ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Chapter 3.2. Functions

Objectives

- Use several PHP functions for Web application development
- Write and use your own functions

Content

1. Basic PHP Functions
2. Write your own functions
3. Using External Script Files

Content

1. Basic PHP Functions
2. Write your own functions
3. Using External Script Files

1. Basic PHP Functions

- We previously discussed functions such as strlen(), trim(), strtolower(), strtoupper(), and substr().
- In this section we examine several other useful functions including
  - Some basic numeric PHP functions
    - E.g., the absolute value[abs()], square root [sqrt()], round [round()], integer checker[is_numeric()], and random number generation [rand()] functions.
  - The print() function
    - We will cover in more detail
  - The date() function
    - We will discuss using the date() function to determine date and time information.

Numberic PHP Functions

- Absolute value
- Square root,
- Round,
- Integer checker
- Random number generation

1.1. The abs() Function

- The absolute value function takes a single numerical argument and returns its absolute value.
- For example, the following

```
$x=abs(-5);
$y=abs(42);
print "x=$x y=$y";
```

- Will output
    - **`x=5 y=42`**

1.2. The sqrt() Function

- The square root function takes a single numerical argument and returns its square root.
- For example, the following

```
$x=sqrt(25);
$y=sqrt(24);
print "x=$x y=$y";
```

- Will output
  - **x=5 y=4.898979485566**

1.3. The round() Function

- The round function takes a single numerical argument and returns the number rounded up or down to the nearest integer.
- For example, the following

```
$x=round(-5.456);
$y=round(3.7342);
print "x=$x y=$y";
```

- Will output **`x=-5 y=4`**

1.4. The round() Function

- You can include 2nd argument to define the number of digits after the decimal point to round to.
- For example,

```
$x=round(-5.456,2);
$y=round(3.7342,3);
print "x=$x y=$y";
```

- would output
    - `x=-5.46 y=3.734`

1.5. The is_numeric() Function

- **`is_numeric()`** is useful for determining whether a variable is a valid number or a numeric string.
  - It returns *true* or *false*.
- Consider the following example...

```
if (is_numeric($input)) {
    print "Got Valid Number=$input";
} else {
  print "Not Valid Number=$input";
}
```

- If $input was “6” then would : **`Got Valid Number=6`**
- If $input was “Happy” then would output: **`Not Valid Number=Happy`**

1.6. The rand() Function

- Use rand() to generate a random number.
  - You can use random numbers to simulate a dice roll or a coin toss or to randomly select an advertisement banner to display.
- rand() typically uses 2 arguments to define the range of numbers it should return (min and max limits),
  - For example the following returns a number 1 - 15
    - `$num = rand(1, 15);`

1.6. The rand() Function (2)

- Use the srand and microtime to seed rand() and ensure it returns a random number, for example,

```
srand ((double) microtime() * 10000000);
$dice = rand(1, 6);
print "Your random dice toss is $dice";
```

- The random number generated in this case can be a 1, 2, 3, 4, 5, or 6.

1.7. More information on the `print()` Function

- You don’t need to use parenthesis with print()
- Double quotes means output the value of any variable:

```
$x = 10;
print ("Mom, please send $x dollars");
```

- Single quotes means output the actual variable name

```
$x = 10;
print ('Mom, please send $x dollars');
```

- To output a single variable’s value or expression, omit the quotation marks.

```
$x=5;
print $x*3;
```

Generating HTMLTags with print()

- Using single or double quotation statements can be useful when generating HTML tags
  - `print '<font color="blue">';`
- This above is easier to understand and actually runs slightly faster than using all double quotation marks and the backslash (\) character:
  - `print "<font color=\"blue\">";`

A Full Example ...

- Consider the following application:
  - Uses an HTML form to ask the end-user to guess the results of a coin flip:

```
<input type="radio" name="pick" value="0">
  Heads
<input type="radio" name="pick" value="1">
  Tails
```

Receiving Code

```
<html>
<head><title> Coin Flip Results </title></head>
  <body> <?php
 srand ((double) microtime() * 10000000);
$flip = rand( 0, 1 );
if ( $flip == 0 && $pick == 0 ) {
     print "The flip=$flip, which is heads! <br> ";
     print '<font color="blue"> You got it
  right!</font>';
} elseif ( $flip == 0 && $pick == 1 ) {
       print "The flip=$flip, which is heads! <br> ";
     print '<font color="red"> You got it
  wrong!</font>';
} elseif ( $flip == 1 && $pick == 1 ) {
      print "The flip=$flip, which is tails! <br>";
      print '<font color="blue"> You got it
  right!</font>';
```

Check whether both the coin flip and the guess are heads.

Check whether the coin flip is heads but the guess is tails.

Check whether both the coin flip and the guess are tails.

Receiving Code continued...

```
} elseif ( $flip == 1 && $pick == 0 ) {
    print "The flip=$flip, which is tails! <br>";
    print '<font color="red"> You got it wrong!</font>';
} else {
    print "<br>Illegal state error!";
}
?>  </body></html>
```

Check whether the coin flip is tails but the guess is heads.

Receiving Code With REGISTER_GLOBALS Off

```
<html>
<head><title> Coin Flip Results </title></head> <body> <?php
 srand ((double) microtime() * 10000000);
$pick = $_POST[“PICK”];
$flip = rand( 0, 1 );
if ( $flip == 0 && $pick == 0 ) {
     print "The flip=$flip, which is heads! <br> ";
     print '<font color="blue"> You got it right!</font>';
} elseif ( $flip == 0 && $pick == 1 ) {
        print "The flip=$flip, which is heads! <br> ";
     print '<font color="red"> You got it wrong!</font>';
} elseif ( $flip == 1 && $pick == 1 ) {
      print "The flip=$flip, which is tails! <br>";
```

Check whether both the coin flip and the guess are heads.

