WEB APPLICATIONS

LECTURE 5
MARIE BRENNAN

THIS WEEK

- Some forms recap
- User defined functions
- Scope of variables
- Include files
- Server Environment
- Sticky Forms

FORMS: RECAP

- HTML form input fields send data as key-value pairs where the input is the key, and the input field content is the value.

- Form action attribute specifies a form that will process the data when the data is submitted.

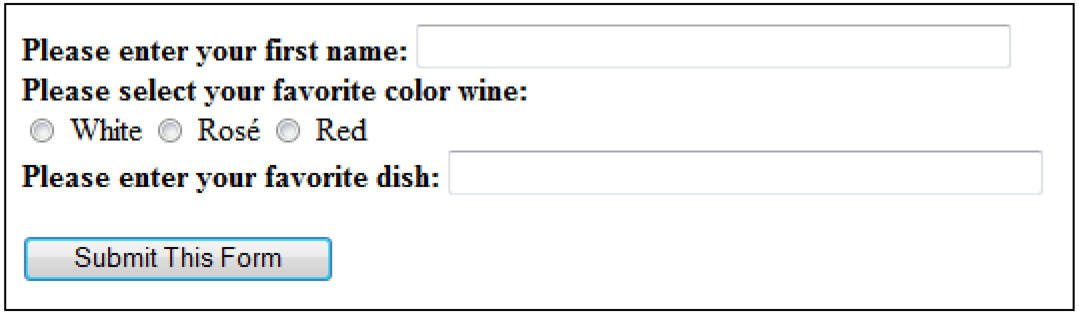
- Form method attribute can be “get” or “post”

GETTING FORM VALUES --- GET

fav.html

Using ‘GET’

```
<html><head><title>Your Favorites</title></head><body>
 <form action="fav.php" method=“get">
  <b>Please enter your first name:</b>
  <input type="text" size="45" name="username">  <br>
  <b>Please select your favorite color wine:</b> <br>
  <input type="radio" name="color" value="white">  White
  <input type="radio" name="color" value="rosé">   Rosé
  <input type="radio" name="color" value="red">    Red <br>
  <b>Please enter your favorite dish:</b>
  <input type="text" size="45" name="dish"><br><br>
  <input type="submit" name = “button” value="Submit This Form">
</form></body></html>
```



Please enter your first name:

Please select your favorite color wine:

○ White ○ Rosé ○ Red

Please enter your favorite dish:

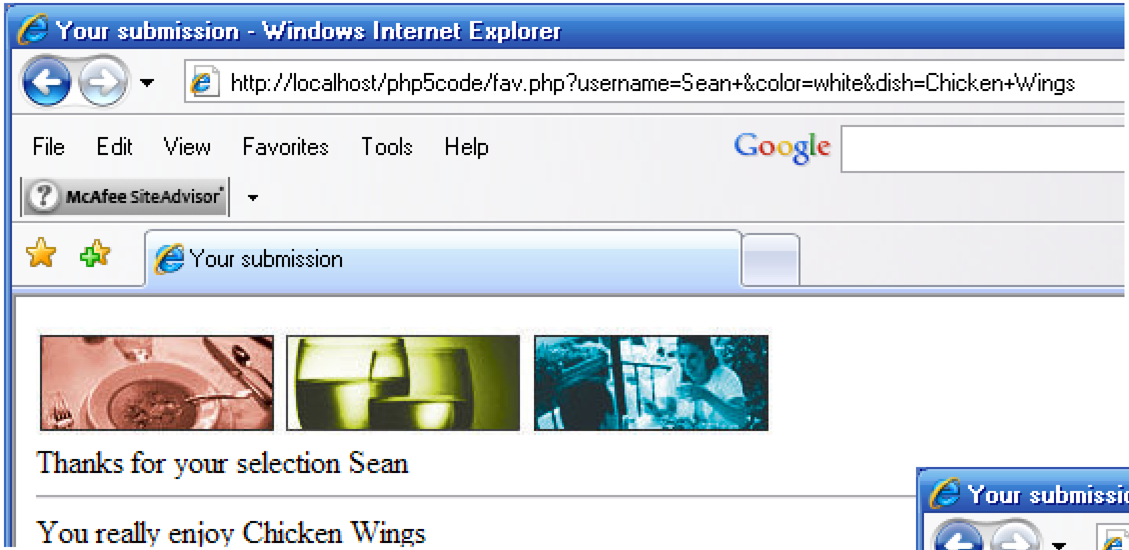
Getting Form Values --- post

fav.html

Using POST

```
<html><head><title>Your Favorites</title></head><body>
 <form action="fav.php" method=“post">
  <b>Please enter your first name:</b>
  <input type="text" size="45" name="username">  <br>
  <b>Please select your favorite color wine:</b> <br>
  <input type="radio" name="color" value="white">  White
  <input type="radio" name="color" value="rosé">   Rosé
  <input type="radio" name="color" value="red">    Red
   <br>
  <b>Please enter your favorite dish:</b>
  <input type="text" size="45" name="dish"><br><br>
  <input type="submit" name = “button” value="Submit This
   Form">
</form></body></html>
```

Displaying the contents



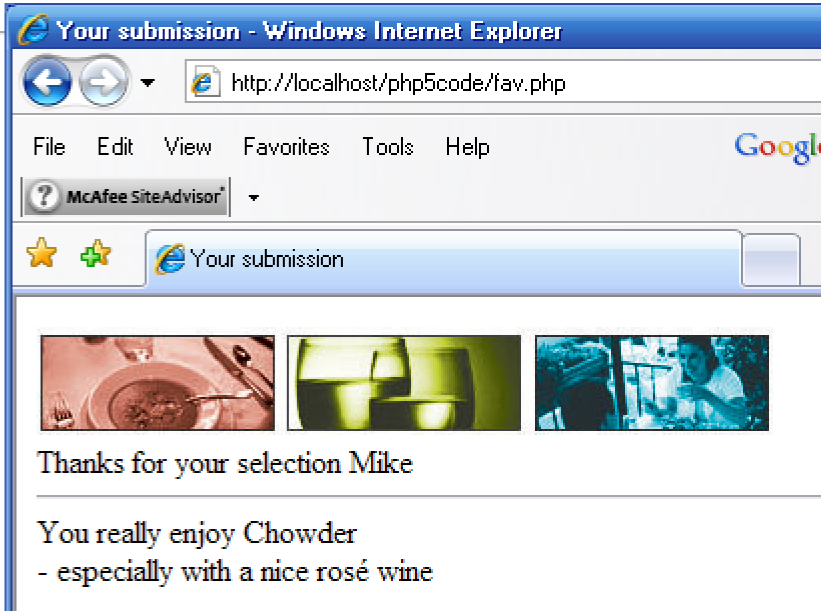
Your submission - Windows Internet Explorer
File Edit View Favorites Tools Help
Google
McAfee SiteAdvisor
Your submission

http://localhost/php5code/fav.php?username=Sean+&color=white&dish=Chicken+Wings

Thanks for your selection Sean

You really enjoy Chicken Wings

“get” used



Your submission - Windows Internet Explorer
File Edit View Favorites Tools Help
Googl
McAfee SiteAdvisor
Your submission

http://localhost/php5code/fav.php

Thanks for your selection Mike

You really enjoy Chowder

- especially with a nice rosé wine

“post”

Forms: Get V Post methods

- There are limitations to the GET method:
  - long strings may exceed browser limitations 1024 bytes
  - Name value pairs are visible to anyone
  - Not secure
  - However, URL's can be bookmarked
  - More….

Forms: Post method

- POST method doesn't have these limitations.
- Forms can specify either GET or POST

Form processing

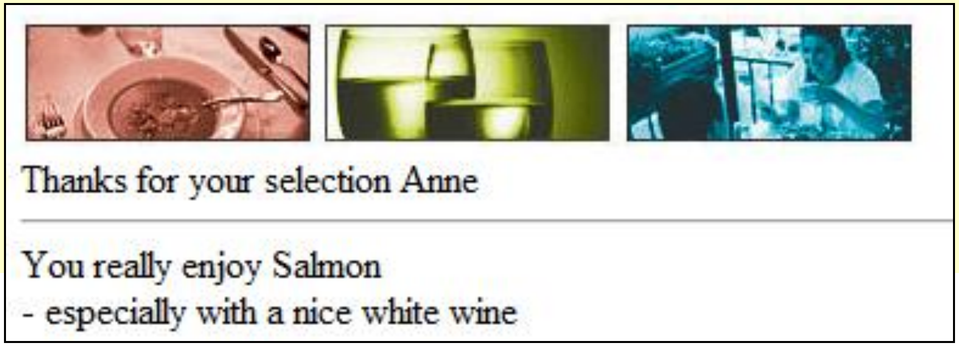
- When you submit the form the name-value pairs are sent to the server-side script

- In the case of PHP these values are available to the script in the arrays `$_GET`, `$_POST`, `$_REQUEST`

DISPLAYING SUBMITTED VALUES – VERSION 1

fav.php

```
<html> <head><title>Your submission</title> </head> <body>
 <img src="foodbnr.jpg" width="368" height="54"> <br>
<?php
                $username = $_REQUEST['username'];
                $color =    $_REQUEST['color'];
                $dish =     $_REQUEST['dish'];
                if( $username != null ) {
                  echo  "Thanks for your selection $username <hr/>" ;
                }
                if( ( $color != null ) && ( $dish != null ) ) {
                  $msg = "You really enjoy $dish <br/>";
                  $msg .= "- especially with a nice $color wine";
                  echo( $msg );
                }
 ?>
 </body></html>
```



Thanks for your selection Anne

You really enjoy Salmon
- especially with a nice white wine

THE ALL-IN-ONE FORM & RESPONSE

Several common patterns in processing forms

1. Put the form in one file & the PHP to generate the response in another form.
2. Specify the second form as the action of the first Example: fav.html, fav.php
3. Form & response are put in the same PHP file. favForm.php

THE ALL-IN-ONE FORM & RESPONSE

A test is made to see if the main form variable is set i.e. Submit button pressed

If it isn’t, the HTML form is generated as output

If it is, the form is processed and a response is generated

```
if  (isset($_POST[“buttonName”])) {
 // do the form processing and & generate the response here
}
else {
   // generate the form here
}
```

Idea is to put as much HTML code outside main block of PHP code

THE ALL-IN-ONE FORM & RESPONSE

isset() function checks for the existence of

….. is it set?

Use $SERVER[‘PHP_SELF’] as the action in the form

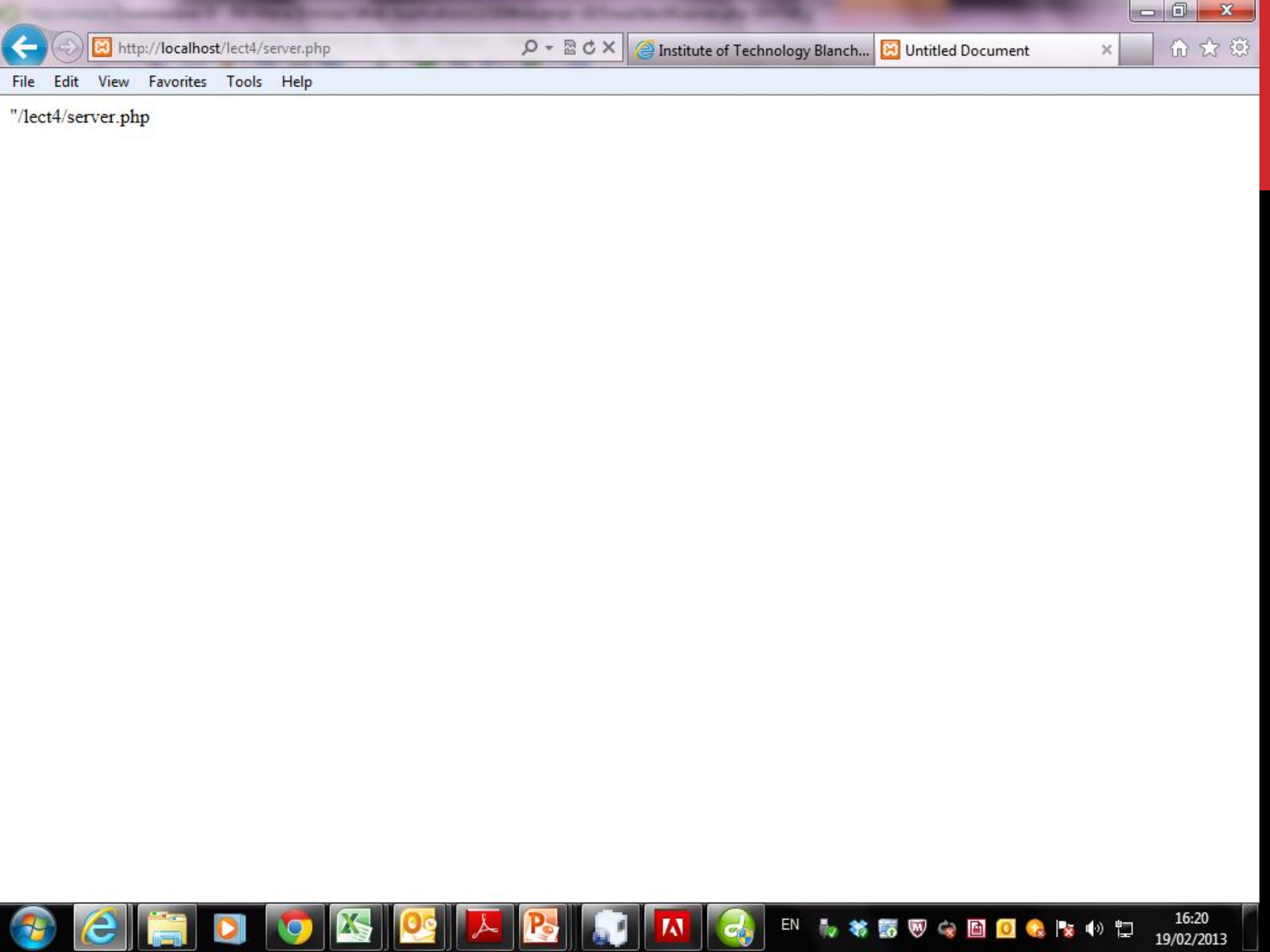
$SERVER[‘PHP_SELF’] evaluates to filename of the PHP script currently running.

It gives the script location relative to the document root.

`$self = $_SERVER['PHP_SELF'];` puts the file path into a variable called $self

THE ALL-IN-ONE FORM & RESPONSE

```
<html><head><title>Your Favorites</title></head><body>

<form action= “


<?php echo $_SERVER['PHP_SELF']; ?>"


    method="post">

Check <input type="submit" name = "button"
value="Submit">

</form>...
```



http://localhost/lect4/server.php
Institute of Technology Blanch...
Untitled Document
File
Edit
View
Favorites
Tools
Help
EN
16:20
19/02/2013

```
"/lect4/server.php
```

DISPLAYING SUBMITTED VALUES – VERSION 2

- **What if the user tries to get the form processed without entering any form data?**
- **Check whether a field e.g. username has been filled in or not**
  - Use empty() or !isset() functions
- **If not, re-direct the user to the form**
  - Use the header() function

```
if (empty($_REQUEST[‘username’]))
    {
       header(“Location: url or filename.php”);
       exit;
       }
```

THE ALL-IN-ONE FORM & RESPONSE

```
<?php
if (isset($_REQUEST[‘button']))
  {
 $self =  $_SERVER['PHP_SELF'];
 if(empty($_REQUEST[‘username’]))  {
                    header(“Location: $self”); exit;
                    }
   // process the form (using PHP) & display the form response
}
else
  {
?>
   <html>  create a HTML form here, using plain HTML  </html>
<?php
}   // end of else-block
?>
```

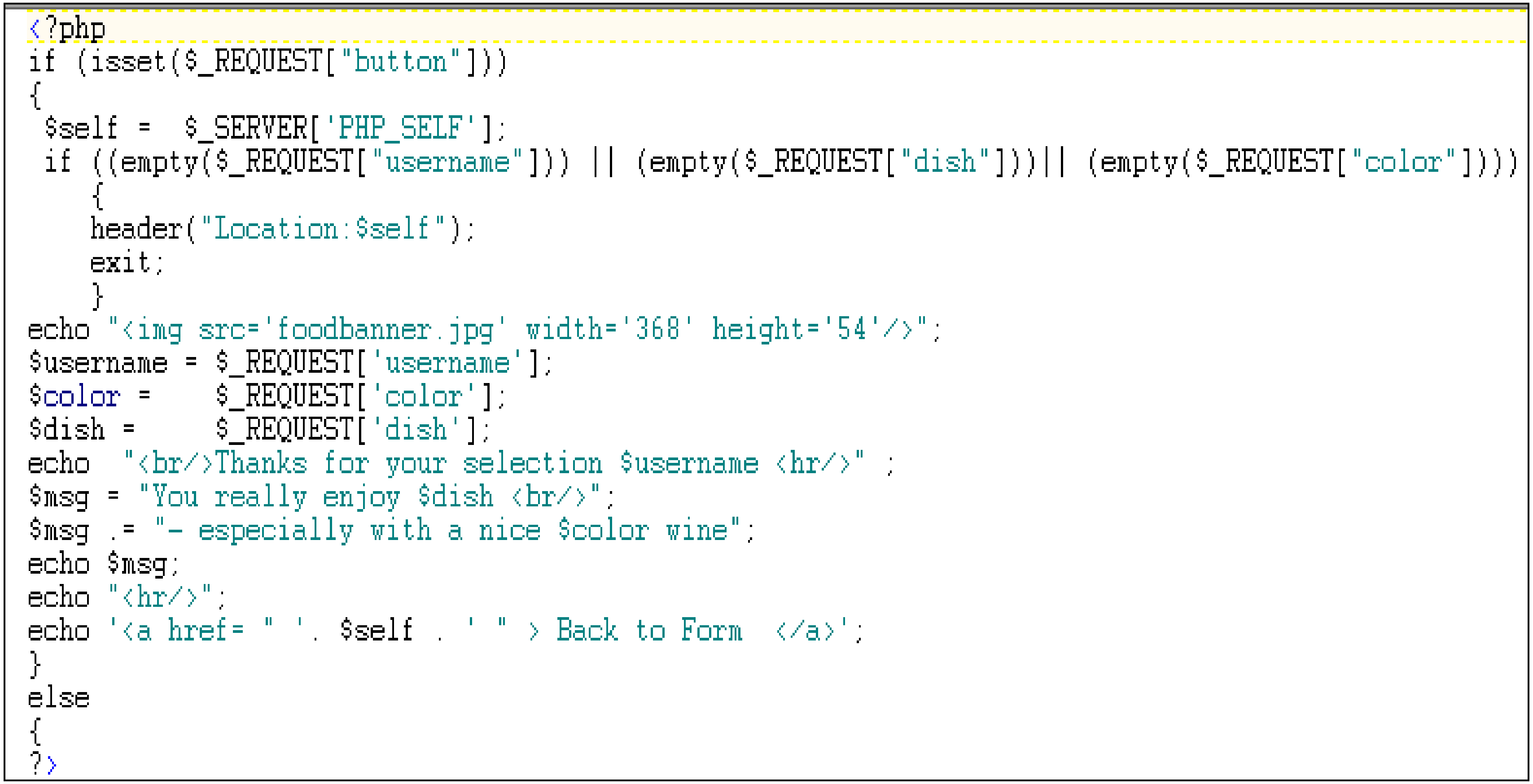
THE ALL-IN-ONE FORM & RESPONSE

Pseudocode for All-in-one Form

```
If Submit button pressed
    Check if any fields in the form are blank
        If any fields are blank, display the HTML form again & exit
    If not blank -Get values sent in by user & process form & display results
Else
    Display HTML form
```

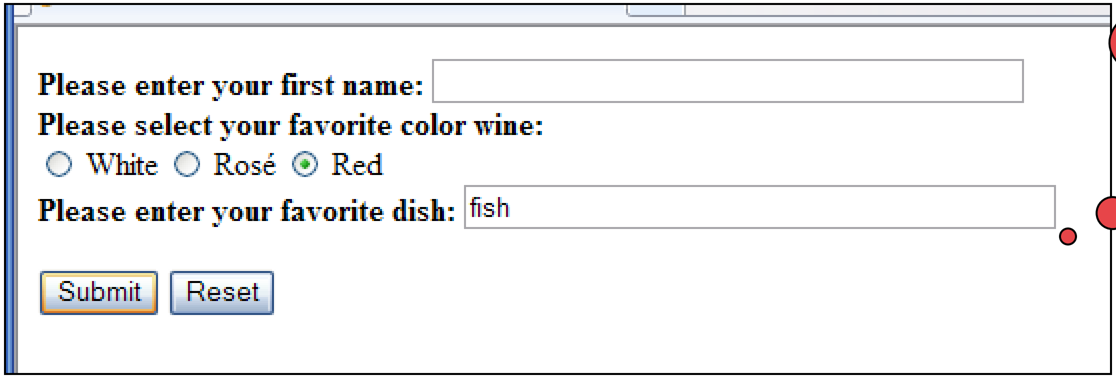
THE ALL-IN-ONE FORM & RESPONSE

Favform.php



```
<?php
if (isset($_REQUEST["button"]))
{
 $self =  $_SERVER['PHP_SELF'];
 if ((empty($_REQUEST["username"])) || (empty($_REQUEST["dish"]))|| (empty($_REQUEST["color"])))
    {
    header("Location:$self");
    exit;
    }
echo "<img src='foodbanner.jpg' width='368' height='54'/>";
$username = $_REQUEST['username'];
$color =    $_REQUEST['color'];
$dish =     $_REQUEST['dish'];
echo  "<br/>Thanks for your selection $username <hr/>" ;
$msg = "You really enjoy $dish <br/>";
$msg .= "- especially with a nice $color wine";
echo $msg;
echo "<hr/>";
echo '<a href= " '. $self . ' " > Back to Form  </a>';
}
else
{
?>
```

