CIS 4004: Web-Based Information Technology
Spring 2011

Introduction to PHP – Part 3 - Arrays

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cis4004/spr2011

Department of Electrical Engineering and Computer Science
University of Central Florida

Arrays In PHP

- Most of our PHP examples to this point have involved scalar variables (we did see a couple of example in the first section of notes that made use of one of PHP's global associative arrays).
- Scalar variables can only hold a single value at a time. For example, a variable `$color` could hold only a single value such as `red`, at any point in time. The variable could not be used to hold more than one color.
- Arrays are special types of variables that enable you to store as many values as you want.

Note: Although you can technically make an array as large as you'd like, some built-in array handling functions in PHP have an upper limit of 100,000 values. If you are storing more data that this in your arrays and you need to use one of these functions, you will either need to write your own function or split the data into multiple arrays.

Arrays In PHP

- Arrays are indexed, which means that each entry in the array, called an element, is made up of a key and a value.
- The key is the index position, beginning with 0 and increasing incrementally by 1 with each new element in the array.
- The value is whatever value you associate with that position – a string, an integer, or whatever you want.
- In PHP you can think of an array as a filing cabinet and each key/value pair as a file folder. The key is the label written on the tab of the folder, and the value is what is inside. What's inside each folder can vary from folder to folder.

Creating Arrays In PHP

- You can create an array using either the `array()` function or the array operator `[]`.
- The `array()` function is usually used when you want to create a new array and populate it with more than one element, all at the same time.
- The array operator is more often used when you want to create a new array with just one element at the outset or when you want to add to an existing array element.
- The examples on the following couple of pages illustrate creating an array in PHP using these two techniques.

C:\xampp\htdocs\CIS 4004\CNT 4714\array definition V1.php - Notepad++

File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?

config.php | functionWithReturnValue.php | array definition V1.php

```
1  <html>
2  <head>
3  <title>Creating An Array - Version 1</title>
4  </head>
5  <body  style = "font-family: arial, sans-serif;
6        background-color: #856363"  background=image1.jpg>
7  <?php
8      $rainbow = array("red", "orange", "yellow", "grren", "blue", "indigo", "violet");
9      for ($i=0; $i<=6; $i++) {
10        echo 'Rainbow['. $i.'] color is: '. $rainbow[$i]."<br />";
11     }
12 ?>
13 </body>
14 </html>
```

PHP Hypertext Pre | length : 390 lines : 14 | Ln : 13 Col : 8 Sel : 0 | Dos\Windows

This version uses the `array()` function to create the array.

Creating An Array - Version 1 ...

Menu | Cre...

Rainbow[0] color is: red
Rainbow[1] color is: orange
Rainbow[2] color is: yellow
Rainbow[3] color is: grren
Rainbow[4] color is: blue
Rainbow[5] color is: indigo
Rainbow[6] color is: violet

View (90%)

C:\xampp\htdocs\CIS 4004\CNT 4714\array definition V1.php - Notepad++

File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?

config.php | functionWithReturnValue.php | array definition V1.php

```
1  <html>
2  <head>
3  <title>Creating An Array - Version 2</title>
4  </head>
5  <body  style = "font-family: arial, sans-serif;
6        background-color: #856363"  background=image1.jpg>
7  <?php
8      $rainbow[] = "red";
9      $rainbow[] = "orange";
10     $rainbow[] = "yellow";
11     $rainbow[] = "green";
12     $rainbow[] = "blue";
13     $rainbow[] = "indigo";
14     $rainbow[] = "violet";
15     for ($i=0; $i<=6; $i++) {
16         echo 'Rainbow['. $i.'] color is: '. $rainbow[$i]."<br /
17     }
18 ?>
19 </body>
20 </html>
```

length : 475 lines : 20 | Ln : 19 Col : 8 Sel : 0 | Dos\Windows | ANSI

This version uses the array operator [ ] to create the array.
Note that no index values are specified, PHP will auto number for you

Creating An Array - Version 2 - Ope...

Menu | Creating A...