Check whether the coin flip is heads but the guess is tails.

Check whether both the coin flip and the guess are tails.

Receiving Code With REGISTER_GLOBALS Off, cont. ...

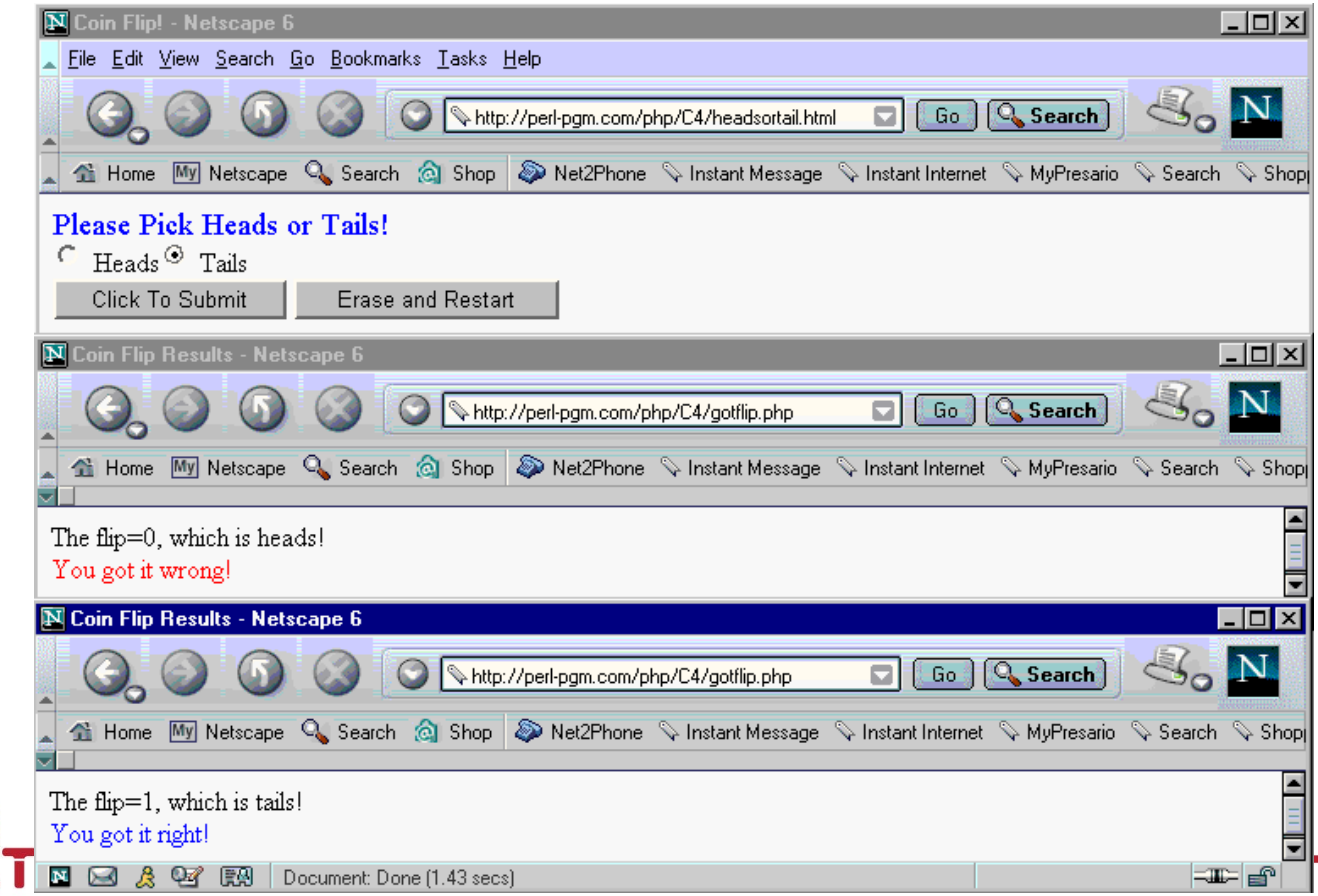
```
    print '<font color="blue"> You got it right!</font>';
} elseif ( $flip == 1 && $pick == 0 ) {
    print "The flip=$flip, which is tails! <br>";
    print '<font color="red"> You got it wrong!</font>';
} else {
    print "<br>Illegal state error!";
}
?>  </body></html>
```

Check whether the coin flip is tails but the guess is heads.

The Output ...

The previous code can be executed at

http://webwizard.aw.com/~phppgm/C3/headsortail.html


Coin Flip! - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://perl-pgm.com/php/C4/headsortail.html
Go
Search
Home
Netscape
Search
Shop
Net2Phone
Instant Message
Instant Internet
MyPresario
Search
Shop
Coin Flip Results - Netscape 6
http://perl-pgm.com/php/C4/gotflip.php
Coin Flip Results - Netscape 6
http://perl-pgm.com/php/C4/gotflip.php
Document: Done (1.43 secs)

Please Pick Heads or Tails!

○ Heads ● Tails

Click To Submit | Erase and Restart

The flip=0, which is heads!

You got it wrong!

The flip=1, which is tails!