ALL-IN-ONE FORM



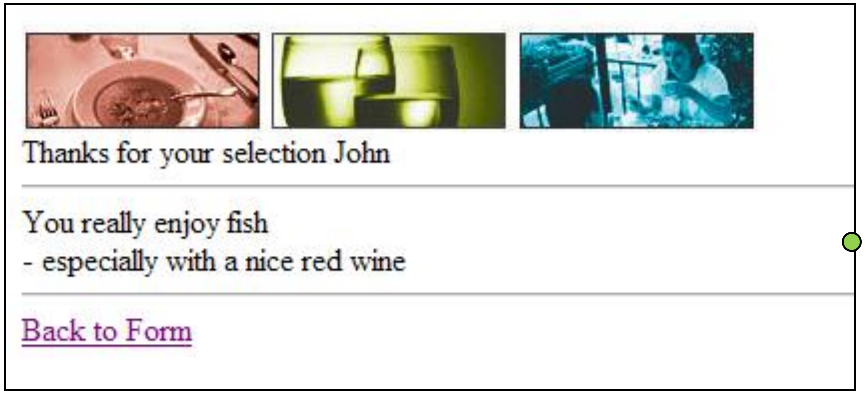
Please enter your first name:

Please select your favorite color wine:

○ White ○ Rosé ● Red

Please enter your favorite dish: fish

Form will display again
If form incomplete



Thanks for your selection John

You really enjoy fish
- especially with a nice red wine

Back to Form

Response to completed form

USER-DEFINED FUNCTIONS

- A function is a piece of PHP code that can be executed once or many times by the PHP script.
- User defined functions have the form

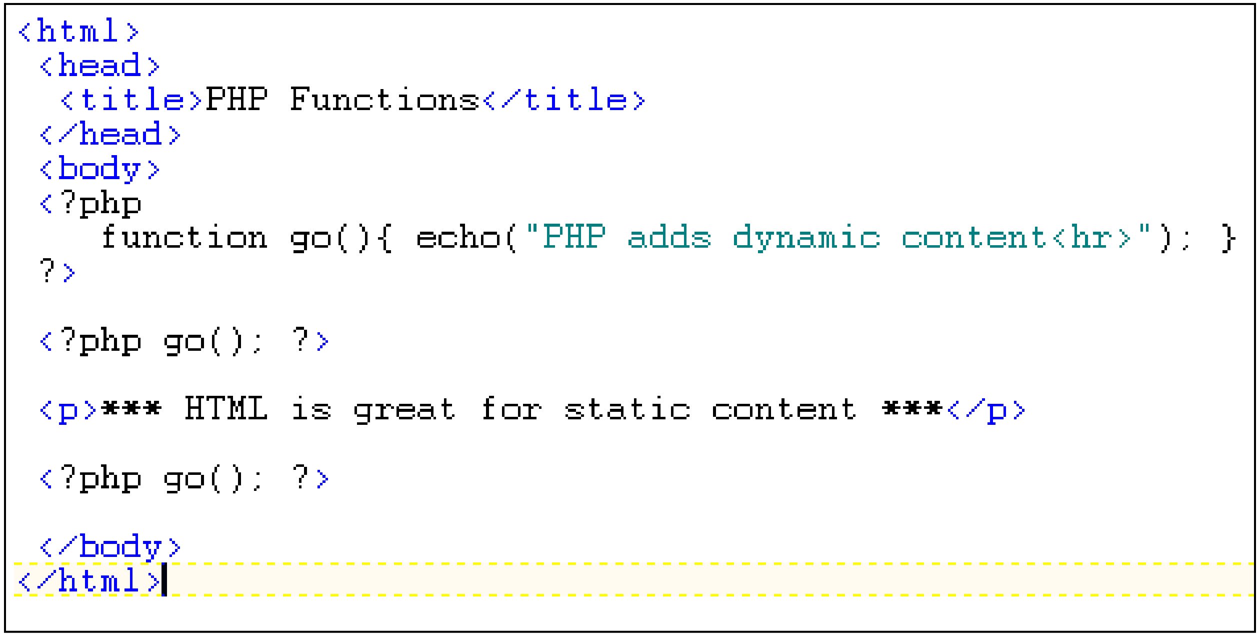
Use keyword function()

```
function name(arg_list)
{

    statements;

}
```

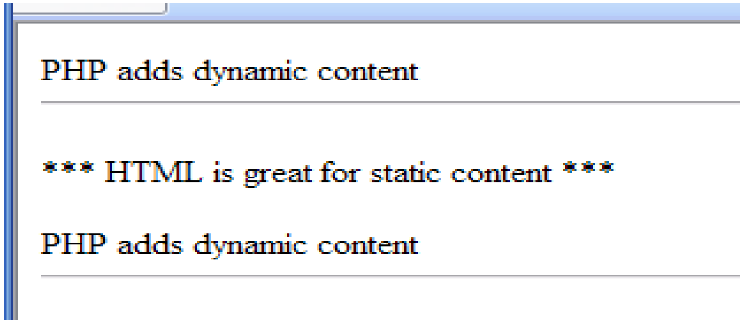
Pass the arguments here

FUNCTION: EXAMPLE 1

Function called go()



```
<html>
 <head>
  <title>PHP Functions</title>
 </head>
 <body>
 <?php
    function go(){ echo("PHP adds dynamic content<hr>"); }
 ?>

 <?php go(); ?>

 <p>*** HTML is great for static content ***</p>

 <?php go(); ?>

 </body>
</html>
```

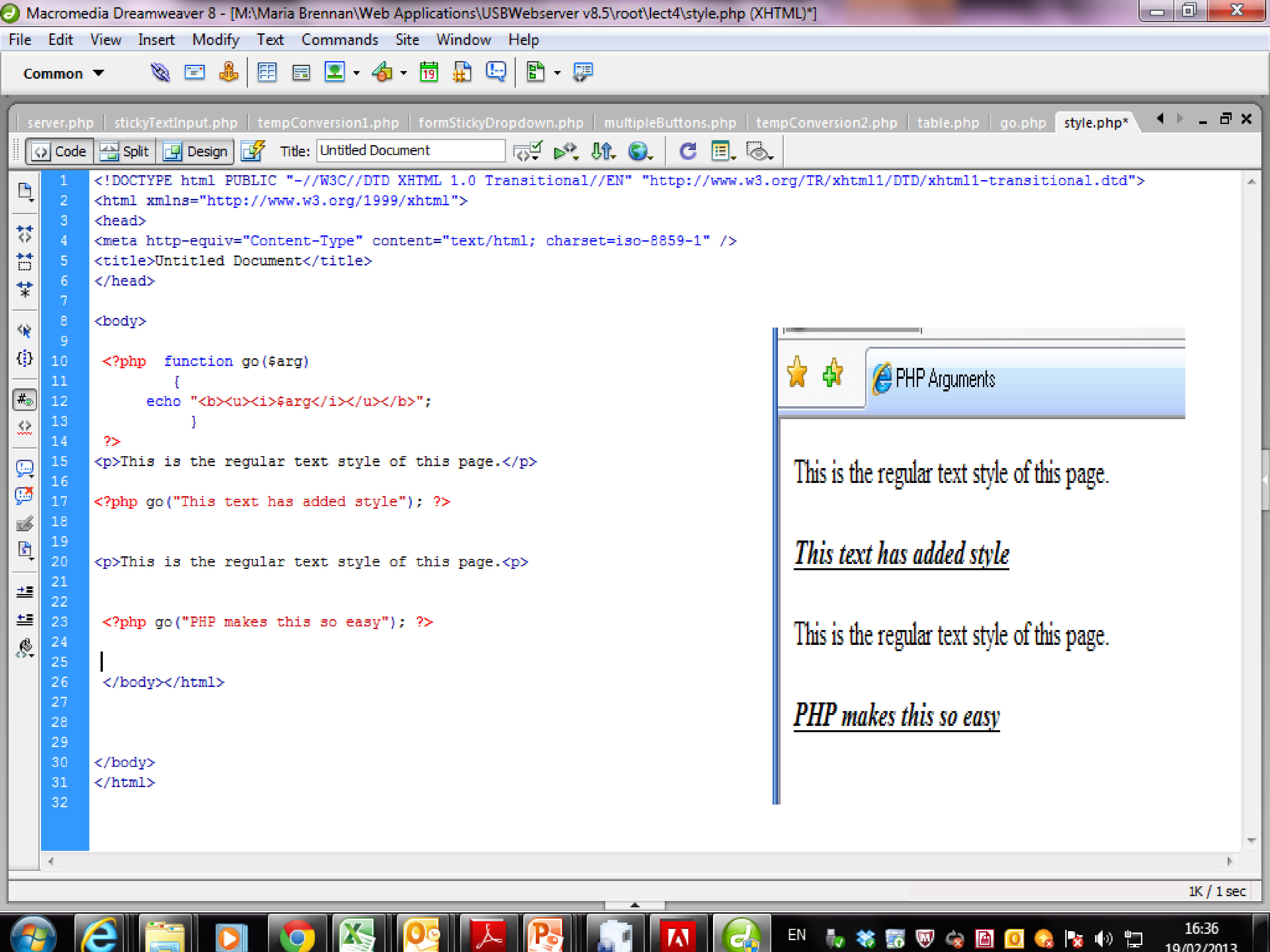


```
PHP adds dynamic content
________________________________

*** HTML is great for static content ***

PHP adds dynamic content
________________________________
```

FUNCTION; EXAMPLE 2

Function passes string arguments to the function go()

```
<html><head> <title>PHP Arguments</title></head> <body>
 <?php  function go($arg)
        {
              echo "<b><u><i>$arg</i></u></b>";
         }
 ?>
<p>This is the regular text style of this page.</p>
<?php go("This text has added style"); ?>
<p>This is the regular text style of this page.<p>
 <?php go("PHP makes this so easy"); ?>
 </body></html>
```


Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\style.php (XHTML)*]
File Edit View Insert Modify Text Commands Site Window Help
Common
server.php stickyTextInput.php tempConversion1.php formStickyDropdown.php multipleButtons.php tempConversion2.php table.php go.php style.php*
Code Split Design
Title: Untitled Document
PHP Arguments
1K / 1 sec
EN
16:36
19/02/2013

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
</head>

<body>

 <?php  function go($arg)
         {
      echo "<b><u><i>$arg</i></u></b>";
          }
 ?>
<p>This is the regular text style of this page.</p>

<?php go("This text has added style"); ?>


<p>This is the regular text style of this page.<p>


 <?php go("PHP makes this so easy"); ?>


 </body></html>



</body>
</html>

```

This is the regular text style of this page.

<u>This text has added style</u>

This is the regular text style of this page.

<u>PHP makes this so easy</u>

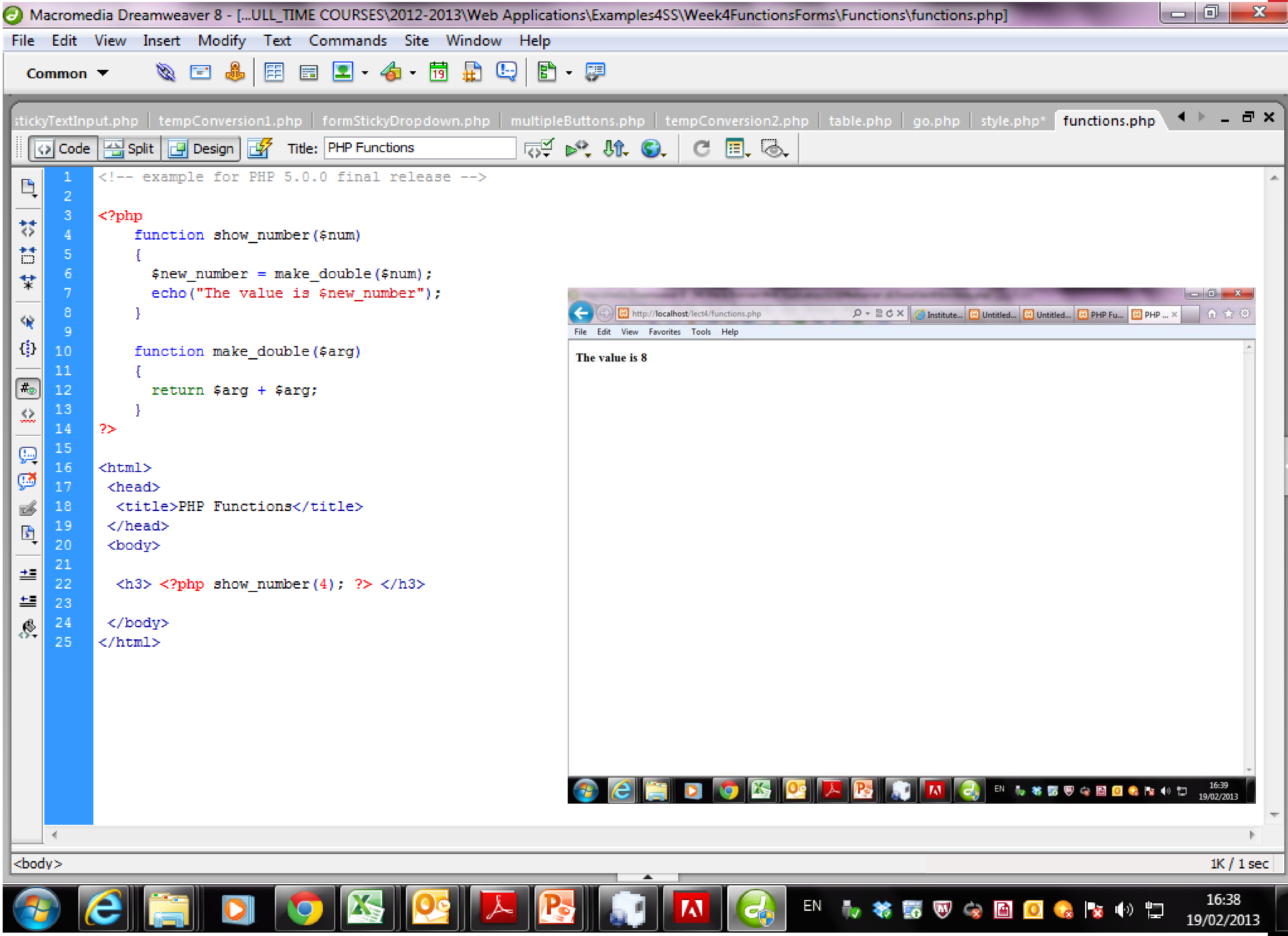
FUNCTION; EXAMPLE 3

Functions may call other functions when their code is being executed

```
<?php
function show_number($num)
          {
            $new_number = make_double($num);
            echo "The value is $new_number";
          }
function make_double($arg) {
            return $arg + $arg;
          }
?>
<html> <head>  <title>PHP Functions</title></head>
<body> <h3> <?php show_number(4); ?> </h3>
 </body></html>
```

The value is ____

functions.php



Macromedia Dreamweaver 8 - [...ULL_TIME COURSES\2012-2013\Web Applications\Examples4SS\Week4FunctionsForms\Functions\functions.php]
File Edit View Insert Modify Text Commands Site Window Help
Common
stickyTextInput.php | tempConversion1.php | formStickyDropdown.php | multipleButtons.php | tempConversion2.php | table.php | go.php | style.php* | functions.php
Code Split Design
Title: PHP Functions
http://localhost/lect4/functions.php
Institute... Untitled... Untitled... PHP Fu... PHP ...
File Edit View Favorites Tools Help
<body>
1K / 1 sec
EN
16:39 19/02/2013
16:38 19/02/2013

```
<!-- example for PHP 5.0.0 final release -->

<?php
    function show_number($num)
    {
      $new_number = make_double($num);
      echo("The value is $new_number");
    }

    function make_double($arg)
    {
      return $arg + $arg;
    }
?>

<html>
 <head>
  <title>PHP Functions</title>
 </head>
 <body>

  <h3> <?php show_number(4); ?> </h3>

 </body>
</html>
```

```
The value is 8
```

VARIABLE SCOPE

- Defines which parts of a PHP script have access to a variable
- Variables defined inside functions are local
- Local variables can only be used in the function in which they are declared
- Variables defined outside functions are global variables
- PHP requires a function to declare explicitly if it wants to use a global variable.

VARIABLE SCOPE

- In programming, scope refers to the context you declare a variable in.

- Most variables in PHP have a single scope: global.

- Using this scope means that a variable is available in the script that declares it, as well as in any script that is included after the variable is declared or in any script that includes the file in which the variable is declared.

VARIABLE SCOPE

For example, try opening test.php again and experimenting with variable scope:

```
<?php
$foo = "some value";
include_once 'extras.php'; // $foo is available in extras.php
$bar = "another value"; // $bar is not available in extras.php
echo "test.php: Foo is $foo, and bar is $bar. <br />";
?>
```

Now open up extras.php and insert the following code:

```
<?php
echo "extras.php: Foo is $foo, and bar is $bar. <br />";
?>
```

VARIABLE SCOPE

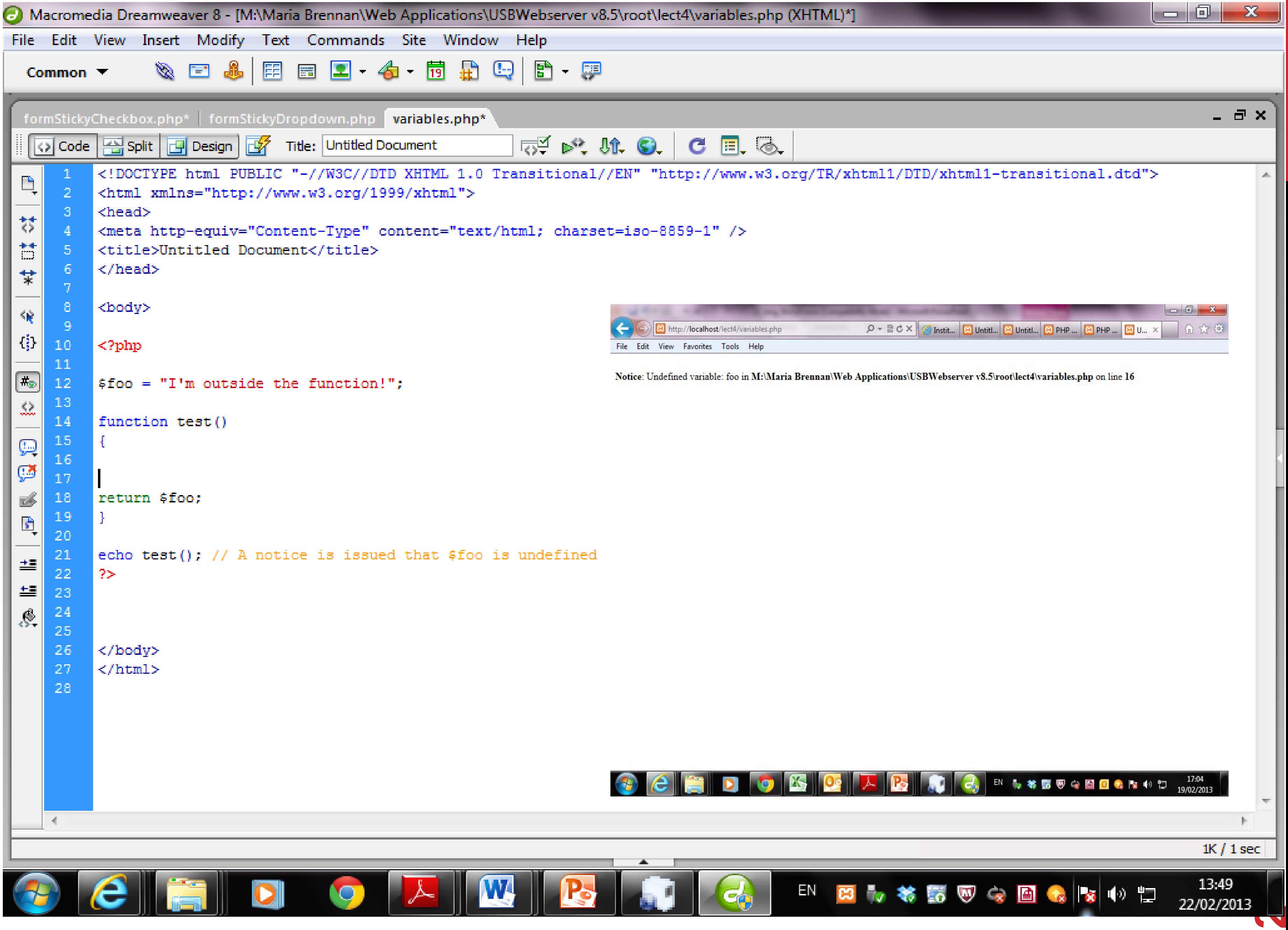
- Scope changes a bit when you start using functions, because variables declared within a function have local scope, meaning they’re only available within the function that declares them.

- Additionally, variables declared in the global scope are only available if they are explicitly declared as global inside the function.

VARIABLE SCOPE

