CHAPTER 6

MANIPULATING ARRAYS

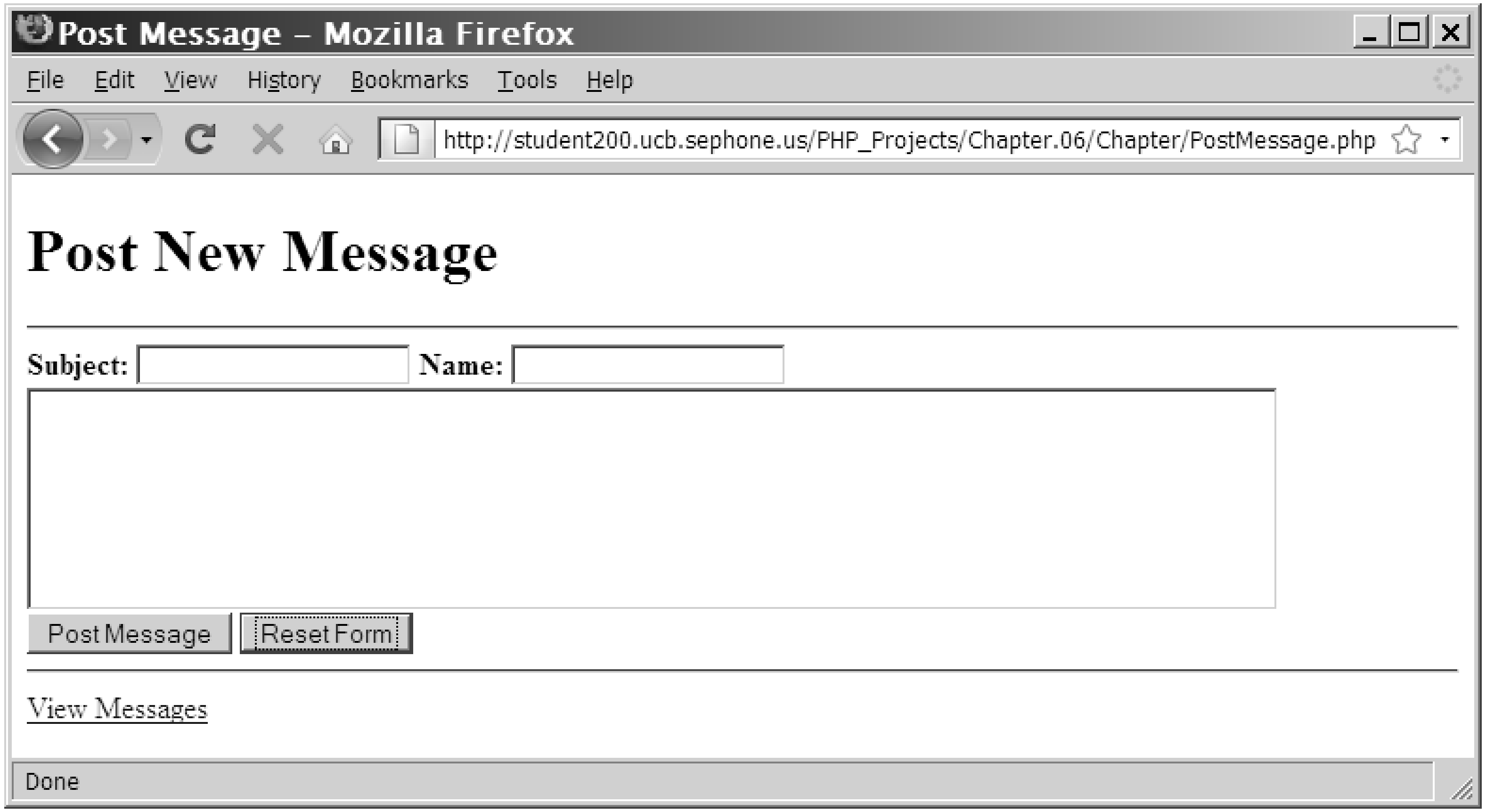
PHP PROGRAMMING WITH MYSQL
2ND EDITION

Objectives

In this chapter, you will:

- Manipulate array elements
- Declare and initialize associative arrays
- Iterate through an array
- Find and extract elements and values
- Sort, combine, and compare arrays
- Understand multidimensional arrays
- Use arrays in Web forms

Manipulating Elements (continued)


Post Message – Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://student200.ucb.sephone.us/PHP_Projects/Chapter.06/Chapter/PostMessage.php

Post New Message

Subject: Name:

Post Message Reset Form

View Messages

Figure 6-1 Post New Message page of the Message Board

Manipulating Elements (continued)

```
<h1>Post New Message</h1>
<hr />
<form action="PostMessage.php" method="POST">
<strong>Subject:</strong> <input type="text" name="subject" />
<strong>Name:</strong> <input type="text" name="name" /><br />
<textarea name="message" rows="6" cols="80"></textarea><br />
<input type="submit" name="submit" value="Post Message" />
<input type="reset" name="reset" value="Reset Form" />
</form>
<hr />
<a href="MessageBoard.php">View Messages</a>
```

Manipulating Elements

```
if (isset($_POST['submit'])) {
     $Subject = stripslashes($_POST['subject']);
     $Name = stripslashes($_POST['name']);
     $Message = stripslashes($_POST['message']);
     // Replace any '~' characters with '-' characters
     $Subject = str_replace("~", "-", $Subject);
     $Name = str_replace("~", "-", $Name);
     $Message = str_replace("~", "-", $Message);
     $MessageRecord = "$Subject~$Name~$Message\n";
     $MessageFile = fopen("MessageBoard/messages.txt", "ab");
     if ($MessageFile === FALSE)
          echo "There was an error saving your message!\n";
     else {
          fwrite($MessageFile, $MessageRecord);
          fclose($MessageFile);
          echo "Your message has been saved.\n";
     }
}
```

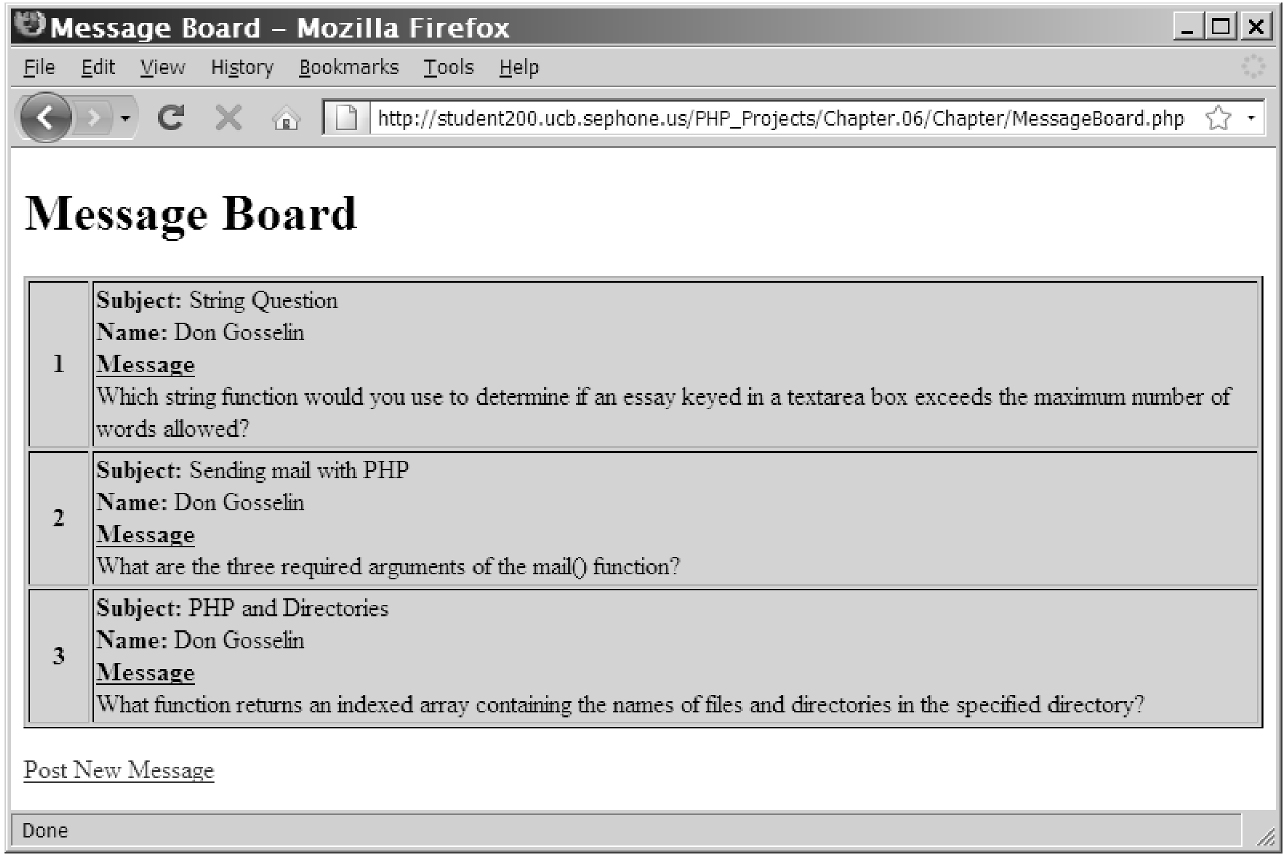
Manipulating Elements (continued)

```
<h1>Message Board</h1>
<?php
?>
<p>
<a href="PostMessage.php">Post New Message</a>
</p>
if ((!file_exists("MessageBoard/messages.txt")) ||
   (filesize("MessageBoard/messages.txt") == 0))
     echo "<p>There are no messages posted.</p>\n";
}
else {
     $MessageArray = file("MessageBoard/messages.txt");
     echo "<table style=\"background-color:lightgray\"
     border=\"1\" width=\"100%\">\n";
     $count = count($MessageArray);
```