You got it right!

printf() function

- outputs a string built by substituting values into a template (the format string).
- Derived from the function of the same name in the standard C library.

echo() function

- put a string into the HTML of a PHP-generated page

```
echo "Printy";
echo("Printy"); // also valid
// Display: Firstsecondthird
echo "First", "second", "third";
// this is a parse error
echo("Hello", "world");
```

echo and print

- echo is not a true function, faster

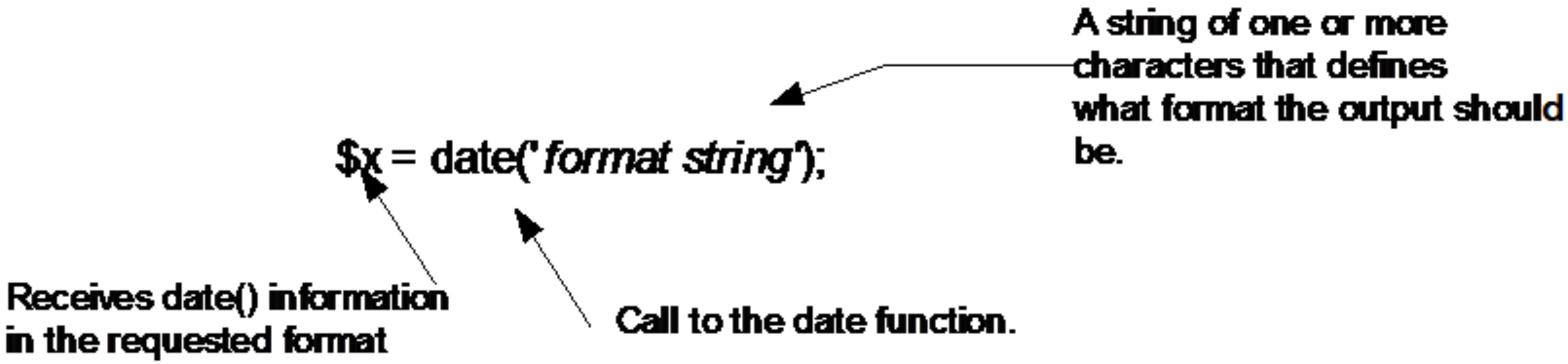
```
// parse error
if (echo("test")) {
   echo("it worked!");
}
```

- Print or printf can remedy this error

```
if (! print("Hello, world")) {
   die("you're not listening to me!");
}
```

1.8. The date() Function

- The date() function is a useful function for determining the current date and time



$x = date('*format string*');

A string of one or more characters that defines what format the output should be.

Receives date() information in the requested format

Call to the date function.

- The format string defines the format of the date() function’s output:
  - `$day = date('d');`
  - `print "day=$day";`

Request date() to return the numerical day of the month.

- If executed on September 16, 2010, then it would output “day=16”.

Selected character formats for date()

Format String	Meaning	Format String	Meaning
D	Three-letter indication of day of week (for example, Mon, Tue)	M	Current month of year in short three-letter format (for example, Jan, Feb)
d	Numerical day of month returned as two digits (for example, 01, 02)	s	Seconds in current minute from 00 to 59 (for example, 07, 50)
F	Current month in long format (for example, January, February)	t	Number of days in current month (28, 29, 30, or 31)
h	Current hour in day from 01 to 12 (for example, 02, 11)	U	Number of seconds since the epoch (usually since January 1, 1970)
H	Current hour in day from 00 to 23 (for example, 01, 18).	w	Current day of week from 0 to 6 (where 0 is Sunday, 1 is Monday, and so on)
i	Current minute from 00 to 59 (for example, 05, 46)	y	Current year returned in two digits (for example, 01, 02)
l	Current day of week in long format (for example, Sunday, Monday)	Y	Current year returned in four digits (for example, 2001, 2002)
L	Returns 1 if it is a leap year or 0 otherwise	z	Day number of the year from 0 to 365 (where January 1 is day 0, January 2 is day 1, and so on)
m	Current month of year from 01 to 12		

More About date()

- You can combine multiple character formats return more than one format from the date()
  - For example,

    ```
    $today = date( 'l, F d, Y');
    print "Today=$today";
    ```

- On September 10, 2009, would output
  - “Today=Thursday, September 10, 2009”.

A Full Example ...

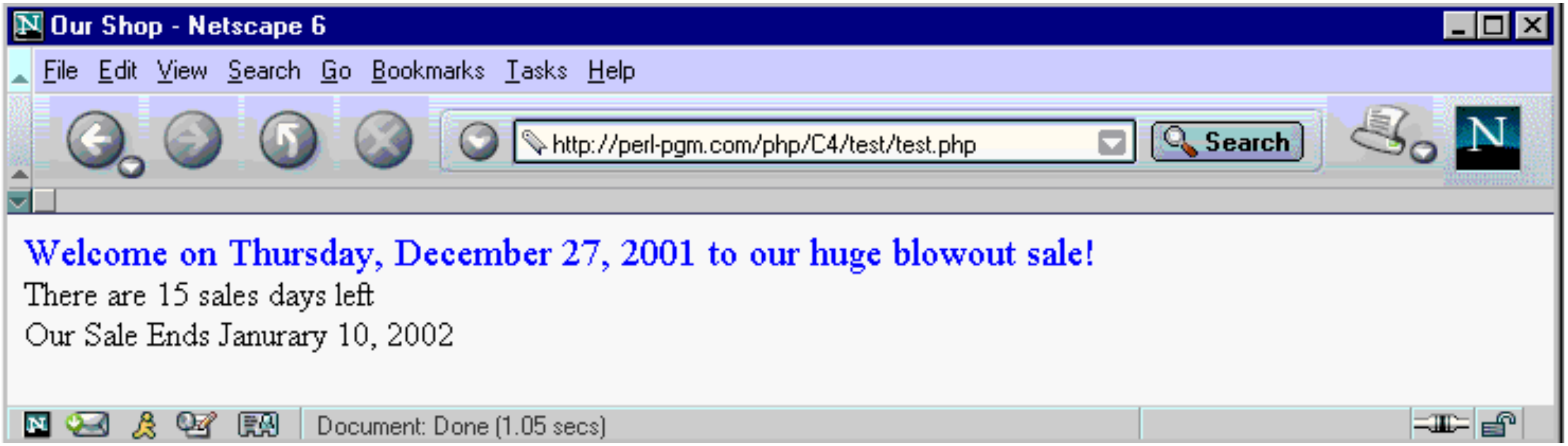
- Consider the following Web application that uses date() to determine the current date and the number of days remaining in a store’s sale event.

