Introduction to Server-Side Development with PHP

Abdallah Karakra & Sobhi Ahmed

Chapter 8

Objectives

1. Server-Side Development
2. Web Server's Responsabilities
3. Quick Tour of PHP
4. Program Control
5. Functions

Section 1 of 5

WHAT IS SERVER-SIDE DEVELOPMENT

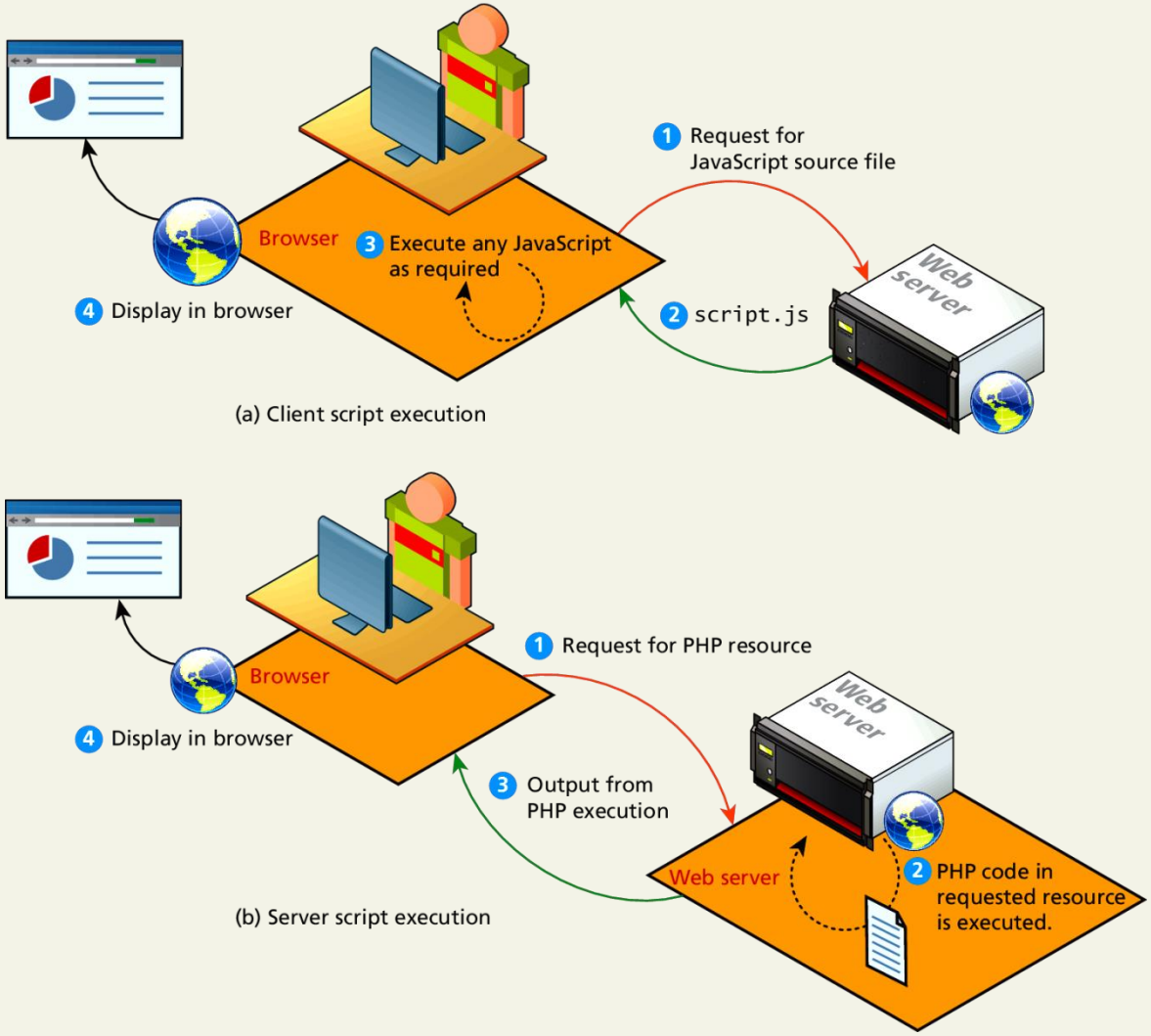
What is Server-Side Development

The basic hosting of your files is achieved through a web server.

Server-side development is much more than web hosting: it involves the use of a programming technology like PHP or ASP.NET to create scripts that dynamically generate content

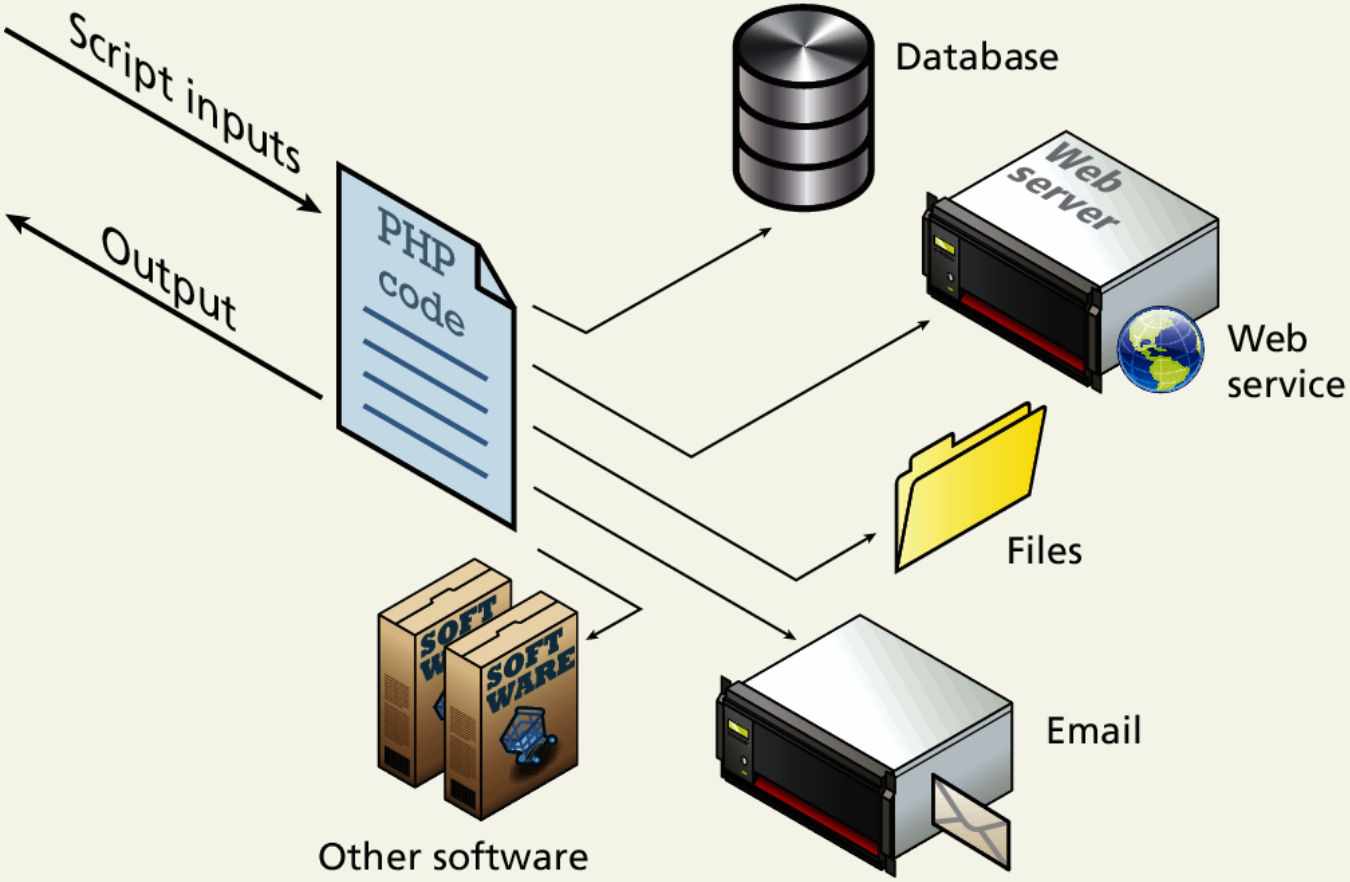
Consider distinction between client side and server side…