```
<?php
$foo = "I'm outside the function!";

function test()
{
return $foo;
}

echo test(); // A notice is issued that $foo is undefined
?>
```

$foo is undefined if you run your test() function, which issues a notice.



Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\variables.php (XHTML)*]
File Edit View Insert Modify Text Commands Site Window Help
Common
formStickyCheckbox.php* formStickyDropdown.php variables.php*
Code Split Design Title: Untitled Document
http://localhost/lect4/variables.php
File Edit View Favorites Tools Help
1K / 1 sec
EN 13:49 22/02/2013

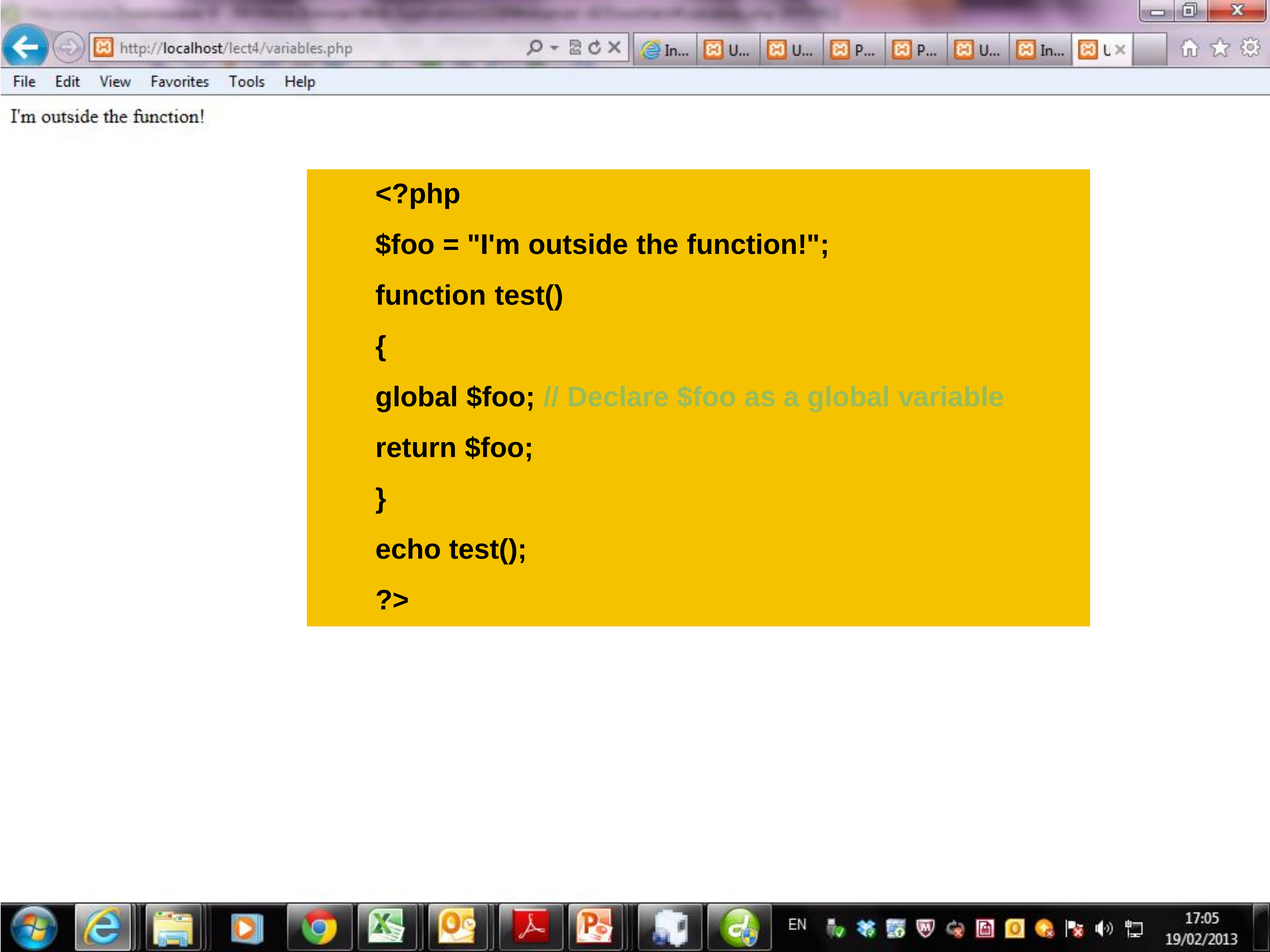
```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
5  <title>Untitled Document</title>
6  </head>
7
8  <body>
9
10 <?php
11
12 $foo = "I'm outside the function!";
13
14 function test()
15 {
16
17
18 return $foo;
19 }
20
21 echo test(); // A notice is issued that $foo is undefined
22 ?>
23
24
25
26 </body>
27 </html>
28
```

```
Notice: Undefined variable: foo in M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\variables.php on line 16
```

VARIABLE SCOPE

You can clear this up by declaring $foo as global within your test() function; this means that $foo in local scope will now refer to the variable $foo in global scope:

```
<?php
$foo = "I'm outside the function!";
function test()
{
global $foo; // Declare $foo as a global variable
return $foo;
}
echo test();
?>
```



I'm outside the function!

```
<?php
$foo = "I'm outside the function!";
function test()
{
global $foo; // Declare $foo as a global variable
return $foo;
}
echo test();
?>
```

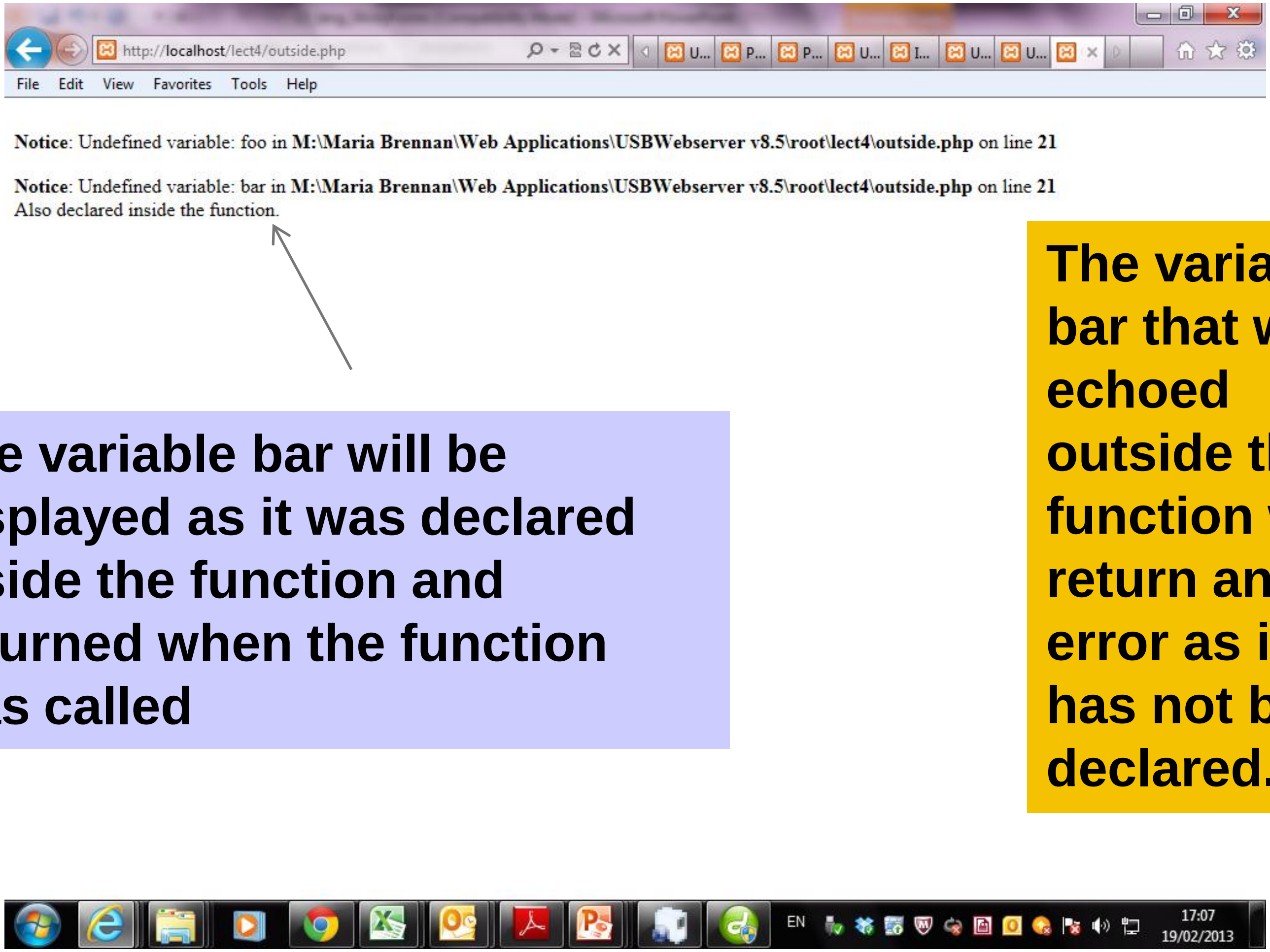
VARIABLE SCOPE

A variable declared within a function is not available outside that function unless it is specified as the function’s return value. For instance, consider the following code:

```
<?php

function test() {
    $foo = "Declared inside the function. <br />";
    $bar = "Also declared inside the function. <br />";
    return $bar;
}

$baz = test();

echo $foo, $bar, $baz;
?>
```

VARIABLE SCOPE

This code gives the following results:



```
Notice: Undefined variable: foo in M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\outside.php on line 21

Notice: Undefined variable: bar in M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\outside.php on line 21
Also declared inside the function.
```

The variable bar will be displayed as it was declared inside the function and returned when the function was called

The variable bar that was echoed outside the function will return an error as it has not been declared.

VARIABLE SCOPE

You need to declare two variables within a function and return both; next, you use an array and

the list() function to access the values easily.

```
<?php
function test()
{
$foo = "Value One";
$bar = "Value Two";
return array($foo, $bar);
}
/*
* The list() function allows us to assign a variable
* to each array index as a comma-separated list
*/
list($one, $two) = test();
echo $one, "<br />", $two, "<br />";
?>
```

VARIABLE SCOPE

Running this code produces the desired output:

```
Value One
Value Two
```

Using list() is a way to declare multiple variables in one line; for example this line declares the variables $one and $two:

```
list($one, $two) = test();
```

The following handful of lines accomplishes the same thing:

```
$array = test();
$one = $array[0];
$two = $array[1];
```

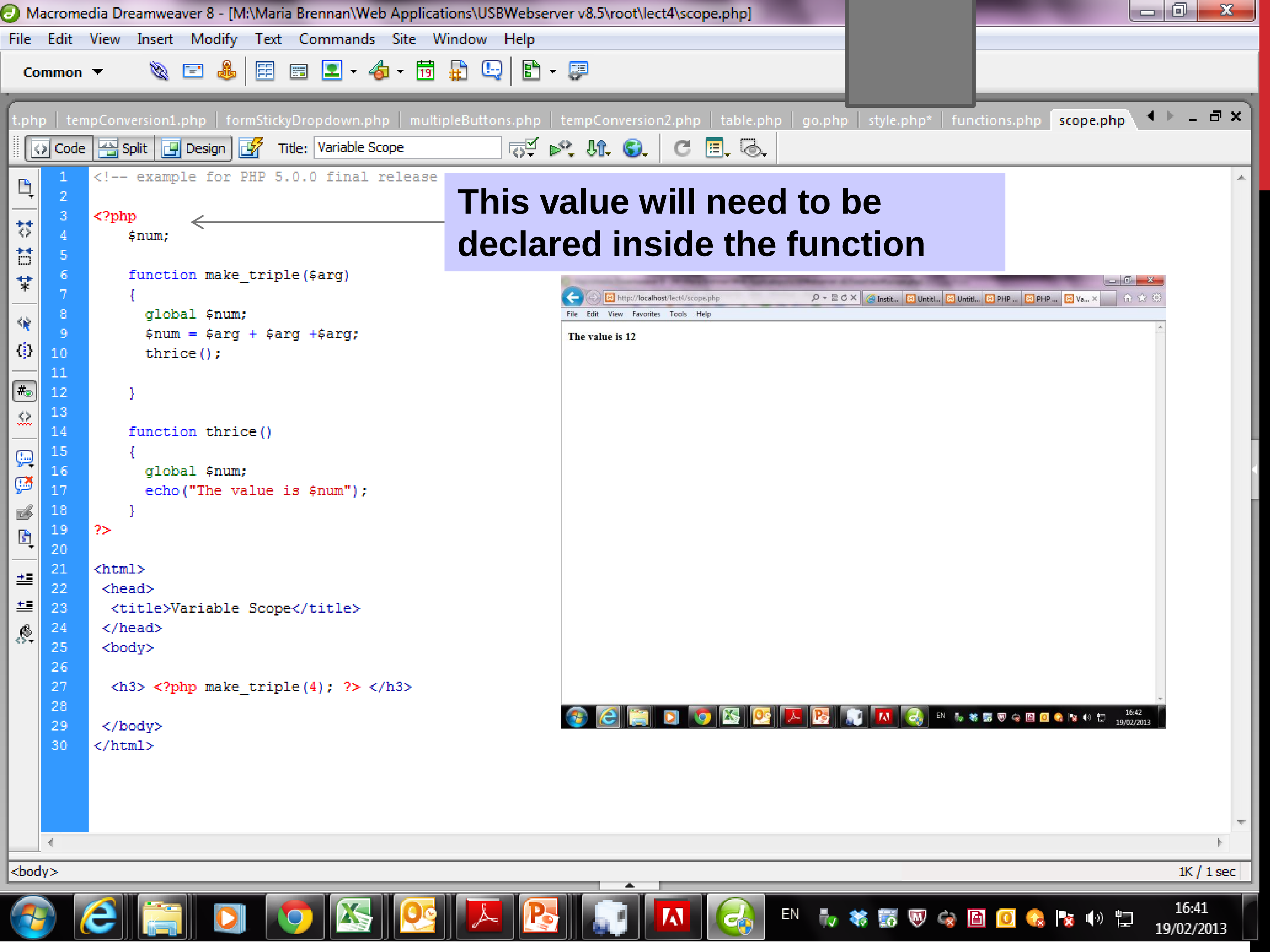
VARIABLE SCOPE

```
<?php   $num;
function make_triple($arg){
            global $num;
            $num = $arg + $arg +$arg;
            thrice();
}
function thrice() {
            global $num;
            echo "The value is $num";} ?>
<html><head><title>Variable
Scope</title>
 </head> <body>
  <h3> <?php make_triple(4); ?> </h3>
 </body> </html>
```

`$num` is defined outside the functions & must be called explicitly in the two functions.

The value is ____

scope.php


Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\scope.php]
File Edit View Insert Modify Text Commands Site Window Help
Common
t.php tempConversion1.php formStickyDropdown.php multipleButtons.php tempConversion2.php table.php go.php style.php* functions.php scope.php
Code Split Design
Title: Variable Scope
<body>
1K / 1 sec
http://localhost/lect4/scope.php
File Edit View Favorites Tools Help

```
<!-- example for PHP 5.0.0 final release

<?php
    $num;

    function make_triple($arg)
    {
      global $num;
      $num = $arg + $arg +$arg;
      thrice();

    }

    function thrice()
    {
      global $num;
      echo("The value is $num");
    }
?>

<html>
 <head>
  <title>Variable Scope</title>
 </head>
 <body>

  <h3> <?php make_triple(4); ?> </h3>

 </body>
</html>
```

This value will need to be declared inside the function

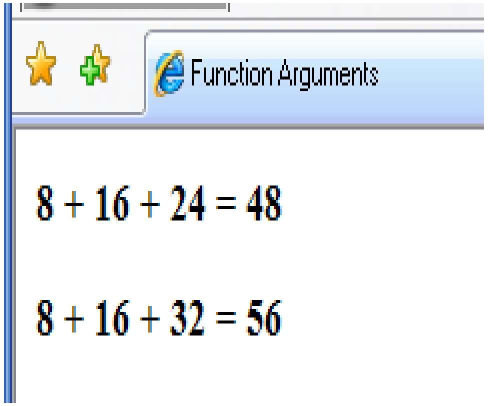
```
The value is 12
```

FUNCTIONS: MULTIPLE ARGUMENTS

Multiple arguments can be passed to a function – use commas to separate

Three arguments specified, with a default value if none passed from caller

```
<?php
                function addup( $a = 32, $b = 32, $c = 32){
                  $total = $a + $b + $c;
                  echo "$a + $b + $c = $total";
}?>
<html> <head> <title>Function Arguments
</title> </head><body>
 <h3> <?php addup(8, 16, 24); ?> </h3>
 <h3> <?php addup(8, 16); ?> </h3>
 </body></html>
```



```
8 + 16 + 24 = 48

8 + 16 + 32 = 56
```

args.php

INCLUDING FILES

A file can be included in another file using include and require

`include()`, `include_once()`

- inherits the scope of the include point
- parsing is in HTML mode so code must use the php tags `<?php ... ?>`

`require`, `require_once`

- like include but can cause a fatal error if file doesn't exist

INCLUDE, INCLUDE_ONCE

- **A great feature provided by PHP is the ability to load a script from an external file**
- **This makes it much easier to organize your code in larger projects.**
- **PHP provides four constructs you can use to load an external script:**
  - include,
  - include_once,
  - Require
  - Require_once

INCLUDE, INCLUDE_ONCE

- The PHP manual recommends that developers use include_once and require_once because these constructs first check whether the file has already been loaded before either will load a given script.

- This saves resources and can increase the performance of your applications

INCLUDE, INCLUDE_ONCE

Now let’s take a look at an exercise that illustrates the power of loading external scripts.

Lets create a file called extras.php with the following content

```
<?php
$foo = "green";
$bar = "red";
?>
```

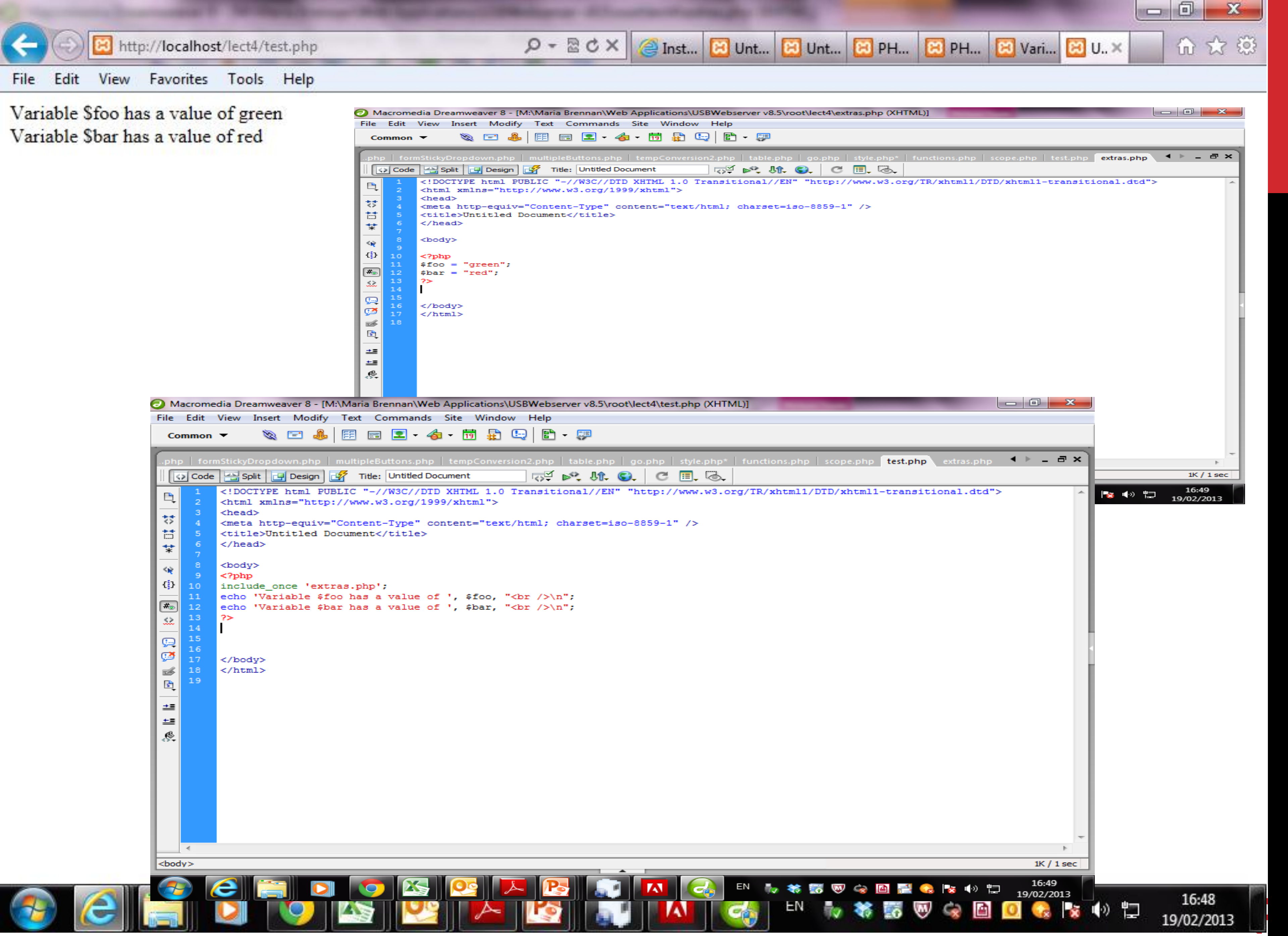
INCLUDE, INCLUDE_ONCE

Lets create a file called test.php with the following content