Receiving Code

```
<html> <head><title> Our Shop </title> </head>
<body> <font size=4 color="blue">
<?php
$today = date( 'l, F d, Y');
print "Welcome on $today to our huge blowout sale! </font>";
$month = date('m');
$year = date('Y');
$dayofyear = date('z');
if ($month == 12 && $year == 2001) {
   $daysleft = (365 - $dayofyear + 10);
   print "<br> There are $daysleft sales days left";
   } elseif ($month == 01 && $year == 2002) {
   if ($dayofyear <= 10) {
     $daysleft = (10 - $dayofyear);
     print "<br> There are $daysleft sales days left";
  } else {
      print "<br>Sorry, our sale is over.";
   }
 } else {
    print "<br>Sorry, our sale is over.";
 }
 print "<br>Our Sale Ends January 10, 2002";
 ?> </body></html>
```

The Output ...

The previous code can be executed at http://webwizard.aw.com/~phppgm/C3/date.php


Our Shop - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://perl-pgm.com/php/C4/test/test.php
Search
Document: Done (1.05 secs)

Welcome on Thursday, December 27, 2001 to our huge blowout sale!
There are 15 sales days left
Our Sale Ends Janurary 10, 2002

Content

1. Basic PHP Functions
2. Write your own functions
3. Using External Script Files

2. Writing your own functions

- User Defined Functions provide a way to group a set of statements, set them aside, and turn them into mini-scripts within a larger script.
  - *Scripts that are easier to understand and change.*
  - *Reusable script sections.*
  - *Smaller program size*

2.2. Function definition

- Use the following general format

Include parentheses at the end of the function name

```
function function_name() {
   set of statements
}
```

The function runs these statements when called

Enclose in curly brackets.

For example …

- Consider the following:

```
function OutputTableRow() {
        print
  '<tr><td>One</td><td>Two</td></tr>';
}
```

- You can run the function by executing

```
OutputTableRow();
```

As a full example …

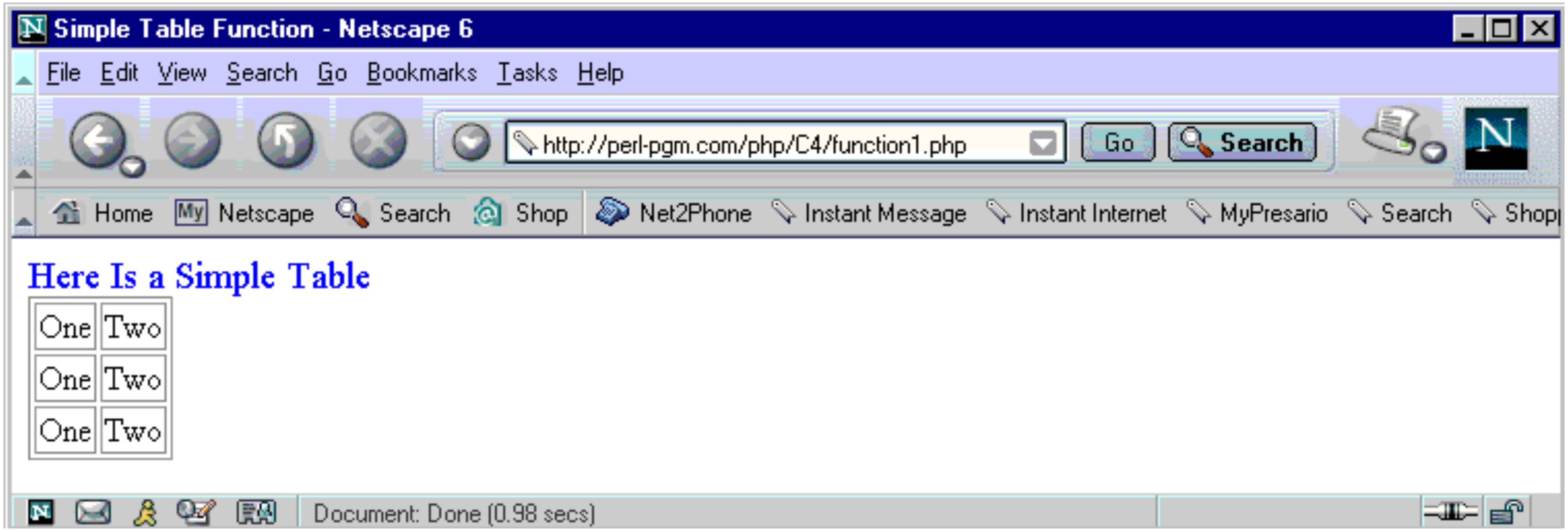
```
<html>
<head><title> Simple Table Function </title> </head>
<body>
<font color="blue" size="4"> Here Is a Simple Table
<table border=1>
<?php
   function OutputTableRow() {
        print '<tr><td>One</td><td>Two</td></tr>';
    }
  OutputTableRow();
  OutputTableRow();
  OutputTableRow();
?>
</table></body></html>
```

OutputTableRow()

function definition.