Comparing Client and Server Scripts

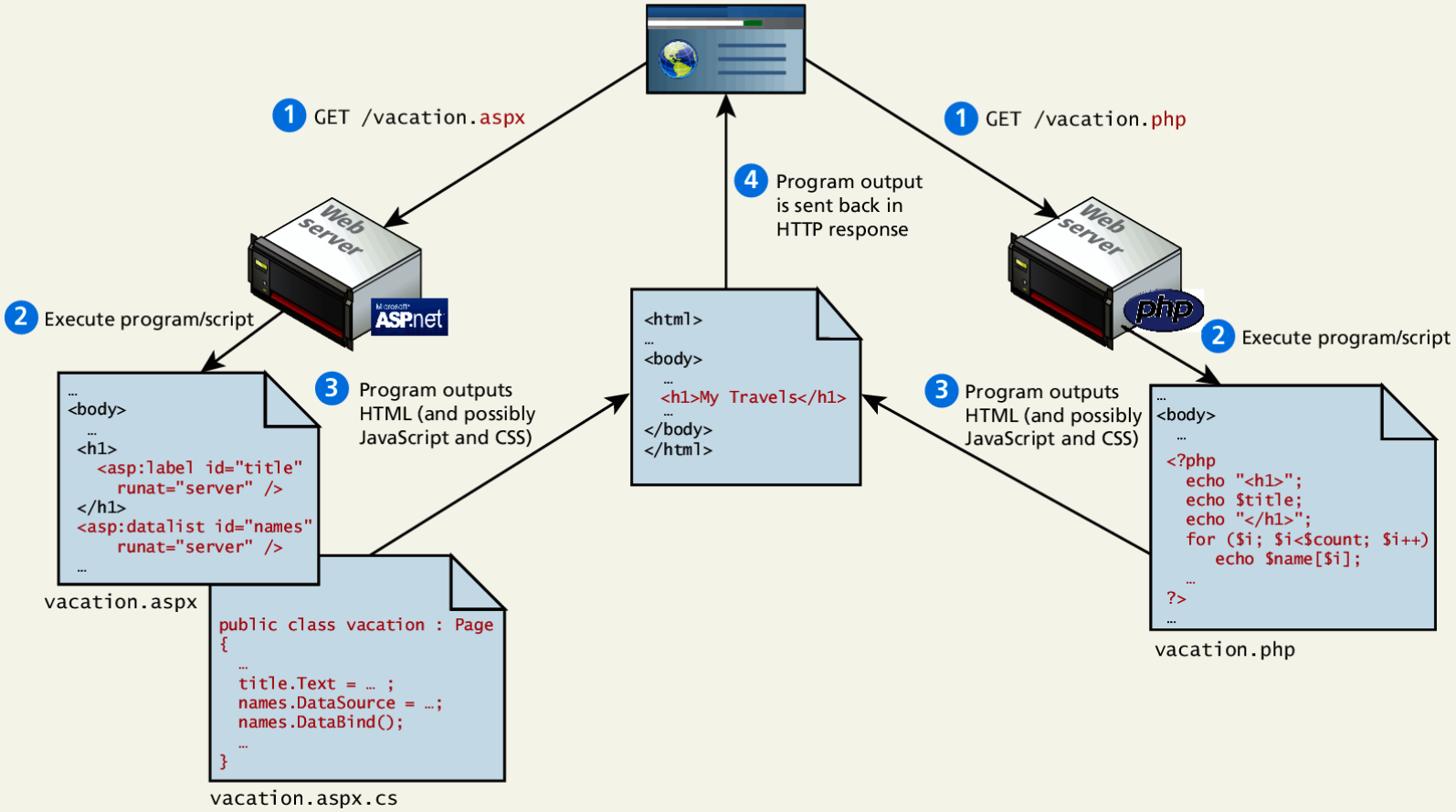


Server-Side Script Resources

So many tools in your kit

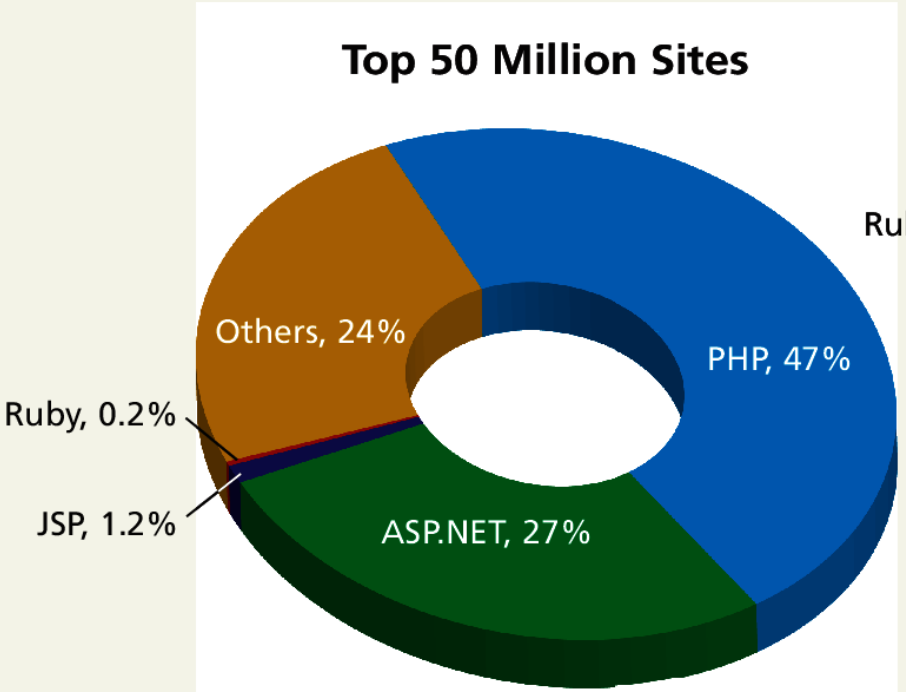


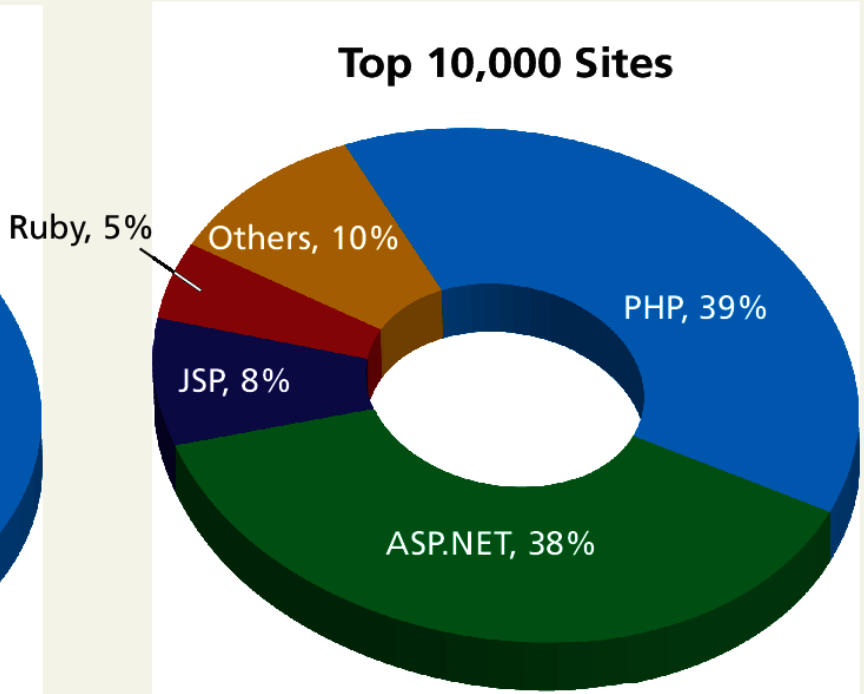
Web Development Technologies



Market Share

Of web development environments





PHP stood for *personal home pages*, although it now is a recursive acronym that means *PHP: Hypertext Processor*.

Server-Side web programming

- Server-side pages are programs written using one of many web programming languages/frameworks
  - examples: PHP, Java/JSP, Ruby on Rails, ASP.NET, Python, Perl

Server-Side web programming

- Web server:
  - contains software that allows it to run server side programs
  - sends back their output as responses to web requests

LAMP stack

You will be using the LAMP software stack

- **L**inux operating system
- **A**pache web server
- **M**ySQL DBMS
- **P**HP scripting language

PHP Internals

PHP itself is written in C

There are 3 main modules

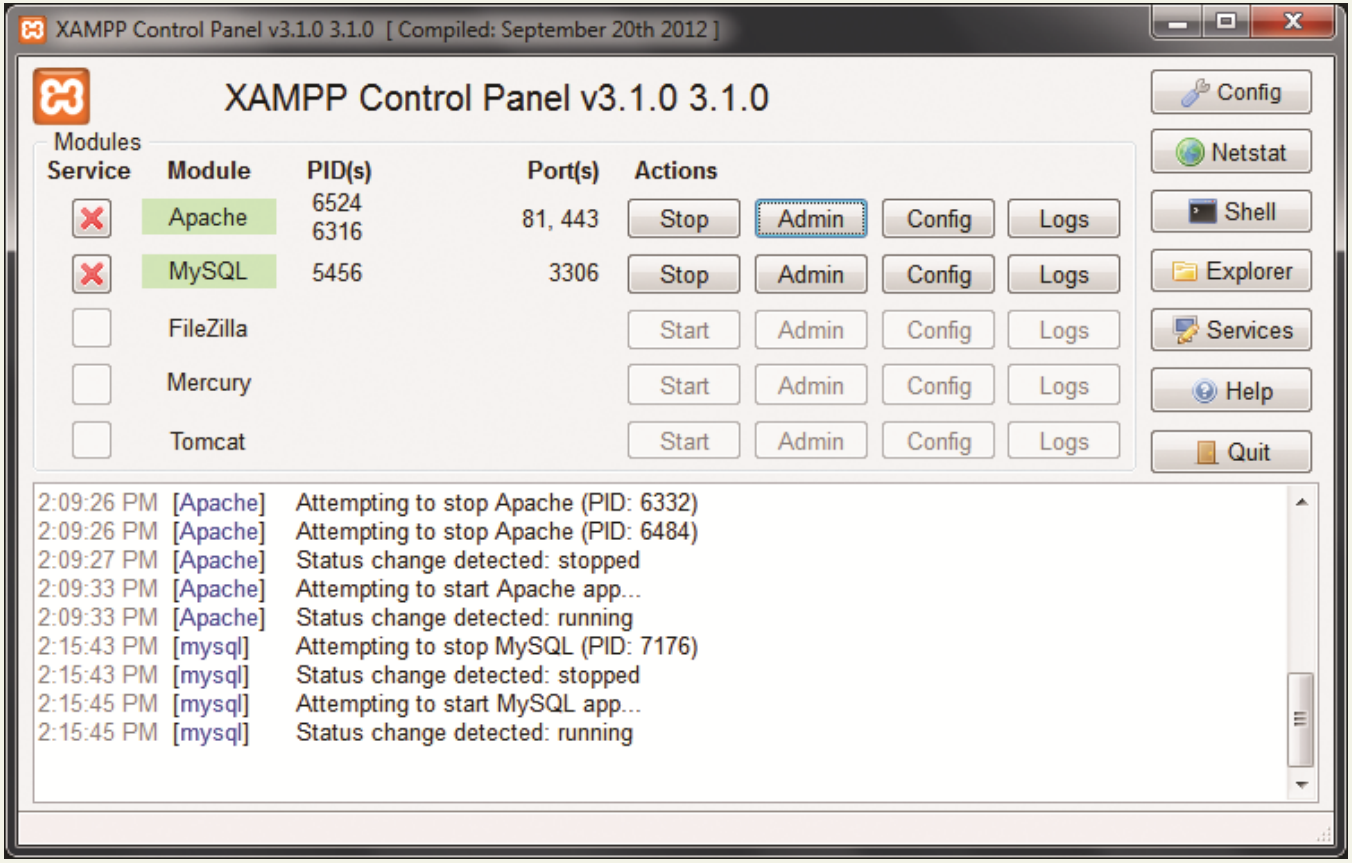
1. **PHP core**. The Core module defines the main features of the PHP environment, including essential functions for variable handling, arrays, strings, classes, math, and other core features.

2. **Extension layer**. This module defines functions for interacting with services outside of PHP. This includes libraries for MySQL, FTP, SOAP web services, and XML processing, among others.

3. **Zend Engine**. This module handles the reading in of a requested PHP file, compiling it, and executing it.

