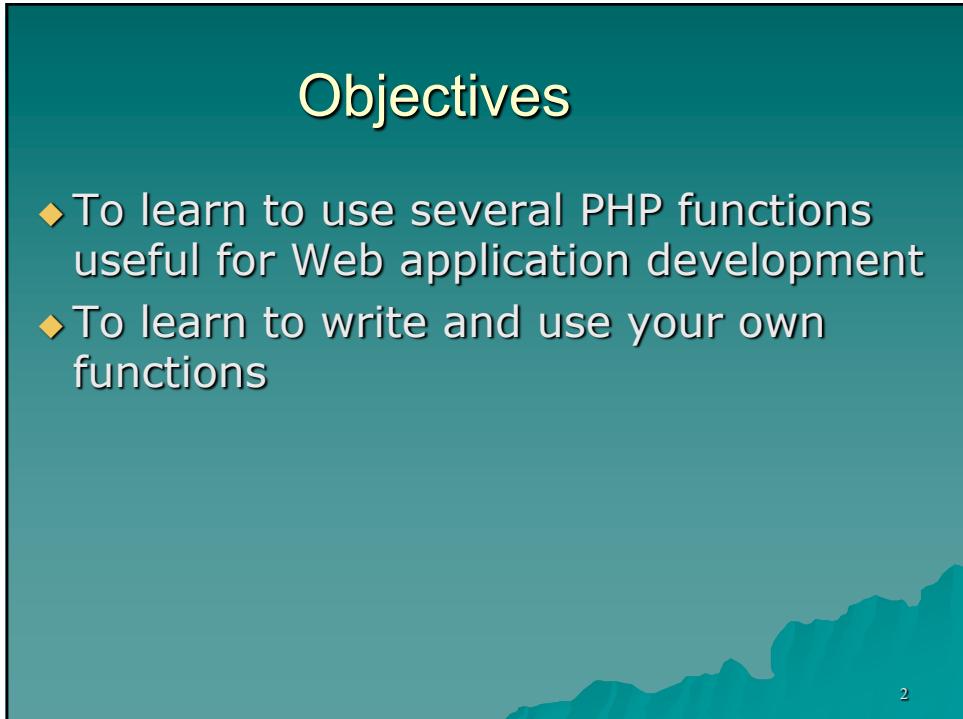


Web Programming

Chapter 3.2. Functions

1



Objectives

- ◆ To learn to use several PHP functions useful for Web application development
- ◆ To learn to write and use your own functions

2

2

Content

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

3

Content

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

4

4

2

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.

5

Numeric PHP Functions

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

6

6

3

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

7

7

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

8

8

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

9

9

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

10

10

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`

11

11

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);`

12

12

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.

13

13

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;
```

14

14

Generating HTML Tags with print()

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

15

15

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
```

16

16

Receiving Code

```
1. <html>
2. <head><title> Coin Flip Results </title></head> <body> <?php
3. srand ((double) microtime() * 10000000);
4. $flip = rand( 0, 1 );
5. if ( $flip == 0 && $pick == 0 ) { Check whether both
   the coin flip and the
   guess are heads.
6.     print "The flip=$flip, which is heads! <br> ";
7.     print '<font color="blue"> You got it right!</font>';
8. } elseif ( $flip == 0 && $pick == 1 ) {
9.     print "The flip=$flip, which is heads! <br> ";
10.    print '<font color="red"> You got it wrong!</font>';
11. } elseif ( $flip == 1 && $pick == 1 ) { Check whether the
   coin flip is heads but
   the guess is tails.
12.     print "The flip=$flip, which is tails! <br>";
13.     print '<font color="blue"> You got it right!</font>' Check whether both
   the coin flip and the
   guess are tails.
```

17

Receiving Code continued...

```
14. } elseif ( $flip == 1 && $pick == 0 ) { Check whether the coin flip
   is tails but the guess is heads.
15.     print "The flip=$flip, which is tails! <br>";
16.     print '<font color="red"> You got it wrong!</font>';
17. } else {
18.     print "<br>Illegal state error!";
19. }
20. ?> </body></html>
```