Three consecutive calls to the OutputTableRow() function

Would have the following output …


Simple Table Function - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://perl-pgm.com/php/C4/function1.php
Go
Search
Home
Netscape
Search
Shop
Net2Phone
Instant Message
Instant Internet
MyPresario
Search
Shop
Document: Done (0.98 secs)

Here Is a Simple Table

One	Two
One	Two
One	Two

TIP: Use Comments at the Start of a Function

- It is good practice to place comments at the start of a function
- For example,

```
function OutputTableRow() {
        // Simple function that outputs 2 table
  cells
        print
  '<tr><td>One</td><td>Two</td></tr>';
}
```

2.3. Passing Arguments to Functions

- Input variables to functions are called *arguments to the function*
- For example, the following sends 2 arguments
  - `OutputTableRow("A First Cell", "A Second Cell");`
- Within function definition can access values

```
function OutputTableRow($col1, $col2) {
        print
  "<tr><td>$col1</td><td>$col2</td></tr>";
}
```

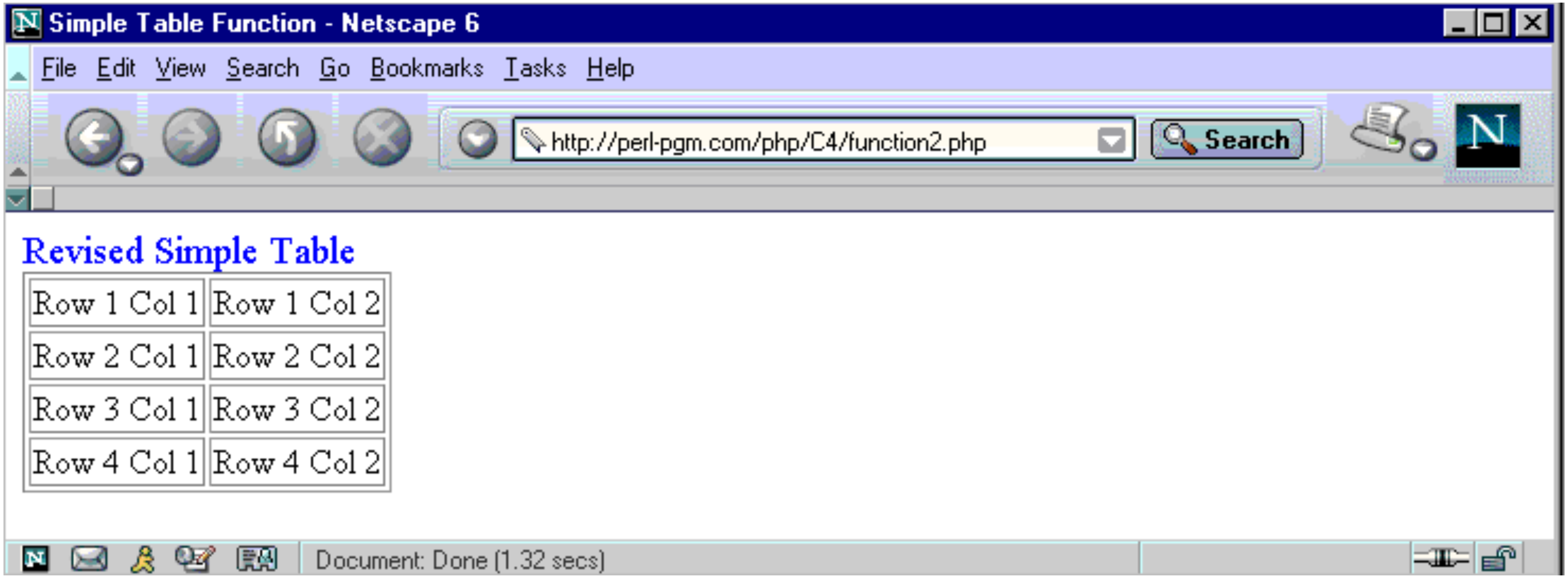
Consider the following code …

```
<html>
<head><title> Simple Table Function </title> </head> <body>
<font color="blue" size=4> Revised Simple Table
<table border=1>
<?php
function OutputTableRow( $col1, $col2 ) {
    print "<tr><td>$col1</td><td>$col2</td></tr>";
}
for ( $i=1; $i<=4; $i++ ) {
    $message1="Row $i Col 1";
    $message2="Row $i Col 2";
    OutputTableRow( $message1, $message2 );
}
?>
</table></body></html>
```

OutputTableRow()
Function definition.

Four calls to
OuputTableRow()

Would output the following …



Simple Table Function - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://perl-pgm.com/php/C4/function2.php
Search
Document: Done (1.32 secs)

Revised Simple Table

Row 1 Col 1	Row 1 Col 2
Row 2 Col 1	Row 2 Col 2
Row 3 Col 1	Row 3 Col 2
Row 4 Col 1	Row 4 Col 2

2.4. Returning Values

- Your functions can return data to the calling script.
  - For example, your functions can return the results of a computation.

- You can use the PHP return statement to return a value to the calling script statement:

```
    return $result;
```

This variable's value will be returned to the calling script.

