## ❶ STANDARD DELIMITERS

```
<?php  statements; ?>
```

- This could be embedded anywhere in your page.

Example:

```
<?php echo “Explore Africa”; ?>
```

- **echo** is similar to print.

## 2 SHORT DELIMITERS

```
<? statements; ?>
```

- Not always supported
- To be safe, use the standard

## EXAMPLE 1

```
<body>

<h1> Multiple Script Sections </h1>
<h2> First Script Section </h2>

<?php echo "<p>Output from the first script section.
</p>"; ?>

<h2> Second Script Section </h2>

<?php echo "<p>Output from the second script section.
</p>"; ?>

</body>
```

## EXAMPLE 2

```
<h1>First Phplesson</h1>
<p>
<?php
      echo "Hi there.";
      $answer = 6 * 7;
      echo "The answer is $answer, what ";
      echo "was the question again?";
  ?>

</p>
<p> Yes another paragraph.</p>
```

## COMMENTS

```
// this is a comment

# this is a comment

/* this is
  All a
  Comment
  */
```

PART 2

## Variables, Arrays, & functions

## VARIABLES

- Case sensitive
- Start with a dollar ($) sign

```
$variable_name = variable_value
$myname = “john”;
$x=$y;
echo $x;
echo “<p>the age is”, $x, “.</p>”;
```

## EXAMPLE 2

```
<?php
$x = "15" + 27;
echo($x);
echo("\n");
?>
```

```
42
```

## ARRAYS

```
$team = array(‘abc’,’def’,’ghi’);
```

```
echo $team[2] //will give ‘ghi’
```

```
$team[] = “xyz”; //this will add to the first available index.
```

## VARIABLES (CONT.)

- ✓ Variable names are case sensitive
- ✓ Operators are the same: +,-,++,--, +=,….,
- ✓ .= (for strings)
- ✓ Comparison operators are the same as well: ==, !=, …
- ✓ Logical operators: &&, ||, !, and, or, xor
- ✓ The and, or and xor have low precedence but used for databases (later)

## STRINGS

The "." operator concatenates strings

```
echo "hello "."I am here"
```
//produces "hello I am here"

```
echo "I have ".$x."apples"
```
//produces I have 5 apples

`$b .= $c` will concatenate c to b.

## LOOK AT THIS

`$info=‘I have $x apples’` → I have $x apples

`$info=“I have $x apples”` → I have 5 apples

The double quotes behaves like “.” operators.

```
$info = ‘My sister\’s car is a \”Mercedes\”.’
$info =‘Date \t Name \t Payment’
```

\t is tab

\n is newline

\r is carriage return

# VARIABLE TYPING AND IMPLICIT CASTING

PHP is similar to JavaScript in terms of implicit casting

```
<? $number=12345*76890;   //= 949207050

echo substr($number,3,1);
// will return 2, $number becomes string.

?>
```

- `Substr` will be covered later.

```
$pi =“3.14”;

$radius=5;

echo $pi*($radius*$radius)  // turns string into number.
```

## CONSTANTS

```
define(“Root_location”,”/usr/local”);
$dir = Root_location;
```

## FUNCTIONS

```
function longdate($timestamp)
{ return date(“l F j S y”, $timestamp);}
Echo longdate( time());
```

// time(), date() are php functions.(will be covered later)

## GLOBAL VARIABLES

```
global $is_logged_in;
```

## EXPLICIT CASTING

```
$speed=“55mph”;
$speed=(int)$speed*2;


echo gettype($x);
echo is_numeric($x);
echo is_string($x);
echo is_int($x);
```

## LOOPS

Same as all others:

- `if() {} else {}`
- `do {} while {}`
- `while() {}`
- `switch() { case: …}`
- `for() {}`

## LOOPS (PART 2)

Additional loops:

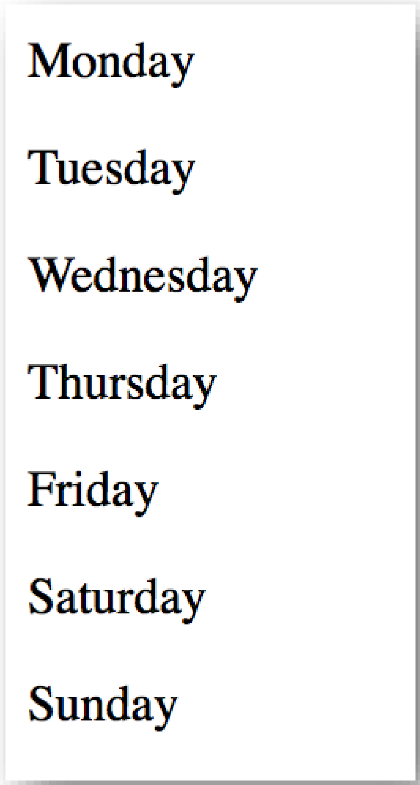
`foreach`: used with arrays

Two syntaxes available:

```
foreach($arrayname as $variablename){}

foreach($Aname as $indexname => $vname){}
```

## foreach SYNTAX 1

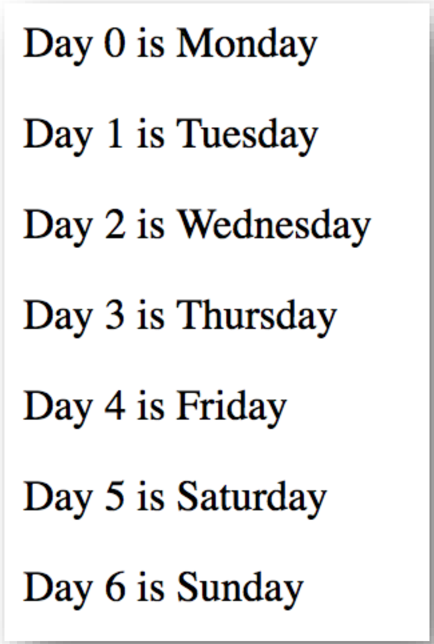
```
$DaysOfWeek =
      array("Monday", "Tuesday",
      "Wednesday”, "Thursday", "Friday",
      "Saturday", "Sunday");
foreach ($DaysOfWeek as $Day)
{
echo "<p>$Day</p>";
}
```



```
Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday
```

## foreach SYNTAX 2

```
foreach ($DaysOfWeek as $DayNumber => $Day)
{
echo "<p>Day $DayNumber is $Day</p>";
}
```



```
Day 0 is Monday

Day 1 is Tuesday

Day 2 is Wednesday

Day 3 is Thursday

Day 4 is Friday

Day 5 is Saturday

Day 6 is Sunday
```

PART 3

## Strings & String functions

## STRINGS

```
$quote=“<p>hello</p>”
echo $quote → hello

$quote=‘<p>”hello”</p>’
echo $quote → “hello”
```

Vice versa:

```
$quote=“<p>’hello’</p>”
echo $quote → ‘hello’
```

## STRINGS

```
$vegetable =“tomato”;
Echo “<p>I have a $vegetable? </p>”
→ I have a tomato?


Echo “<p>I have 5 $vegetables </p>”
→ I have 5
```

(empty because ***$vegetables*** is considered 1 word and the variable is undefined)

# SOLUTION TO THE PREVIOUS PROBLEM

“<p> I have 5 {$vegetable}s </p>”

Or

<p>I have 5 ${vegetable}s </p>

Or

echo "<p> I have 5 ",$vegetable,"s </p>";

CAUTION

```
I have 5 { $vegetable}s
```

Returns

```
I have 5 { tomato}s
```

## STRING FUNCTIONS

You can check http://www.php.net/manual/en/ref.strings.php for complete API.