```
<?php
include_once 'extras.php';
echo 'Variable $foo has a value of ', $foo, "<br />\n";
echo 'Variable $bar has a value of ', $bar, "<br />\n";
?>
```

We will get the following output

```
Variable $foo has a value of green
Variable $bar has a value of red
```


http://localhost/lect4/test.php
File Edit View Favorites Tools Help
Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\extras.php (XHTML)]
Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\test.php (XHTML)]
File Edit View Insert Modify Text Commands Site Window Help
Common
.php formStickyDropdown.php multipleButtons.php tempConversion2.php table.php go.php style.php* functions.php scope.php test.php extras.php
Code Split Design Title: Untitled Document
<body>
1K / 1 sec
16:49 19/02/2013
16:48 19/02/2013
EN

Variable $foo has a value of green
Variable $bar has a value of red

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
</head>

<body>

<?php
$foo = "green";
$bar = "red";
?>


</body>
</html>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
</head>

<body>
<?php
include_once 'extras.php';
echo 'Variable $foo has a value of ', $foo, "<br />\n";
echo 'Variable $bar has a value of ', $bar, "<br />\n";
?>



</body>
</html>
```

INCLUDE, INCLUDE_ONCE

- By including the extras.php file you created using include_once, you are able to access the information stored in the file.

- This proves especially useful when you're working with a large set of functions, which allows common functions to be stored in a file that is included in other areas of your site, rather than requiring that you copy-and-paste those functions into each file.

- Adopting this approach reduces the size of your applications and can play a part in optimizing your application's performance.

INCLUDE, INCLUDE_ONCE

This next short example illustrates how using include_once can reduce the load on your server;

begin by adding this code to extras.php:

```
<?php
$var += 1;
?>
```

INCLUDE, INCLUDE_ONCE

Next, add this code to test.php:

```
<?php
$var = 0;

include 'extras.php';
echo $var, "<br />";

include 'extras.php';
echo $var, "<br />";
?>
```

This code produces the following output when loaded into a browser:

```
1
2
```

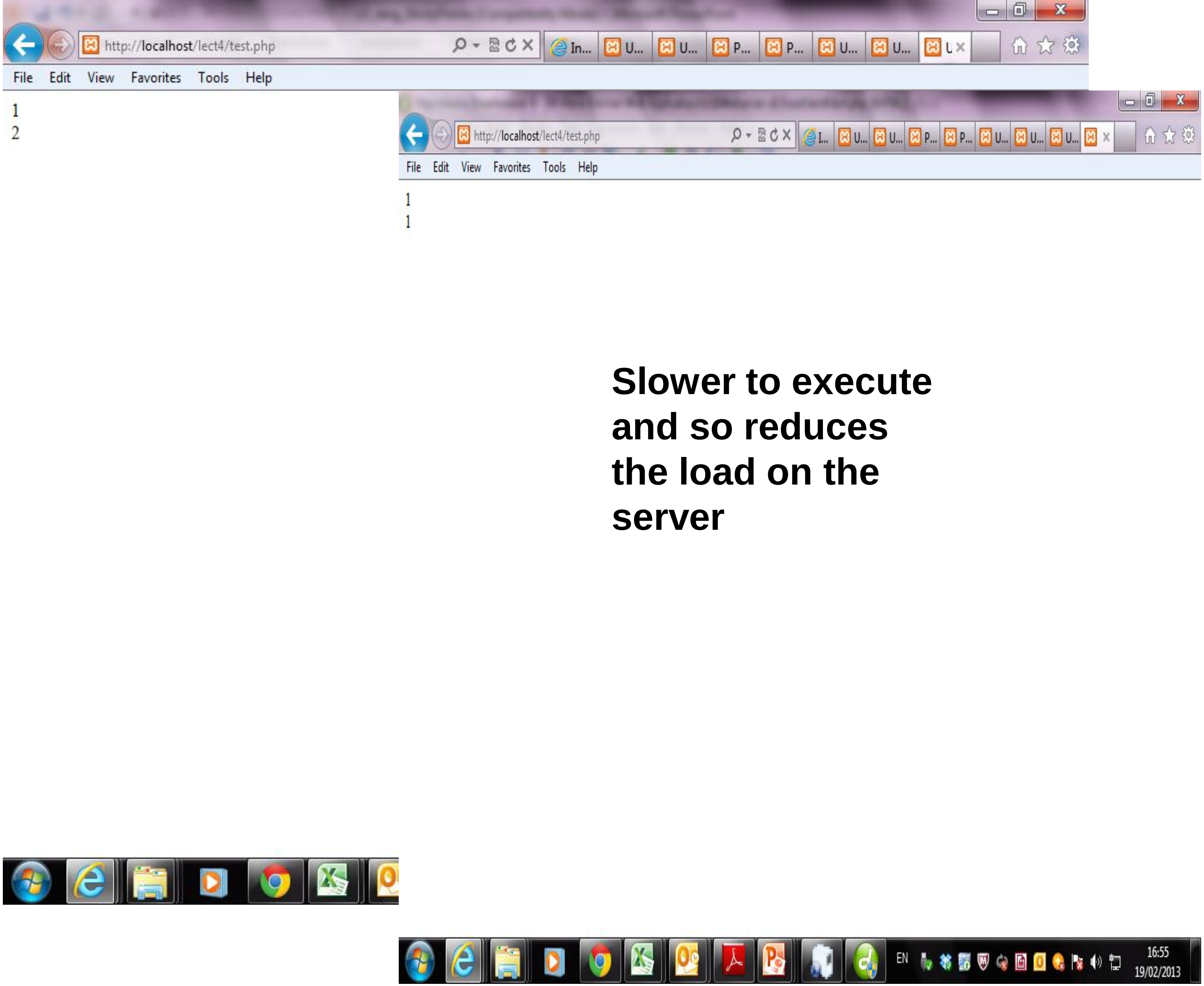
INCLUDE, INCLUDE_ONCE

Now, change test.php so it uses include_once instead of include:

```
<?php
$var = 0;

include_once 'extras.php';
echo $var, "<br />";

include_once 'extras.php';
echo $var, "<br />";
?>
```

Next, load test.php in a browser to see the result:

```
1
1
```



http://localhost/lect4/test.php
File Edit View Favorites Tools Help

```
1
2
```

http://localhost/lect4/test.php
File Edit View Favorites Tools Help

```
1
1
```

Slower to execute and so reduces the load on the server

SUPERGLOBAL ARRAYS

PHP offers several types of superglobal arrays to developers, each with a different useful purpose. A superglobal array is a special variable that are always available in scripts, regardless of the current scope of the script.

PHP includes several superglobals:

$GLOBALS: Variables available in the global scope
$_SERVER: Information about the server
$_GET: Data passed using the HTTP GET method
$_POST: Data passed using the HTTP POST method
$_REQUEST: Data passed via an HTTP request
$_FILES: Data passed by an HTML file input
$_SESSION: Current session data specific to the user
$_COOKIE: Data stored on the user’s browser as a cookie

$GLOBALS

- PHP provides another option for accessing variables in the global scope: the $GLOBALS superglobal array.

- All variables in the global scope are loaded into the $GLOBALS array, enabling you to access them using the variable name as the array key.

- You can try out this array in test.php:

$GLOBALS

```php
<?php

$foo = "Some value.";

function test()
{
echo $GLOBALS['foo'];
}

test();

?>
```

This code produces the following output:

```
Some value.
```

$GLOBALS

- **Tip is generally a good practice to avoid using globals wherever possible.**

- **The preferred method of accessing global variables inside functions is to pass them as arguments.**

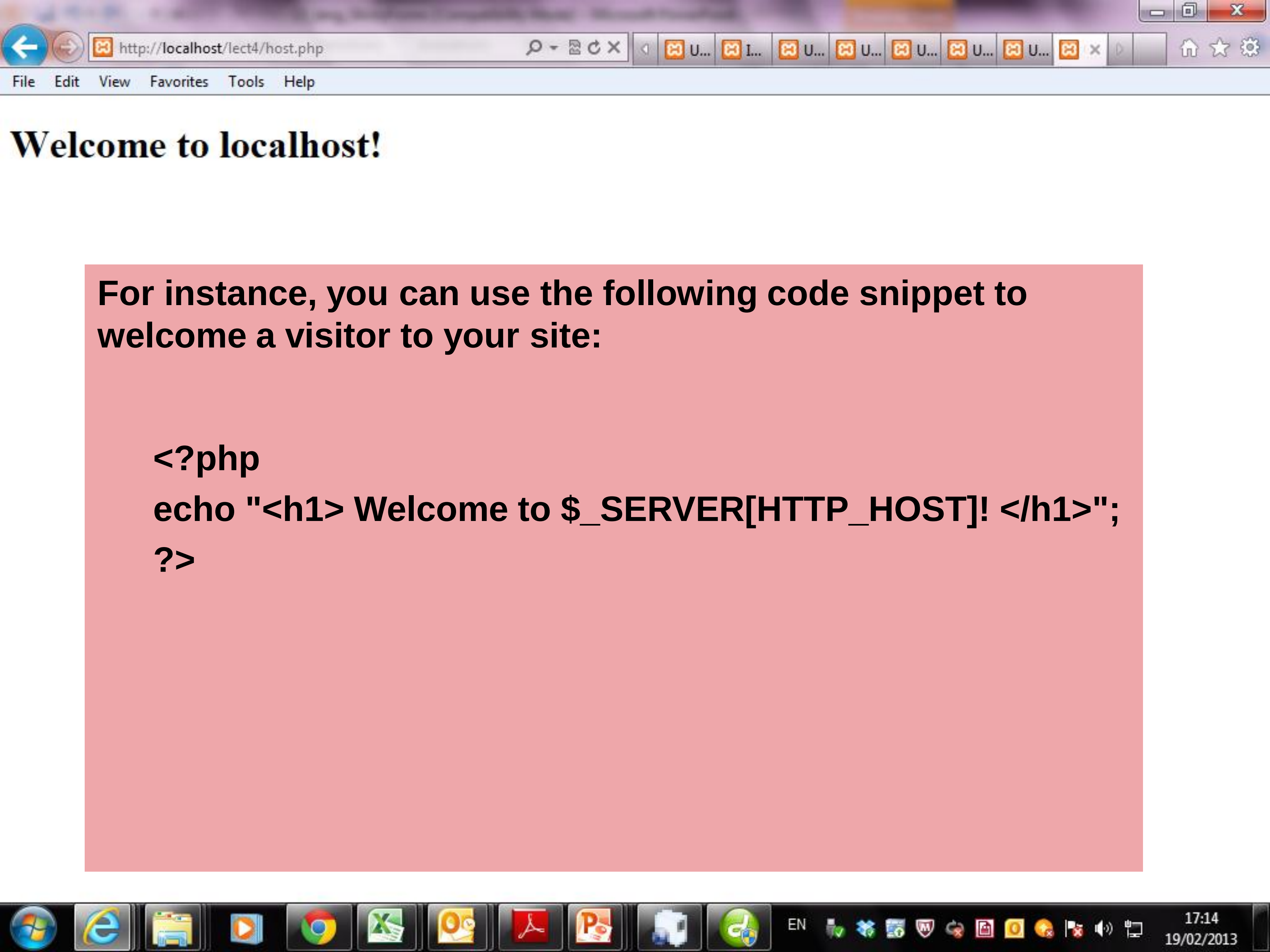
- **This makes your scripts more readable, which simplifies maintenance over the long term.**

$_SERVER

- The $_SERVER superglobal stores information about the server and the current script.

- It also has features that allow you to access the IP address of a site visitor, what site referred the visitor to this script, and many other useful pieces of information.

- One of the most useful pieces of information available in the $_SERVER superglobal is the name of the host site, which is stored in HTTP_HOST.



Welcome to localhost!

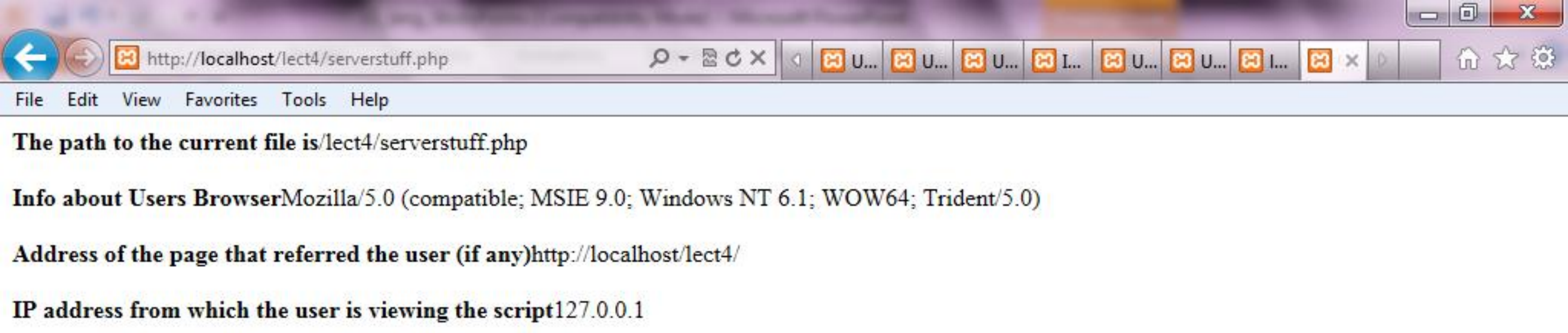
For instance, you can use the following code snippet to welcome a visitor to your site:

```
<?php
echo "<h1> Welcome to $_SERVER[HTTP_HOST]! </h1>";
?>
```

$_SERVER

```
<?php
// Path to the current file (i.e. '/simple_blog/test.php')
echo $_SERVER['PHP_SELF'], "\n\n";
// Information about the user's browser
echo $_SERVER['HTTP_USER_AGENT'], "\n\n";
// Address of the page that referred the user (if any)
echo $_SERVER['HTTP_REFERER'], "\n\n";
// IP address from which the user is viewing the script
echo $_SERVER['REMOTE_ADDR'], "\n\n";
// Human-readable export of the contents of $_SERVER
print_r($_SERVER);
?>
```

Note: the use of print_r() at the bottom of the script. This is a great way to debug code, especially arrays, because it outputs a “human-readable” display of a variable’s contents. You can see this at work when you load your test script in test.php:

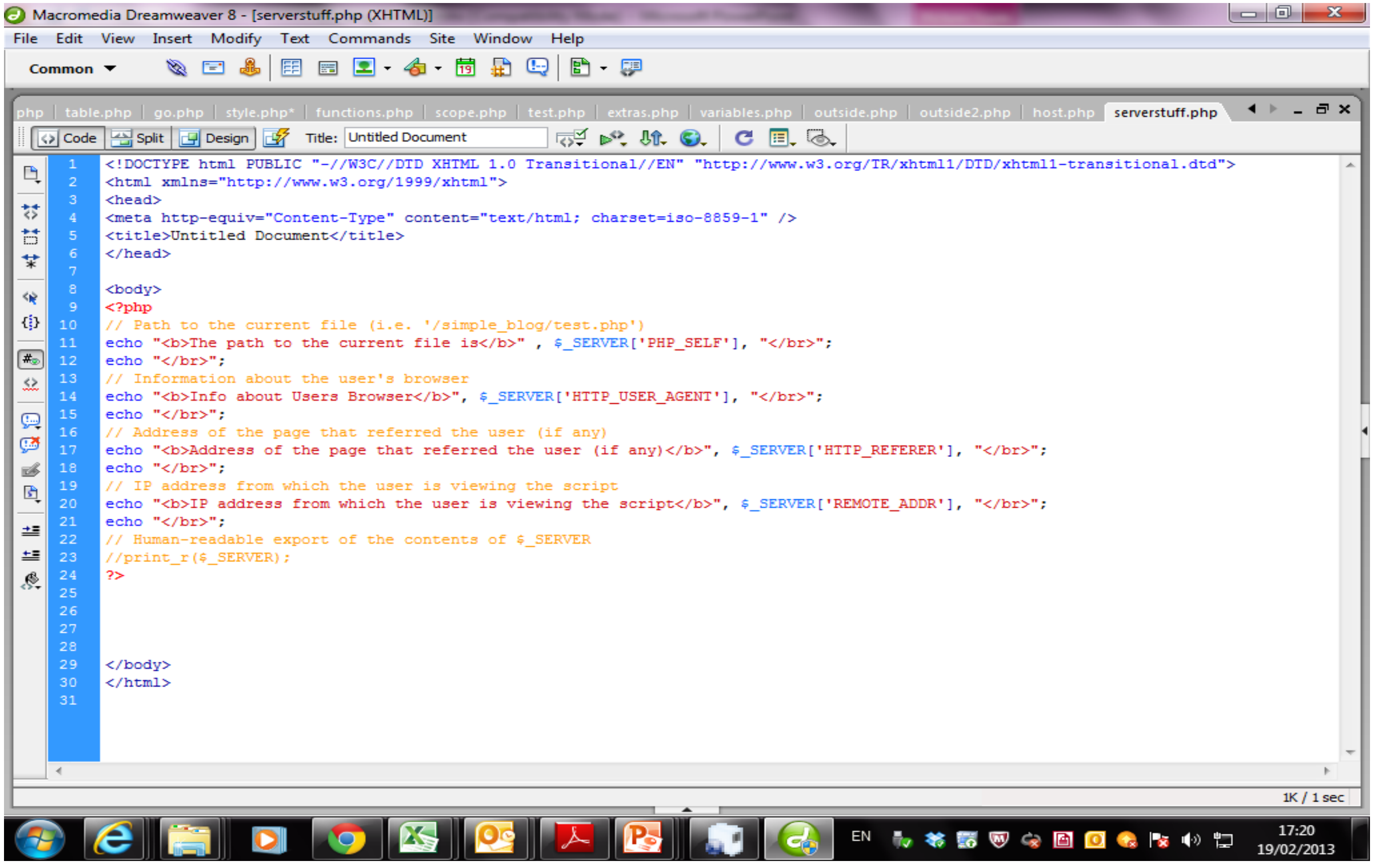


The path to the current file is/lect4/serverstuff.php

Info about Users Browser Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)

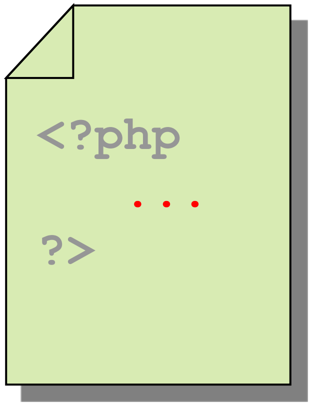
Address of the page that referred the user (if any)http://localhost/lect4/