Manipulating Elements (continued)

```
for ($i = 0; $i < $count; ++$i) {
       $CurrMsg = explode("~", $MessageArray[$i]);
       echo "      <tr>\n";
       echo "           <td width=\"5%\"
       align=\"center\"><strong>" . ($i + 1) .
      "</strong></td>\n";
       echo "           <td
       width=\"95%\"><strong>Subject:</strong> " .
       htmlentities($CurrMsg[0]) . "<br />";
       echo "<strong>Name:</strong> " .
       htmlentities($CurrMsg[1]) . "<br />";
       echo "<u><strong>Message</strong></u><br />" .
       htmlentities($CurrMsg[2]) . "</td>\n";
       echo "      </tr>\n";
 }
  echo "</table>\n";
```

Manipulating Elements (continued)


Message Board – Mozilla Firefox
File Edit View History Bookmarks Tools Help
Done

http://student200.ucb.sephone.us/PHP_Projects/Chapter.06/Chapter/MessageBoard.php

Message Board

1	**Subject:** String Question **Name:** Don Gosselin **Message** Which string function would you use to determine if an essay keyed in a textarea box exceeds the maximum number of words allowed?
2	**Subject:** Sending mail with PHP **Name:** Don Gosselin **Message** What are the three required arguments of the mail() function?
3	**Subject:** PHP and Directories **Name:** Don Gosselin **Message** What function returns an indexed array containing the names of files and directories in the specified directory?

Post New Message

Figure 6-2 Message Board page of the Message Board

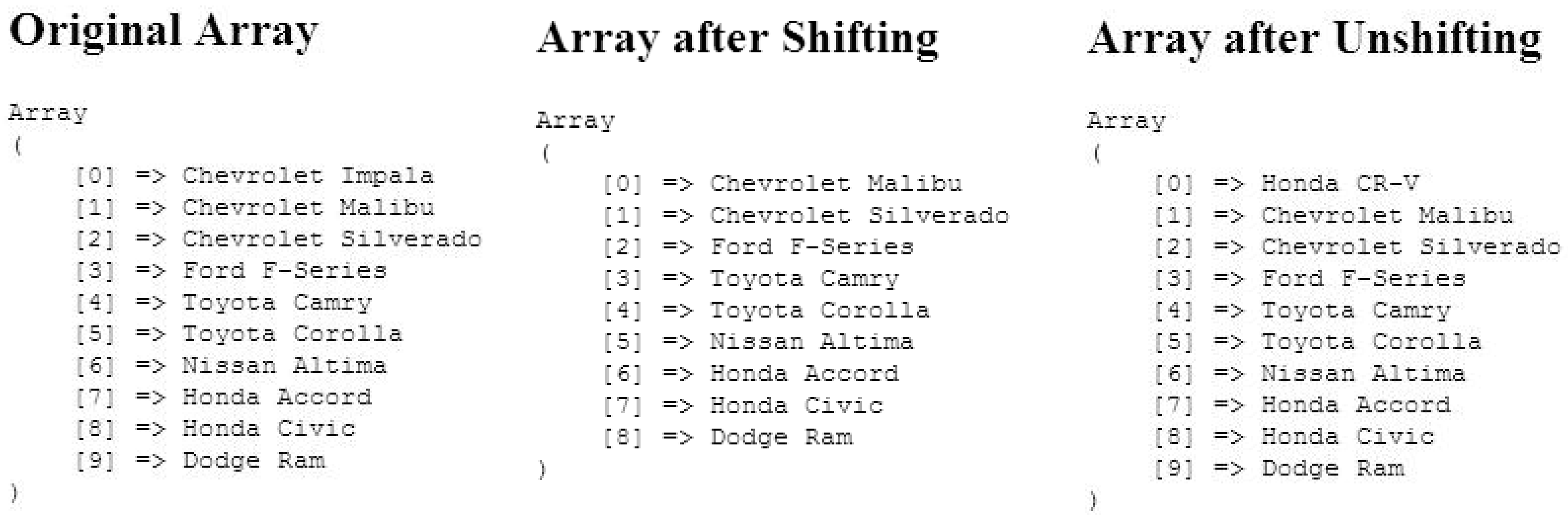
Adding and Removing Elements from the Beginning of an Array

- The `array_shift()` function removes the first element from the beginning of an array
  - Pass the name of the array whose first element you want to remove
- The `array_unshift()` function adds one or more elements to the beginning of an array
  - Pass the name of an array followed by comma-separated values for each element you want to add

Adding and Removing Elements from the Beginning of an Array (continued)

```
$TopSellers = array(
     "Chevrolet Impala",
     "Chevrolet Malibu",
     "Chevrolet Silverado",
     "Ford F-Series",
     "Toyota Camry",
     "Toyota Corolla",
     "Nissan Altima",
     "Honda Accord",
     "Honda Civic",
     "Dodge Ram");
array_shift($TopSellers);
array_unshift($TopSellers, "Honda CR-V");
echo "<pre>\n";
print_r($TopSellers);
echo "</pre>\n";
```

Adding and Removing Elements from the Beginning of an Array (continued)



Original Array

```
Array
(
    [0] => Chevrolet Impala
    [1] => Chevrolet Malibu
    [2] => Chevrolet Silverado
    [3] => Ford F-Series
    [4] => Toyota Camry
    [5] => Toyota Corolla
    [6] => Nissan Altima
    [7] => Honda Accord
    [8] => Honda Civic
    [9] => Dodge Ram
)
```

Array after Shifting

```
Array
(
    [0] => Chevrolet Malibu
    [1] => Chevrolet Silverado
    [2] => Ford F-Series
    [3] => Toyota Camry
    [4] => Toyota Corolla
    [5] => Nissan Altima
    [6] => Honda Accord
    [7] => Honda Civic
    [8] => Dodge Ram
)
```

Array after Unshifting

```
Array
(
    [0] => Honda CR-V
    [1] => Chevrolet Malibu
    [2] => Chevrolet Silverado
    [3] => Ford F-Series
    [4] => Toyota Camry
    [5] => Toyota Corolla
    [6] => Nissan Altima
    [7] => Honda Accord
    [8] => Honda Civic
    [9] => Dodge Ram
)
```

Figure 6-3 Output of an array modified with the `array_shift()` and `array_unshift()` functions

Adding and Removing Elements from the End of an Array

- The `array_pop()` function removes the last element from the end of an array
  - Pass the name of the array whose last element you want to remove
- The `array_push()` function adds one or more elements to the end of an array
  - Pass the name of an array followed by comma-separated values for each element you want to add

Adding and Removing Elements from the End of an Array (continued)

```
$HospitalDepts = array(
     "Anesthesia",
     "Molecular Biology",
     "Neurology",
     "Pediatrics");
array_pop($HospitalDepts); // Removes "Pediatrics"
array_push($HospitalDepts, "Psychiatry", "Pulmonary
  Diseases");
```

Adding and Removing Elements Within an Array

- The `array_splice()` function adds or removes array elements
- The `array_splice()` function renumbers the indexes in the array
- The syntax for the `array_splice()` function is:

```
array_splice(array_name, start,
  characters_to_delete, values_to_insert);
```

Adding and Removing Elements Within an Array (continued)

- To add an element within an array, include a value of 0 as the third argument of the `array_splice()` function

```
$HospitalDepts = array(
     "Anesthesia",            // first element (0)
     "Molecular Biology",     // second element (1)
     "Neurology",             // third element (2)
     "Pediatrics");           // fourth element (3)
array_splice($HospitalDepts, 3, 0, "Ophthalmology");
```

Adding and Removing Elements Within an Array (continued)

- To add more than one element within an array, pass the `array()` construct as the fourth argument of the `array_splice()` function
- Separate the new element values by commas

```
$HospitalDepts = array(
   "Anesthesia",              // first element (0)
   "Molecular Biology",       // second element (1)
   "Neurology",               // third element (2)
   "Pediatrics");             // fourth element (3)
array_splice($HospitalDepts, 3, 0, array("Opthalmology",
   "Otolaryngology"));
```

Adding and Removing Elements Within an Array (continued)

- Delete array elements by omitting the fourth argument from the `array_splice()` function

```
$HospitalDepts = array(
     "Anesthesia",           // first element (0)
     "Molecular Biology",    // second element (1)
     "Neurology",            // third element (2)
     "Pediatrics");          // fourth element (3)
array_splice($HospitalDepts, 1, 2);
```

Adding and Removing Elements Within an Array (continued)

- The `unset()` function removes array elements and other variables
- Pass to the `unset()` function the array name and index number of the element you want to remove
- To remove multiple elements, separate each index name and element number with commas

```
unset($HospitalDepts[1], $HospitalDepts[2]);
```

- Does not renumber the remaining elements in the array

Removing Duplicate Elements

- To renumber an indexed array's elements, you need to use the `array_values()` function
- The `array_values()` function does not operate directly on an array, instead it returns a new array with the renumbered indexes
- For this reason, you need to write a statement that assigns the array returned from the `array_values()` function to either a new variable or to the original variable

```
$HospitalDepts = array_values($HospitalDepts);
```