strlen — Get string length

str_word_count — Return information about words used in a string

ucfirst — Make a string's first character uppercase

lcfirst — Make a string's first character lowercase

strtok — Tokenize string

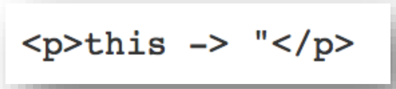
strtolower — Make a string lowercase

strtoupper — Make a string uppercase

## STRING FUNCTIONS

**htmlspecialchars_decode** — Convert special HTML entities back to characters

```
<?php
$str = "<p>this -&gt; &quot;</p>\n";

echo htmlspecialchars_decode($str);
```



```
<p>this -> "</p>
```

**htmlspecialchars** — Convert special characters to HTML entities: ***&*** becomes ***&amp;***, ***<*** becomes ***&lt***

**md5** — Calculate the md5 hash of a string

- used to store passwords in databases for security reasons

**trim** — Strip whitespace (or other characters) from the beginning and end of a string

## STRING FUNCTIONS

**`substr_compare`** — Binary safe comparison of two strings from an offset, up to length characters

**`substr_count`** — Count the number of substring occurrences

**`substr_replace`** — Replace text within a portion of a string

**`substr`** — Return part of a string

## SUBSTR

```
string substr ( string $string , int $start [,
int $length ] )
```

**Start:**

- If **start** is non-negative, the returned string will start at the **start**'th position in **string**, counting from zero. For instance, in the string '*abcdef*', the character at position *0* is '*a*', the character at position *2* is '*c*', and so forth.

- If **start** is negative, the returned string will start at the **start**'th character from the end of **string**.

- If **string** is less than or equal to **start** characters long, FALSE will be returned.

## SUBSTR

```
<?php
$rest = substr("abcdef", -1);     // returns "f"
$rest = substr("abcdef", -2);     // returns "ef"
$rest = substr("abcdef", -3, 1); // returns "d"
?>
```

- **Length:**
- If **length** is given and is positive, the string returned will contain at most **length** characters beginning from **start** (depending on the length of **string**).
- If **length** is given and is negative, then that many characters will be omitted from the end of **string** (after the start position has been calculated when a **start** is negative). If **start** denotes the position of this truncation or beyond, false will be returned.
- If **length** is given and is *0*, FALSE or NULL an empty string will be returned.
- If **length** is omitted, the substring starting from **start** until the end of the string will be returned.

## SUBSTR EXAMPLES

```php
<?php
$rest = substr("abcdef", 0, -1);  // returns "abcde"
$rest = substr("abcdef", 2, -1);  // returns "cde"
$rest = substr("abcdef", 4, -4);  // returns false
$rest = substr("abcdef", -3, -1); // returns "de"
?>
```

# EXAMPLE ON SUBSTR

```
<?php
echo substr('abcdef', 1);     // bcdef
echo substr('abcdef', 1, 3);  // bcd
echo substr('abcdef', 0, 4);  // abcd
echo substr('abcdef', 0, 8);  // abcdef
echo substr('abcdef', -1, 1); // f

// Accessing single characters in a string
// can also be achieved using "square brackets"
$string = 'abcdef';
echo $string[0];              // a
echo $string[3];              // d
echo $string[strlen($string)-1]; // f

?>
```

## SOME ADDITIONAL FUNCTIONS

`strrev` — Reverse a string.

`strpos` — Find the position of the first occurrence of a substring in a string or returns FALSE.

`str_replace` — Replace all occurrences of the search string with the replacement string

`substr_replace` — Replace text within a portion of a string

## EXAMPLES

```
$email = “mghantous@liu.edu.lb”;
$newemail=str_replace(“mghantous”, “miladghantous”, $email);
```

```
$nameend=strpos($email,’@’);
$newemail=substr_replace($email,”miladghantous”, 0,$nameend)
```

## STRING COMPARISONS

`strcmp(“Dan”,”Don”)` returns a value < 0

`strcmp(“Don”, “Dan”)` returns positive value.