Rainbow[0] color is: red
Rainbow[1] color is: orange
Rainbow[2] color is: yellow
Rainbow[3] color is: green
Rainbow[4] color is: blue
Rainbow[5] color is: indigo
Rainbow[6] color is: violet

View (90%)

C:\xampp\htdocs\CIS 4004\CNT 4714\array definition V3.php - Notepad++

File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?

config.php | functionWithReturnValue.php | array definition V3.php

```
1  <html>
2  <head>
3  <title>Creating An Array - Version 3</title>
4  </head>
5  <body  style = "font-family: arial, sans-serif;
6        background-color: #856363"  background=image1.jpg>
7  <?php
8      $rainbow[0] = "red";
9      $rainbow[1] = "orange";
10     $rainbow[2] = "yellow";
11     $rainbow[3] = "green";
12     $rainbow[4] = "blue";
13     $rainbow[5] = "indigo";
14     $rainbow[6] = "violet";
15     for ($i=0; $i<=6; $i++) {
16         echo 'Rainbow['. $i.'] color is: '. $rainbow[$i]."<b
17     }
18 ?>
19 </body>
20 </html>
```

length : 482 lines : 20 | Ln : 20 Col : 8 Sel : 0 | Dos\Windows | ANSI

This version also uses the array operator [ ] to create the array.
Note that index values are specified in this case.

Creating An Array - Version 3 - Opera

Menu | Creating An...

Rainbow[0] color is: red
Rainbow[1] color is: orange
Rainbow[2] color is: yellow
Rainbow[3] color is: green
Rainbow[4] color is: blue
Rainbow[5] color is: indigo
Rainbow[6] color is: violet

View (90%)

Creating Arrays In PHP

- As shown in the example on page 6, PHP can automatically index the array for you when you use the [ ] operator to create the array.
- This is useful in that it eliminates the possibility that you might misnumber the elements. The example on the next page illustrates what happens if you misnumber the elements in an array.

C:\xampp\htdocs\CIS 4004\CNT 4714\array definition V3.php - Notepad++

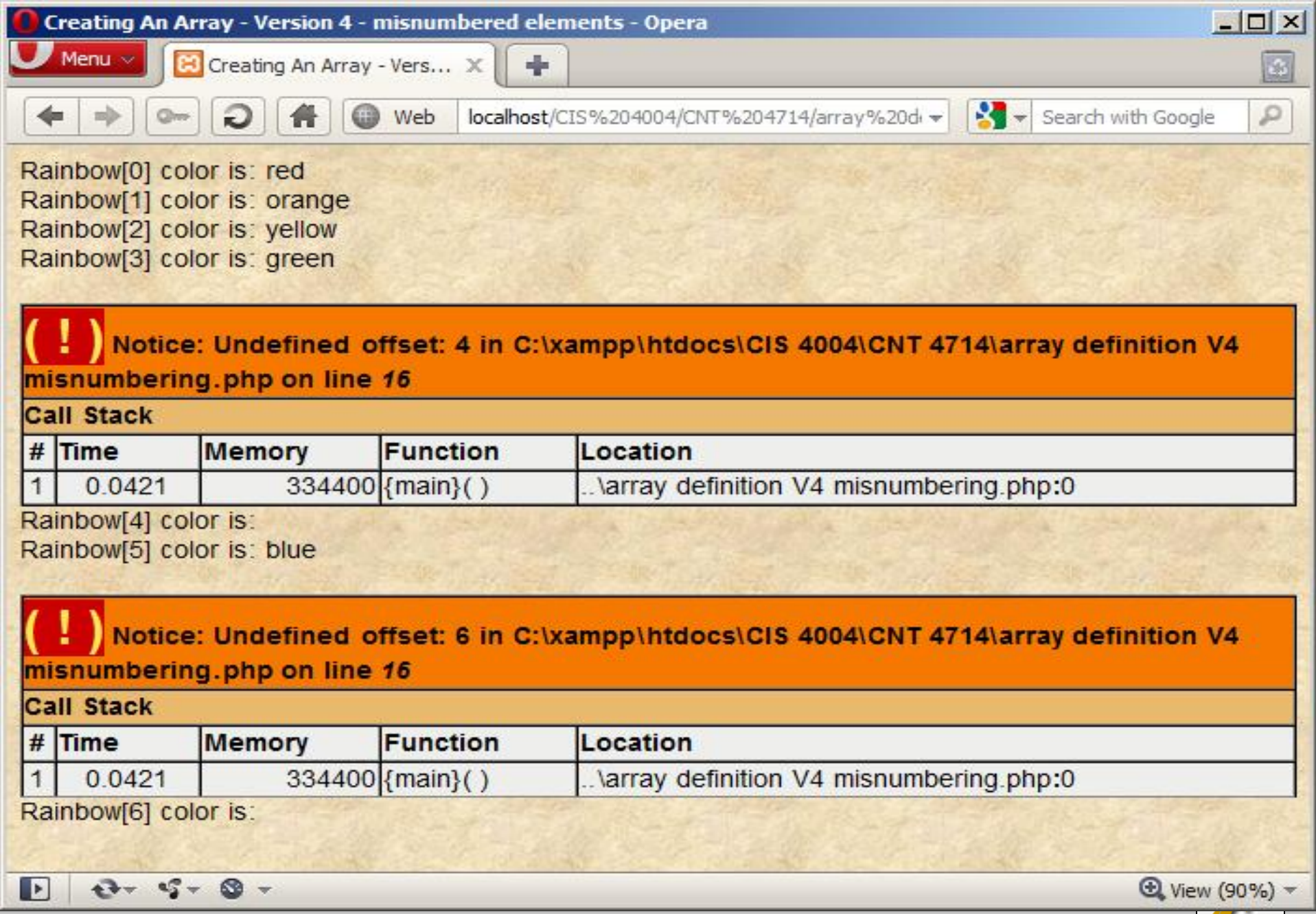
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ? X