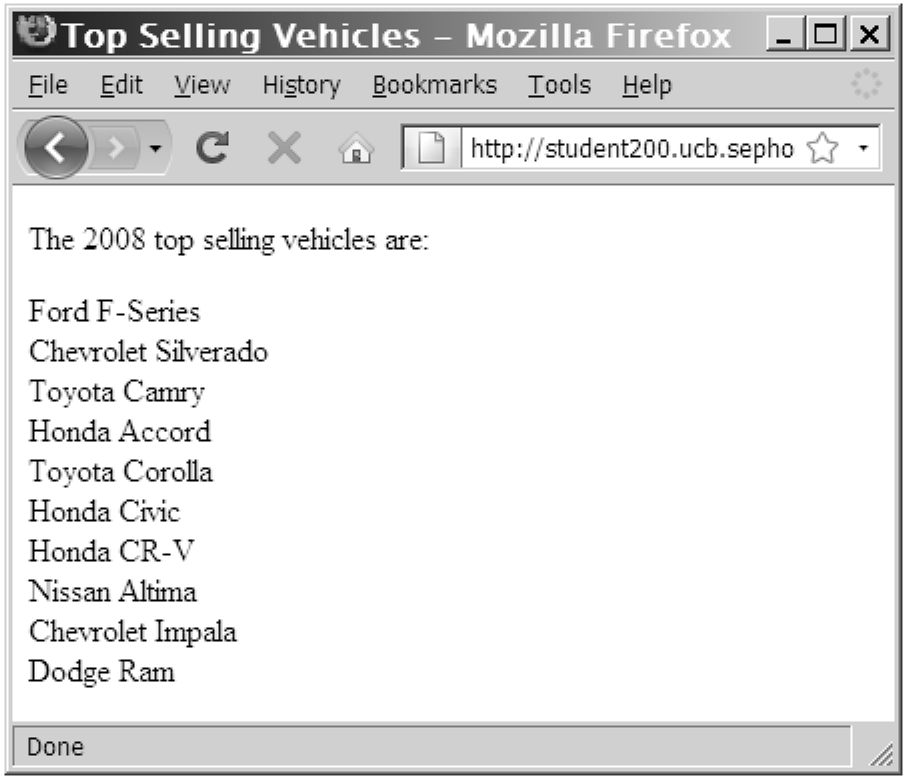
Removing Duplicate Elements

- The `array_unique()` function removes duplicate elements from an array
- Pass to the `array_unique()` function the name of the array from which you want to remove duplicate elements
- Like the `array_values()`, the `array_unique()` function does not operate directly on an array, therefore, you will need to write a statement that assigns the array returned

Removing Duplicate Elements (continued)

```
$TopSellers = array(
     "Ford F-Series", "Chevrolet  Silverado", "Toyota Camry",
     "Honda Accord", "Toyota Corolla", "Ford F-Series", "Honda
  Civic",
     "Honda CR-V", "Honda Accord", "Nissan Altima", "Toyota
  Camry",
     "Chevrolet Impala", "Dodge Ram", "Honda CR-V");
echo "<p>The 2008 top selling vehicles are:</p><p>";
$TopSellers = array_unique($TopSellers);
$TopSellers = array_values($TopSellers);
for ($i=0; $i<count($ TopSellers); ++$i) {
     echo "{$TopSellers[$i]}<br />";
}
echo "</p>";
```

Removing Duplicate Elements (continued)


Top Selling Vehicles – Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://student200.ucb.sepho
Done

```
The 2008 top selling vehicles are:

Ford F-Series
Chevrolet Silverado
Toyota Camry
Honda Accord
Toyota Corolla
Honda Civic
Honda CR-V
Nissan Altima
Chevrolet Impala
Dodge Ram
```

Figure 6-4 Output of an array after removing duplicate values with the `array_unique()` function

Short Quiz, p. 316

1. What two functions are used to add or remove elements from the beginning of an array?
2. Briefly describe the `array_pop()` and `array_push()` functions.
3. What function is used to add a new element at any position in an array?
4. Explain the process of using the `array_splice()` function to delete an array element?
5. What function must be used in conjunction with the `array_unique()` function to renumber the indexes after the duplicates have been removed?

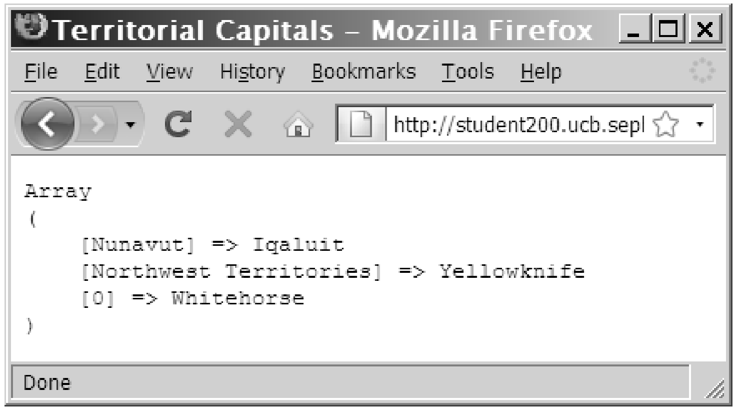
Declaring and Initializing Associative Arrays

- With associative arrays, you specify an element's key by using the array operator (=>)
  - The syntax for declaring and initializing an associative array is:

```
$array_name = array(key=>value, ...);
```

Declaring and Initializing Associative Arrays (continued)

```
$TerritorialCapitals["Nunavut"] = "Iqaluit";
$TerritorialCapitals["Northwest Territories"] = "Yellowknife";
$TerritorialCapitals[] = "Whitehorse";
echo "<pre>\n";
print_r($TerritorialCapitals);
echo "</pre>\n";
```

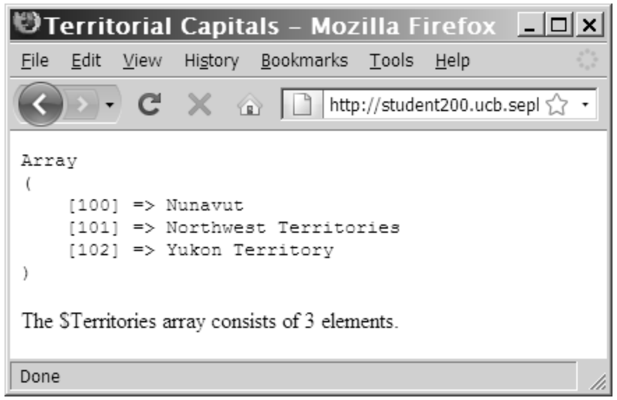

Territorial Capitals – Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://student200.ucb.sepl
Done

```
Array
(
    [Nunavut] => Iqaluit
    [Northwest Territories] => Yellowknife
    [0] => Whitehorse
)
```

Figure 6-5 Output of an array associative and indexed elements

Declaring and Initializing Associative Arrays (continued)

```
$Territories[100] = "Nunavut";
$Territories[] = "Northwest Territories";
$Territories[] = "Yukon Territory";
echo "<pre>\n";
print_r($Territories);
echo "</pre>\n";
echo '<p>The $Territories array consists of ',
     count($Territories), " elements.</p>\n";
```


Territorial Capitals – Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://student200.ucb.sepl
Done

```
Array
(
    [100] => Nunavut
    [101] => Northwest Territories
    [102] => Yukon Territory
)
```

The $Territories array consists of 3 elements.

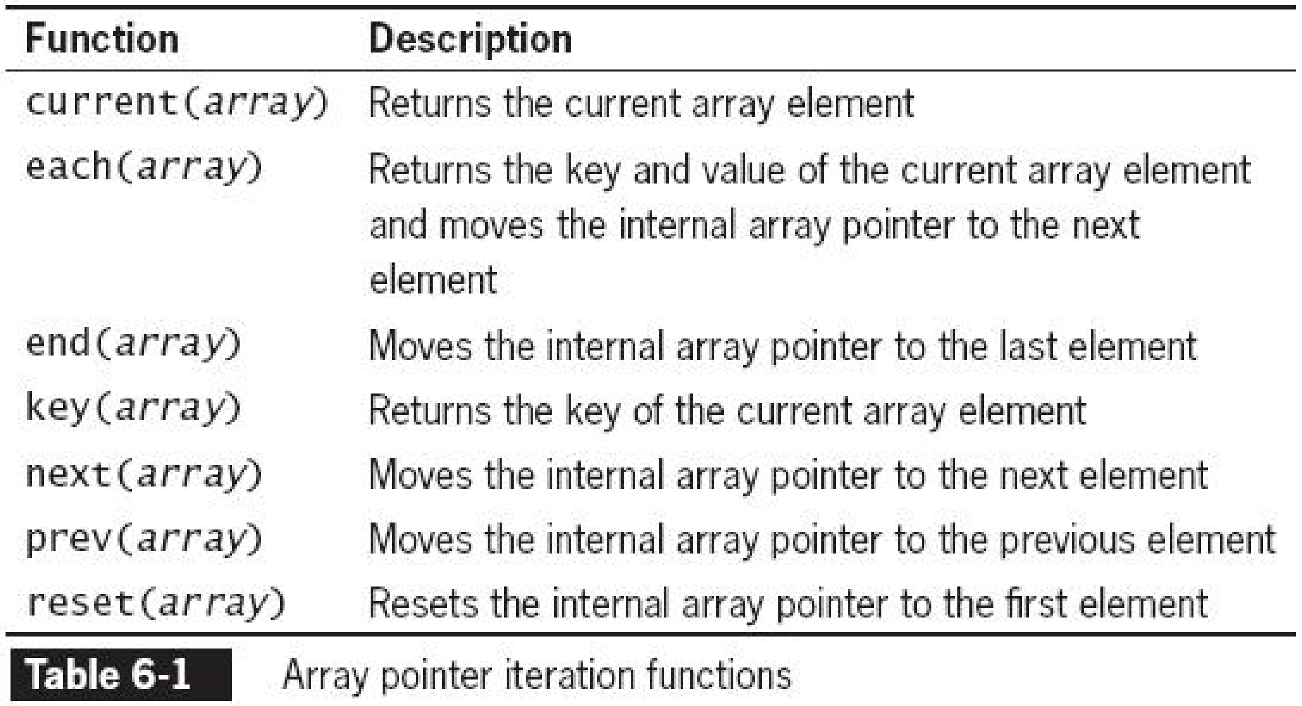
Figure 6-6 Output of an array with a starting index of 100

Short Quiz, p. 322-323

1. Describe the difference in assigning a key with an indexed array versus an associate array.
2. Explain what happens if you do not assign a key to an associative array.
3. What operator is used to define associative array keys within the `array()` construct.
4. What function is used to determine the number of elements in an associative array?