XAMPP Control Panel

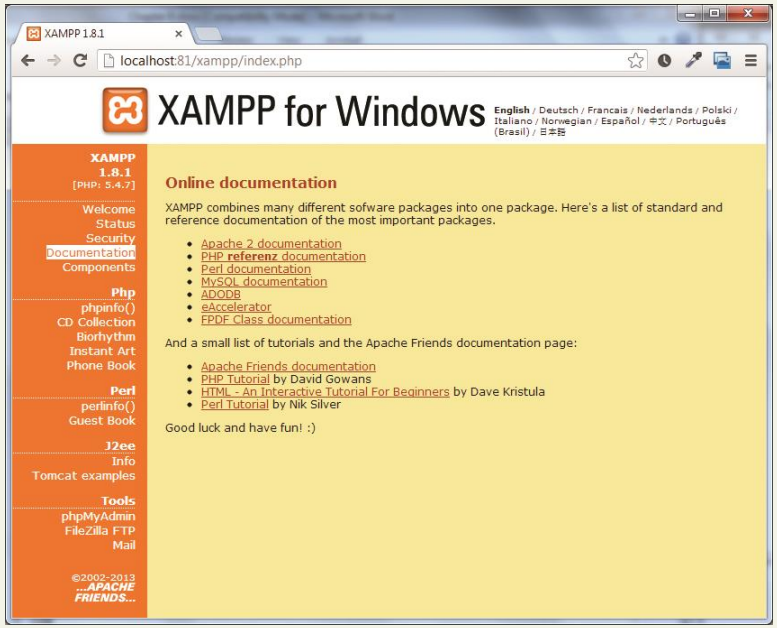
Turn this key



XAMPP Settings

Defaults are

- PHP requests in your browser will need to use the **localhost** domain (127.0.0.1)
- PHP files will have to be saved somewhere within the **C:\xampp\htdocs** folder



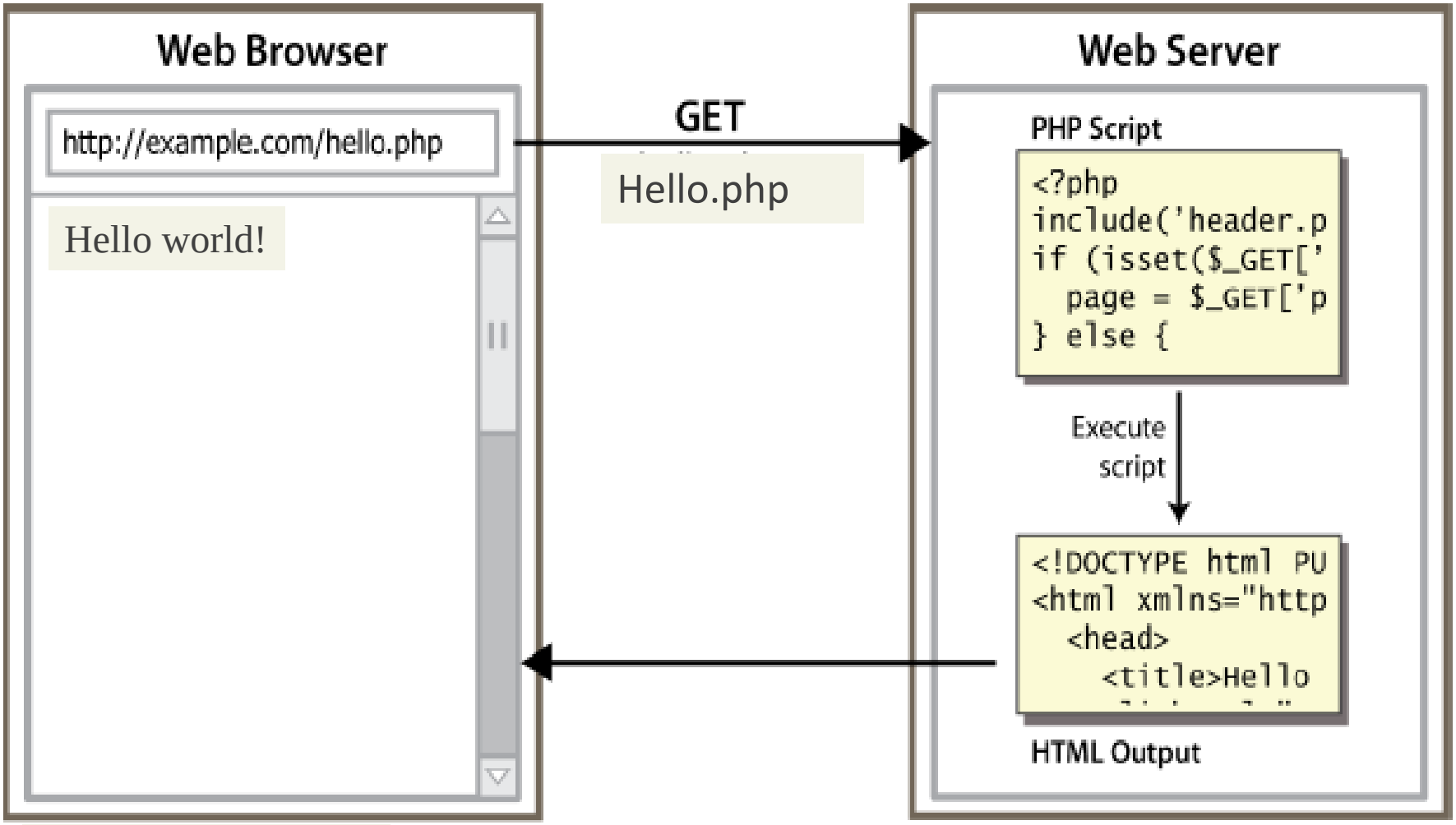
Section 3 of 5

QUICK TOUR OF PHP

Quick Tour

- PHP, like JavaScript, is a dynamically typed language.
- it uses classes and functions in a way consistent with other object-oriented languages such as C++, C#, and Java
- The syntax for loops, conditionals, and assignment is identical to JavaScript
- Differs when you get to functions, classes, and in how you define variables

Lifecycle of a PHP web request



PHP Tags

The most important fact about PHP is that the programming code can be embedded directly within an HTML file.

- A PHP file will usually have the extension **.php**
- programming code must be contained within an opening **<?php** tag and a matching closing **?>** tag
- any code outside the tags is echoed directly out to the client

Hello World!

```php
<?php
print "Hello, world!";
?>
```

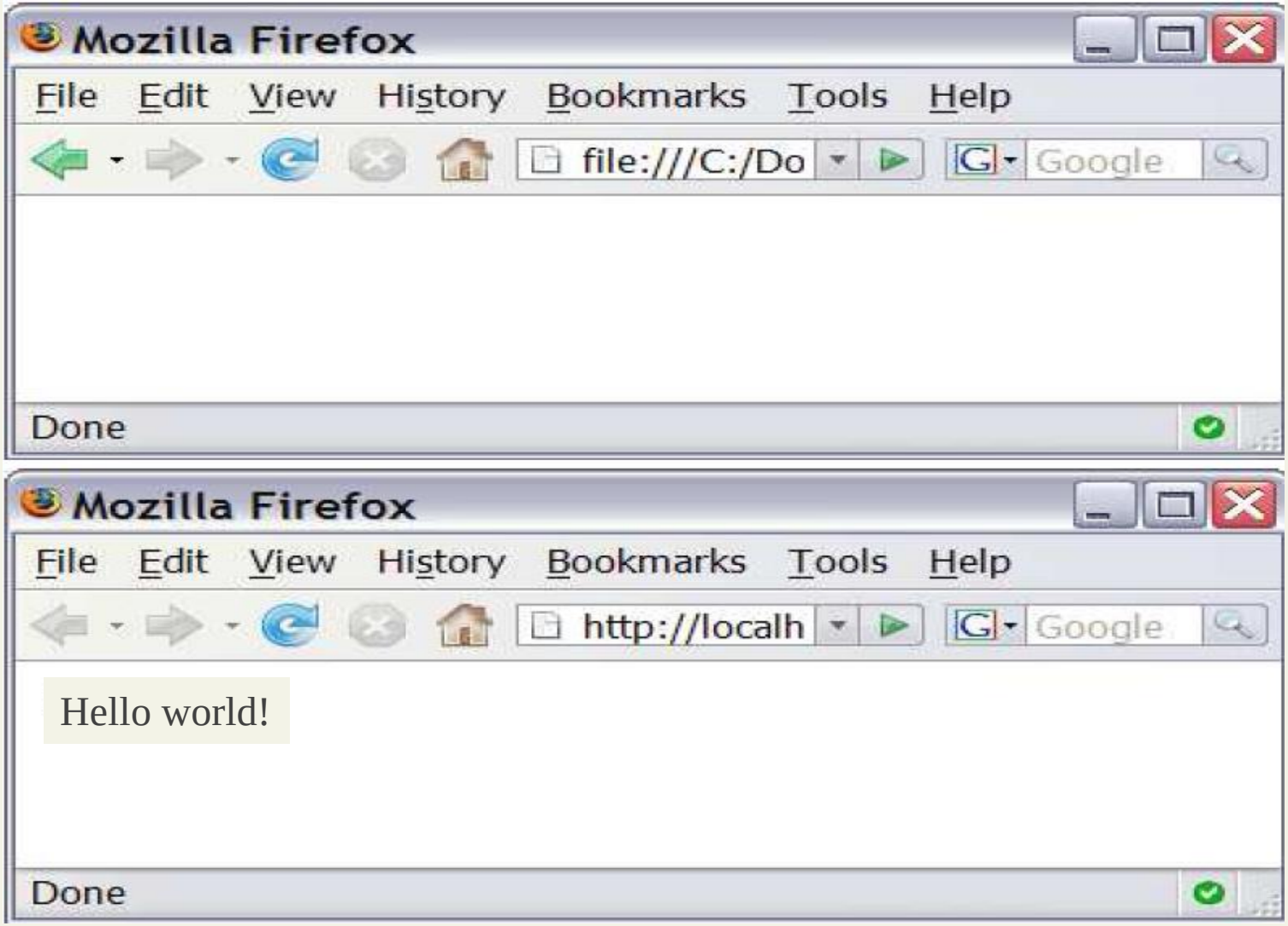
PHP

```
Hello, world!
```

output

block or file of PHP code begins with <?php and ends with ?>
PHP statements, function declarations, etc. appear between these endpoints

Viewing PHP output



PHP BASIC SYNTAX

PHP syntax template

```
HTML content
<?php
PHP code
?>
HTML content
<?php
PHP code
?>
HTML content ...
```
PHP

- Contents of a .php file between `<?php` and `?>` are executed as PHP code
- All other contents are output as pure HTML
- We can switch back and forth between HTML and PHP "modes"

PHP Tags

```
<?php
$user = "Randy";
?>
<!DOCTYPE html>
<html>
<body>
<h1>Welcome <?php echo $user; ?></h1>
<p>
The server time is
<?php
echo "<strong>";
echo date("H:i:s");
echo "</strong>";
?>
</p>
</body>
</html>
```

LISTING 8.1 PHP tags

```
<!DOCTYPE html>
<html>
<body>
<h1>Welcome Randy</h1>
<p>
The server time is <strong>02:59:09</strong>
</p>
</body>
</html>
```

LISTING 8.2 Listing 8.1 in the browser

PHP Comments

3 kinds

The types of comment styles in PHP are:

- **Single-line comments**. Lines that begin with a # are comment lines and will not be executed.
- **Multiline (block) comments**. These comments begin with a /* and encompass everything that is encountered until a closing */ tag is found.
- **End-of-line comments**. Whenever // is encountered in code, everything up to the end of the line is considered a comment.