ArrayScript.ps1 | array definition V3.php

```
1  <html>
2  <head>
3  <title>Creating An Array - Version 4 - misnumbered elements</title>
4  </head>
5  <body  style = "font-family: arial, sans-serif;
6        background-color: #856363"  background=image1.jpg>
7  <?php
8      $rainbow[0] = "red";
9      $rainbow[1] = "orange";
10     $rainbow[2] = "yellow";
11     $rainbow[3] = "green";
12     $rainbow[5] = "blue";
13     $rainbow[7] = "indigo";
14     $rainbow[8] = "violet";
15     for ($i=0; $i<=6; $i++) {
16         echo 'Rainbow['. $i.'] color is: '. $rainbow[$i]."<br />";
17     }
18 ?>
19 </body>
20 </html>
```

Misnumbering starts here with no element 4 defined and then 6 too is missed.

length : 505 line | Ln : 14 Col : 15 Sel : 0 | Dos\Windows | ANSI | INS


Creating An Array - Version 4 - misnumbered elements - Opera
Menu
Creating An Array - Vers...
Web
localhost/CIS%204004/CNT%204714/array%20d
Search with Google
Rainbow[0] color is: red
Rainbow[1] color is: orange
Rainbow[2] color is: yellow
Rainbow[3] color is: green
(!) Notice: Undefined offset: 4 in C:\xampp\htdocs\CIS 4004\CNT 4714\array definition V4 misnumbering.php on line 16
Call Stack
Time Memory Function Location
1 0.0421 334400 {main}( ) ..\array definition V4 misnumbering.php:0
Rainbow[4] color is:
Rainbow[5] color is: blue
(!) Notice: Undefined offset: 6 in C:\xampp\htdocs\CIS 4004\CNT 4714\array definition V4 misnumbering.php on line 16
Call Stack
Time Memory Function Location
1 0.0421 334400 {main}( ) ..\array definition V4 misnumbering.php:0
Rainbow[6] color is:
View (90%)

Creating Associative Arrays In PHP

- The arrays we've seen so far have been numerically indexed, meaning that they use an integer index position as the key.
- Associative arrays utilize actual named keys. In PHP, the named keys of an associative array are character strings rather than numerical values. The string value is used to look up or provide a cross-reference to the data value.
- The following example creates an associative array named `$instructor` with three elements.

```
$instructor["CIS 4004"] = "Llewellyn";

$instructor["CIS 3003"] = "Eisler";

$instructor["CIS 3360"] = "Guha";
```

Creating Associative Arrays In PHP

- The same array could also be created using the `array()` function instead of the array operator [ ]. This is shown below:

```
$instructor = array ("CIS 4004" => "Llewellyn",
"CIS 3003" => "Eisler", "CIS 3360" => "Guha");
```

- When using the `array()` function, items are assigned in index/value pairs using the => operator.