Iterating Through an Array

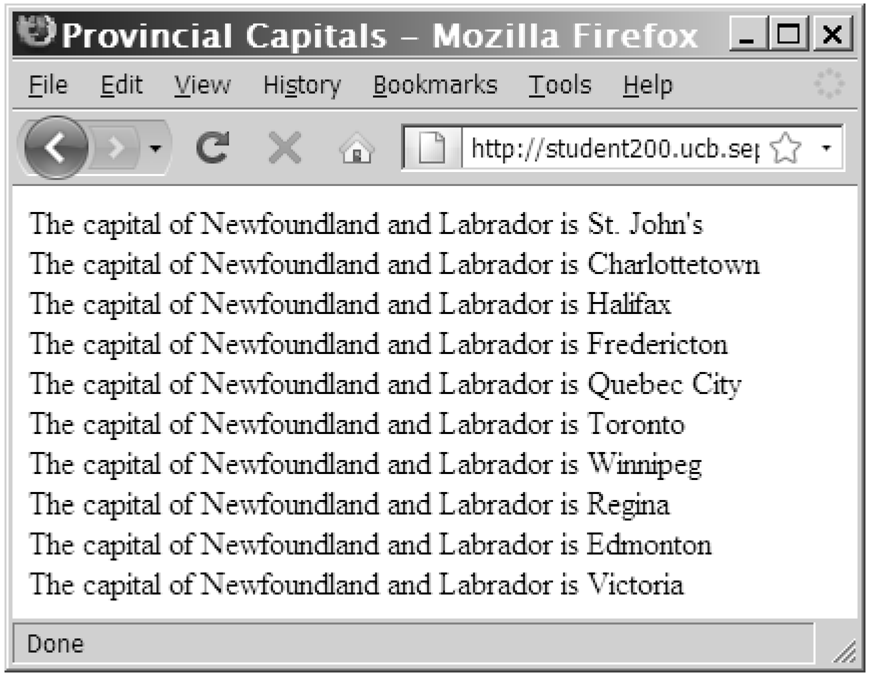
- The **internal array pointer** refers to the currently selected element in an array



Function	Description
`current(`*array*`)`	Returns the current array element
`each(`*array*`)`	Returns the key and value of the current array element and moves the internal array pointer to the next element
`end(`*array*`)`	Moves the internal array pointer to the last element
`key(`*array*`)`	Returns the key of the current array element
`next(`*array*`)`	Moves the internal array pointer to the next element
`prev(`*array*`)`	Moves the internal array pointer to the previous element
`reset(`*array*`)`	Resets the internal array pointer to the first element

Table 6-1 Array pointer iteration functions

Iterating Through an Array (continued)


Provincial Capitals - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://student200.ucb.se
Done

```
The capital of Newfoundland and Labrador is St. John's
The capital of Newfoundland and Labrador is Charlottetown
The capital of Newfoundland and Labrador is Halifax
The capital of Newfoundland and Labrador is Fredericton
The capital of Newfoundland and Labrador is Quebec City
The capital of Newfoundland and Labrador is Toronto
The capital of Newfoundland and Labrador is Winnipeg
The capital of Newfoundland and Labrador is Regina
The capital of Newfoundland and Labrador is Edmonton
The capital of Newfoundland and Labrador is Victoria
```

Figure 6-8 Output of an array <u>without</u> advancing the internal array pointer

Short Quiz, p. 327-328

1. Describe the purpose of the internal array pointer.
2. Explain why you might need to use an internal array pointer when working with associative arrays.
3. What is the purpose of the `key()` function?
4. When using a `foreach` statement to iterate though the elements of any array, what function must be used to move to the next element in the array?
5. What two functions are used to move an internal array pointer to the beginning or end of an array?

Finding and Extracting Elements and Values

- One of the most basic methods for finding a value in an array is to use a looping statement to iterate through the array until you find the value – example code p. 328.
- Rather than write custom code to find a value, use the `in_array()` and `array_search()` functions to determine whether a value exists in an array

Determining if a Value Exists

- The `in_array()` function returns a Boolean value of true if a given value exists in an array
- The `array_search()` function determines whether a given value exists in an array and:
  - Returns the index or key of the first matching element if the value exists, or
  - Returns `FALSE` if the value does not exist

```
if (in_array("Neurology", $HospitalDepts))
   echo "<p>The hospital has a Neurology
department.</p>";
```

Determining if a Key Exists

- The `array_key_exists()` function determines whether a given index or key exists
- You pass two arguments to the `array_key_exists()` function:
  - The first argument represents the key to search for
  - The second argument represents the name of the array in which to search

Determining if a Key Exists (continued)

```
$ScreenNames["Dancer"] = "Daryl";
$ScreenNames["Fat Man"] = "Dennis";
$ScreenNames["Assassin"] = "Jennifer";
if (array_key_exists("Fat Man", $ScreenNames))
     echo "<p>{$ScreenNames['Fat Man']} is already
     'Fat Man'.</p>\n";
else {
     $ScreenNames["Fat Man"] = "Don";
     echo "<p>{$ScreenNames['Fat Man']} is now
     'Fat Man'.</p>";
}
```

Determining if a Key Exists

- The `array_keys()` function returns an indexed array that contains all the keys in an associative array

Determining if a Key Exists (continued)

```
$ScreenNames["Dancer"] = "Daryl";
$ScreenNames["Fat Man"] = "Dennis";
$ScreenNames["Assassin"] = "Jennifer";
$UsedScreenNames = array_keys($ScreenNames);
echo "<p>The following screen names are already
  assigned:</p>\n";
for ($i = 0; $i < count($UsedScreenNames); ++i) {
  echo "<p>{$UsedScreenNames[$i]}</p>\n";


}
```

Returning a Portion of an Array

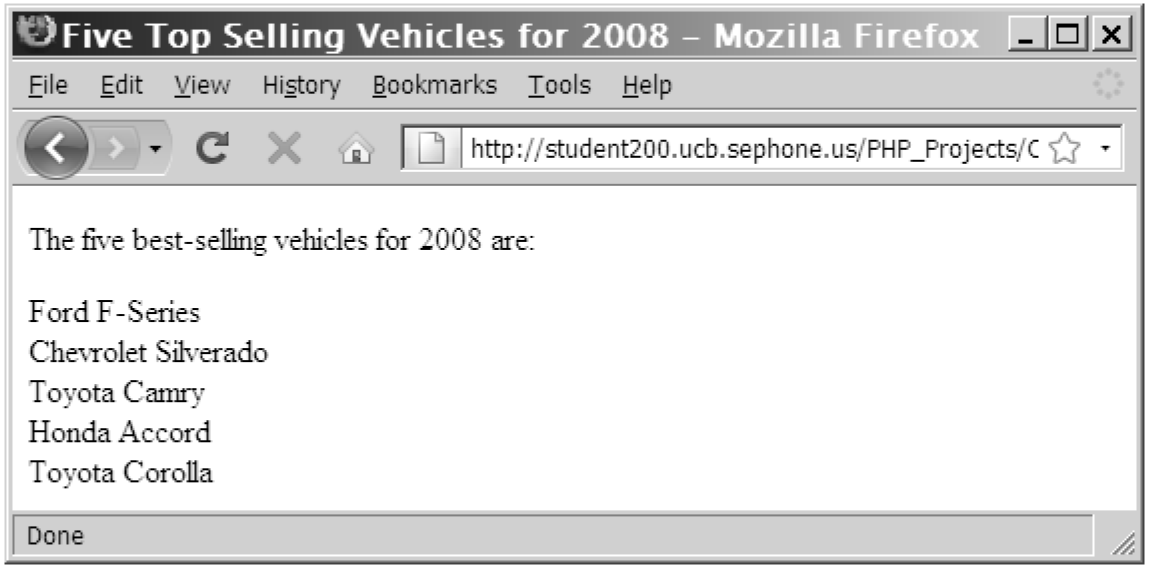
- The `array_slice()` function returns a portion of an array and assigns it to another array
- The syntax for the `array_slice()` function is:

```
array_slice(array_name, start, characters_to_return);
```

Returning a Portion of an Array (continued)

```
// This array is ordered by sales, high to low.
$TopSellers = array("Ford F-Series", "Chevrolet Silverado",
  "Toyota Camry", "Honda Accord", "Toyota Corolla", "Honda
  Civic", "Nissan Altima", "Chevrolet Impala", "Dodge Ram",
  "Honda CR-V");
$FiveTopSellers = array_slice($TopSellers, 0, 5);
echo "<p>The five best-selling vehicles for 2008
  are:</p>\n";
for ($i=0; $i<count($FiveTopSellers); ++$i) {
    echo "{$FiveTopSellers[$i]}<br />\n";
}
```

Returning a Portion of an Array (continued)


Five Top Selling Vehicles for 2008 – Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://student200.ucb.sephone.us/PHP_Projects/C
Done