PHP Comments

3 kinds

```
<?php

# single-line comment

/*

This is a multiline comment.

They are a good way to document functions or
complicated blocks of code

*/

$artist = readDatabase(); // end-of-line comment

?>
```

Variables

Variables in PHP are **dynamically typed.**

Variables are also **loosely typed** in that a variable can be assigned different data types over time

To declare a variable you must preface the variable name with the dollar ($) symbol.

$count = 42;

Data Types

Data Type	Description
Boolean	A logical true or false value
Integer	Whole numbers
Float	Decimal numbers
String	Letters
Array	A collection of data of any type (covered in the next chapter)
Object	Instances of classes

Constants

A **constant** is somewhat similar to a variable, except a constant's value never changes . . . in other words it stays constant.

- Typically defined near the top of a PHP file via the **define()** function
- once it is defined, it can be referenced without using the $ symbol

Constants

```php
<?php

# Uppercase for constants is a programming convention
define("DATABASE_LOCAL", "localhost");
define("DATABASE_NAME", "ArtStore");
define("DATABASE_USER", "Fred");
define("DATABASE_PASSWD", "F5^7%ad");
...
# notice that no $ prefaces constant names
$db = new mysqli(DATABASE_LOCAL, DATABASE_NAME, DATABASE_USER,
  DATABASE_NAME);

?>
```

LISTING 8.4 PHP constants

Writing to Output

Hello World

To output something that will be seen by the browser, you can use the echo() function.

echo ("hello"); //long form

echo "hello"; //shortcut

String Concatenation

Easy

Strings can easily be appended together using the concatenate operator, which is the period (.) symbol.

```
$username = "World";

echo "Hello". $username;
```

Will Output **Hello World**

String Concatenation

Example

```
$firstName = "Pablo";

$lastName = "Picasso";

/*

Example one:

The first four lines are equivalent. Notice that you can reference PHP variables within a string literal defined with double quotes.

The resulting output for the first four lines is: <em>Pablo Picasso</em>
The last one displays: <em> $firstName $lastName </em>

*/

echo "<em>" . $firstName . " ". $lastName . "</em>";

echo '<em>' . $firstName . ' '. $lastName. '</em>';

echo '<em>' . $firstName . ' '. $lastName. "</em>";

echo "<em> $firstName $lastName </em>";

echo '<em> $firstName $lastName </em>';  Won't Work!!
```

String Concatenation

Example

```
/*

Example two:

These two lines are also equivalent. Notice that you
can use either the single quote symbol or double
quote symbol for string literals.

*/

echo "<h1>";

echo '<h1>';
```

String Concatenation

Example

```
/*

Example three:

These two lines are also equivalent. In the second
example, the escape character (the backslash) is used
to embed a double quote within a string literal defined
within double quotes.

*/

echo '<img src="23.jpg" >';

echo "<img src=\"23.jpg\" >";
```

PrintF

Good ol' printf

As an alternative, you can use the **printf()** function.

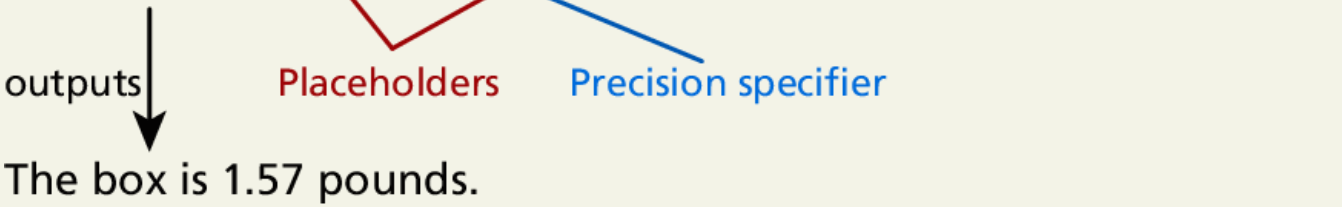
- derived from the same-named function in the C programming language
- includes variations to print to string and files (sprintf, fprintf)

PrintF

Illustrated example

```
$product = "box";
$weight = 1.56789;

printf("The %s is %.2f pounds", $product, $weight);
```



outputs

The box is 1.57 pounds.

PrintF

Type specifiers

Each placeholder requires the percent (%) symbol in the first parameter string followed by a type specifier.

- b for binary
- d for signed integer
- f for float
- o for octal
- x for hexadecimal

PrintF

Precision

Precision allows for control over how many decimal places are shown. Important for displaying calculated numbers to the user in a “pretty” way.

Precision is achieved in the string with a period (.) followed by a number specifying how many digits should be displayed for floating-point numbers.

Section 4 of 5

PROGRAM CONTROL

print

```php
<?php
print "Hello, Comp334 students";
print " Have a look to this Escape \"chars\" ";
print nl2br("hello\nComp334");


echo ' A string can use "single-quotes".';

?>
```

Hello, Comp334 students Have a look to this Escape "chars" hello Comp334 A string can use "single-quotes".

some PHP programmers use the equivalent echo instead of print

If…else

The syntax for conditionals in PHP is almost identical to that of JavaScript

```
// if statement with condition
if ( $hourOfDay > 6 && $hourOfDay < 12 ) {
   $greeting = "Good Morning";
}
else if ($hourOfDay == 12) {   // optional else if
   $greeting = "Good Noon Time";
}
else {                         // optional else branch
   $greeting = "Good Afternoon or Evening";
}
```

LISTING 8.7 Conditional statement using if . . . else

If…else

Alternate syntax

```
<?php

   if ($userStatus == "loggedin") {
      echo '<a href="account.php">Account</a> ';
      echo '<a href="logout.php">Logout</a>';
   }
   else {
      echo '<a href="login.php">Login</a> ';
      echo '<a href="register.php">Register</a>';
   }
?>
```

LISTING 8.8 Combining PHP and HTML in the same script

Switch…case

Nearly identical

```
switch ($artType) {
   case "PT":
   $output = "Painting";
     break;
     case "SC":
   $output = "Sculpture";
     break;
     default:
   $output = "Other";
}

// equivalent
if ($artType == "PT")
   $output = "Painting";
else if ($artType == "SC")
   $output = "Sculpture";
else
   $output = "Other";
```

LISTING 8.9 Conditional statement using switch

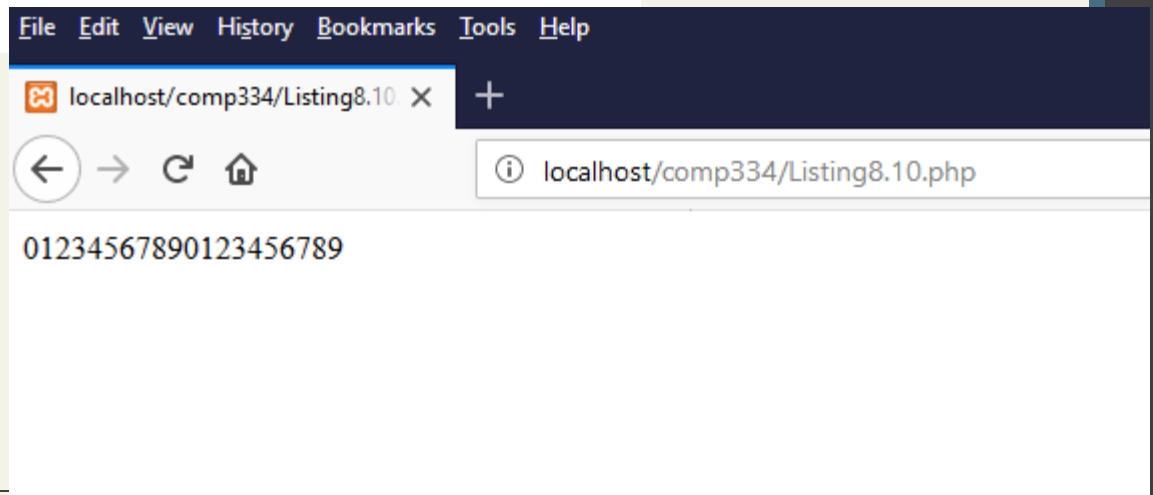
While and Do..while

Identical to other languages

```php
$count = 0;
while ($count < 10)
{
   echo $count;
   $count++;
}

$count = 0;
do
{
   echo $count;
   $count++;
} while ($count < 10);
```

LISTING 8.10 while loops



For

Identical to other languages

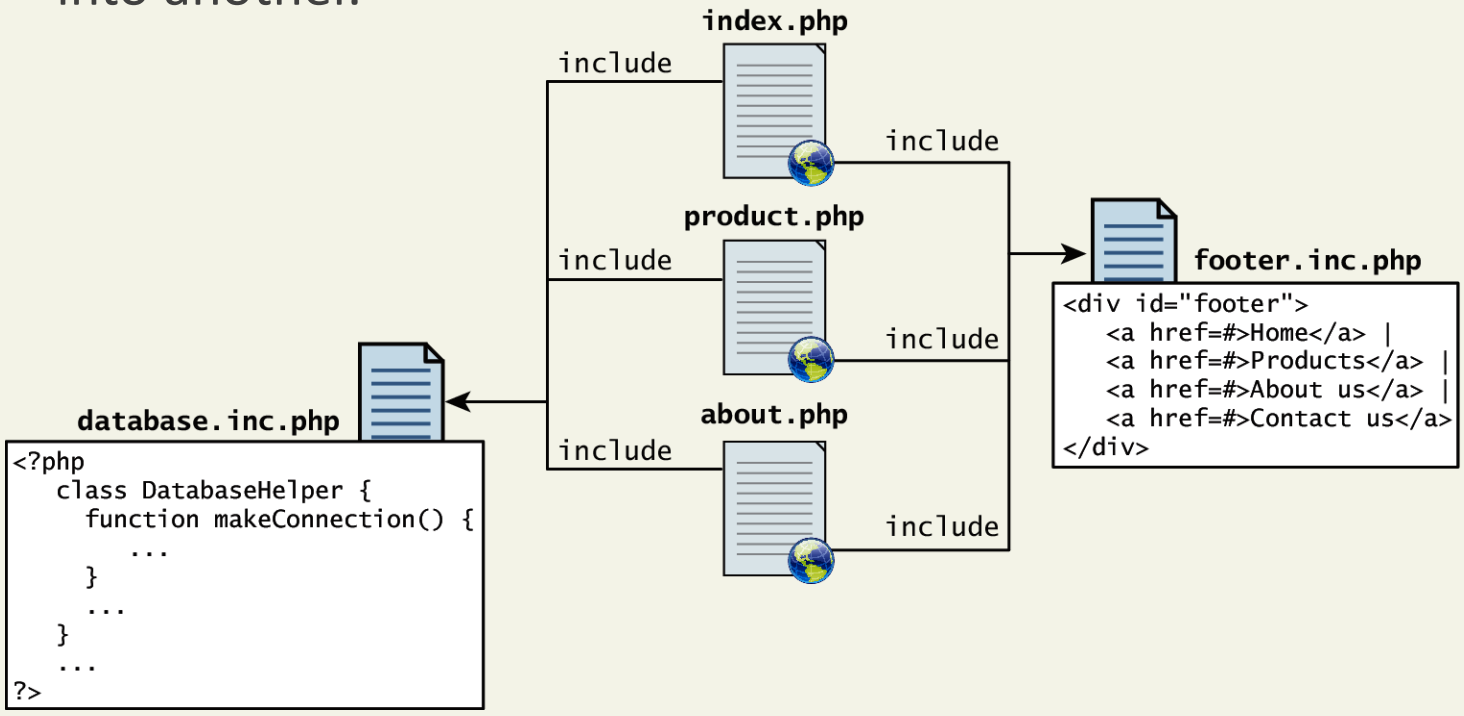
```
for ($count=0; $count < 10; $count++)
{
    echo $count;
}
```