- When you want to access an item in an associative array, a syntax similar to that used with sequential (numerically indexed) arrays is employed, however, a string value or variable is used for the index.

Creating Associative Arrays In PHP

- Suppose you wanted to retrieve the instructor for CIS 4004. The following expression would achieve this:

```
$teacher = $instructor["CIS 4004"];
```

- The variable `$teacher` would be assigned the data value associated with "CIS 4004" which would be "Llewellyn".

Note: You might be tempted to do the following with an associative array, where you are trying to determine which course is taught by the instructor named "Llewellyn":

```
$course = $instructor["Llewellyn"];
```

Don't do this! An associative array can fetch data values only via the keys and not the values associated with the keys. Therefore, it cannot find and entry in the array with an index value of "Llewellyn" and will return nothing and the value of $course will be undefined. The example on the following page illustrates this.

C:\xampp\htdocs\CIS 4004\CNT 4714\associative array - incorrect version.php - Notepad++

File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?

ArrayScript.ps1 | array definition V4 misnumbering.php | associative array - incorrect version.php

```
1  <html>
2  <head>
3  <title>Creating An Associative Array - Incorrect Version</title>
4  </head>
5  <body  style = "font-family: arial, sans-serif;
6        background-color: #856363"  background=image1.jpg>
7  <?php
8      $instructor = array( "CIS 4004" => "Llewellyn",
9                           "CIS 3003" => "Eisler",
10                          "CIS 3360" => "Guha" );
11     echo 'The instructor of CIS 4004 is: '. $instructor["CIS 4004"]. "<br />";
12     echo 'The course taught by Eisler is: '. $instructor["Eisler"]. "<br />";
13
14 ?>
15 </body>
16
```

Creating An Associative Array - Incorrect Version - Opera

Menu | Creating An Associative ... | Web | localhost/CIS%204004/CNT%204714/associative | Search with Google

The instructor of CIS 4004 is: Llewellyn

(!) Notice: Undefined index: Eisler in C:\xampp\htdocs\CIS 4004\CNT 4714\associative array - incorrect version.php on line *12*

Call Stack

#	Time	Memory	Function	Location
1	0.0346	332432	{main}( )	..\associative array - incorrect version.php:0

The course taught by Eisler is:

View (90%)

Using Associative Arrays In PHP

- A common iterative statement used with both sequential and associative arrays is the `foreach` statement.
- The general syntax of the `foreach` statement is:

```
foreach ( arrayname as variable ) {

    . . . Statements to repeat

}
```

- The first variable inside the parentheses is the variable name representing the array and the second variable is automatically set to the next array item at each iteration of the loop. An example using a sequential array is shown on the next page and one with an associative array on the following page.

C:\xampp\htdocs\CIS 4004\CNT 4714\foreach statement.php - Notepad++

File Edit Search View Encoding Language Settings Macro Run TextFX Plugins

ArrayScript.ps1 | array definition V4 misnumbering.php | foreach statement.php

```
1  <html>
2  <head>
3  <title>Foreach statement</title>
4  </head>
5  <body  style = "font-family: arial, sans-serif;
6        background-color: #856363"  background=image1.jpg>
7  <?php
8      $languages = array( "PHP", "C", "Java", "Python", "Ruby", "Pascal", "Fortran");
9      echo "A list of computer languages <br /> <Br />";
10     foreach ($languages as $item) {
11         echo $item . "<br />";
12     }
13 ?>
14 </body>
15 </html>
```

Foreach statement - Op...

Menu

A list of computer languages

PHP
C
Java
Python
Ruby
Pascal
Fortran

View (90%)

length : 399 lines : 15 | Ln : 15 Col : 8 Sel : 0 | Dos\Windows | ANSI | INS

C:\xampp\htdocs\CIS 4004\CNT 4714\foreach statement - associative array.php - Notepad++

File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ? X

ArrayScript.ps1 | array definition V4 misnumbering.php | foreach statement - associative array.php

```
1  <html>
2  <head>
3  <title>Foreach statement with an associative array</title>
4  </head>
5  <body  style = "font-family: arial, sans-serif;
6        background-color: #856363"  background=image1.jpg>
7  <?php
8      $inventory = array( "hard drives" => 10, "printers" => 4, "monitors" => 23);
9      echo "Current Inventory <br /> <br />";
10     foreach ($inventory as $index => $item) {
11         echo $index . ' = ' . $item . "<br />";
12     }
13 ?>
14 </body>
15 </html>
```

length : 438 lines : 15 | Ln : 15 Col : 8 Sel : 0 | Dos\Wir

Foreach statement wit...