18

18

Receiving Code With REGISTER_GLOBALS Off

```
1. <html>
2. <head><title> Coin Flip Results </title></head> <body> <?php
3.   srand ((double) microtime() * 10000000);
4.   $pick = $_POST["PICK"];
5.   $flip = rand( 0, 1 );
6.   if ( $flip == 0 && $pick == 0 ) {           Check whether both
                                                the coin flip and the
                                                guess are heads.
7.     print "The flip=$flip, which is heads! <br> ";
8.     print '<font color="blue"> You got it right!</font>';
9.   } elseif ( $flip == 0 && $pick == 1 ) {
10.     print "The flip=$flip, which is heads! <br> ";
11.     print '<font color="red"> You got it wrong!</font>';
12.   } elseif ( $flip == 1 && $pick == 1 ) {
13.     print "The flip=$flip, which is tails! <br> ";
                                                Check whether both
                                                the coin flip and the
                                                guess are tails.
                                                Check whether the
                                                coin flip is heads but
                                                the guess is tails.
```

19

19

Receiving Code With REGISTER_GLOBALS Off, cont. ...

```
14.   print '<font color="blue"> You got it right!</font>';
15. } elseif ( $flip == 1 && $pick == 0 ) {
16.   print "The flip=$flip, which is tails! <br>";
17.   print '<font color="red"> You got it
            wrong!</font>';
18. } else {
19.   print "<br>Illegal state error!";
20. }
21. ?> </body></html>
```

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

20

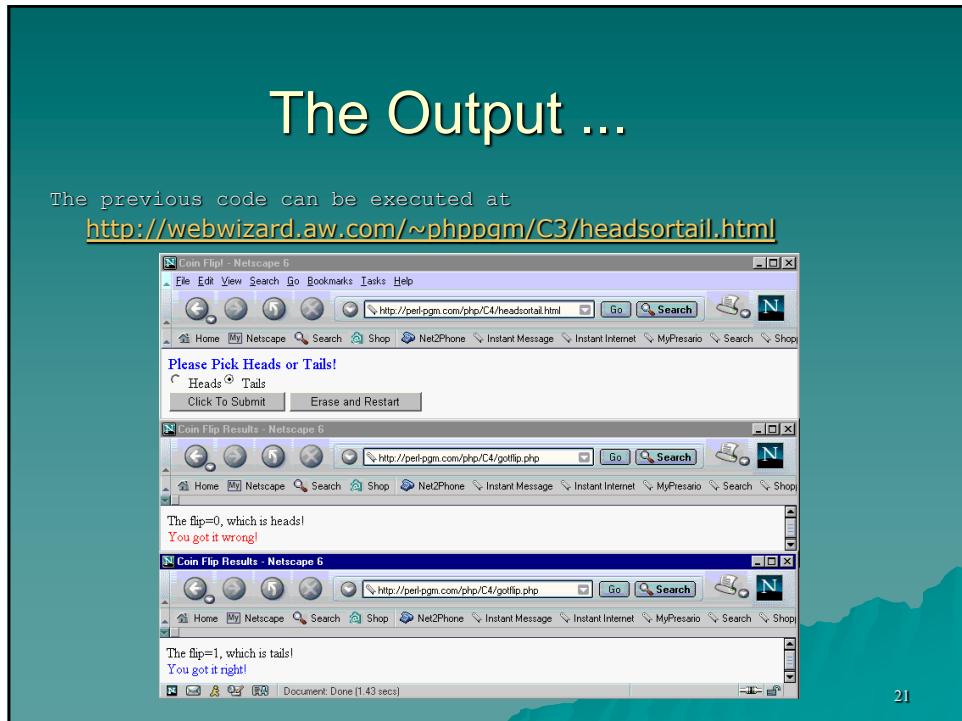
20

10

The Output ...

The previous code can be executed at

<http://webwizard.aw.com/~phppqm/C3/headsortail.html>



21

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.

22

22

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");
```

23

23

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

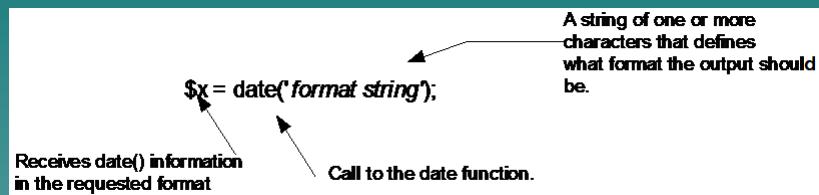
```
if (! print("Hello, world")) {
    die("you're not listening to me!");
}
```

24

24

1.8. The date() Function

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



- ◆ The format string defines the format of the date() function's output:

◆ \$day = date('d'); ← Request date() to return the numerical day of the month.
◆ print "day=\$day";

- ◆ If executed on September 16, 2010, then it would output "day=16".

25

25

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)
I	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		

26

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".

27

27

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.

28

28