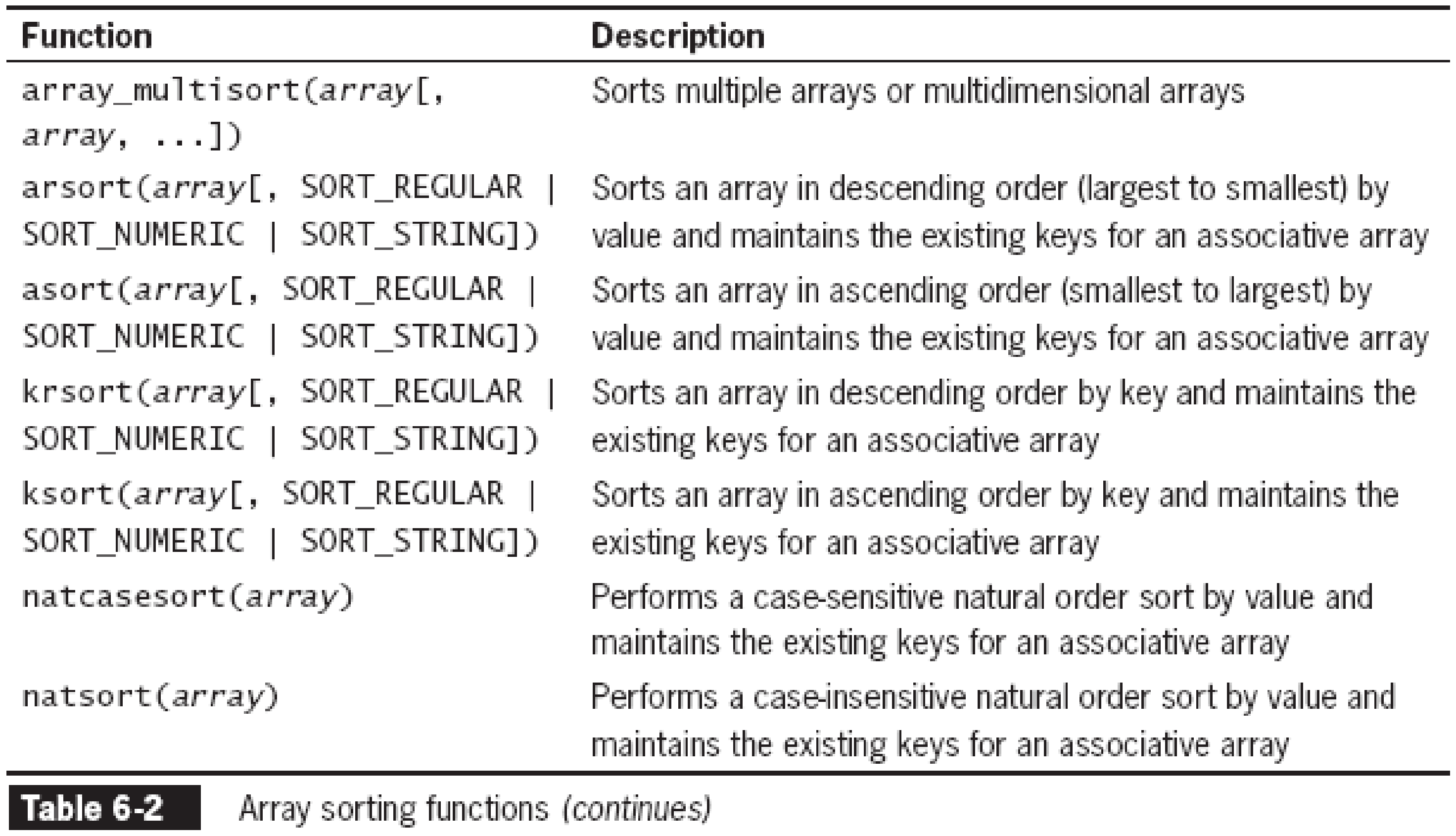
```
The five best-selling vehicles for 2008 are:

Ford F-Series
Chevrolet Silverado
Toyota Camry
Honda Accord
Toyota Corolla
```

Figure 6-11 Output of an array returned with the `array_slice()` function

Short Quiz, p. 336

1. Differentiate between the value returned by the `in_array()` function and the `array_search()` function.
2. What function can be used to return an indexed array of all keys in an associate array?
3. What does the `array_key_exists()` function do?
4. What function is used to return a portion of an array and assign it to another array?

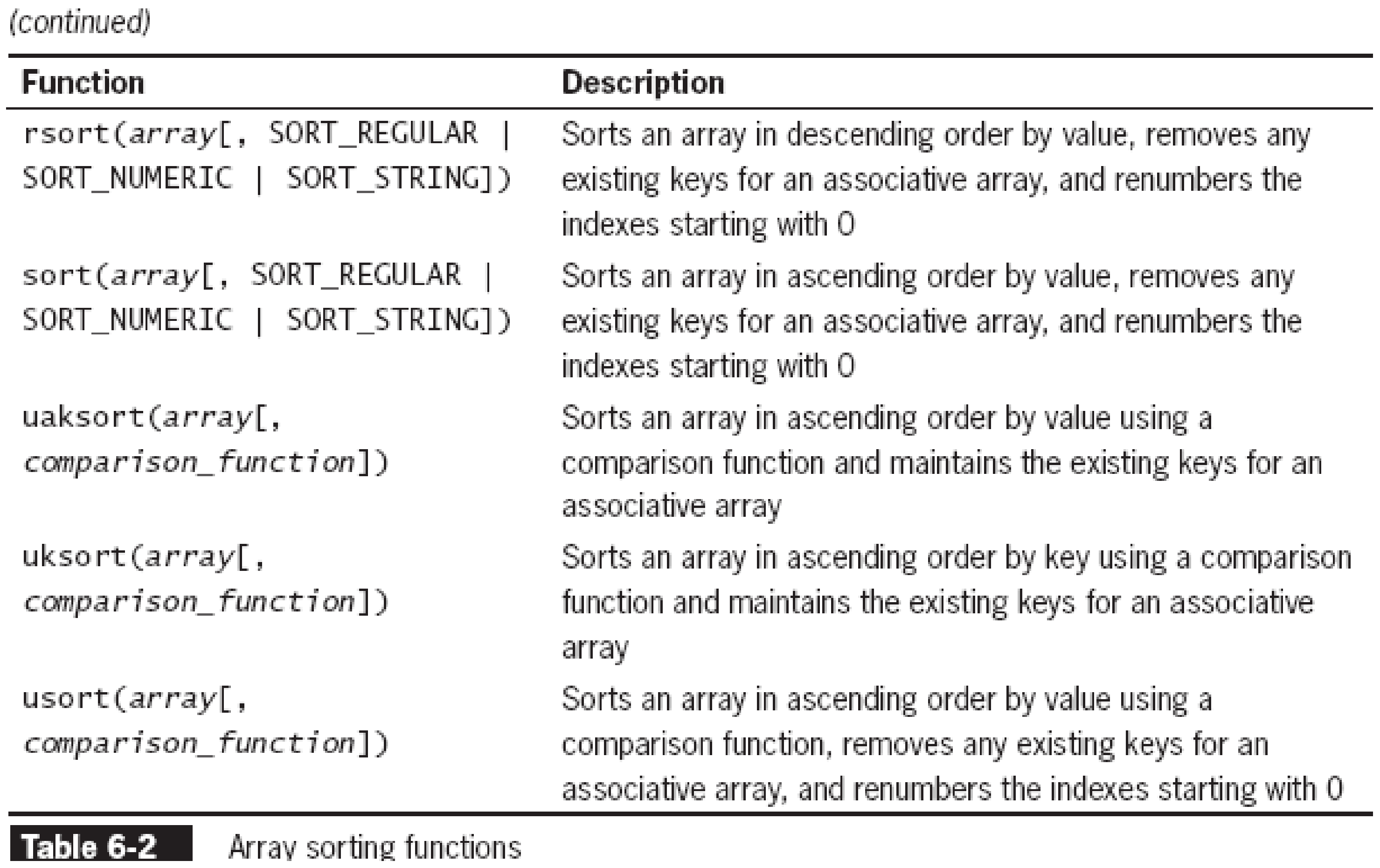
Sorting Arrays (continued)



Function	Description
`array_multisort(array[, array, ...])`	Sorts multiple arrays or multidimensional arrays
`arsort(array[, SORT_REGULAR | SORT_NUMERIC | SORT_STRING])`	Sorts an array in descending order (largest to smallest) by value and maintains the existing keys for an associative array
`asort(array[, SORT_REGULAR | SORT_NUMERIC | SORT_STRING])`	Sorts an array in ascending order (smallest to largest) by value and maintains the existing keys for an associative array
`krsort(array[, SORT_REGULAR | SORT_NUMERIC | SORT_STRING])`	Sorts an array in descending order by key and maintains the existing keys for an associative array
`ksort(array[, SORT_REGULAR | SORT_NUMERIC | SORT_STRING])`	Sorts an array in ascending order by key and maintains the existing keys for an associative array
`natcasesort(array)`	Performs a case-sensitive natural order sort by value and maintains the existing keys for an associative array
`natsort(array)`	Performs a case-insensitive natural order sort by value and maintains the existing keys for an associative array

Table 6-2 Array sorting functions *(continues)*

Sorting Arrays (continued)



(continued)