IP address from which the user is viewing the script127.0.0.1



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
</head>

<body>
<?php
// Path to the current file (i.e. '/simple_blog/test.php')
echo "<b>The path to the current file is</b>" , $_SERVER['PHP_SELF'], "</br>";
echo "</br>";
// Information about the user's browser
echo "<b>Info about Users Browser</b>", $_SERVER['HTTP_USER_AGENT'], "</br>";
echo "</br>";
// Address of the page that referred the user (if any)
echo "<b>Address of the page that referred the user (if any)</b>", $_SERVER['HTTP_REFERER'], "</br>";
echo "</br>";
// IP address from which the user is viewing the script
echo "<b>IP address from which the user is viewing the script</b>", $_SERVER['REMOTE_ADDR'], "</br>";
echo "</br>";
// Human-readable export of the contents of $_SERVER
//print_r($_SERVER);
?>




</body>
</html>

```

MORE FORM PROCESSING

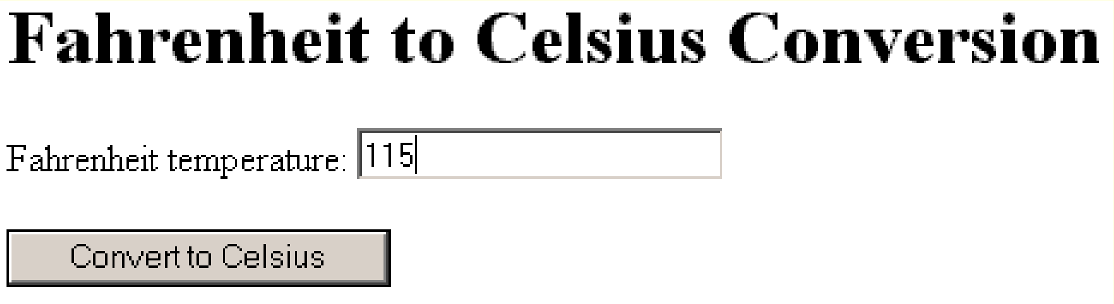


TOPICS

- Form Processing
- Temperature Conversion Program
- Using PHP_SELF
- Multiple Buttons
- Error Checking using Regular Expressions
- Sticky Input Fields – Text, Checkboxes, Radio Buttons, Dropdown List

TEMPERATURE CONVERSION

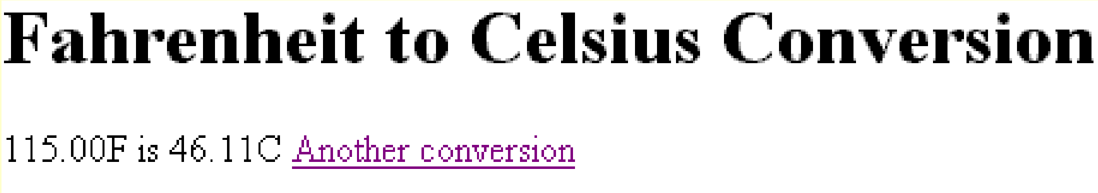
Program to convert Fahrenheit to Celsius



Fahrenheit to Celsius Conversion

Fahrenheit temperature: 115

input page



Fahrenheit to Celsius Conversion

115.00F is 46.11C Another conversion

output page

TEMPERATURE CONVERSION

Approach

- Put all the code in one file
- Code the form including the submit button
  - <input type = “submit” name = “convert” value = “Convert to Celsius”/>
- Form calls itself
- <form action = “<?php echo $self ?>” method = “post”>
- Insert code to check if the Convert button has been pressed
  - If pressed – perform the calculations – provide a link to allow the user to do another conversion
  - If not pressed – display the form

APPROACH

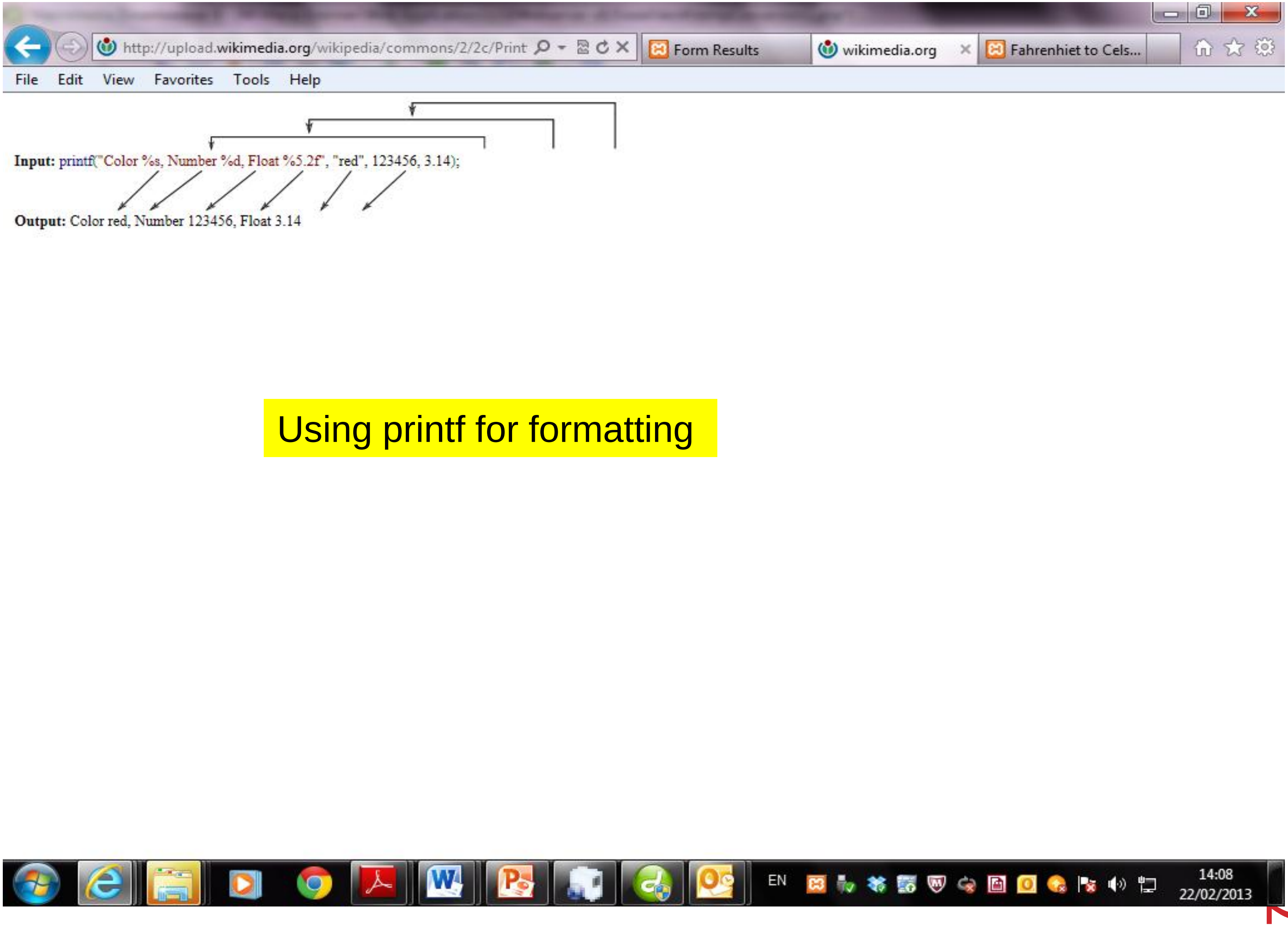
- Must decide whether to display the form or do the calculations and display the results:
- Use button name - convert
- It will be sent to the PHP script only if it is clicked.

```
if (isset($_REQUEST['convert'])) // button clicked
{
   // do the conversion and display results
}
else
{
   // display the form
}
```

TEMPCONVERSION.PHP

```
<html><head>
<title>Fahrenheit to Celsius Conversion</title>
</head><body>
<h1>Fahrenheit to Celsius Conversion</h1>
<?php
   $self = $_SERVER['PHP_SELF'];
   if (isset($_REQUEST['convert']))
   {  // do the conversion and display results

     $fahr = $_REQUEST['fahrenheit'];
      $celsius = ($fahr - 32)*(5.0/9.0);
      printf("%.2fF is %.2fC", $fahr, $celsius);
    echo " <a href= ‘$self’ >Another conversion</a>"
   }
```

Using printf for formatting



Input: printf("Color %s, Number %d, Float %5.2f", "red", 123456, 3.14);

Output: Color red, Number 123456, Float 3.14

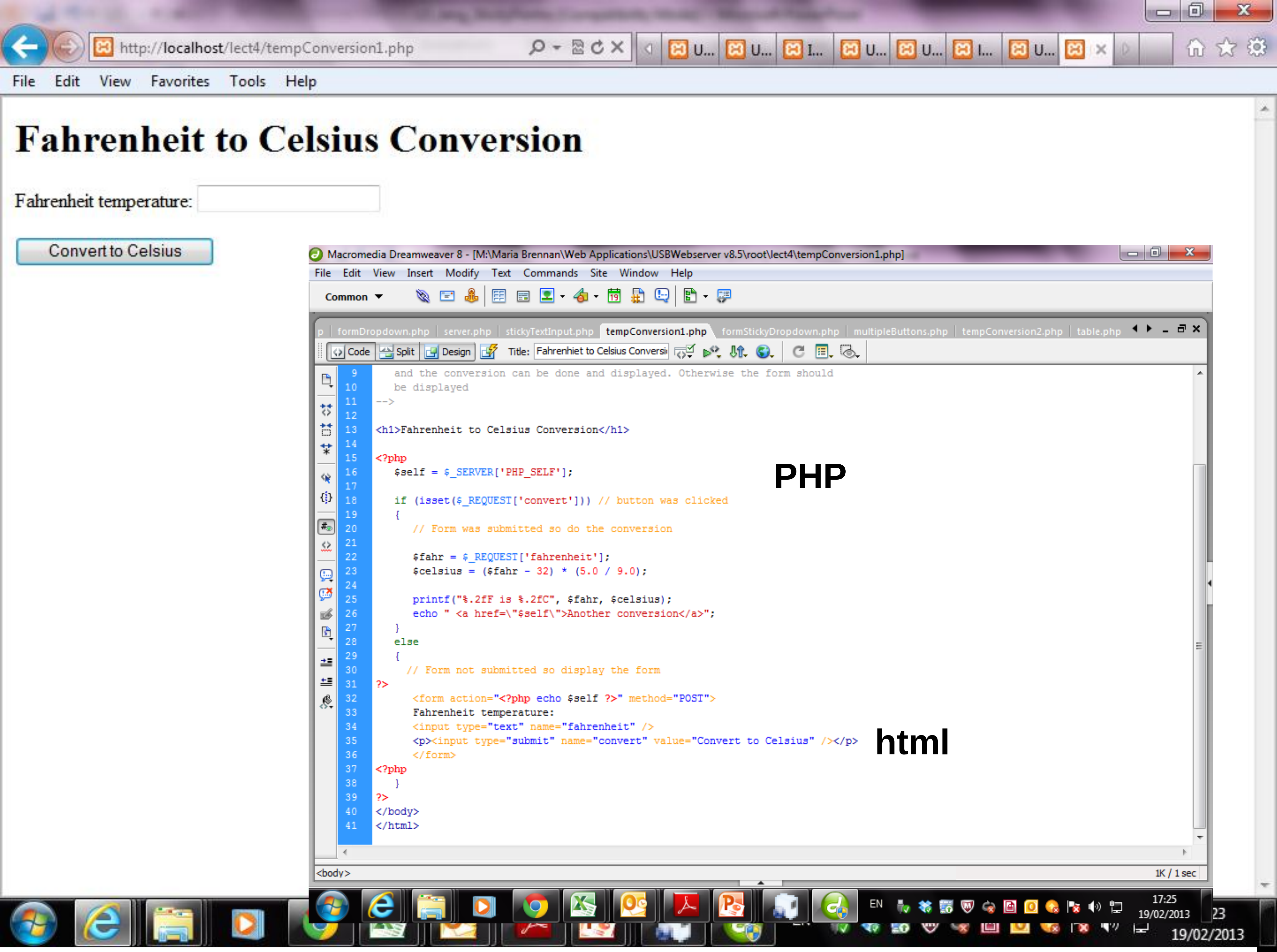
Using printf for formatting

….Code continued

TEMPCONVERSION.PHP

```
   else
   {
<?php
   <form action="<?php echo $self ?>" method="POST">
   Fahrenheit Temperature:
   <input type="text" name="fahrenheit" />
   <p><input type="submit" name="convert"
      value="Convert to Celsius" /></p>
   </form>


   }
?>
</body></html>
```

forms/tempConversion1.php



http://localhost/lect4/tempConversion1.php
File Edit View Favorites Tools Help
Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\tempConversion1.php]
File Edit View Insert Modify Text Commands Site Window Help
Common
formDropdown.php server.php stickyTextInput.php tempConversion1.php formStickyDropdown.php multipleButtons.php tempConversion2.php table.php
Code Split Design Title: Fahrenhiet to Celsius Conversi
<body>
1K / 1 sec
17:25
19/02/2013

Fahrenheit to Celsius Conversion

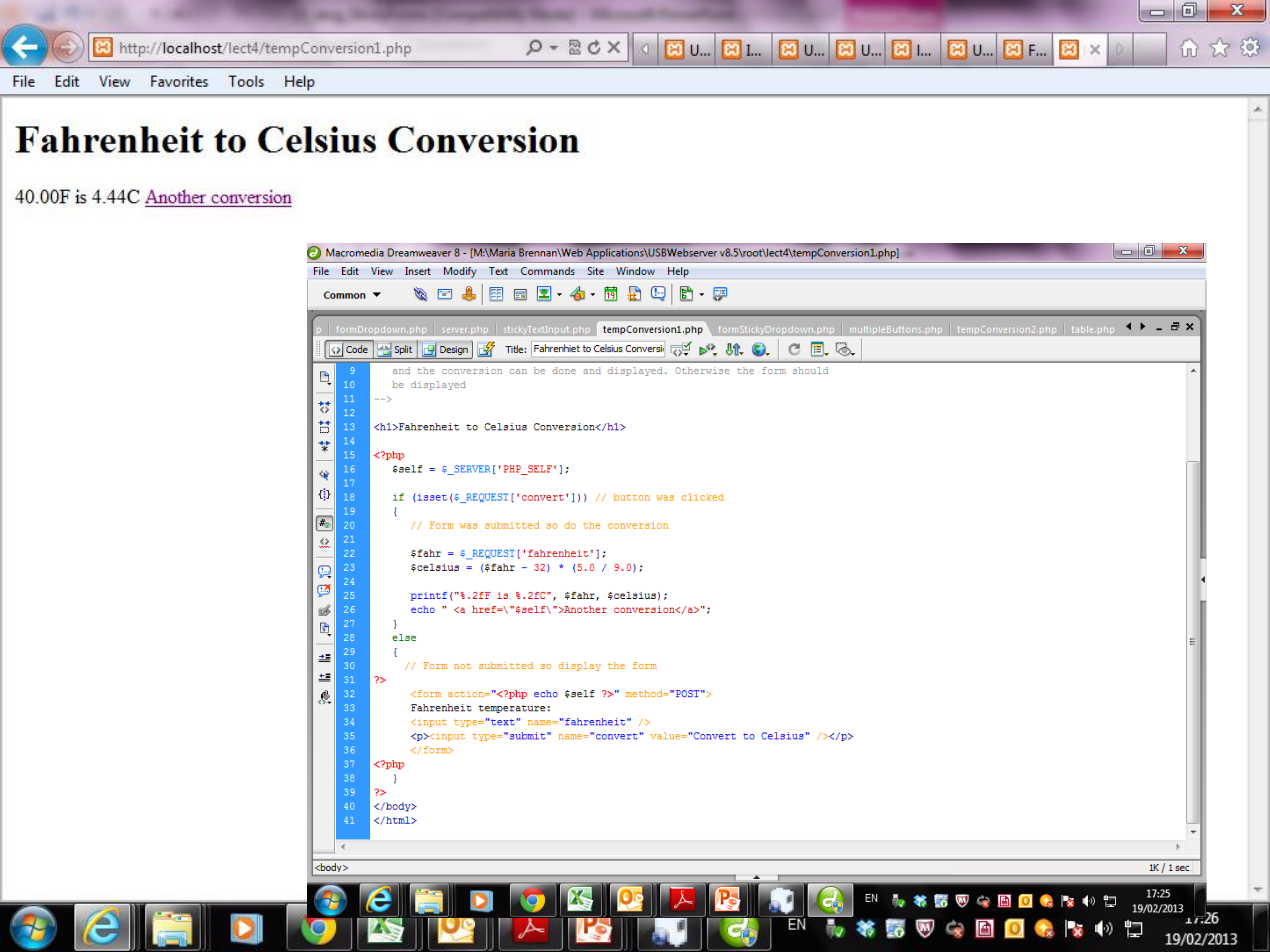
Fahrenheit temperature:

Convert to Celsius

```
   and the conversion can be done and displayed. Otherwise the form should
   be displayed
-->

<h1>Fahrenheit to Celsius Conversion</h1>

<?php
   $self = $_SERVER['PHP_SELF'];

   if (isset($_REQUEST['convert'])) // button was clicked
   {
      // Form was submitted so do the conversion

      $fahr = $_REQUEST['fahrenheit'];
      $celsius = ($fahr - 32) * (5.0 / 9.0);

      printf("%.2fF is %.2fC", $fahr, $celsius);
      echo " <a href=\"$self\">Another conversion</a>";
   }
   else
   {
     // Form not submitted so display the form
?>
      <form action="<?php echo $self ?>" method="POST">
      Fahrenheit temperature:
      <input type="text" name="fahrenheit" />
      <p><input type="submit" name="convert" value="Convert to Celsius" /></p>
      </form>
<?php
   }
?>
</body>
</html>
```

PHP

html


http://localhost/lect4/tempConversion1.php
File Edit View Favorites Tools Help
Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\tempConversion1.php]
File Edit View Insert Modify Text Commands Site Window Help
Common
formDropdown.php server.php stickyTextInput.php tempConversion1.php formStickyDropdown.php multipleButtons.php tempConversion2.php table.php
Code Split Design
Title: Fahrenheit to Celsius Conversi
<body>
1K / 1 sec
EN
17:25
19/02/2013
17:26
19/02/2013

Fahrenheit to Celsius Conversion

40.00F is 4.44C Another conversion

```
   and the conversion can be done and displayed. Otherwise the form should
   be displayed
-->

<h1>Fahrenheit to Celsius Conversion</h1>

<?php
   $self = $_SERVER['PHP_SELF'];

   if (isset($_REQUEST['convert'])) // button was clicked
   {
      // Form was submitted so do the conversion

      $fahr = $_REQUEST['fahrenheit'];
      $celsius = ($fahr - 32) * (5.0 / 9.0);

      printf("%.2fF is %.2fC", $fahr, $celsius);
      echo " <a href=\"$self\">Another conversion</a>";
   }
   else
   {
     // Form not submitted so display the form
?>
      <form action="<?php echo $self ?>" method="POST">
      Fahrenheit temperature:
      <input type="text" name="fahrenheit" />
      <p><input type="submit" name="convert" value="Convert to Celsius" /></p>
      </form>
<?php
   }
?>
</body>
</html>
```

FORM PROCESSING LOGIC

- You Will have a Form with a Submit button.
- When the Submit button is pressed, the data input is sent for processing and is often redisplayed for the user
- All this is done in one .php program
- How does the program know whether to display the Form or the Data Processed?
- Check whether the Submit button has been pressed – call it by its “name”

FORM PROCESSING LOGIC

If the input element for the button is

```
<input type="submit" name="button"
   value = "Submit Name" />
```

Must name the button

then the PHP script logic has the form

```
if (the submit button is pressed)
{
      display_output_page();
}
      else
{
      display_form_page();
}
```

Call this Function

Call this Function

FORM PROCESSING LOGIC

- The PHP script logic has the form

```
if (isset($_REQUEST['button']))
{
   display_output_page();
}
else
{
   display_form_page();
}
```

isset() function checks for the existence of ….. is it set?

OUTPUTFCN.PHP

```
<?php
if (isset($_REQUEST['button']))
{   display_output_page();
}
else
{   display_form_page();
}

<?php function display_form_page() goes here ?>
<?php function display_output_page() goes here ?>