Menu

Current Inventory

hard drives = 10
printers = 4
monitors = 23

View (90%)

Using Associative Arrays In PHP

- Changing values, adding elements, deleting elements, and verifying an element are all among the common operations that you'll need to perform on an associative array.
- Changing an existing value is done through simple assignment. For example, to update the number of monitors in the previous example from 23 to 5, the following statement would be used: `$inventory["monitors"] = 5;`
- To add a new element to an associative array, use the array operator [] as in: `$inventory["keyboards"] = 12;`
- Deleting an element from an associative array is done using the `unset()` function.

C:\xampp\htdocs\CIS 4004\CNT 4714\associative array operations.php - Notepad++

File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ? X

ArrayScript.ps1 | array definition V4 misnumbering.php | associative array operations.php

```
5   <body  style = "font-family: arial, sans-serif;
6         background-color: #856363"  background=image1.jpg>
7   <?php
8       $inventory = array( "hard drives" => 10, "printers" => 4, "monitors" => 23);
9       echo "Current Inventory <br /> <br />";
10      foreach ($inventory as $index => $item) {
11          echo $index . ' = ' . $item . "<br />";
12      }
13      $inventory["monitors"] = 5;
14      $inventory["keyboards"] = 12;
15      echo "<br /><br /> Current Inventory <br /> <br />";
16      foreach ($inventory as $index => $item) {
17          echo $index . ' = ' . $item . "<br />";
18      }
19      unset($inventory["printers"]);
20      echo "<br /><br /> Current Inventory <br /> <br />";
21      foreach ($inventory as $index => $item) {
22          echo $index . ' = ' . $item . "<br />";
23      }
24  ?>
25  </body>
26  </html>
```

length : 827 lines : 26 | Ln : 24 Col : 3 Sel : 0 | Dos\Windows

Some operations o... — Menu

Current Inventory

hard drives = 10
printers = 4
monitors = 23

Current Inventory

hard drives = 10
printers = 4
monitors = 5
keyboards = 12

Current Inventory

hard drives = 10
monitors = 5
keyboards = 12

View (90%)

Using Associative Arrays In PHP

- To verify if a particular index exists in an associative array, use the `isset()` function.
- The `isset()` function returns true if index passed as an argument appears in the associative array and false otherwise.
- The example on the following page illustrates using the `isset()` function.

C:\xampp\htdocs\CIS 4004\CNT 4714\using the isset() function.php - Notepad++

File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ? X

ArrayScript.ps1 | array definition V4 misnumbering.php | using the isset() function.php

```
1  <html>
2  <head>
3  <title>Using the isset() function on an associative array</title>
4  </head>
5  <body  style = "font-family: arial, sans-serif;
6        background-color: #856363"  background=image1.jpg>
7  <?php
8      $inventory = array( "hard drives" => 10, "printers" => 4, "monitors" => 23);
9      echo "Current Inventory <br /> <br />";
10     foreach ($inventory as $index => $item) {
11         echo $index . ' = ' . $item . "<br />";
12     }
13     echo "<br />";
14     if ( isset($inventory["monitors"]) ) {
15         echo "monitors is in the array. <br />";
16     }else { echo "monitors is not in the array. <br />";
17     }
18     if ( isset($inventory["keyboards"]) ) {
19         echo "keyboards is in the array. <br />";
20     }else { echo "keyboards is not in the array. <br />";
21     }
22 ?>
23 </body>
```

Using the isset() function on ...

Menu | Usin...

Current Inventory

hard drives = 10
printers = 4
monitors = 23

monitors is in the array.
keyboards is not in the array.

View (90%)

length : 759 lines : 24 | Ln : 22 Col : 3 Sel : 0 | Dos\Windows | ANSI | INS