LISTING 8.11 for loops

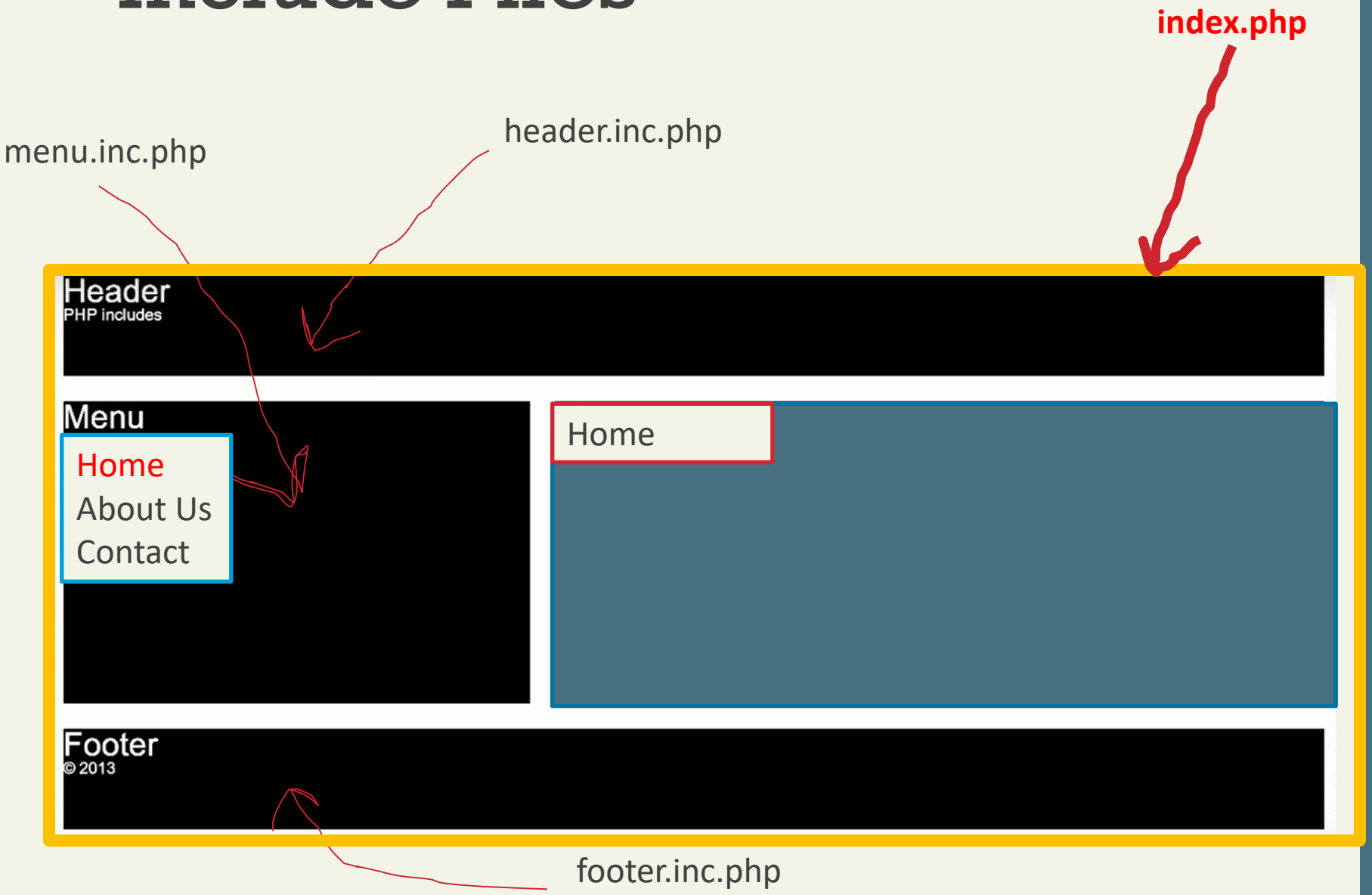
Include Files

Organize your code

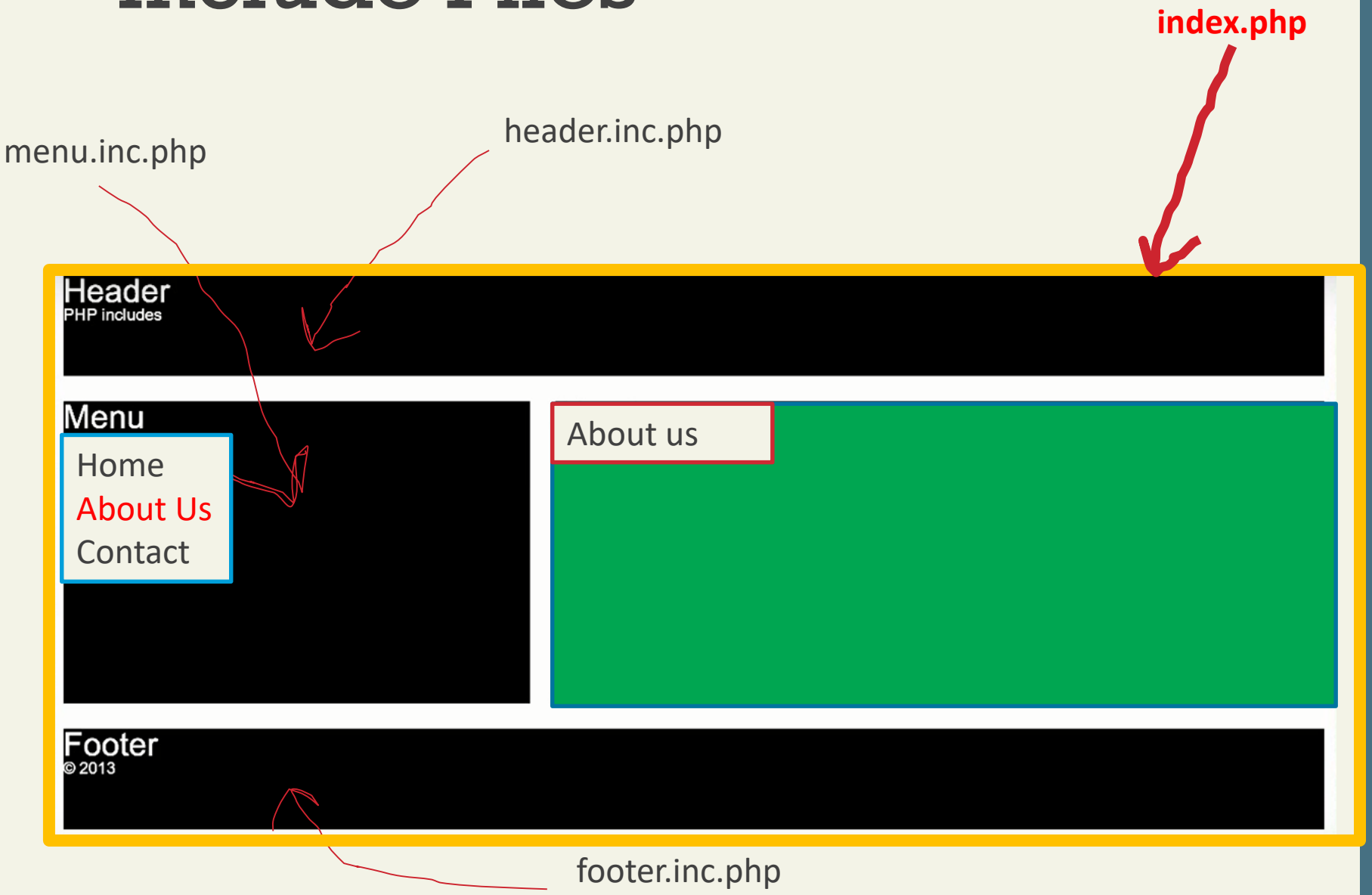
PHP does have one important facility that is generally unlike other non web programming languages, namely the ability to include or insert content from one file into another.