```

1. <html> <head><title> Our Shop </title> </head>
2. <body> <font size=4 color="blue">
3. <?php
4. $today = date( 'l, F d, Y');
5. print "Welcome on $today to our huge blowout sale! </font>";
6. $month = date('m');
7. $year = date('Y');
8. $dayofyear = date('z');
9. if ($month == 12 && $year == 2001) {
10.   $daysleft = (365 - $dayofyear + 10);
11.   print "<br> There are $daysleft sales days left";
12. } elseif ($month == 01 && $year == 2002) {
13.   if ($dayofyear <= 10) {
14.     $daysleft = (10 - $dayofyear);
15.     print "<br> There are $daysleft sales days left";
16.   } else {
17.     print "<br>Sorry, our sale is over.";
18.   }
19. } else {
20.   print "<br>Sorry, our sale is over.";
21. }
22. print "<br>Our Sale Ends January 10, 2002";
23. ?
24. > </body></html>

```

Receiving Code

29

29

The Output ...

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



30

30

Content

1. Basic PHP Functions

⇒ 2. Write your own functions

3. Using External Script Files

31

31

2. Writing your own functions

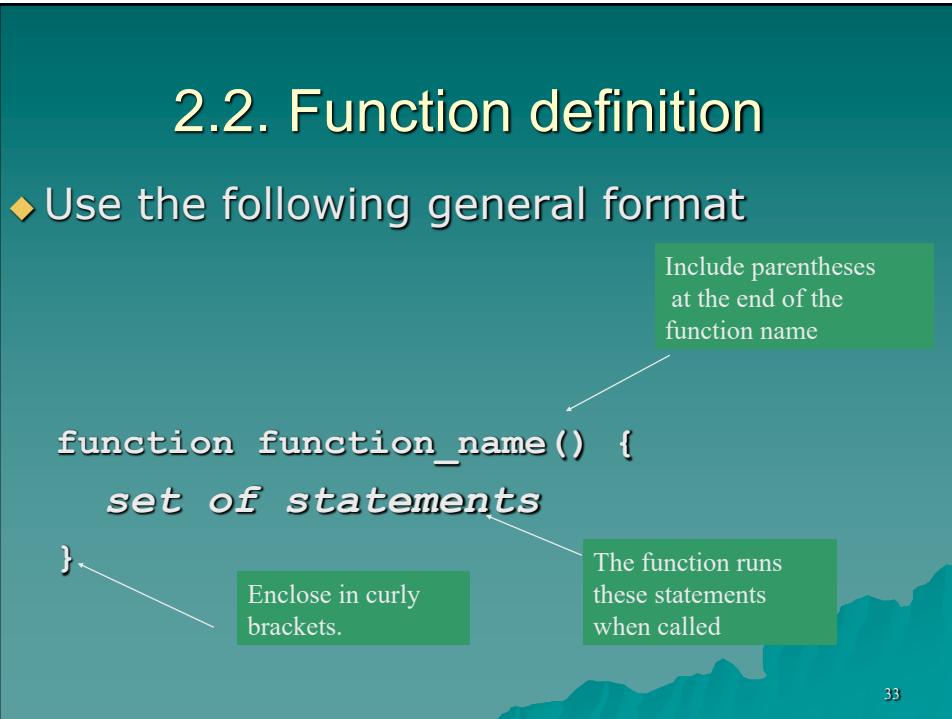
- ◆ User Defined Functions provide a way to group a set of statements, set them aside, and turn them into mini-scripts within a larger script.
 - *Scripts that are easier to understand and change.*
 - *Reusable script sections.*
 - *Smaller program size*

32

32

2.2. Function definition

- ◆ Use the following general format



33

33

For example ...

- ◆ Consider the following:

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

- ◆ You can run the function by executing
`OutputTableRow();`

34

34

As a full example ...