Example function

```
function Simple_calc( $num1, $num2 ) {
   // PURPOSE: returns largest of 2 numbers
   // ARGUMENTS: $num1 -- 1st number, $num2 -- 2nd number
  if ($num1 > $num2) {
    return($num1);
  } else {
    return($num2);
  }
}
```

Return $num1 when it is the larger value.

Return $num2 when it is the larger value.

What is output if called as follows:

```
$largest = Simple_calc(15, -22);
```

A Full Example ...

- Consider a script that calculates the percentage change from starting to an ending value
- Uses the following front-end form:

```
Starting Value: <input type="text" size="15”
                maxlength="20" name="start">
Ending Value: <input type="text" size="15”
                maxlength="20" name="end">
```

The Source Code ...

```
<html>
<head><title> Your Percentage Calculation </title></head><body>
<font color="blue" size=4> Percentage Calculator </font>
<?php
function Calc_perc($buy, $sell) {
    $per = (($sell - $buy) / $buy) *100;
    return($per);
}
print "<br>Your starting value was $start.";
print "<br>Your ending value was $end.";
if (is_numeric($start) && is_numeric($end) ) {
    if ($start != 0) {
          $per = Calc_perc($start, $end);
          print "<br> Your percentage change was $per %.";
    } else { print "<br> Error! Starting values cannot be zero "; }
} else {
     print "<br> Error! You must have valid numbers for start and end ";
 }
?> </body></html>
```

Calculate the percentage change from the starting value to the ending value.

The call to Calc_perc() returns the percentage change into $per.

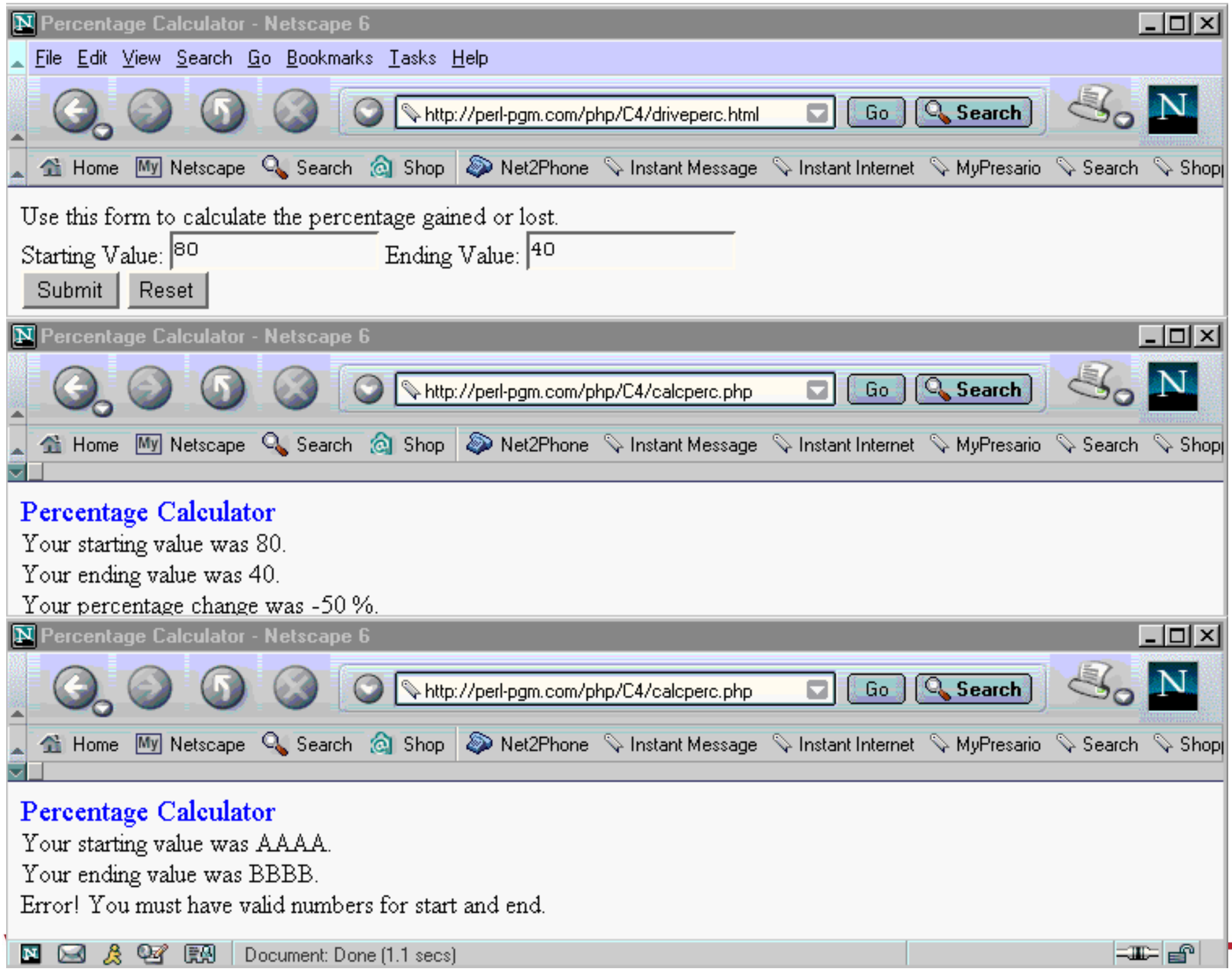
The Source Code with REGISTER_GLOBALS Off

```
<html>
<head><title> Your Percentage Calculation </title></head>
<font color="blue" size=4> Percentage Calculator </font>
<?php
function Calc_perc($buy, $sell) {
    $per = (($sell - $buy) / $buy) *100;
    return($per);
}
$start = $_POST[“start”]; $end = $_POST[“end”];
print "<br>Your starting value was $start.";
print "<br>Your ending value was $end.";
if (is_numeric($start) && is_numeric($end) ) {
    if ($start != 0) {
         $per = Calc_perc($start, $end);
         print "<br> Your percentage change was $per %.";
    } else { print "<br> Error! Starting values cannot be zero "; }
} else {
  print "<br> Error! You must have valid numbers for start and end ";
 }
?> </body></html>
```

Calculate the percentage change from the starting value to the ending value.