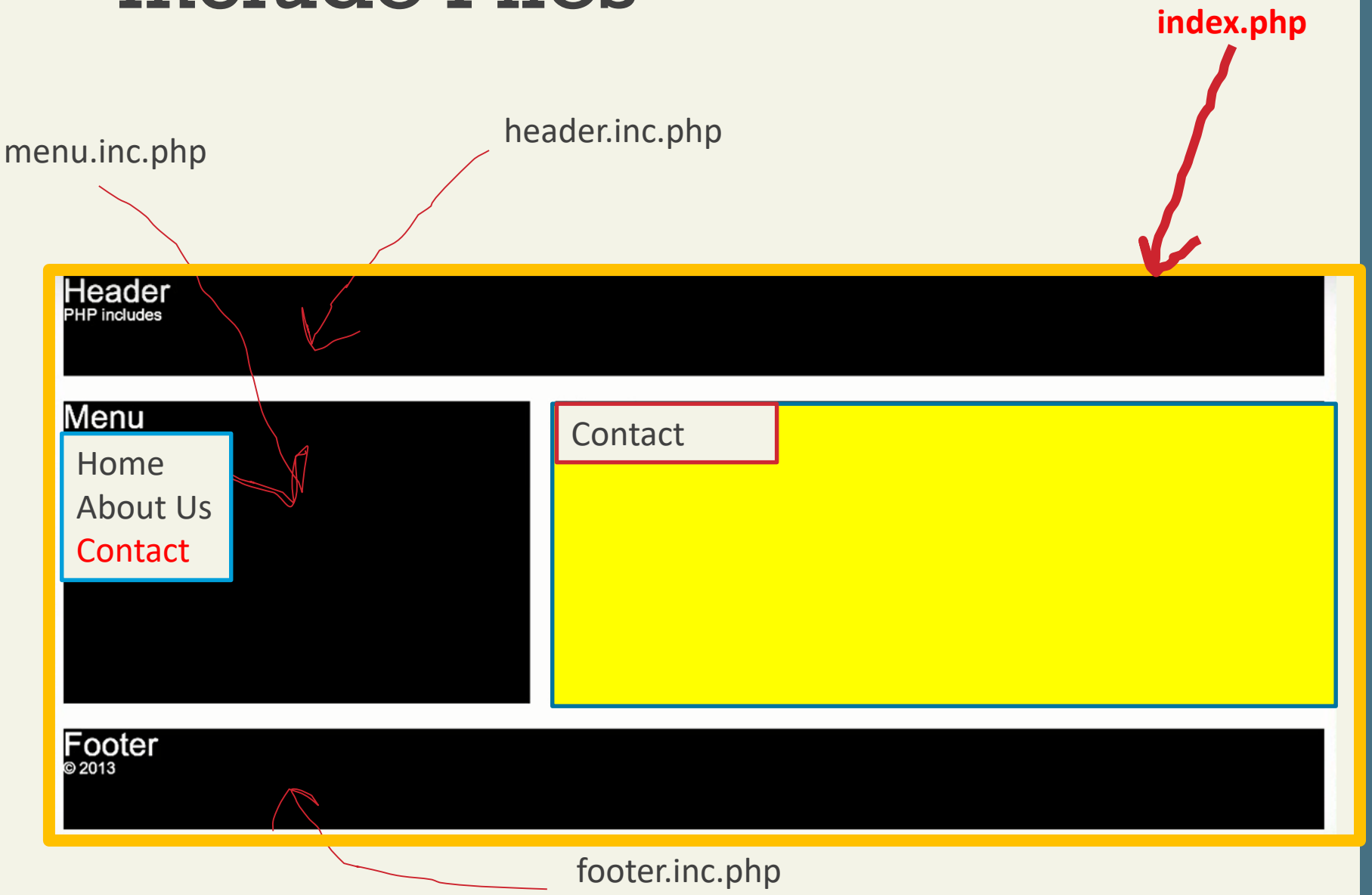
Include Files



Include Files



Include Files



Include Files

Scope

Include files are the equivalent of copying and pasting.

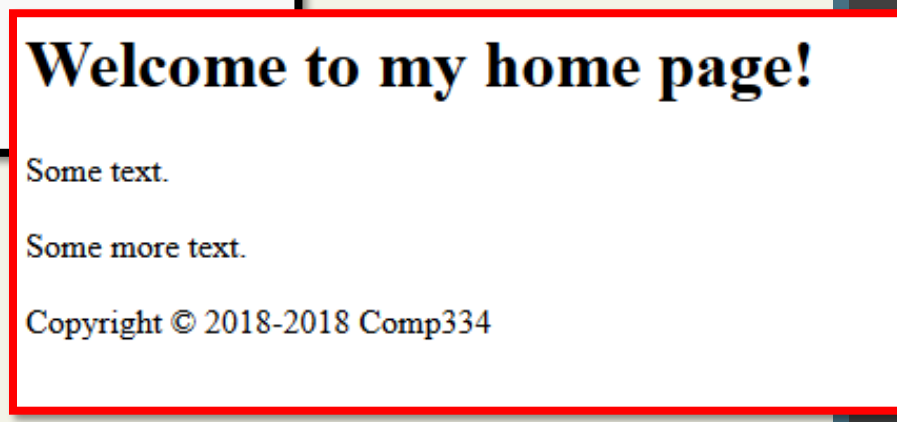
include **require**

With include, a warning is displayed and then execution continues.
With require, an error is displayed and execution stops.

Include Files

mypage.php

```
1  <html>
2  <body>
3
4    <h1>Welcome to my home page!</h1>
5    <p>Some text.</p>
6    <p>Some more text.</p>
7    <?php include 'footer.inc.php';?>
8
9    </body>
10   </html>
```



footer.inc.php

```
1  <?php
2  echo "<p>Copyright &copy; 2018-" . date("Y") . " Comp334</p>";
3  ?>
```

Section 5 of 5

FUNCTIONS

Functions

You mean we don't write everything in main?

Just as with any language, writing code in the main function (which in PHP is equivalent to coding in the markup between <?php and ?> tags) is not a good habit to get into.

A **function** in PHP contains a small bit of code that accomplishes one thing. In PHP there are two types of function: user-defined functions and built-in functions.

1. A **user-defined function** is one that you the programmer define.
2. A **built-in function** is one of the functions that come with the PHP environment

Functions

syntax

```
/**
* This function returns a nicely formatted string using the current
* system time.
*/
function getNiceTime() {
   return date("H:i:s");
}
```

LISTING 8.13 The definition of a function to return the current time as a string

While the example function in Listing 8.13 returns a value, there is no requirement for this to be the case.

Functions

```php
<?php
print "Hello, Comp334 students";
print "This example print the current time as a string  ";



echo nl2br("\n").date("H:i:s");
?>
```

Functions

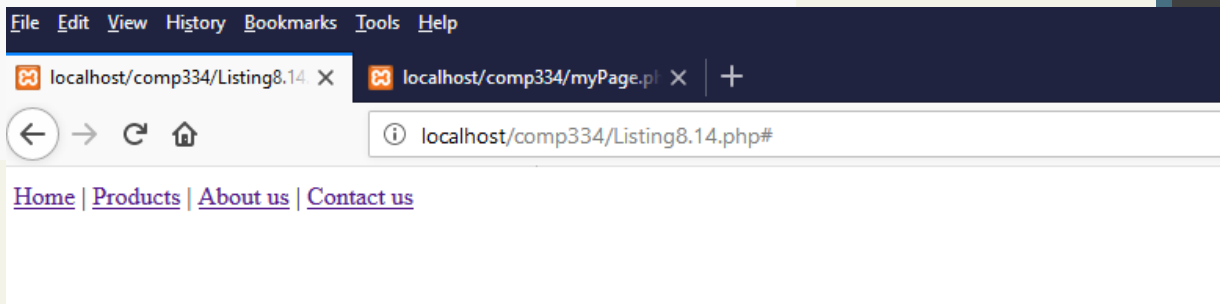
No return – no big deal.

```php
<?php

/**
* This function outputs the footer menu
*/
function outputFooterMenu() {
echo '<div id="footer">';
echo '<a href=#>Home</a> | <a href=#>Products</a> | ';
echo '<a href=#>About us</a> | <a href=#>Contact us</a>';
echo '</div>';
}

//demo the function
outputFooterMenu();

?>
```



Function call

Call a function

Now that you have defined a function, you are able to use it whenever you want to. To call a function you must use its name with the () brackets.

Since getNiceTime() returns a string, you can assign that return value to a variable, or echo that return value directly, as shown below.

```
$output = getNiceTime();

echo getNiceTime();
```

Return value

If the function doesn't return a value, you can just call the function:

```
outputFooterMenu();
```

Doesn't return a value

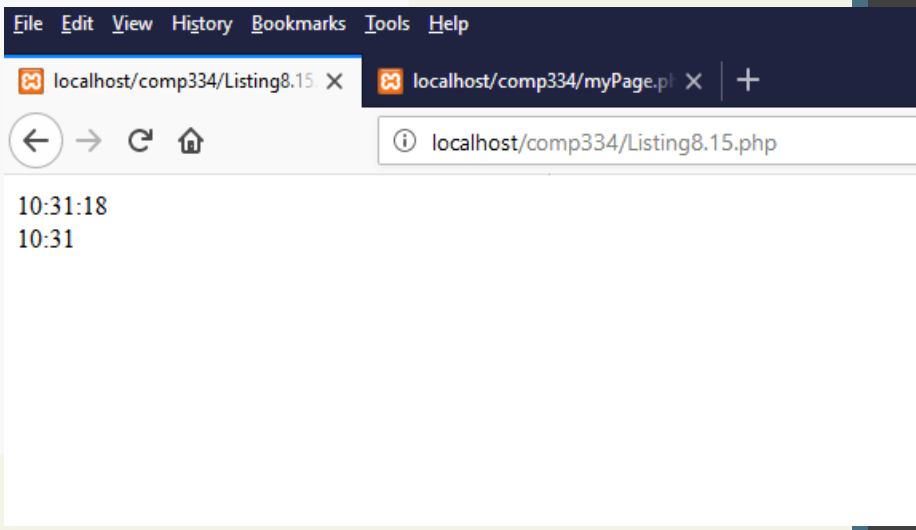
Parameters

```php
<?php

/**
* This function returns a nicely formatted string using the current
* system time. The showSeconds parameter controls whether or not to
* include the seconds in the returned string.
*/
function getNiceTime($showSeconds) {
if ($showSeconds==true)
return date("H:i:s");
else
return "<br>".date("H:i");
}


echo getNiceTime(1); // this will print seconds
echo getNiceTime(0); // will not print seconds

?>
```



Thus to call our function, you can now do it in two ways:

echo getNiceTime(1); *// this will print seconds*
echo getNiceTime(0); *// will not print seconds*

Parameter Default Values

```php
/**
* This function returns a nicely formatted string using the current
* system time. The showSeconds parameter controls whether or not
* to show the seconds.
*/
function getNiceTime($showSeconds=1){
   if ($showSeconds==true)
      return date("H:i:s");
   else
      return date("H:i");
}
```

LISTING 8.16 A function to return the current time with a parameter that includes a default

Now if you were to call the function with no values, the $showSeconds parameter would take on the default value, which we have set to 1, and return the string with seconds.