?>
```

forms/outputfcn.php

FUNCTION DISPLAY_FORM_PAGE()

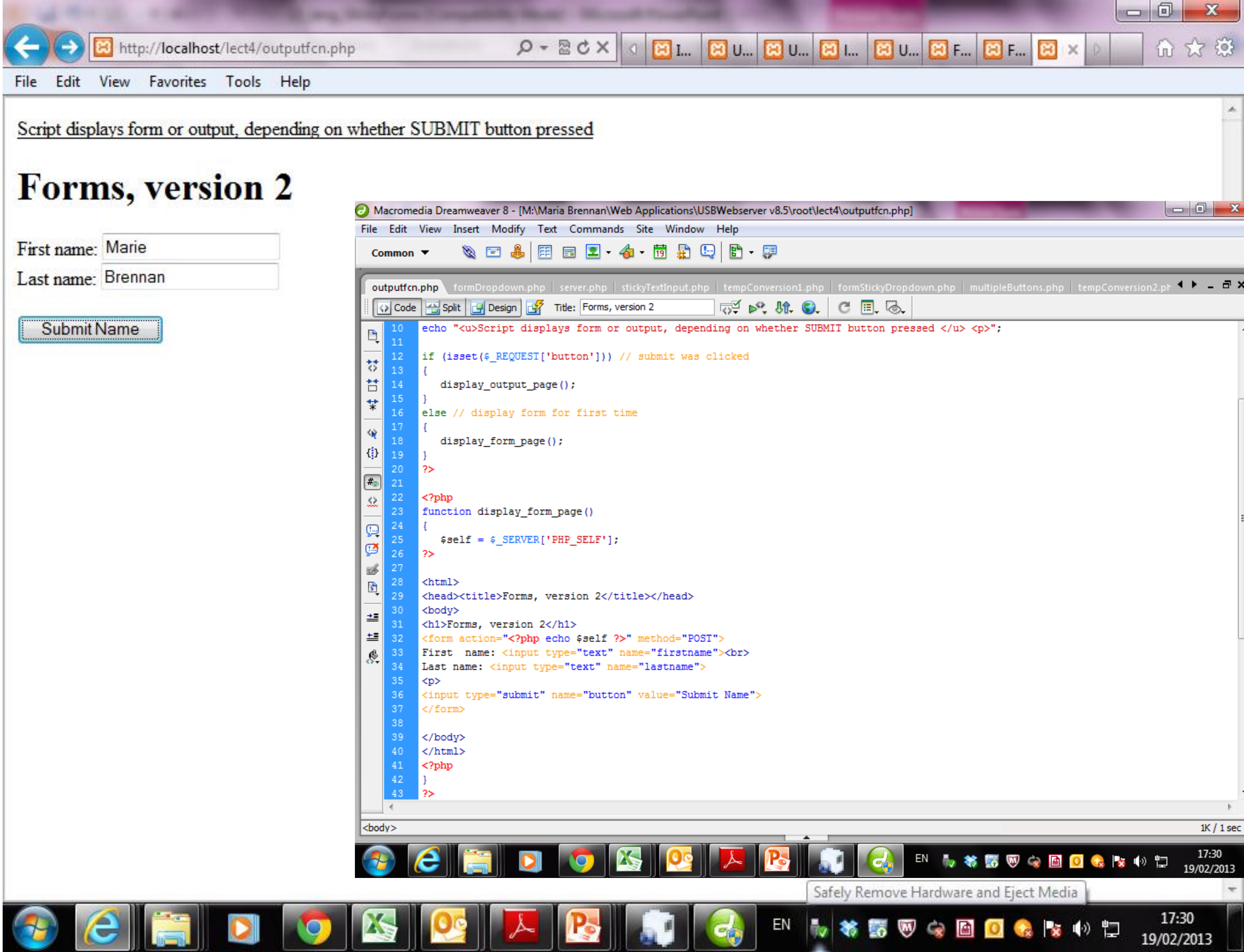
```
<?php
function display_form_page()
{  $self = $_SERVER['PHP_SELF'];
?>

<html>
   <head><title>Forms, version 2</title></head>
   <body><h1>Forms, version 2</title></h1>
   <form action="<?php echo $self ?>" method="POST">
  First name: <input type="text" name="firstname"><br>
  Last name: <input type="text" name="lastname">
   <p><input type="submit" name="button"
        value="Submit Name"></p>
   </form>
   </body></html>

<?php
}
?>
```

forms/outputfcn.php

FUNCTION DISPLAY_OUTPUT_PAGE()

```
<?php
function display_output_page()
{
   $first_name = $_REQUEST['firstname'];
   $last_name = $_REQUEST['lastname'];
?>
   <html>
   <head><title>Form Results</title></head>
   <body>
   <h1>Form Results</h1>
   <?php echo "Hello $first_name $last_name<br/> "; ?>
   </body>
   </html>
<?php
}
?>
```

forms/outputfcn.php



http://localhost/lect4/outputfcn.php
File Edit View Favorites Tools Help
Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\outputfcn.php]
File Edit View Insert Modify Text Commands Site Window Help
Common
outputfcn.php formDropdown.php server.php stickyTextInput.php tempConversion1.php formStickyDropdown.php multipleButtons.php tempConversion2.pl
Code Split Design
<body>
1K / 1 sec
Safely Remove Hardware and Eject Media
EN
17:30
19/02/2013

Script displays form or output, depending on whether SUBMIT button pressed

Forms, version 2

First name: Marie

Last name: Brennan

Submit Name

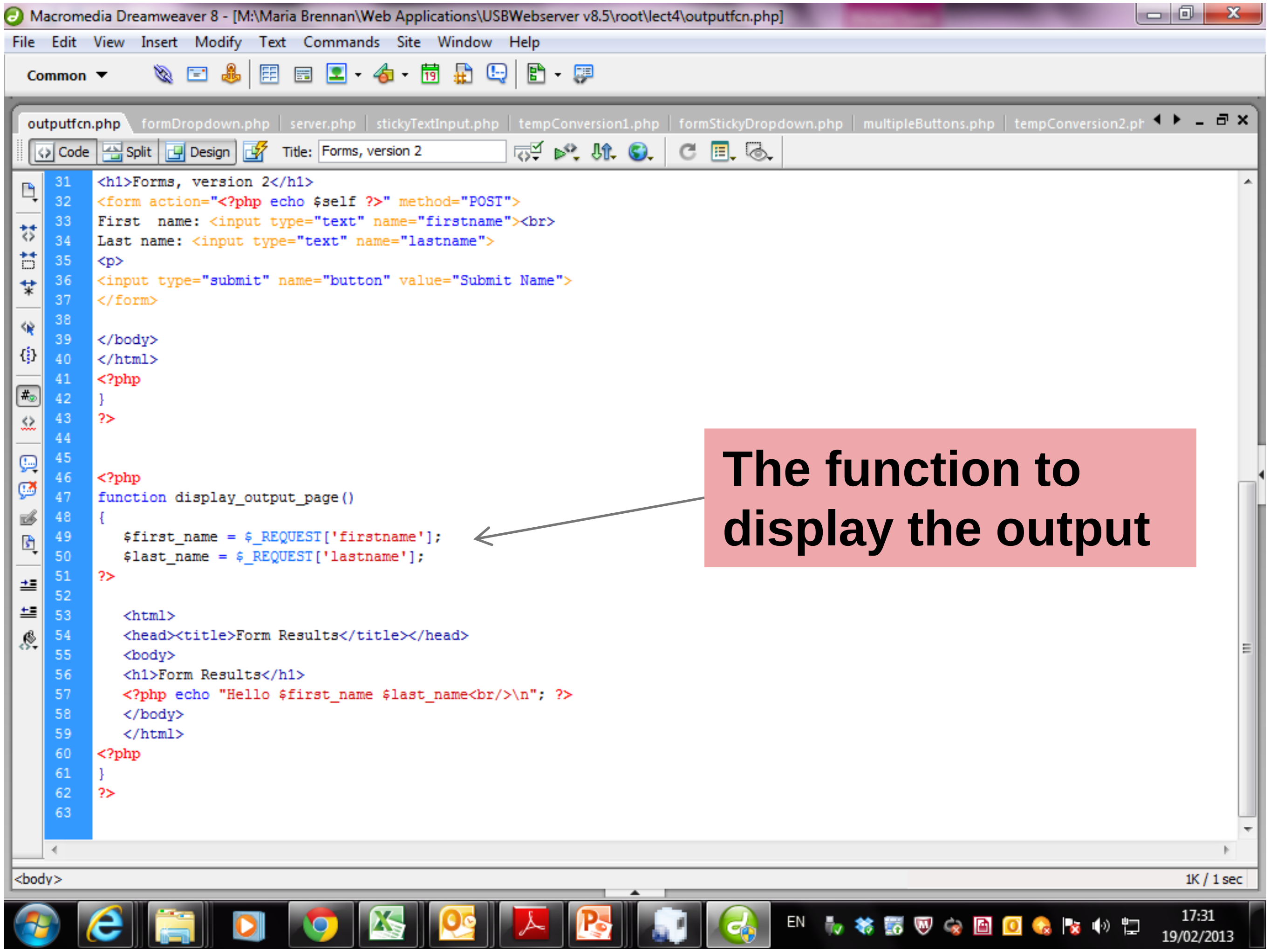
Title: Forms, version 2

```
echo "<u>Script displays form or output, depending on whether SUBMIT button pressed </u> <p>";

if (isset($_REQUEST['button'])) // submit was clicked
{
   display_output_page();
}
else // display form for first time
{
   display_form_page();
}
?>

<?php
function display_form_page()
{
   $self = $_SERVER['PHP_SELF'];
?>

<html>
<head><title>Forms, version 2</title></head>
<body>
<h1>Forms, version 2</h1>
<form action="<?php echo $self ?>" method="POST">
First  name: <input type="text" name="firstname"><br>
Last name: <input type="text" name="lastname">
<p>
<input type="submit" name="button" value="Submit Name">
</form>

</body>
</html>
<?php
}
?>
```

Script displays form or output, depending on whether SUBMIT button pressed

Form Results

Hello Marie Brennan


Macromedia Dreamweaver 8 - [M:\Maria Brennan\Web Applications\USBWebserver v8.5\root\lect4\outputfcn.php]
File Edit View Insert Modify Text Commands Site Window Help
Common
outputfcn.php formDropdown.php server.php stickyTextInput.php tempConversion1.php formStickyDropdown.php multipleButtons.php tempConversion2.ph
Code Split Design
Title: Forms, version 2
<body>
1K / 1 sec

```
<h1>Forms, version 2</h1>
<form action="<?php echo $self ?>" method="POST">
First  name: <input type="text" name="firstname"><br>
Last name: <input type="text" name="lastname">
<p>
<input type="submit" name="button" value="Submit Name">
</form>

</body>
</html>
<?php
}
?>


<?php
function display_output_page()
{
   $first_name = $_REQUEST['firstname'];
   $last_name = $_REQUEST['lastname'];
?>

   <html>
   <head><title>Form Results</title></head>
   <body>
   <h1>Form Results</h1>
   <?php echo "Hello $first_name $last_name<br/>\n"; ?>
   </body>
   </html>
<?php
}
?>
```

The function to display the output

USING PHP_SELF

Find the script using

- `$self = $_SERVER['PHP_SELF'];`

This gives the script location relative to the document root. For example

- `/www/forms/outputfcn.php`

To see this, view the html before submitting the form.

Then use $self in the form method to refresh the page

```
<form action="<?php echo $self ?>" method="POST">
```

SUMMARY

View outputfcn.php

Note:

- isset()
- Function to display the Form
- Function to display the output
- How HTML and PHP are interleaved
- These functions could be put in an external file and called using include() function

USING MULTIPLE BUTTONS (1)

Give the buttons different names:

```
<input type="submit" name="button1"
   value = "Submit Name 1" />
<input type="submit" name="button2"
   value = "Submit Name 2" />
```

Check which one was clicked

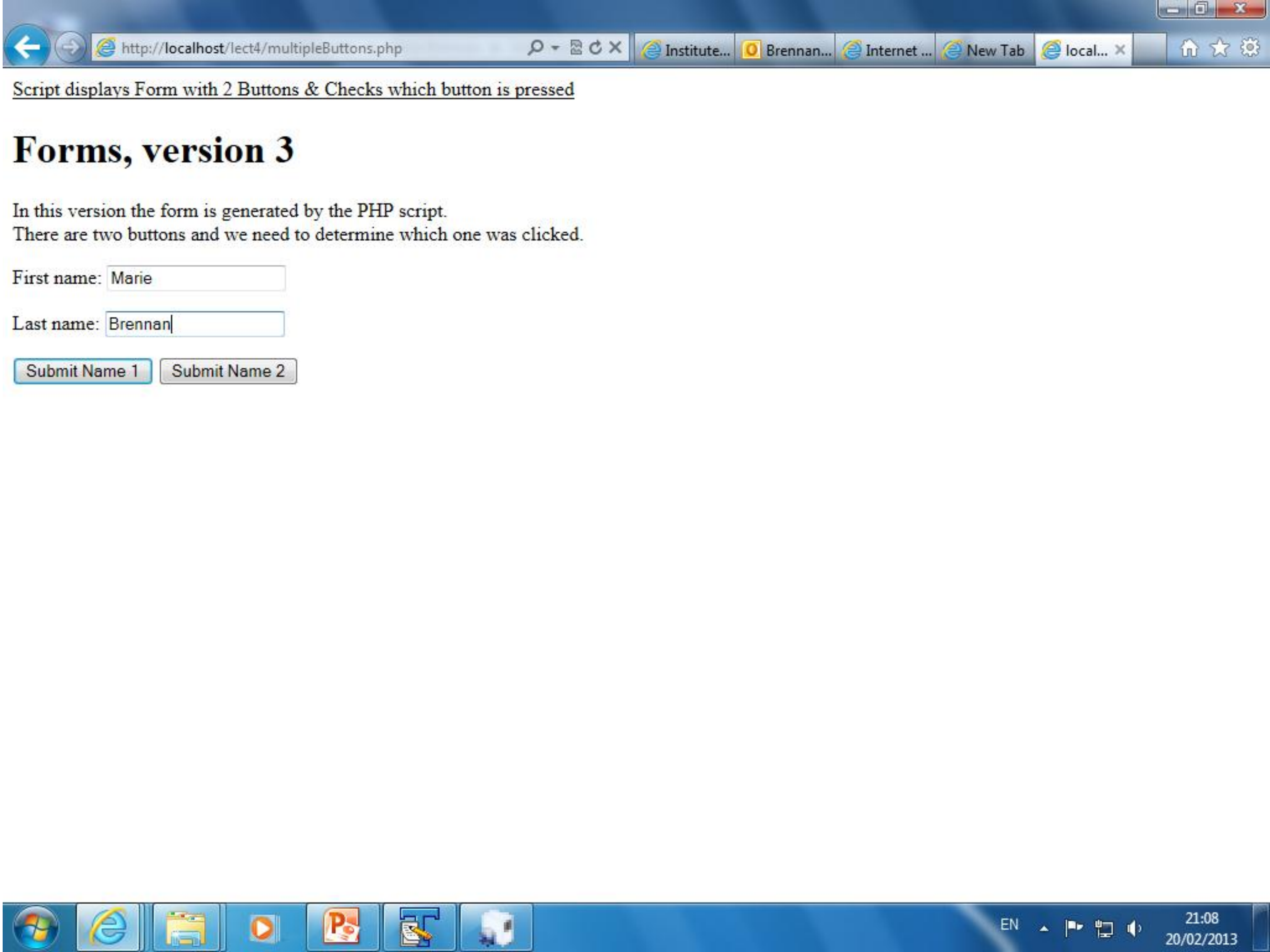
```
if ( isset($_REQUEST['button1']) ||
isset($_REQUEST['button2'])
{  display_output_page();
} else
{  display_form_page();
}
```

USING MULTIPLE BUTTONS (2)

```
$button1 = isset($_REQUEST['button1']
           ? $_REQUEST['button1'] : '';
$button2 = isset($_REQUEST['button2']
           ? $_REQUEST['button2'] : '';
```