`strcmp(“Don”,”Don)` returns 0

Check out:

`Similar_text, levenshtein, strncmp, explode, implode, strsplit`

PART 5

## Array Functions

## ASSOCIATIVE **ARRAYS**

- Instead of using indices (numbers) to access the cells of an array (A[1], A[6]), we can use more descriptive names.
- Example: T[first_name], T[Last_name]
- Think of a database tuple retrieved from a table: we access the attributes by number (first attribute is 0, second is 1, …), or by name ( → Associative)

```php
<?php
$paper['copier'] = "Copier & Multipurpose";
$paper['inkjet'] = "Inkjet Printer";
$paper['laser'] = "Laser Printer";
$paper['photo'] = "Photographic Paper";
echo $paper['laser'];
?>
```

## ASSOCIATIVE ARRAYS (CONT.)

We can also do this:

```
<?php
$p1 = array("Copier", "Inkjet", "Laser", "Photo");

$p2 = array('copier' => "Copier & Multipurpose",
            'inkjet' => "Inkjet Printer",
            'laser' => "Laser Printer",
            'photo' => "Photographic Paper");

echo "p2 element: " . $p2['inkjet'] . "<br>";
?>
```

## PRINT_R

**print_r** — Prints human-readable information about a variable

```
$v = array('a','b','c','d');

print_r($v);
```

```
Array ( [0] => a [1] => b [2] => c [3] => d )
```

```
$movies=array('title' =>
'avatar', 'length'=>'160',
'studio'=>'fox');

print_r($movies);
```

```
Array ( [title] => avatar [length] => 160
[studio] => fox )
```

## IS_ARRAY()

```
echo (is_array($fred)) ? "Is an array" : "Is not an array";
```

## COUNT()

Counts the number of elements in an array.

```
echo count($fred); //single dimensional
echo count($fred, 1); //multi-dimensional
```

# EXPLODE()

take a string containing several items separated by a single character (or string of characters) and then,

place each of these items into an array.

One handy example is to split a sentence up into an array containing all its words

```
<?php
$temp = explode(' ', "This is a sentence with
seven words");
print_r($temp);
?>
```

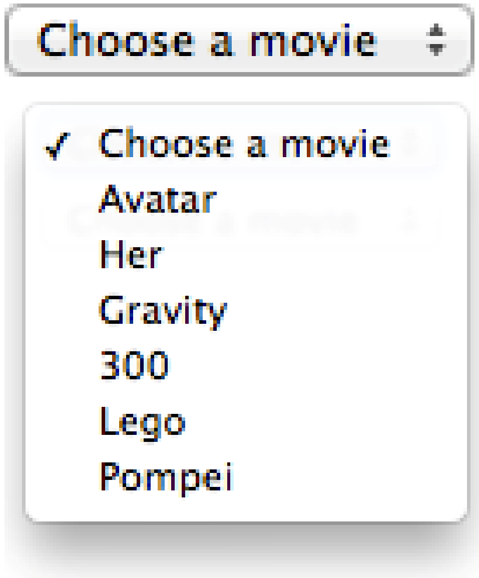
```
Array
(
[0] => This
[1] => is
[2] => a
[3] => sentence
[4] => with
[5] => seven
[6] => words
)
```

```
<?php
$temp = explode('***',
"A***sentence***with***asterisks");
print_r($temp);
?>
```

## PRACTICE

1) Use echo to create an array containing several movies. (in normal cases, this array is a database query answer, but here we will use an array)
2) Create a “SELECT” list populated by these movies.
3) Hint: the <select> syntax is the following:

```
<select>

<option value= -1>choose a movie</option>
<option value=“value1”>some value 1</option>
<option value=“value2”>some value 2</option>
<option value=“value3”>some value 3</option>
</select>
```


Choose a movie
✓ Choose a movie
Avatar
Her
Gravity
300
Lego
Pompei