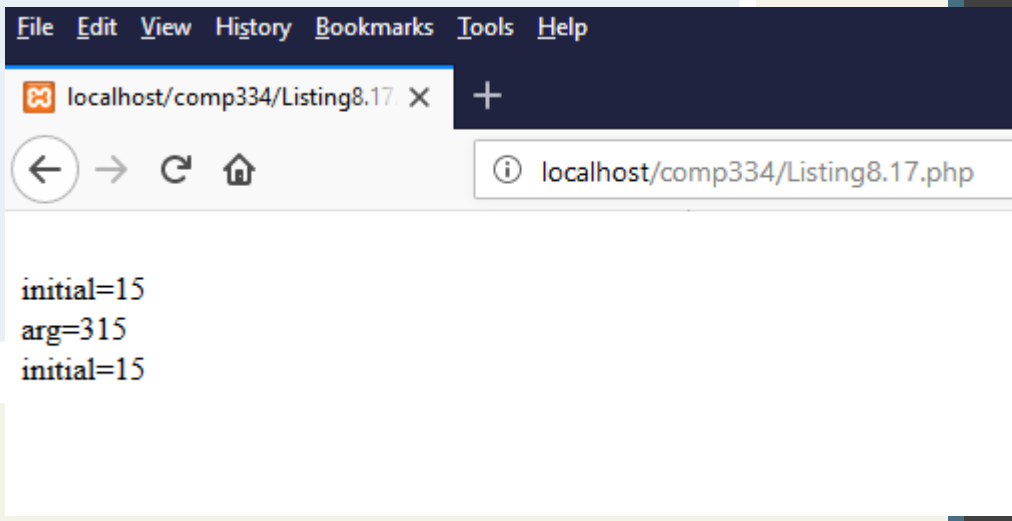
Pass Parameters by Value

By default, arguments passed to functions are **passed by value** in PHP. This means that PHP passes a copy of the variable so if the parameter is modified within the function, it does not change the original.

```
function changeParameter($arg) {
   $arg += 300;
   echo "<br/>arg=" . $arg;
}

$initial = 15;
echo "<br/>initial=" . $initial;
changeParameter($initial);
echo "<br/>initial=" . $initial;
```

LISTING 8.17 Passing a parameter by value



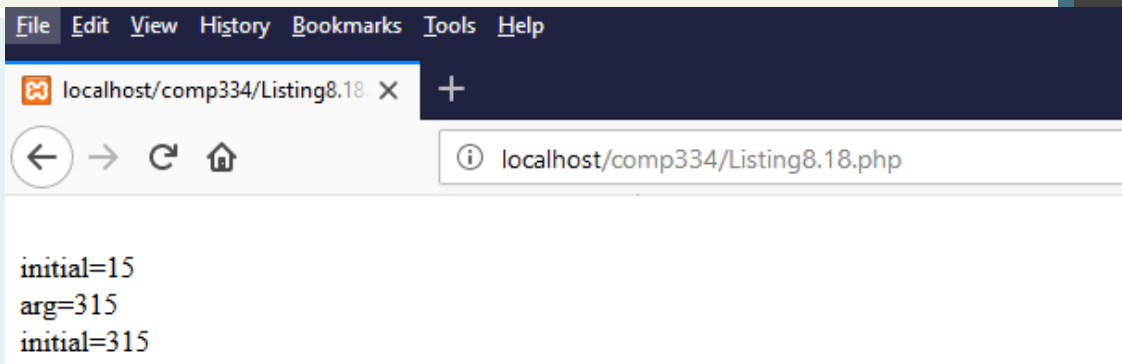
Pass Parameters by Reference

PHP also allows arguments to functions to be **passed by reference**, which will allow a function to change the contents of a passed variable.

The mechanism in PHP to specify that a parameter is passed by reference is to add an ampersand (&) symbol next to the parameter name in the function declaration

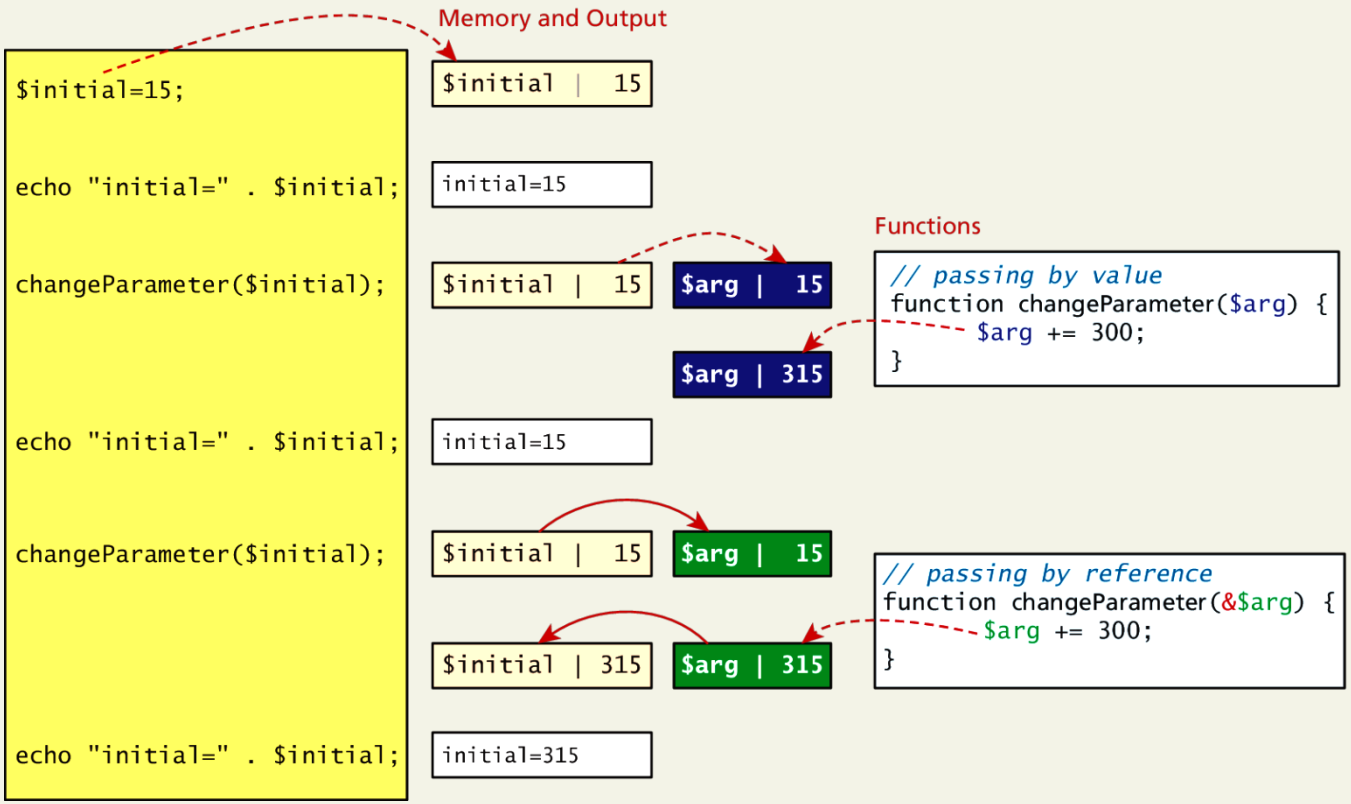
```
function changeParameter(&$arg) {
   $arg += 300;
   echo "<br/>arg=". $arg;
}

$initial = 15;
echo "<br/>initial=" . $initial;
changeParameter($initial);
echo "<br/>initial=" . $initial;
```



LISTING 8.18 Passing a parameter by referer

Value vs Reference



Variable Scope in functions

All variables defined within a function (such as parameter variables) have **function scope**, meaning that they are only accessible within the function.

Any variables created outside of the function in the main script are unavailable within a function.

```
$count= 56;

function testScope() {
        echo $count;      // outputs 0 or generates run-time
                          //warning/error

}

testScope();
echo $count; // outputs 56
```

Global variables

Sometimes unavoidable

Variables defined in the main script are said to have **global scope.**

Unlike in other programming languages, a global variable is not, by default, available within functions.

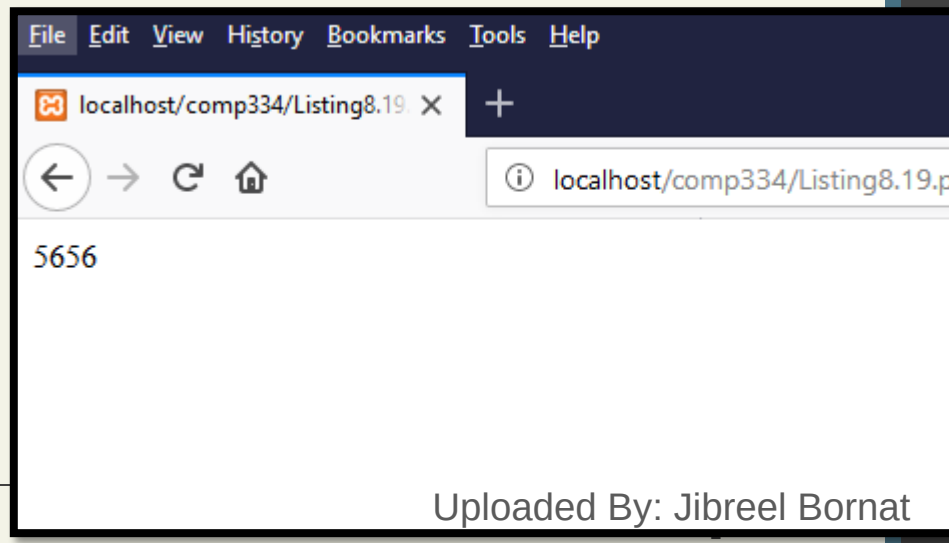
PHP does allow variables with global scope to be accessed within a function using the **global** keyword

```php
<?php

$count= 56;
function testScope() {
    global $count;
    echo $count; // outputs 56
}

testScope();
echo $count; // outputs 56

?>
```



More Examples

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

<?php
echo "Today is " . date("Y/m/d") . "<br>";
echo "Today is " . date("Y.m.d") . "<br>";
echo "Today is " . date("Y-m-d") . "<br>";
echo "Today is " . date("l");
?>