```
if ($button1 != '')
{ // process button1 click event here
}
if ($button2 != '')
{ // process button2 click event here
}
```

forms/multipleButtons.php



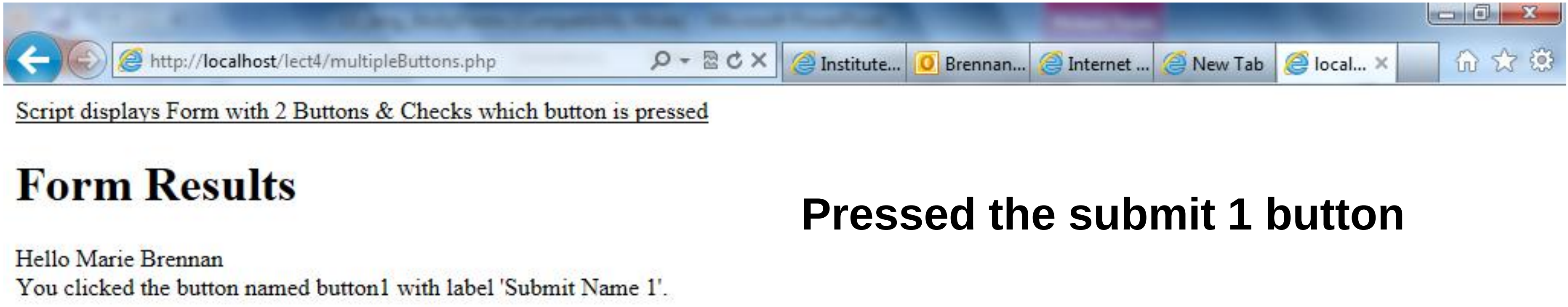
Script displays Form with 2 Buttons & Checks which button is pressed

Forms, version 3

In this version the form is generated by the PHP script.
There are two buttons and we need to determine which one was clicked.

First name: Marie

Last name: Brennan

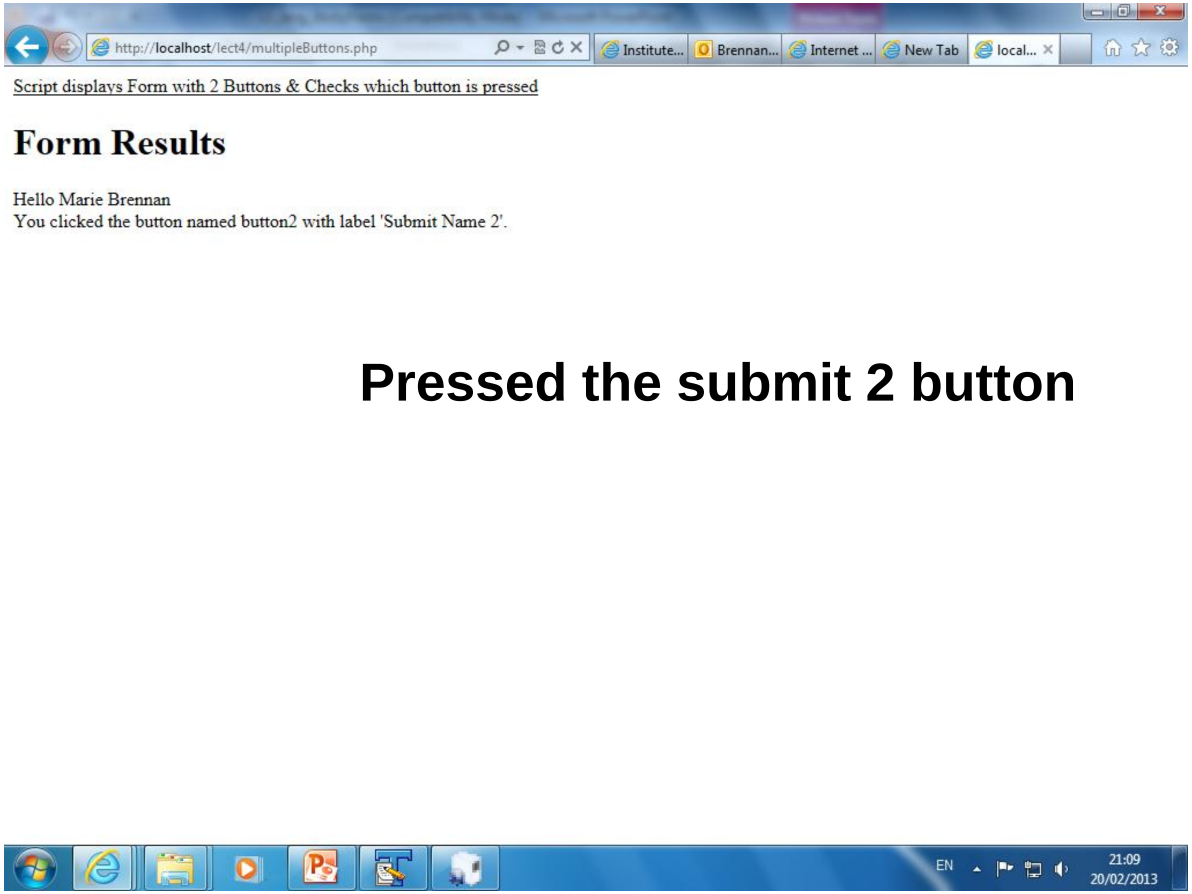


Script displays Form with 2 Buttons & Checks which button is pressed

Form Results

Hello Marie Brennan
You clicked the button named button1 with label 'Submit Name 1'.

Pressed the submit 1 button

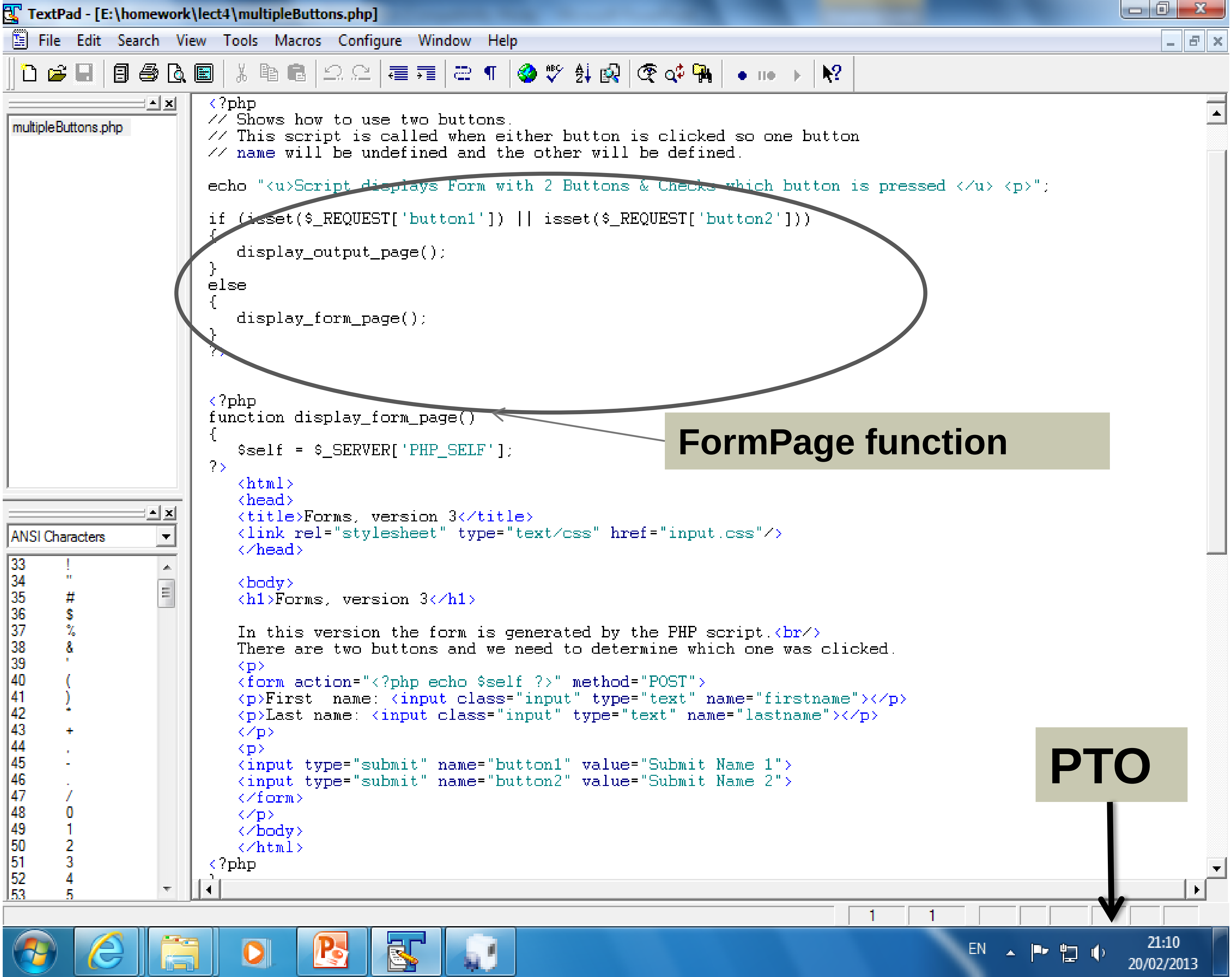


Script displays Form with 2 Buttons & Checks which button is pressed

Form Results

Hello Marie Brennan
You clicked the button named button2 with label 'Submit Name 2'.

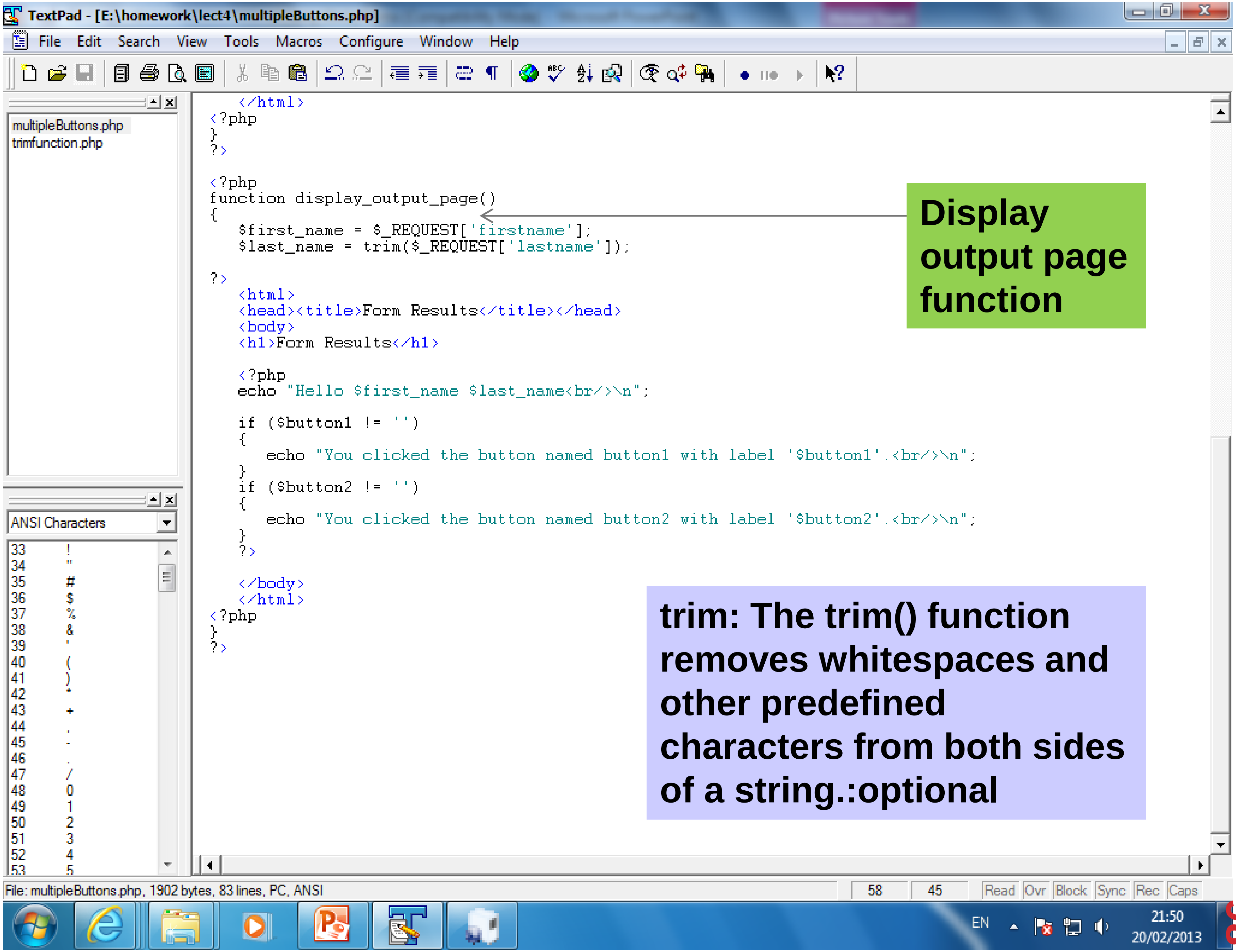
Pressed the submit 2 button


TextPad - [E:\homework\lect4\multipleButtons.php]
File Edit Search View Tools Macros Configure Window Help
multipleButtons.php
ANSI Characters
EN
21:10
20/02/2013

```
<?php
// Shows how to use two buttons.
// This script is called when either button is clicked so one button
// name will be undefined and the other will be defined.

echo "<u>Script displays Form with 2 Buttons & Checks which button is pressed </u> <p>";

if (isset($_REQUEST['button1']) || isset($_REQUEST['button2']))
{
   display_output_page();
}
else
{
   display_form_page();
}
?>


<?php
function display_form_page()
{
   $self = $_SERVER['PHP_SELF'];
?>
   <html>
   <head>
   <title>Forms, version 3</title>
   <link rel="stylesheet" type="text/css" href="input.css"/>
   </head>

   <body>
   <h1>Forms, version 3</h1>

   In this version the form is generated by the PHP script.<br/>
   There are two buttons and we need to determine which one was clicked.
   <p>
   <form action="<?php echo $self ?>" method="POST">
   <p>First  name: <input class="input" type="text" name="firstname"></p>
   <p>Last name: <input class="input" type="text" name="lastname"></p>
   </p>
   <p>
   <input type="submit" name="button1" value="Submit Name 1">
   <input type="submit" name="button2" value="Submit Name 2">
   </form>
   </p>
   </body>
   </html>
<?php
```

FormPage function

PTO



```
   </html>
<?php
}
?>

<?php
function display_output_page()
{
   $first_name = $_REQUEST['firstname'];
   $last_name = trim($_REQUEST['lastname']);

?>
   <html>
   <head><title>Form Results</title></head>
   <body>
   <h1>Form Results</h1>

   <?php
   echo "Hello $first_name $last_name<br/>\n";

   if ($button1 != '')
   {
      echo "You clicked the button named button1 with label '$button1'.<br/>\n";
   }
   if ($button2 != '')
   {
      echo "You clicked the button named button2 with label '$button2'.<br/>\n";
   }
   ?>

   </body>
   </html>
<?php
}
?>
```

File: multipleButtons.php, 1902 bytes, 83 lines, PC, ANSI

58 45

Display output page function

trim: The trim() function removes whitespaces and other predefined characters from both sides of a string.:optional

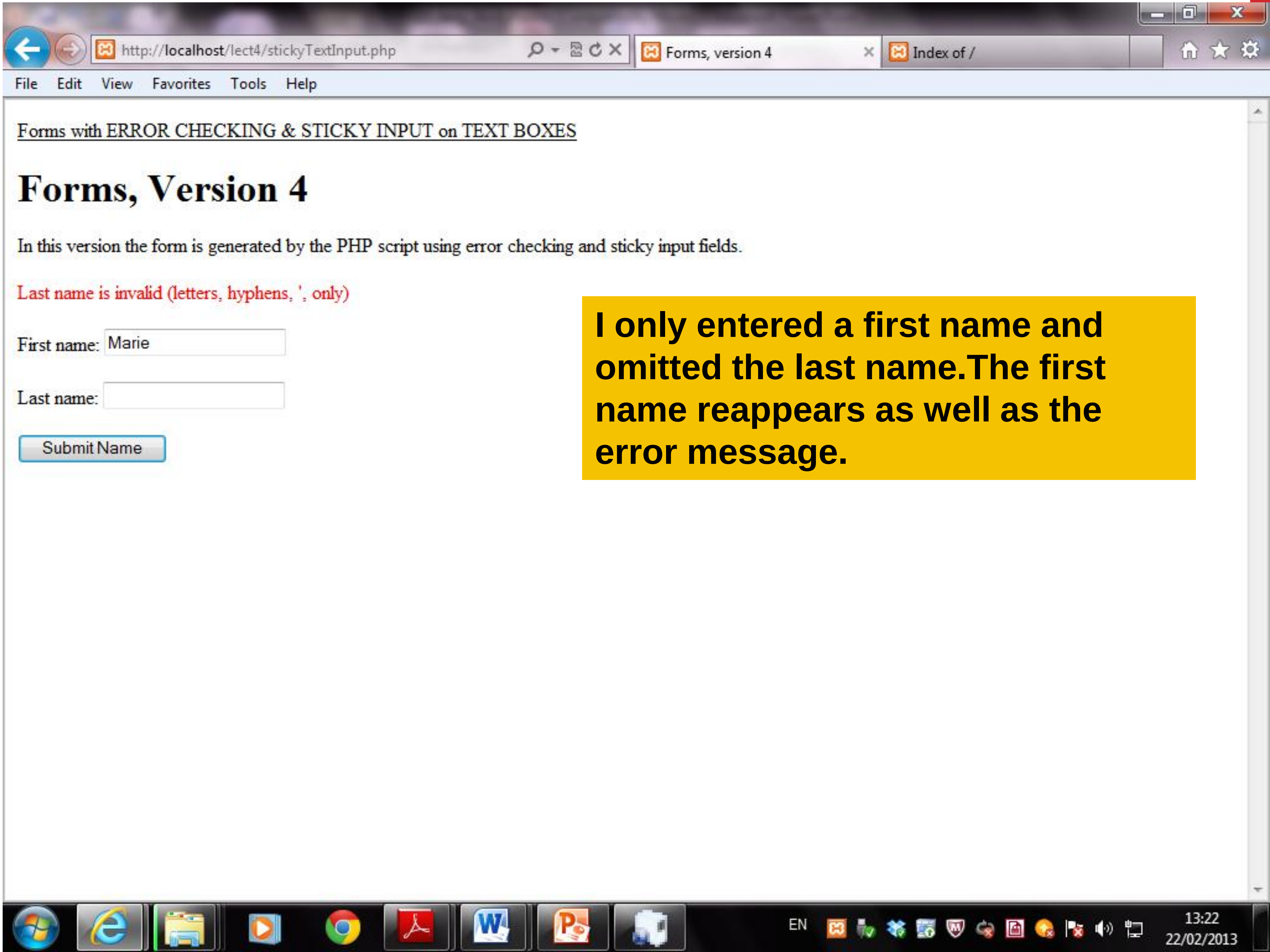
STICKY FORM

HTML form that remembers how you filled it out.

Good feature for end users, especially if you are requiring them to resubmit a form
 (i.e. after filling it out incorrectly in the first place)

STICKY FORMS

- Have you ever entered a form where you might have omitted a value or where you entered an incorrect value.
- What normally happens is you are presented with the form again as well as an error message
- The correct values are normally still there
- This information has been kept stored in some variable
- This information has stuck to the form data!!



Forms with ERROR CHECKING & STICKY INPUT on TEXT BOXES

Forms, Version 4

In this version the form is generated by the PHP script using error checking and sticky input fields.

Last name is invalid (letters, hyphens, ', only)

First name: Marie

Last name:

Submit Name

I only entered a first name and omitted the last name.The first name reappears as well as the error message.

VERIFY USER INPUT

STICKYTEXTINPUT.PHP

- Verify that data, input by user, is in the format expected
- Remove whitepace around the input using the trim() function
- Create a variable for an error message -- $error
- Create a regular expression
- Compare data input with a regular expression to determine it is in the format expected
- Use pattern matching function --- preg_match()

REGULAR EXPRESSIONS IN DETAIL

- Match strings containing only digits
- `$regexp = "^[0-9]+$";`
  - `^` matches the beginning of the string
  - `$` matches the end of the string
  - `[0-9]` specifies a range for a character
  - `+` means 1 or more occurrences

- Matching phone numbers of the form `"ddd-ddd-dddd"`
- `$regexp =`

  ```
  "^[0-9]{3}-[0-9]{3}-[0-9]{4}$";
  ```

  - Here the hyphen is a literal character and {3} indicates exactly three occurrences of the preceding character.

REGULAR EXPRESSIONS – RECAP(2)

```
$reg_exp = "^[a-zA-Z\-\']+$";
```

This definition of a valid name is one containing letters, hyphens and apostrophe's

You should inform the user if input is in an invalid format e.g.

- Text input
- Phone numbers
- Email addresses
- Web addresses
- Etc.

ERROR CHECKING

The validate form is called when the user has pressed the submit button. It checks the data entered is valid and returns an error msg if not.

Names contain only letters, hyphens, and '

```
<?php function validate_form()
{   $first_name = trim($_REQUEST['firstname']);
    $last_name = trim($_REQUEST['lastname']);
    $error = '';
    $reg_exp = “^[a-zA-Z\-\']+$^”;
    if (! preg_match($reg_exp, $first_name))
    {   $error .= "First name is invalid ...";
    }
    if (! preg_match($reg_exp, $last_name))
    {   $error .= "Last name is invalid ...";
    }
    return $error;
} ?>
```

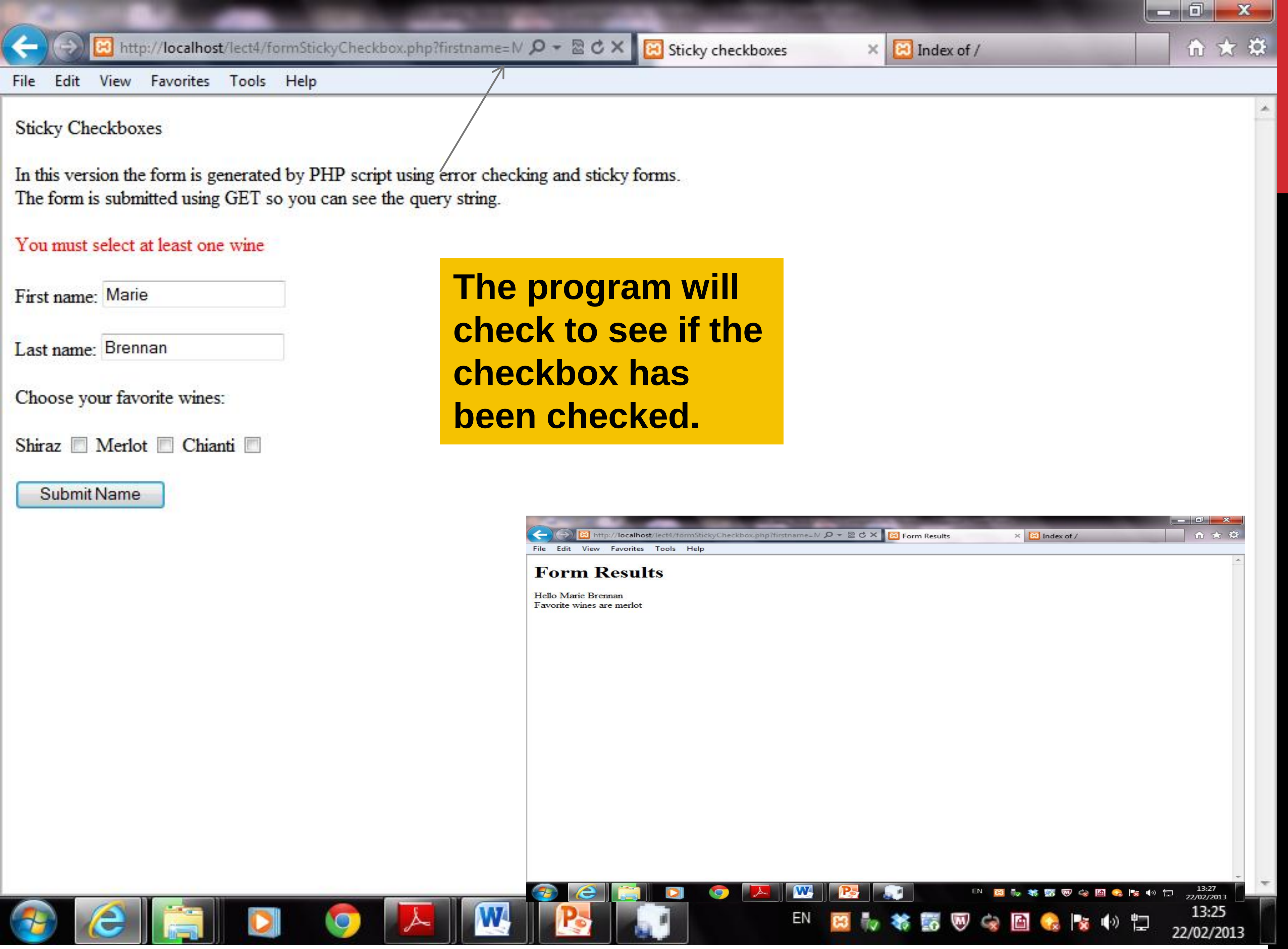
STICKY INPUT FIELDS - TEXT

- If there are errors in a form, the entries should be sent to the user to be corrected
- We can supply the values the user entered as defaults, as follows:

If the user enters the correct username its then returned to them as it will be stored in the variable username