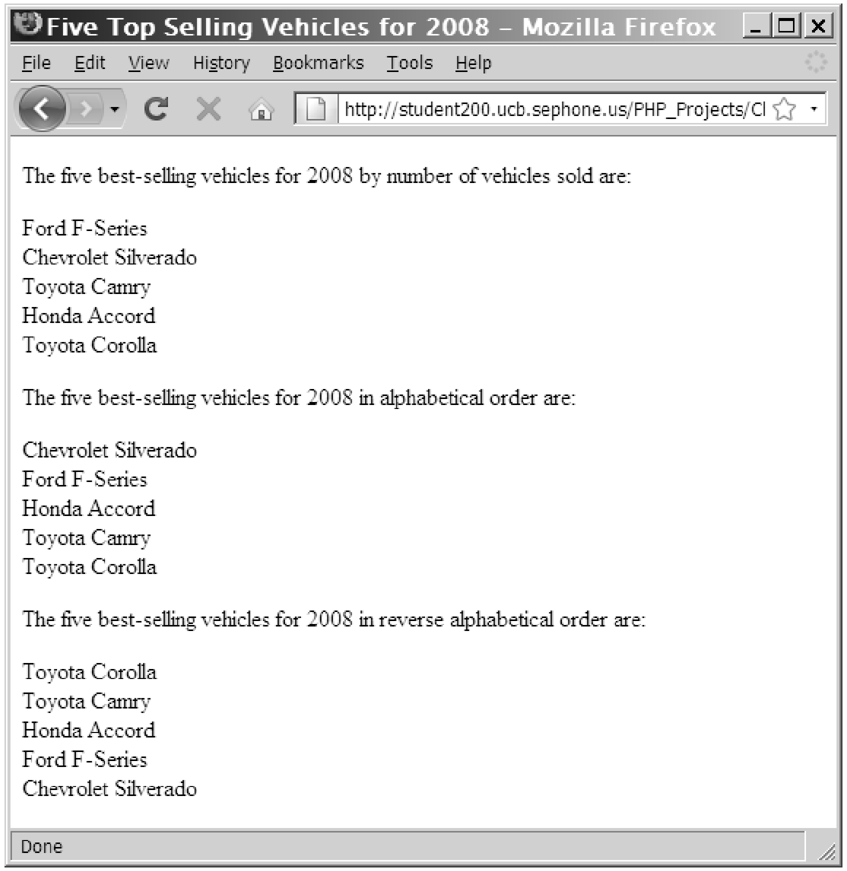
Function	Description
`rsort(array[, SORT_REGULAR | SORT_NUMERIC | SORT_STRING])`	Sorts an array in descending order by value, removes any existing keys for an associative array, and renumbers the indexes starting with 0
`sort(array[, SORT_REGULAR | SORT_NUMERIC | SORT_STRING])`	Sorts an array in ascending order by value, removes any existing keys for an associative array, and renumbers the indexes starting with 0
`uaksort(array[, comparison_function])`	Sorts an array in ascending order by value using a comparison function and maintains the existing keys for an associative array
`uksort(array[, comparison_function])`	Sorts an array in ascending order by key using a comparison function and maintains the existing keys for an associative array
`usort(array[, comparison_function])`	Sorts an array in ascending order by value using a comparison function, removes any existing keys for an associative array, and renumbers the indexes starting with 0

Table 6-2 Array sorting functions

Sorting Arrays

- The most commonly used array sorting functions are:
  - `sort()` and `rsort()` for indexed arrays
  - `asort()`, `arsort()`, `ksort()` and `krsort()` for associative arrays
  - These functions operate directly on an array, not on a new copy of an array, as occurs with the `array_values()` function
- The two "natural order" sort functions, `natsort()` and `natcasesort()`, use a special sorting algorithm

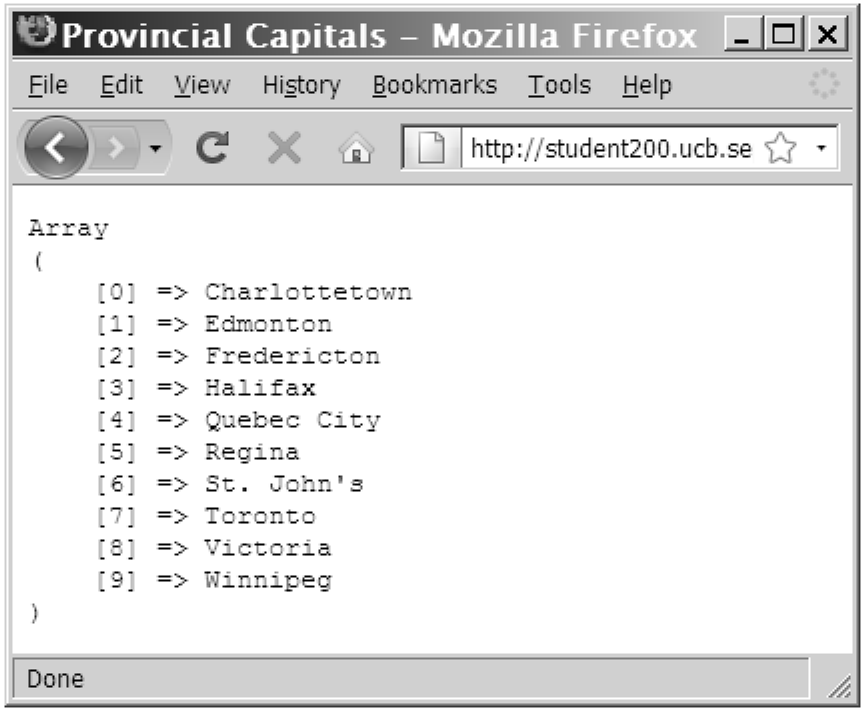
Sorting Arrays (continued)


Five Top Selling Vehicles for 2008 – Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://student200.ucb.sephone.us/PHP_Projects/Cl
Done

```
The five best-selling vehicles for 2008 by number of vehicles sold are:

Ford F-Series
Chevrolet Silverado
Toyota Camry
Honda Accord
Toyota Corolla

The five best-selling vehicles for 2008 in alphabetical order are:

Chevrolet Silverado
Ford F-Series
Honda Accord
Toyota Camry
Toyota Corolla

The five best-selling vehicles for 2008 in reverse alphabetical order are:

Toyota Corolla
Toyota Camry
Honda Accord
Ford F-Series
Chevrolet Silverado
```

Figure 6-12 Output of an array after applying the `sort()` and `rsort()` functions

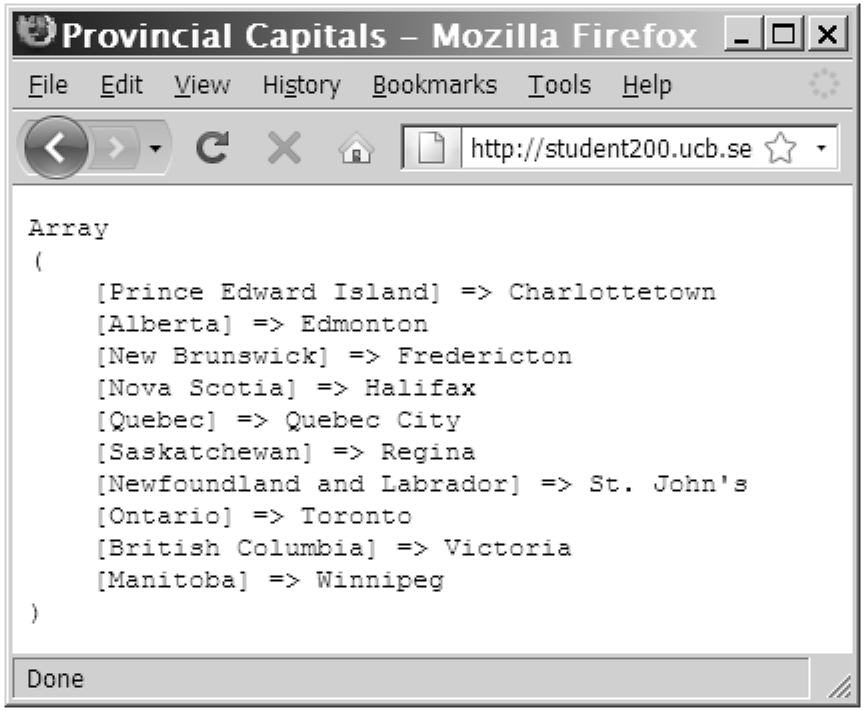
Sorting Arrays (continued)


Provincial Capitals – Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://student200.ucb.se
Done

```
Array
(
    [0] => Charlottetown
    [1] => Edmonton
    [2] => Fredericton
    [3] => Halifax
    [4] => Quebec City
    [5] => Regina
    [6] => St. John's
    [7] => Toronto
    [8] => Victoria
    [9] => Winnipeg
)
```

Figure 6-13 Output of an associative array after sorting with the `sort()` function

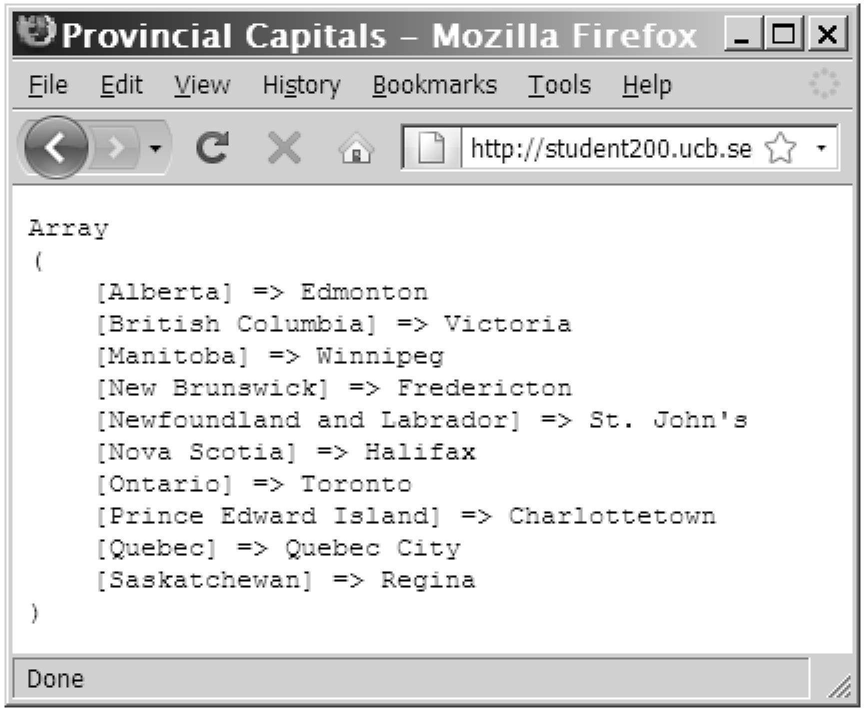
Sorting Arrays (continued)