The call to Calc_perc() returns the percentage change into $per.

Would Output The Following...


Percentage Calculator - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
Go
Search
Home
My Netscape
Search
Shop
Net2Phone
Instant Message
Instant Internet
MyPresario
Search
Shop
Submit
Reset
Document: Done (1.1 secs)

http://perl-pgm.com/php/C4/driveperc.html

Use this form to calculate the percentage gained or lost.

Starting Value: 80 Ending Value: 40

http://perl-pgm.com/php/C4/calcperc.php

Percentage Calculator

Your starting value was 80.

Your ending value was 40.

Your percentage change was -50 %.

http://perl-pgm.com/php/C4/calcperc.php

Percentage Calculator

Your starting value was AAAA.

Your ending value was BBBB.

Error! You must have valid numbers for start and end.

A Full Script Example (with RESGISTER_GLOBALS off)...

```
<html>
<head><title>While Loop</title></head>
<body>
<font size="4" color="blue"> Table of Square and Cube Values </font>
<table border=1>
<th> Numb </th> <th> Sqr </th> <th> Cubed </th>
<?php
   $start = $_POST[“start”];  $end = $_POST[“end”];
   $i = $start;
    while ($i <= $end) {
       $sqr=$i*$i;
       $cubed=$i*$i*$i;
       print ("<tr><td>$i</td><td>$sqr</td><td>$cubed</td></tr>");
       $i = $i + 1;
    }
?></table></body></html>
```

The Output …

The previous code can be executed at
http://webwizard.aw.com/~phppgm/C3/whileloop.php

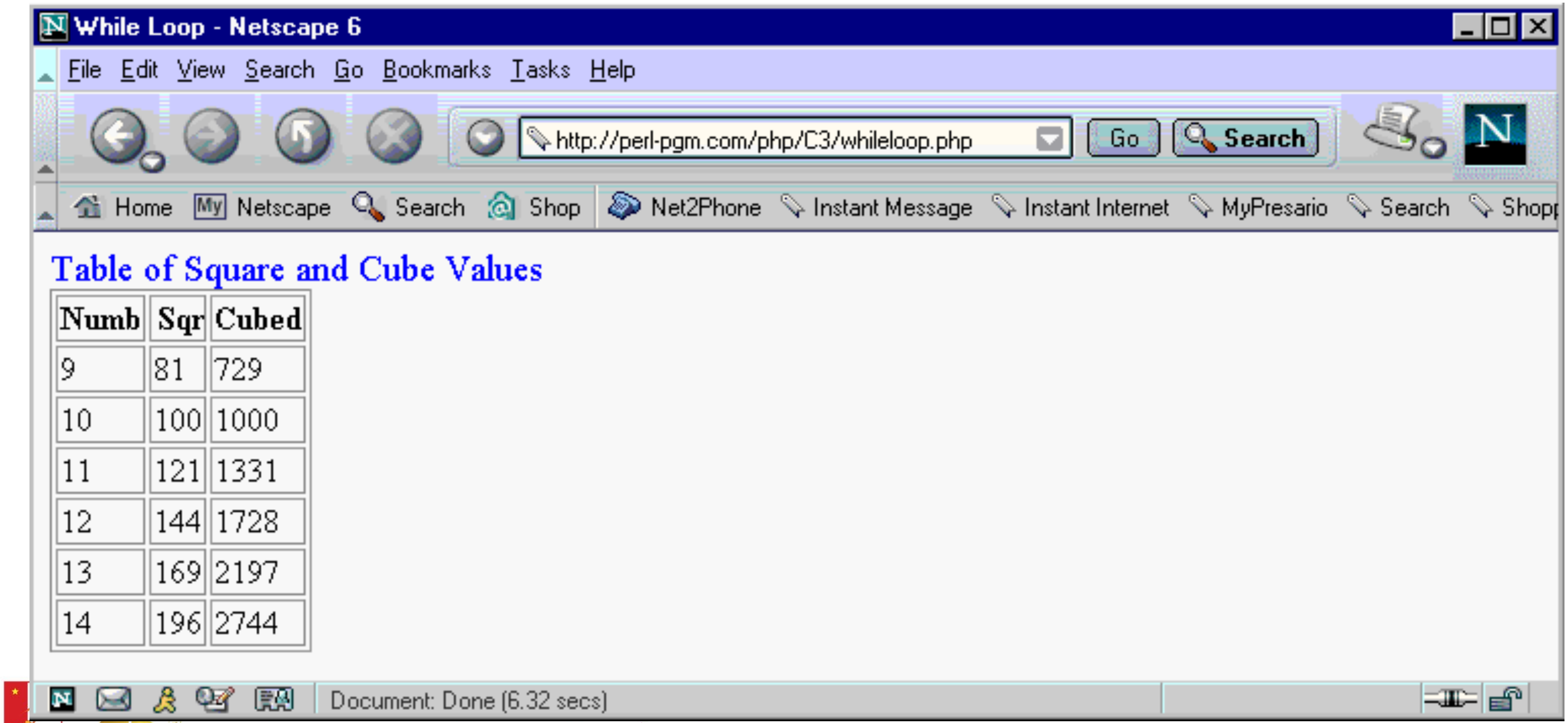


Table of Square and Cube Values

Numb	Sqr	Cubed
9	81	729
10	100	1000
11	121	1331
12	144	1728
13	169	2197
14	196	2744

Content

1. Basic PHP Functions
2. Write your own functions
3. Using External Script Files

3. Using External Script Files

- Sometime you will want to use scripts from external files.
- PHP supports 2 related functions:

The require() function produces a fatal error if it can’t insert the specified file.

```
require ("header.php");
include ("trailer.php");
```

The include() function produces a warning if it can’t insert the specified file.