```
1. <html>
2. <head><title> Simple Table Function </title> </head> <body>
3. <font color="blue" size="4"> Here Is a Simple Table <table border=1>
4. <?php
5.     function OutputTableRow() {
6.         print '<tr><td>One</td><td>Two</td></tr>';
7.     }
8.     OutputTableRow();
9.     OutputTableRow();
10.    OutputTableRow();
11. ?>
12. </table></body></html>
```

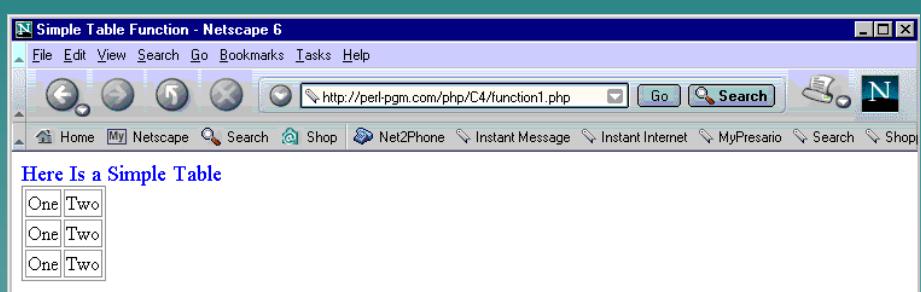
OutputTableRow()
function definition.

Three consecutive calls
to the OutputTableRow()
function

35

35

Would have the following output ...



36

36

18

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>';  
}
```

37

37

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>";  
}
```

38

38

19

Consider the following code ...

```
1. <html>
2. <head><title> Simple Table Function </title> </head>
3. <font color="blue" size=4> Revised Simple Table <table border=1>
4. <?php
5. function OutputTableRow( $col1, $col2 ) {
6.     print "<tr><td>$col1</td><td>$col2</td></tr>";
7. }
8. for ( $i=1; $i<=4; $i++ ) {
9.     $message1="Row $i Col 1";
10.    $message2="Row $i Col 2";
11.    OutputTableRow( $message1, $message2 );
12. }
13.?
14. </table></body></html>
```

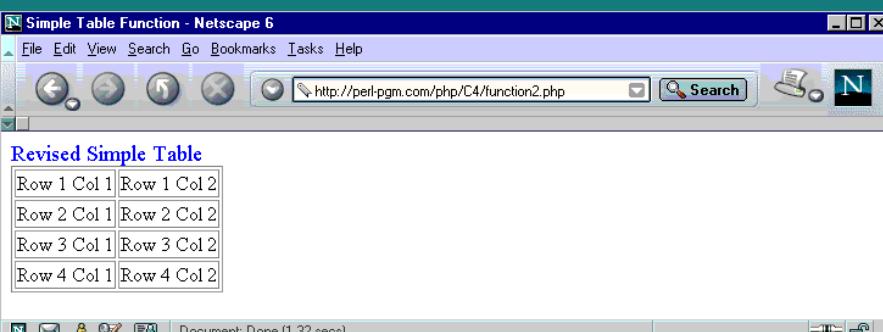
OutputTableRow()
Function definition.

Four calls to
OutputTableRow()

39

39

Would output the following ...



40

40

20

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.

41

41

Example function

```
1. function Simple_calc( $num1, $num2 ) {  
2.     // PURPOSE: returns largest of 2 numbers  
3.     // ARGUMENTS: $num1 -- 1st number, $num2 -- 2nd number  
4.     if ($num1 > $num2) {  
5.         return($num1); ← Return $num1 when it is  
6.     } else {  
7.         return($num2); ← Return $num2 when it is  
8.     }  
9. }
```

What is output if called as follows:

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

42

42

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">
```

43

43

The Source Code ...

```
1. <html>  
2. <head><title> Your Percentage Calculation </title></head><body>  
3. <font color="blue" size=4> Percentage Calculator </font>  
4. <?php  
5. function Calc_perc($buy, $sell) {  
6.     $per = (($sell - $buy) / $buy) *100; ← Calculate the percentage  
7.     return($per); ← change from the starting  
8. } ← value to the ending value.  
9. print "<br>Your starting value was $start.";  
10. print "<br>Your ending value was $end.";  
11. if (is_numeric($start) && is_numeric($end) ) {  
12.     if ($start != 0) {  
13.         $per = Calc_perc($start, $end); ← The call to Calc_perc()  
14.         print "<br> Your percentage change was $per %.";  
15.     } else { print "<br> Error! Starting values cannot be zero " ; }  
16. } else {  
17.     print "<br> Error! You must have valid numbers for start and end " ;  
18. }  
19. ?> </body></html>
```