Provincial Capitals – Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://student200.ucb.se
Done

```
Array
(
    [Prince Edward Island] => Charlottetown
    [Alberta] => Edmonton
    [New Brunswick] => Fredericton
    [Nova Scotia] => Halifax
    [Quebec] => Quebec City
    [Saskatchewan] => Regina
    [Newfoundland and Labrador] => St. John's
    [Ontario] => Toronto
    [British Columbia] => Victoria
    [Manitoba] => Winnipeg
)
```

Figure 6-14 Output of an associative array after sorting with the `asort()` function

Sorting Arrays (continued)


Provincial Capitals – Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://student200.ucb.se
Done

```
Array
(
    [Alberta] => Edmonton
    [British Columbia] => Victoria
    [Manitoba] => Winnipeg
    [New Brunswick] => Fredericton
    [Newfoundland and Labrador] => St. John's
    [Nova Scotia] => Halifax
    [Ontario] => Toronto
    [Prince Edward Island] => Charlottetown
    [Quebec] => Quebec City
    [Saskatchewan] => Regina
)
```

Figure 6-15 Output of an associative array after sorting with the `ksort()` function

Combining Arrays

- 2 options:
  1. To **append** one array to another, use the addition (+) or the compound assignment operator (+=)
     - Order is important when appending one array to another array
     - The array on the left side of the operator is the **primary array**, or the array PHP starts with. The array on the right side of the operator is the **secondary array**, or the array being appended to the primary array
     - Examples, p. 344-346

Combining Arrays

- 2 options (*cont.*):
  2. To **merge** two or more arrays use the `array_merge()` function
     - The syntax for the `array_merge()` function is:

       ```
       new_array = array_merge($array1, $array2,
       $array3, ...);
       ```

     - Examples, p. 346

Combining Arrays

- The `array_combine ()` function creates a new associative array that uses the values from one array as keys and element values from another array
- The syntax for the `array_combine()` function is:
  `new_array = array_combine($array1, $array2,);`
- Examples, p. 347

Comparing Arrays

- The `array_diff()` function returns an array of elements that exist in one array but not in any other arrays to which it is compared
- The syntax for the `array_diff()` function is:

```
new_array = array_diff($array1, $array2,
$array3, ...);
```

Comparing Arrays (continued)

- The `array_intersect()` function returns an array of elements that exist in all of the arrays that are compared
- The syntax for the `array_intersect()` function is:

```
new_array = array_intersect($array1,
$array2, $array3, ...);
```

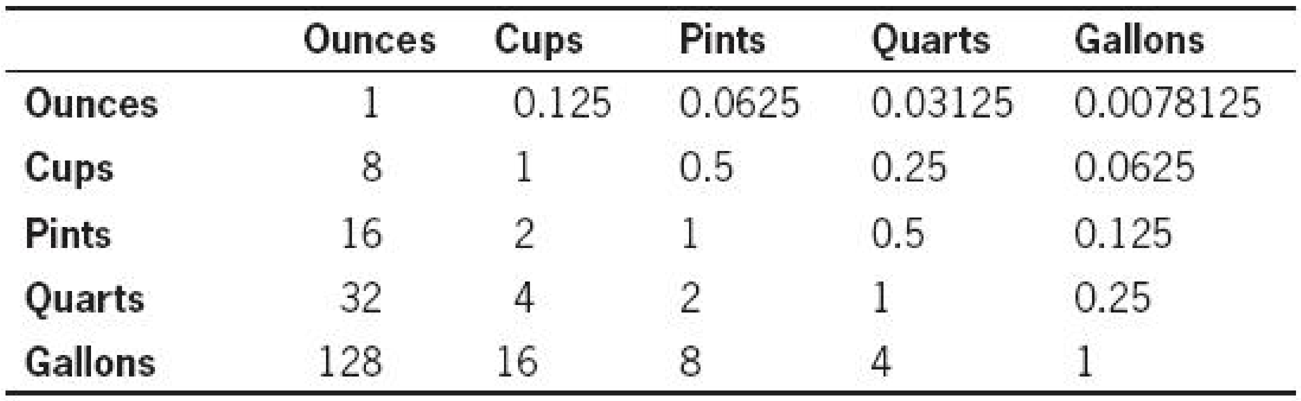
Short Quiz, p. 350

1. Explain the difference between the `sort()` and `asort()` functions.
2. What is the purpose of the `ksort()` and `krsort()` functions?
3. What are the two methods of combining arrays?
4. Explain the difference between the `array_diff()` and `array_intersect()` functions.

Creating Two-Dimensional Indexed Arrays

- A **multidimensional array** consists of multiple indexes or keys
- A **two-dimensional** array has two sets of indexes or keys

Creating Two-Dimensional Indexed Arrays (continued)



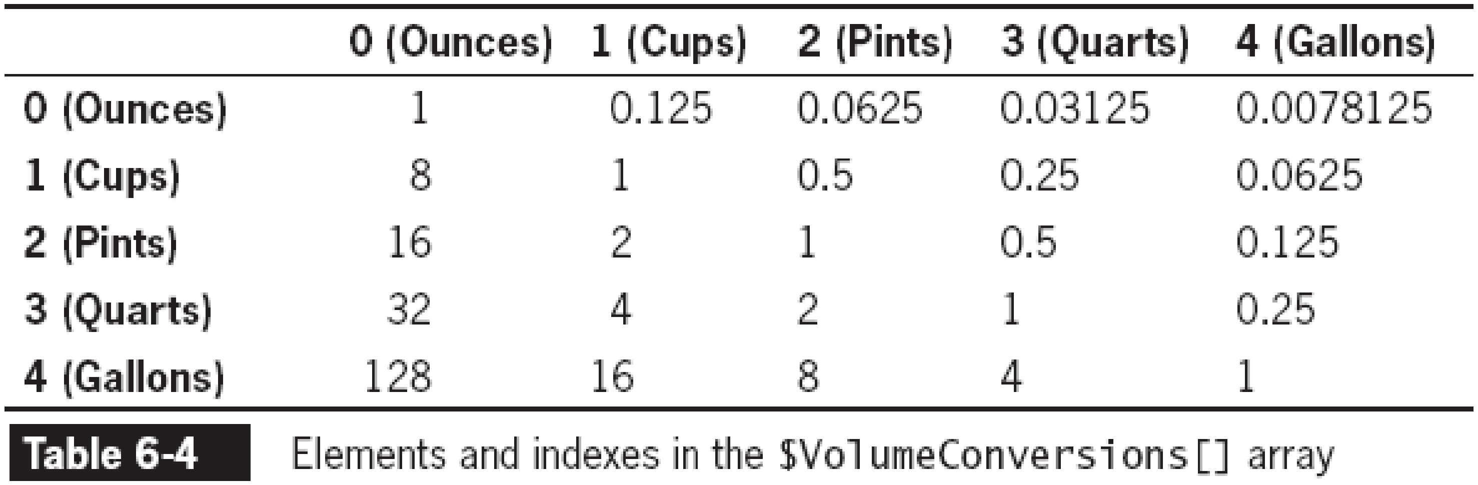
	Ounces	Cups	Pints	Quarts	Gallons
Ounces	1	0.125	0.0625	0.03125	0.0078125
Cups	8	1	0.5	0.25	0.0625
Pints	16	2	1	0.5	0.125
Quarts	32	4	2	1	0.25
Gallons	128	16	8	4	1

Table 6-3 Volume conversion table

```
$Ounces = array(1, 0.125, 0.0625, 0.03125, 0.0078125);
$Cups = array(8, 1, 0.5, 0.25, 0.0625);
$Pints = array(16, 2, 1, 0.5, 0.125);
$Quarts = array(32, 4, 2, 1, 0.25);
$Gallons = array(128, 16, 8, 4, 1);
```

Creating Two-Dimensional Indexed Arrays (continued)

```
$VolumeConversions = array($Ounces, $Cups,
  $Pints, $Quarts, $Gallons);
```



	0 (Ounces)	1 (Cups)	2 (Pints)	3 (Quarts)	4 (Gallons)
0 (Ounces)	1	0.125	0.0625	0.03125	0.0078125
1 (Cups)	8	1	0.5	0.25	0.0625
2 (Pints)	16	2	1	0.5	0.125
3 (Quarts)	32	4	2	1	0.25
4 (Gallons)	128	16	8	4	1

Table 6-4 Elements and indexes in the `$VolumeConversions[]` array

Creating Two-Dimensional Indexed Arrays (continued)

- You refer to the values in a multidimensional indexed array by including two sets of brackets following the array name with the syntax
  `$array_name[index][index]`
  the first set of brackets refers to the row and the second set of brackets refers to the column