- Both search for the file named within the double quotation marks and insert its PHP, HTML, or JavaScript code into the current file.

Consider the following example

```
<font size=4 color="blue">
Welcome to Harry’s Hardware Heaven!
</font><br> We sell it all for you!<br>
<?php
    $time = date('H:i');
    function Calc_perc($buy, $sell) {
        $per = (($sell - $buy ) / $buy) * 100;
        return($per);
    }
?>
```

The script will output these lines when the file is included.

The value of $time will be set when the file is included.

This function will be available for use when the file is included.

header.php

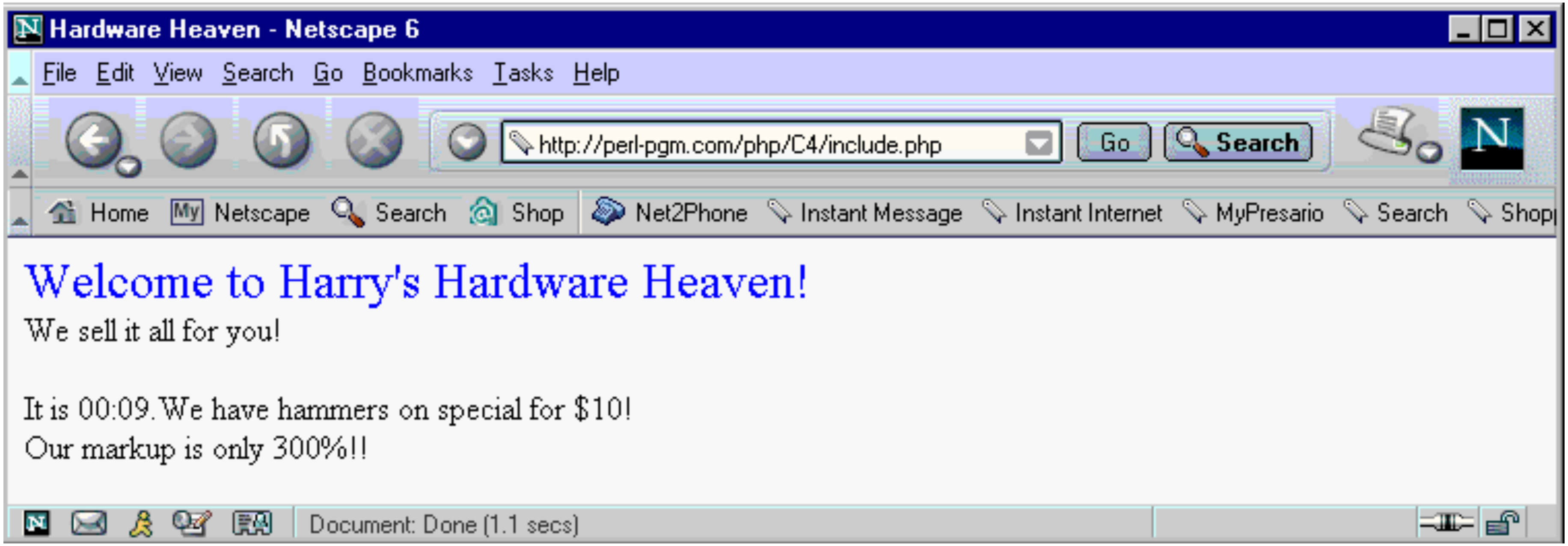
- If the previous script is placed into a file called header.php …

```
<html><head><title> Hardware Heaven </title></head> <body>
<?php
include("header.php");
$buy = 2.50;
$sell = 10.00;
print "<br>It is $time.";
print "We have hammers on special for \$$sell!";
$markup = Calc_perc($buy, $sell);
print "<br>Our markup is only $markup%!!";
?>
</body></html>
```

Include the file header.php

Calc_perc() is defined in header.php

Would output the following ...


Hardware Heaven - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://perl-pgm.com/php/C4/include.php
Go
Search
Home
My Netscape
Search
Shop
Net2Phone
Instant Message
Instant Internet
MyPresario
Search
Shop

Welcome to Harry's Hardware Heaven!

We sell it all for you!

It is 00:09. We have hammers on special for $10!
Our markup is only 300%!!

Document: Done (1.1 secs)

More Typical Use of External Code Files

- More typically might use one or more files with only functions and other files that contain HTML
- For example, might use the following as footer.php.

```
<hr>
Hardware Harry's is located in beautiful downtown
  Hardwareville.
<br>We are open every day from 9 A.M. to
  midnight, 365 days a year.
<br>Call 476-123-4325. Just ask for Harry.
</body></html>
```

- Can include using:

```
<?php include("footer.php"); ?>
```

Summary

- PHP provides several functions useful including abs(), round(), is_numeric(), rand(), date()
- Programmer-defined functions allow you to group a set of statements, set them aside, and turn those grouped statements into mini-scripts.

Question?