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

44

44

The Source Code with REGISTER_GLOBALS Off

```

1. <html>
2. <head><title> Your Percentage Calculation </title></head><body>
3. <font color="blue" size=4> Percentage Calculator </font>
4. <?php
5. function Calc_perc($buy, $sell) {
6.     $per = (($sell - $buy) / $buy) *100; ←
7.     return($per);
8. }
9. $start = $_POST["start"]; $end = $_POST["end"];
10. print "<br>Your starting value was $start.";
11. print "<br>Your ending value was $end.";
12. if (is_numeric($start) && is_numeric($end) ) {
13.     if ($start != 0) {
14.         $per = Calc_perc($start, $end); ←
15.         print "<br> Your percentage change was $per %.";
16.     } else { print "<br> Error! Starting values cannot be zero ";
17. } else {
18.     print "<br> Error! You must have valid numbers for start and end ";
19. }
20. ?> </body></html>

```

45

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

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

45

Would Output The Following...

Percentage Calculator - Netscape 6
 Use this form to calculate the percentage gained or lost.
 Starting Value: 80 Ending Value: 40
 Submit Reset

Percentage Calculator - Netscape 6
 Percentage Calculator
 Your starting value was 80.
 Your ending value was 40.
 Your percentage change was -50 %

Percentage Calculator - Netscape 6
 Percentage Calculator
 Your starting value was AAAA.
 Your ending value was BBBB.
 Error! You must have valid numbers for start and end.

46

46

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

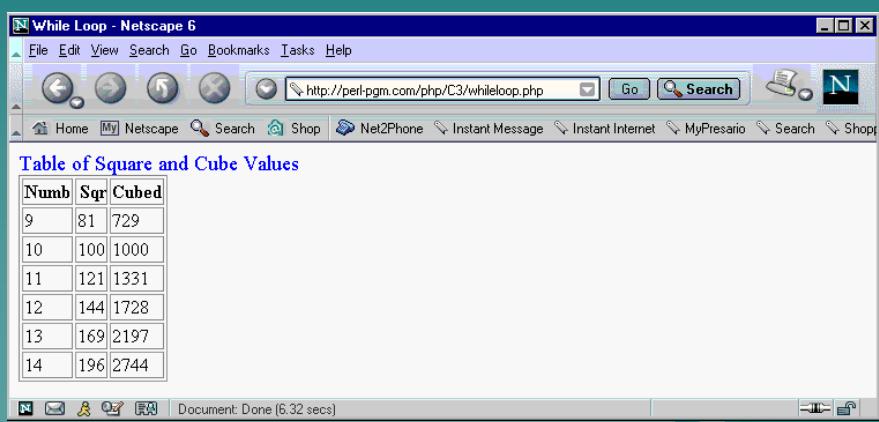
```
1. <html>
2. <head><title>While Loop</title></head>
3. <body>
4. <font size="4" color="blue"> Table of Square and Cube Values </font>
5. <table border=1>
6. <th> Numb </th> <th> Sqr </th> <th> Cubed </th>
7. <?php
8.     $start = $_POST["start"];  $end = $_POST["end"];
9.     $i = $start;
10.    while ($i <= $end) {
11.        $sqr=$i*$i;
12.        $cubed=$i*$i*$i;
13.        print ("<tr><td>$i</td><td>$sqr</td><td>$cubed</td></tr>");
14.        $i = $i + 1;
15.    }
16.?></table></body></html>
```

47

47

The Output ...

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



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

48

48

Content

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

49

49

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.

50

50

Consider the following example

```
1. <font size=4 color="blue">
2. Welcome to Harry's Hardware Heaven!
3. </font><br> We sell it all for you!<br>
4. <?php
5.     $time = date('H:i');
6.     function Calc_perc($buy, $sell) {
7.         $per = (($sell - $buy) / $buy) * 100;
8.         return($per);
9.     }
10. ?>
```

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.

51

51

header.php

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

```
1. <html><head><title> Hardware Heaven </title></head> <body>
2. <?php
3. include ("header.php");
4. $buy = 2.50;
5. $sell = 10.00;
6. print "<br>It is $time.";
7. print "We have hammers on special for \$\$sell!";
8. $markup = Calc_perc($buy, $sell);
9. print "<br>Our markup is only $markup%!!";
10. ?>
11. </body></html>
```

Include the file header.php

Calc_perc() is defined in header.php

52

52

Would output the following ...



53

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"); ?>
```

54

54

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.

55

55

Question?



56

56

28