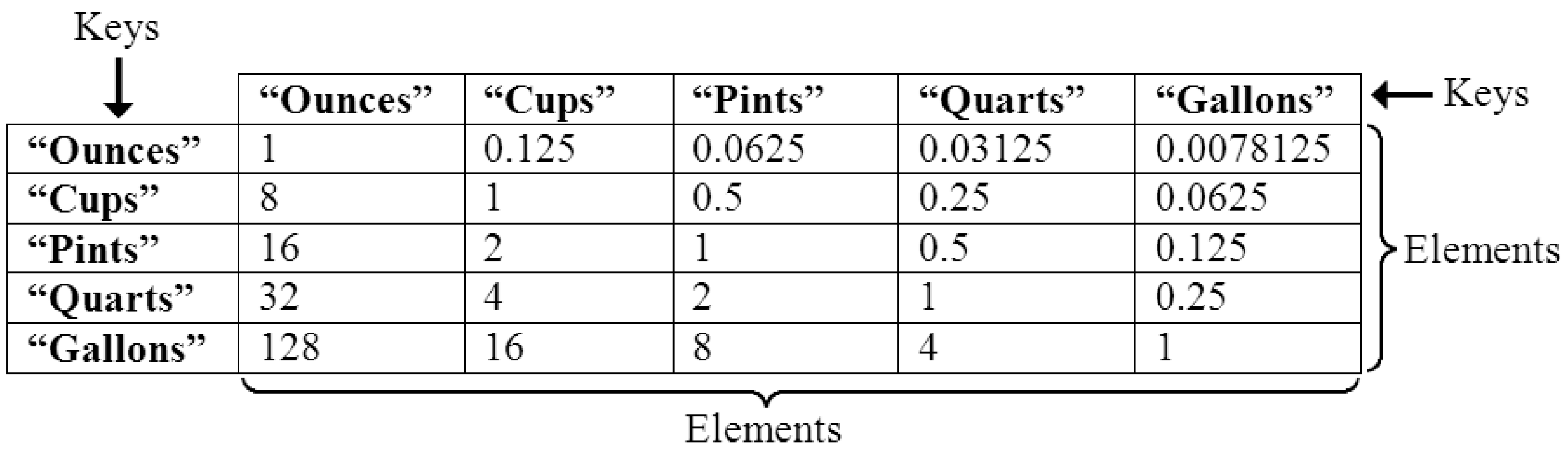
```
$VolumeConversions[3][1]; //quarts to cups
```

Creating Two-Dimensional Associative Arrays

```
$Ounces = array("ounces" => 1, "cups" => 0.125, "pints" =>
   0.0625, "quarts" => 0.03125, "gallons" => 0.0078125);
$Cups = array("ounces" => 8, "cups" => 1, "pints" =>0.5,
   "quarts" => 0.25, "gallons" => 0.0625);
$Pints = array("ounces" => 16, "cups" => 2, "pints" =>1,
   "quarts" => 0.5, "gallons" => 0.125);
$Quarts = array("ounces" => 32, "cups" => 4, "pints" =>2,
   "quarts" => 1, "gallons" => 0.25);
$Gallons = array("ounces" => 128, "cups" => 16, "pints"
   =>8, "quarts" => 4, "gallons" => 1);
```

Creating Two-Dimensional Associative Arrays (continued)

```
$VolumeConversions = array("ounces" => $Ounces,
   "cups" => $Cups, "pints" => $Pints,
   "quarts" => $Quarts, "gallons" => $Gallons);
```


Keys
Keys
Elements
Elements

	"Ounces"	**"Cups"**	**"Pints"**	**"Quarts"**	**"Gallons"**
"Ounces"	1	0.125	0.0625	0.03125	0.0078125
"Cups"	8	1	0.5	0.25	0.0625
"Pints"	16	2	1	0.5	0.125
"Quarts"	32	4	2	1	0.25
"Gallons"	128	16	8	4	1

Figure 6-21 Elements and keys in the `$VolumeConversions[ ]` array

Creating Multidimensional Arrays with a Single Statement

- You can also create a multidimensional array with a single statement
- Instead of writing separate declaration statements, you can include the array construct for each individual array as the value for each element within the declaration statement for the multidimensional array
- Examples, p. 357

Short Quiz, p. 359

1. What is the difference between a one-dimensional array and a multidimensional array?
2. What is the most common type of multidimensional array?
3. In a two-dimensional array, the first set of indexes can be thought of as ____, and the second set of indexes can be thought of as _____.
4. What is the primary difference between creating two-dimensional associate arrays and two-dimensional indexed arrays?
5. Explain how to create a two-dimensional array in a single statement.

Using Arrays in Web Forms

- Store form data in an array by appending an opening and closing ([ ]) to the value of the name attribute
- Data from any element with the same value for the *name* attribute will be appended to an array with that name

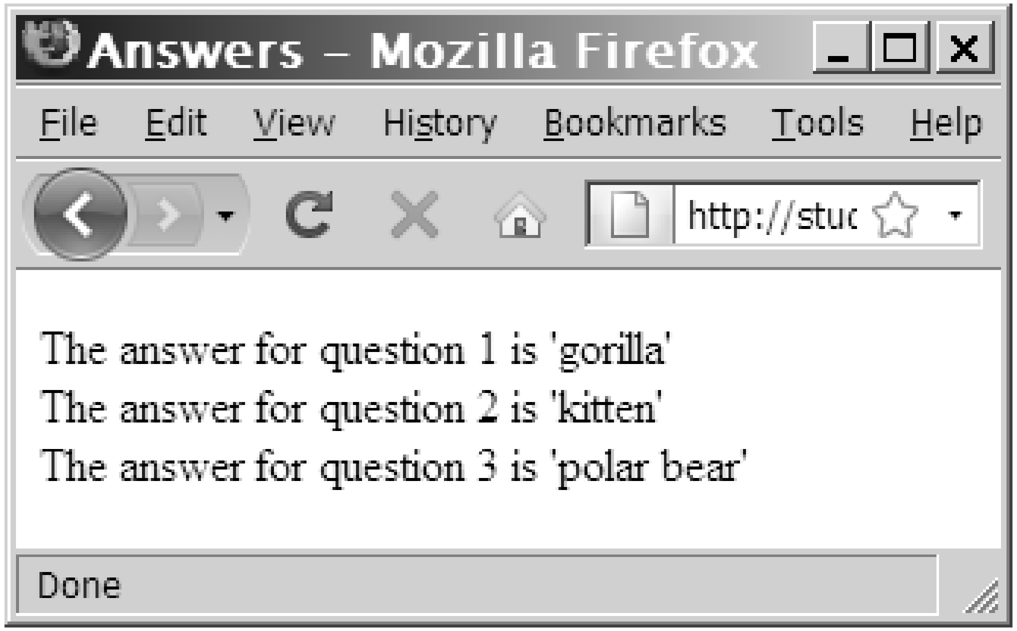
Using Arrays in Web Forms (continued)

```
<form method='post' action='ProcessForm.php'>
<p>Enter the first answer:
<input type='text' name='answers[]' /></p>
<p>Enter the second answer:
<input type='text' name='answers[]' /></p>
<p>Enter the third answer:
<input type='text' name='answers[]' /></p>
<input type='submit' name='submit' value='submit' />
</form>
```

Using Arrays in Web Forms (continued)

```
if (is_array($_POST['answers'])) {
     $Index = 0;
     foreach ($_POST['answers'] as $Answer) {
          ++$Index;
          echo "The answer for question $Index is
  '$Answer'<br />\n";
     }
}
```

Using Arrays in Web Forms (continued)


Answers – Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://stuc
Done

```
The answer for question 1 is 'gorilla'
The answer for question 2 is 'kitten'
The answer for question 3 is 'polar bear'
```

Figure 6-22 Output of an array posted from a Web form

Using Multidimensional Array Notation

- Multidimensional array notation can also be used to process posted form information

```
if (is_array($_POST['answers'])) {
     $count = count($_POST['answers']);
     for ($i=0; $i<$count; ++$i) {
          echo "The answer for question " . ($i+1) .
  " is '{$_POST['answers'][$i]}'<br />\n";
     }
}
```

Creating an Associative Forms Array

```
<form method='post' action='ProcessForm.php'>
<p>Enter the first answer:
<input type='text' name='answers[Question 1]' /></p>
<p>Enter the second answer:
<input type='text' name='answers[Question 2]' /></p>
<p>Enter the third answer:
<input type='text' name='answers[Question 3]' /></p>
<input type='submit' name='submit' value='submit' />
</form>
```

Short Quiz, p. 364

1. What attribute in the Web form `<input>` tag must be changed for the value to be sent as an array element?
2. Can arrays crated from Web forms be indexed arrays, associate array, or both? Explain.
3. Should quotation marks be used in the associative array key name for a Web form? Why or why not?

Summary

- The `array_shift()` function removes the first element from the beginning of an array
- The `array_unshift()` function adds one or more elements to the beginning of an array
- The `array_pop()` function removes the last element from the end of an array
- The `array_push()` function adds one or more elements to the end of an array
- The `array_splice()` function adds or removes array elements

Summary (continued)

- The `unset()` function removes array elements and other variables
- The `array_values()` function renumbers an indexed array's elements
- The `array_unique()` function removes duplicate elements from an array
- The `in_array()` function returns a Boolean value of `TRUE` if a given value exists in an array
- The `array_search()` function determines whether a given value exists in an array

Summary (continued)

- The `array_key_exists()` function determines whether a given index or key exists
- The `array_slice()` function returns a portion of an array and assigns it to another array
- The `array_merge()` function merges two or more arrays
- The `array_diff()` function returns an array of elements that exist in one array but not in any other arrays to which it is compared

Summary (continued)

- The `array_intersect()` function returns an array of elements that exist in all of the arrays that are compared
- A **multidimensional array** consists of multiple sets of indexes or keys
- A **two-dimensional array** has two sets of indexes or keys
- When array notation is used in the name of a Web form input, the value gets stored in a nested array within the `$_POST` or `$_GET` array

Summary (continued)

- When using associative array notation in a Web form, you omit the quotation marks around the key name