</body>
</html>
```

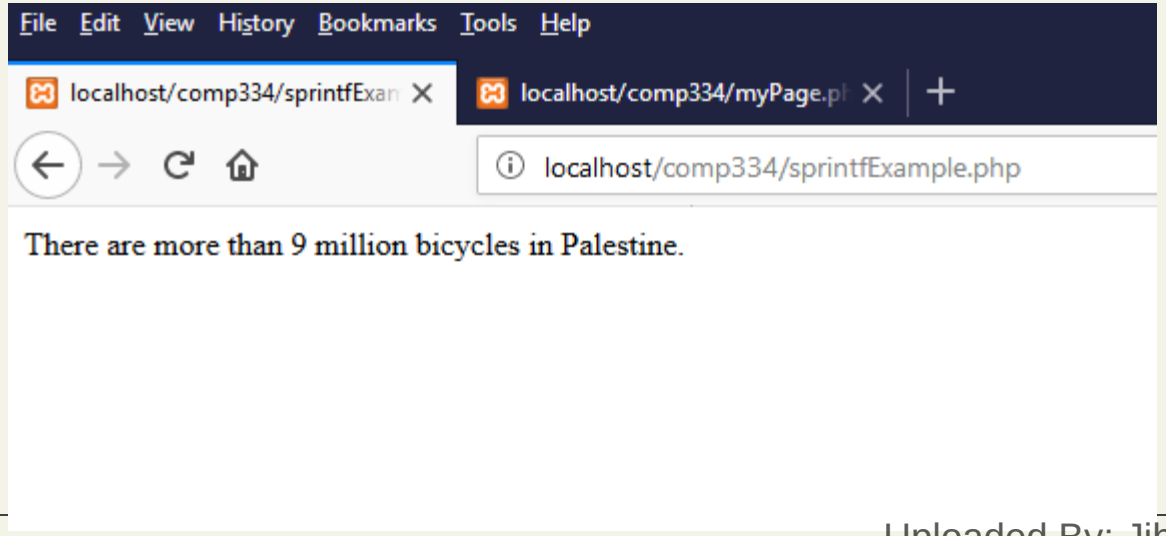
Today is 2018/05/06
Today is 2018.05.06
Today is 2018-05-06
Today is Sunday

- d - Represents the day of the month (01 to 31)
- m - Represents a month (01 to 12)
- Y - Represents a year (in four digits)
- l (lowercase 'L') - Represents the day of the week

https://www.w3schools.com/php/showphp.asp?filename=demo_date1

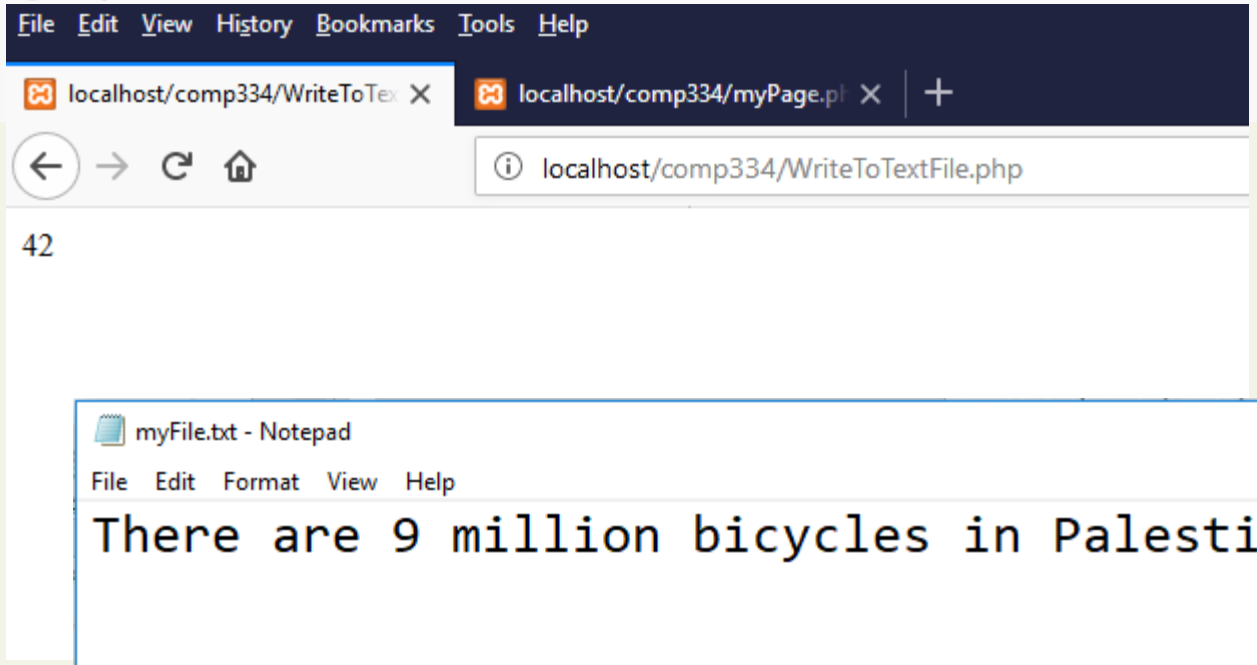
More Examples

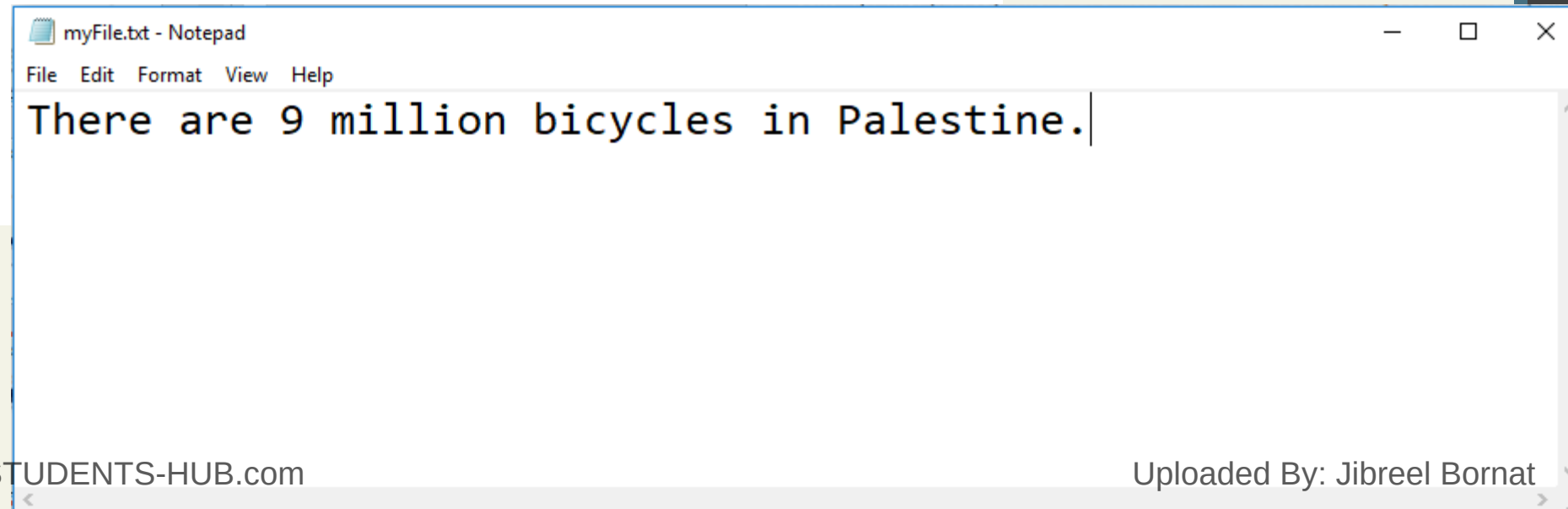
```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <?php
6  $number = 9;
7  $str = "Palestine";
8  $txt = sprintf("There are more than %u million bicycles in %s.",$number,$str);
9  echo $txt;
10 ?>
11
12 </body>
13 </html>
```



More Examples

```php
<?php
$number = 9;
$str = "Palestine";
$file = fopen("myFile.txt","w");
echo fprintf($file,"There are more than %u million bicycles in %s.",$number,$str);
?>
```

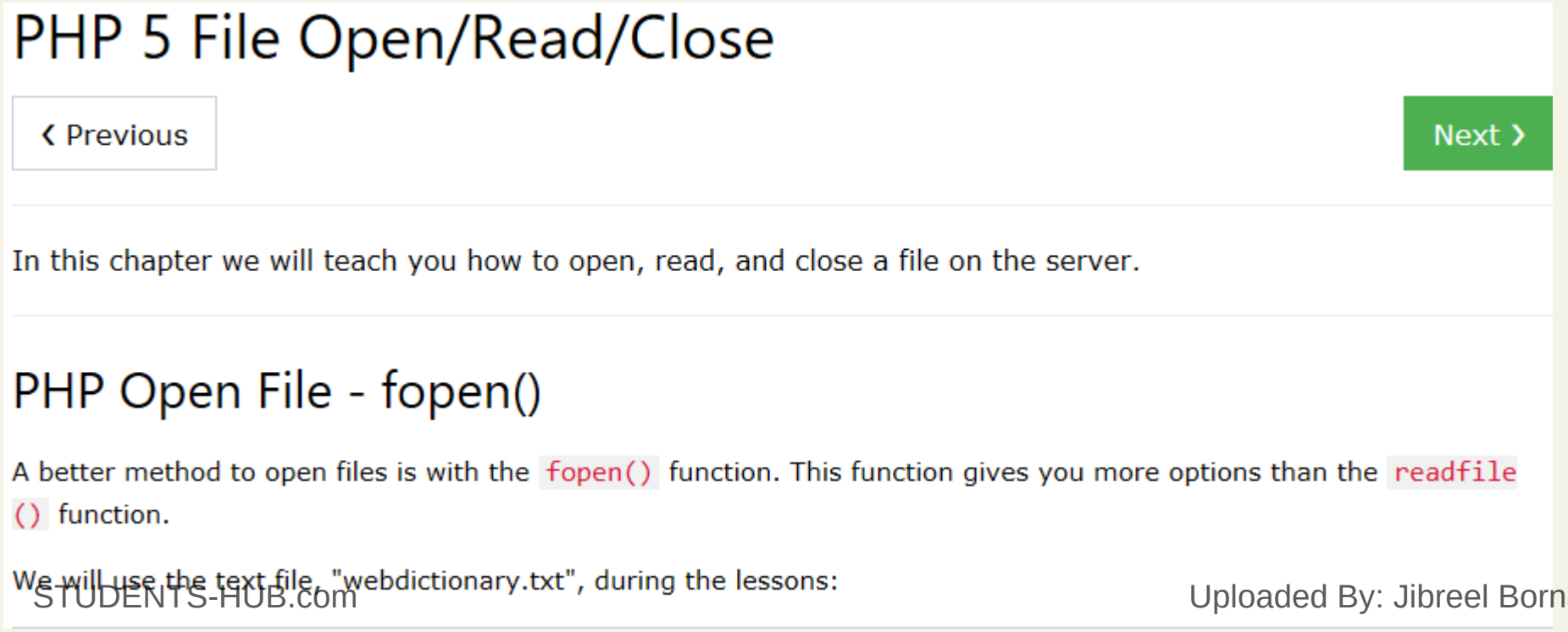




More Examples

To read from file

https://www.w3schools.com/php/php_file_open.asp



PHP 5 File Open/Read/Close

‹ Previous

Next ›

In this chapter we will teach you how to open, read, and close a file on the server.

PHP Open File - fopen()

A better method to open files is with the `fopen()` function. This function gives you more options than the `readfile()` function.

We will use the text file, "webdictionary.txt", during the lessons:

What You've Learned

1. Server-Side Development
2. Web Server's Responsabilities
3. Quick Tour of PHP
4. Program Control
5. Functions