```
First name: <input type="text" name="firstname"
   value="<?php echo $first_name ?>">
Last name: <input type="text" name="lastname"
   value="<?php echo $last_name ?>">
```

forms/stickyTextInput.php


http://localhost/lect4/formStickyCheckbox.php?firstname=M
Sticky checkboxes
Index of /
File Edit View Favorites Tools Help

Sticky Checkboxes

In this version the form is generated by PHP script using error checking and sticky forms.
The form is submitted using GET so you can see the query string.

You must select at least one wine

First name: Marie

Last name: Brennan

Choose your favorite wines:

Shiraz □ Merlot □ Chianti □

Submit Name

The program will check to see if the checkbox has been checked.

http://localhost/lect4/formStickyCheckbox.php?firstname=M
Form Results
Index of /
File Edit View Favorites Tools Help

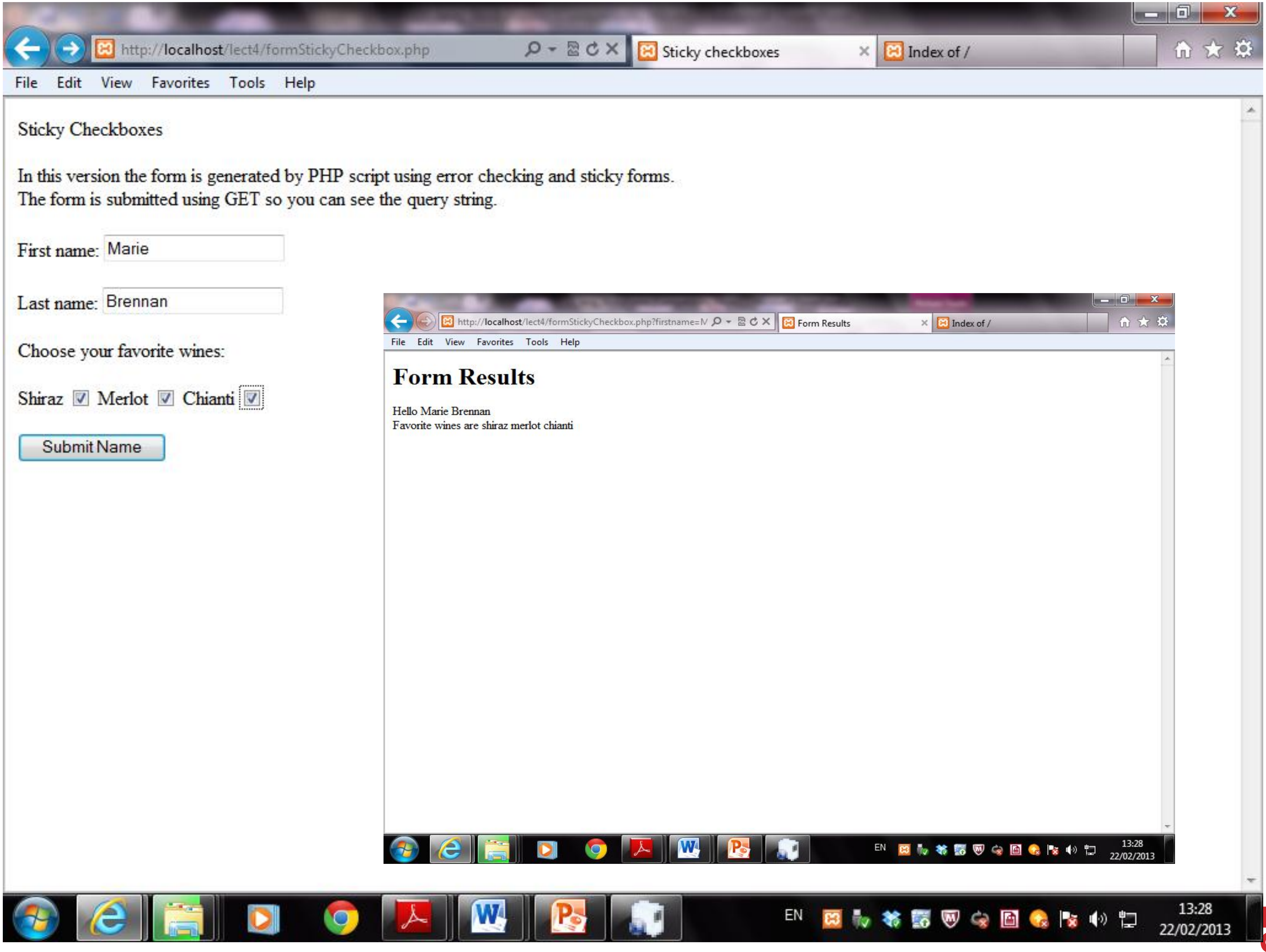
Form Results

Hello Marie Brennan
Favorite wines are merlot

CHECKBOX GROUP (1)

- A user can tick a number of checkboxes
- Use array notation [] in the form to indicate a group of checkboxes

```
Which wines do you like?
Shiraz <input type="checkbox" name="wines[]"
   value="shiraz">
Merlot <input type="checkbox" name="wines[]"
   value="merlot">
Chianti <input type="checkbox" name="wines[]"
  value="chianti">
```



Sticky Checkboxes

In this version the form is generated by PHP script using error checking and sticky forms.
The form is submitted using GET so you can see the query string.

First name: Marie

Last name: Brennan

Choose your favorite wines:

Shiraz ☑ Merlot ☑ Chianti ☑

Form Results

Hello Marie Brennan
Favorite wines are shiraz merlot chianti

CHECKBOX GROUP (2)

- Check entries for wine, using isset()
- $wines is an indexed array
- If no wine had been selected then $wines is blank

```
$wines = isset($_REQUEST['wines']) ? $_REQUEST['wines'] : ‘ ’;
```

CHECKBOX GROUP (3)

Determine which boxes are checked and echo the choices

```
if (! empty($wines))
{   echo "Wines selected were ";
    foreach ($wines as $wine)
    {   echo "$wine ";
    }
}
else
    echo "No wines were selected";
```

Loop through the array and print the values

forms/checkbox.php

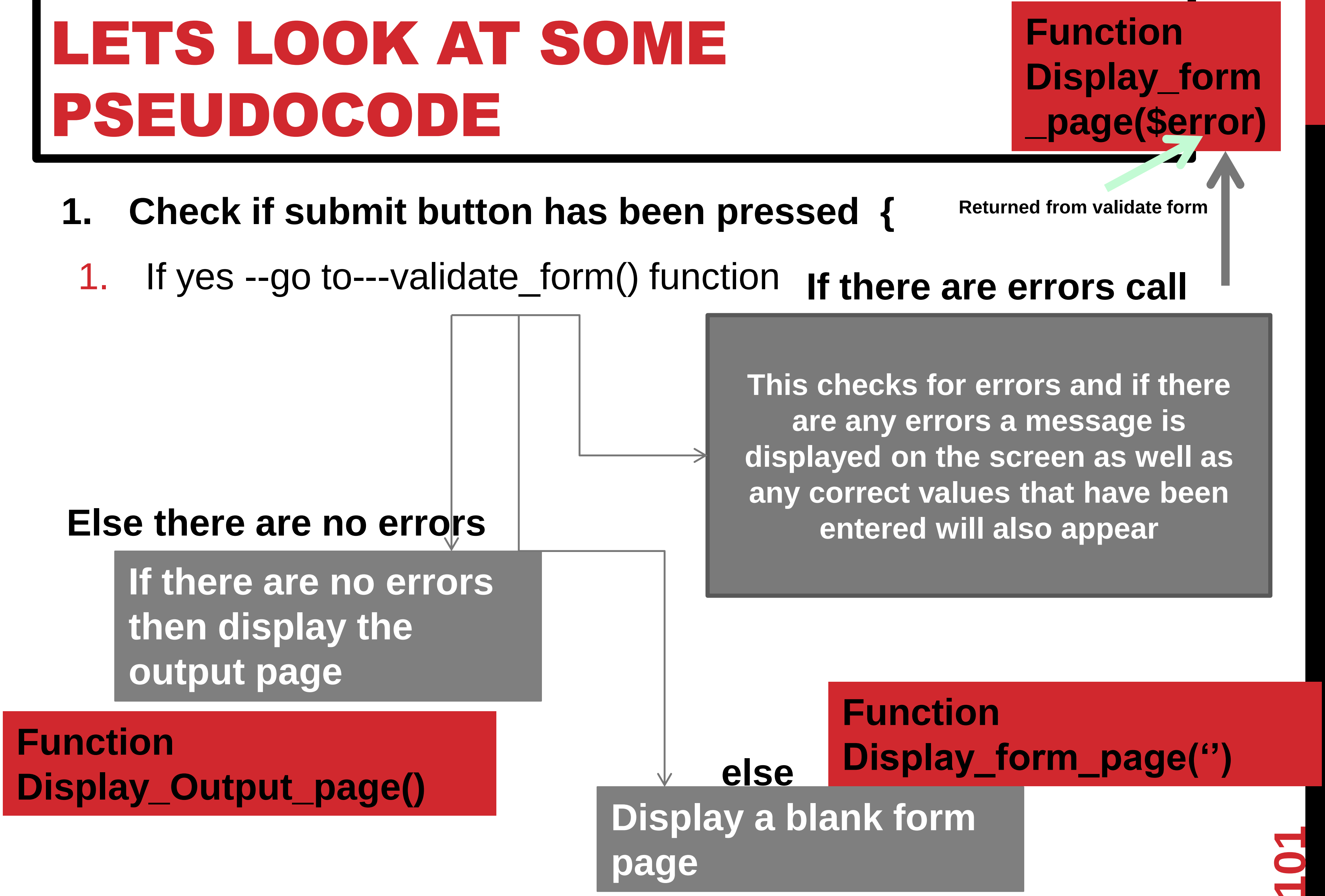
We will look at this code later on

STICKY CHECKBOXES (1)

```
 <p>Choose your favorite wines:</p>
Shiraz <input type="checkbox" name="wines[]" value="shiraz“……
…………………………..
……………………………
$wines = isset($_REQUEST['wines']) ? $_REQUEST['wines'] : '';
………………..
………………..
 if ( empty($wines))
   {
      $error .= "<span class= 'error' >You must select at least one
wine</span></br>";
   }
```

forms/formStickyCheckbox.php

LETS LOOK AT SOME PSEUDOCODE



Function Display_form _page($error)

Returned from validate form

1. Check if submit button has been pressed {

   1. If yes --go to---validate_form() function

If there are errors call

This checks for errors and if there are any errors a message is displayed on the screen as well as any correct values that have been entered will also appear

Else there are no errors

If there are no errors then display the output page

Function Display_Output_page()

else

Function Display_form_page(‘’)

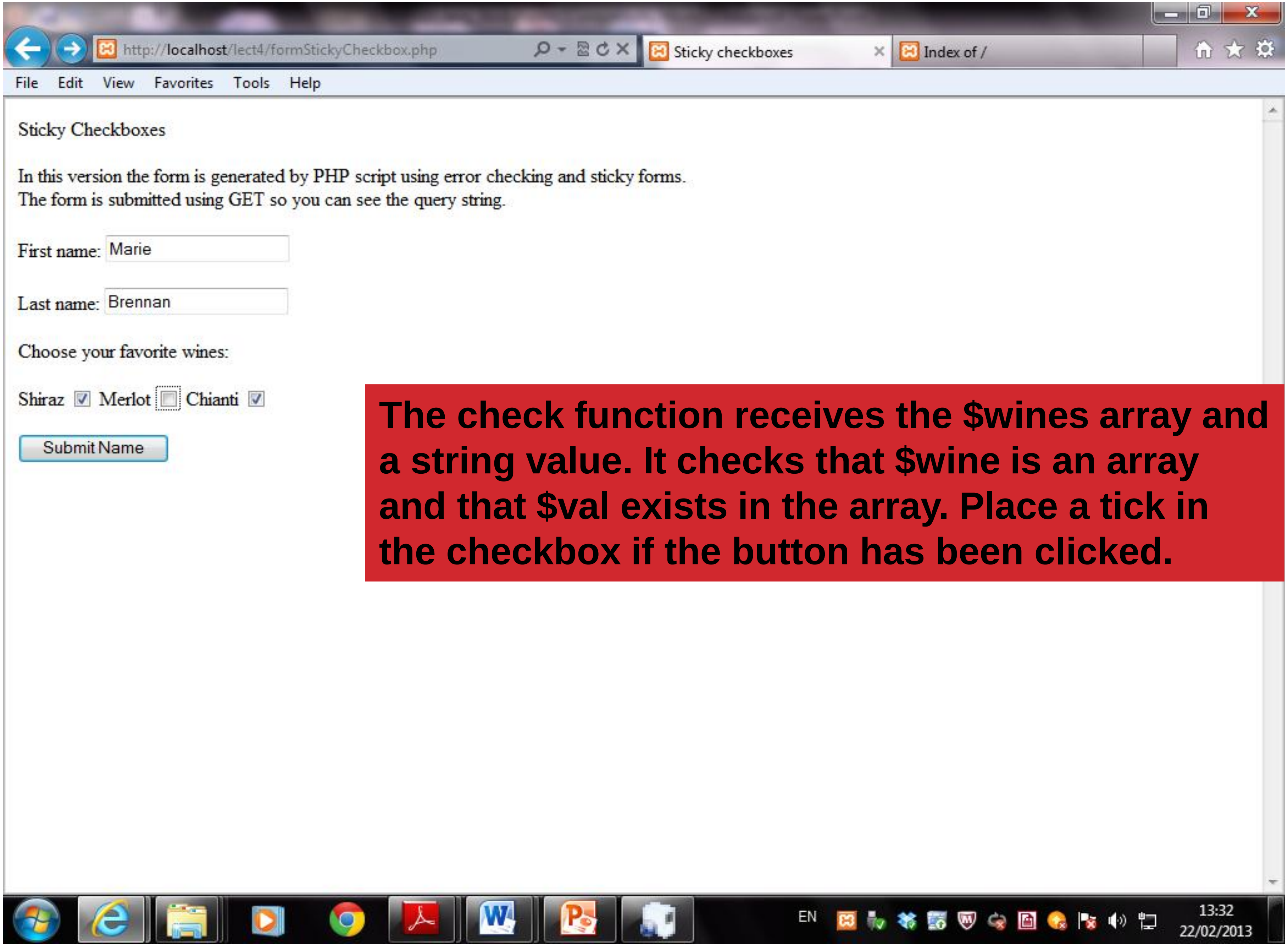
Display a blank form page

STICKY CHECKBOXES

Write a function to find which checkboxes are selected

Use array functions …is_array() and in_array()

```
<?php
function check($group, $val)
{
    if (is_array($group) and in_array($group,$val))
    {
        echo 'checked = "checked"';
    }
}
?>
```



Sticky Checkboxes

In this version the form is generated by PHP script using error checking and sticky forms.
The form is submitted using GET so you can see the query string.

First name: Marie

Last name: Brennan

Choose your favorite wines:

Shiraz ☑ Merlot □ Chianti ☑

Submit Name

The check function receives the $wines array and a string value. It checks that $wine is an array and that $val exists in the array. Place a tick in the checkbox if the button has been clicked.

STICKY CHECKBOXES (2)

Use the check() function as follows

```
Shiraz <input type="checkbox" name="wines[]"
   value="shiraz" <?php check($wines,"shiraz")?>>
Merlot <input type="checkbox" name="wines[]"
   value="merlot" <?php check($wines,"merlot")?>>
Chianti <input type="checkbox" name="wines[]"
   value="chianti" <?php check($wines,"chianti")?>>
```

forms/formStickyCheckbox.php

RADIO BUTTON(1)

```
Shiraz <input type="radio" name="wine" value="shiraz" >

.......

 $wine = isset($_REQUEST['wine']) ? $_REQUEST['wine'] : '';

.................

if $wine is blank

The following error will appear attached to the original form

$error .=

     "<span class= 'error'>Select a wine</span><br>";
```

Check that the wine checkbox has been clicked and is not blank

STICKY RADIO BUTTONS (2)

Use the check() function as follows

```
Shiraz <input type="radio" name="wine"
   value="shiraz" <?php check($wine,"shiraz")?>>
Merlot <input type="radio" name="wine"
   value="merlot" <?php check($wine,"merlot")?>>
Chianti <input type="radio" name="wine"
   value="chianti" <?php check($wine,"chianti")?>>
```

forms/formStickyRadio.php

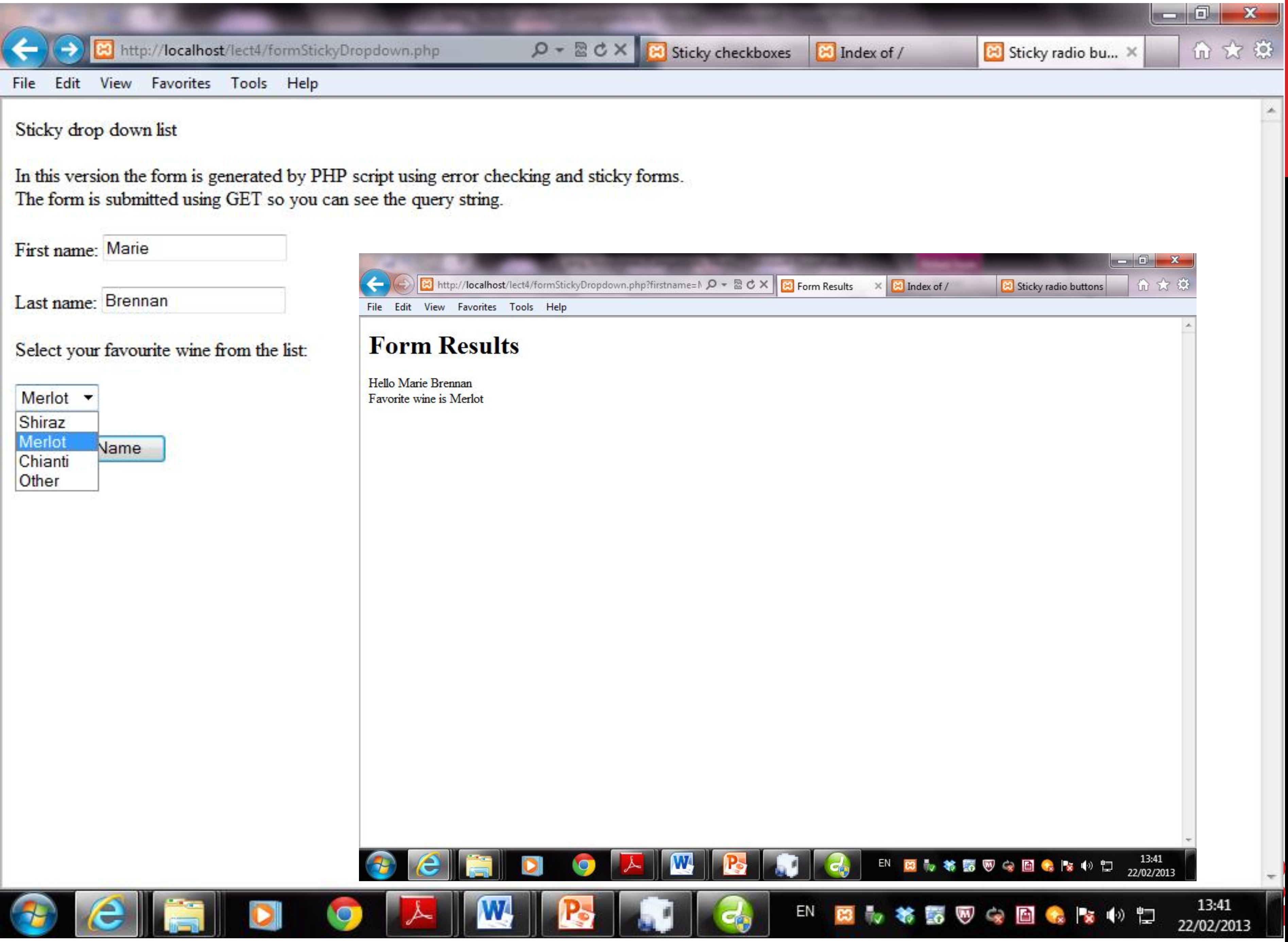
```
<?php
function check($group, $val)
{
   if ($group === $val)
   {
      echo 'checked = "checked"';
   }
}
?>
```

DROPDOWN LIST

- Use select and option.
- The selected attribute indicates which choice has been made.
- If no choice is made, the first one is selected

```
Select your favourite wine from the list:
<select name="wine">
   <option selected="selected">Shiraz</option>
   <option>Merlot</option>
   <option>Chianti</option>
   <option>Other</option>
</select>
```

forms/formDropdown.php



Sticky drop down list

In this version the form is generated by PHP script using error checking and sticky forms.
The form is submitted using GET so you can see the query string.

First name: Marie

Last name: Brennan

Select your favourite wine from the list:

Merlot ▾
Shiraz
Merlot
Chianti
Other

Form Results

Hello Marie Brennan
Favorite wine is Merlot

DROPDOWN LIST (1)

Write a check() function:

```
<?php
function check($group, $val)
{
   if ($group === $val)
   {
      echo 'selected = "selected"';
   }
}
?>
```

STICKY DROPDOWN LIST (2)

Use the check() function as follows

```
<select name="wine">
<option <?php
check($wine,"Shiraz")?>>Shiraz</option>
<option <?php
check($wine,"Merlot")?>>Merlot</option>
<option <?php
check($wine,"Chianti")?>>Chianti</option>
<option <?php
check($wine,"Other")?>>Other</option>
</select>
```

forms/formStickyDropdown.php

STICKY INPUT PROGRAMS

All the programs on Forms and Sticky Input are available, for downloading & studying, from usbwebserver8/roots folder on the Student Share

Summary

Form Processing

Temperature Conversion Program

Using PHP_SELF

Multiple Buttons

Error Checking using Regular Expressions

Sticky Input Fields – Text, Checkboxes, Radio Buttons, Dropdown List