Chapter 11

How to create and use arrays

Objectives

Applied

1. Use any of the functions and techniques presented in this chapter as you use arrays, associative arrays, and arrays of arrays.

Objectives (continued)

Knowledge

1. Distinguish between an array and an associative array, including the difference in the way indexes are used.
2. Explain how gaps can be introduced into an array and how the gaps can be removed.
3. Describe the use of a for loop with an array and the use of a foreach loop with an associative array.
4. Distinguish between a queue and a stack.
5. Describe the use of the functions for creating arrays, working with queues and stacks, performing mathematical calculations, searching arrays, sorting arrays, and modifying arrays.
6. Distinguish between a regular array and an array of arrays, including the difference in the number of indexes that are used to access an element.

The syntax for creating an array

```
$array_name = array([value1[, value2, ... ]])
```

The syntax for referring to an element an array

```
$array_name[index];
```

How to create an array of names

With one statement

```
$names = array('Ted Lewis', 'Sue Jones', 'Ray Thomas');
```

With multiple statements

```
$names = array();         // create an empty array
$names[0] = 'Ted Lewis';  // set 3 values in the array
$names[1] = 'Sue Jones';
$names[2] = 'Ray Thomas';
```

How to create an array of discounts

With one statement

```
$discounts = array(0, 5, 10, 15);
```

With multiple statements

```
$discounts = array(); // create an empty array
$discounts[0] = 0;    // set 4 values in the array
$discounts[1] = 5;
$discounts[2] = 10;
$discounts[3] = 15;
```

The syntax for adding an element to an array

```
$array_name[] = $value;
```

How to add a value to the end of an array

```
$letters = array('a', 'b', 'c', 'd');  // a, b, c, d
$letters[] = 'e';                      // a, b, c, d, e
```

How to set a value at a specific index

```
$letters = array('a', 'b', 'c', 'd');  // a, b, c, d
$letters[0] = 'e';                     // e, b, c, d
$letters[3] = 'f';                     // e, b, c, f
$letters[5] = 'g';                // e, b, c, f, NULL, g
```

How to get values from an array

```
$letters = array('a', 'b', 'c', 'd'); // a, b, c, d
$letter1 = $letters[0];               // $letter1 is 'a'
$letter2 = $letters[1];               // $letter2 is 'b'
$letter4 = $letters[4];               // $letter4 is NULL
```

Functions for removing the values from elements

```
unset($var1[, $var2 ...])
array_values($array)
```

How to delete values from an array

```
$letters = array('a', 'b', 'c', 'd'); // a, b, c, d
unset($letters[2]);                   // a, b, NULL, d
unset($letters);                      // $letters is NULL
```

How to remove NULL elements and reindex

```
$letters = array('a', 'b', 'c', 'd'); // a, b, c, d
unset($letters[2]);                   // a, b, NULL, d
$letters = array_values($letters);    // a, b, d
```

How to use variable substitution with elements

```
$name = array ('Ray', 'Harris');
echo "First Name: $name[0]";     // First Name: Ray
echo "First Name: {$name[0]}";   // First Name: Ray
```

Functions for loops that work with arrays

```
count($array) //The number of elements in an array not counting gaps.

end($array) //Move pointer to the last element

key($array) //The index of the element that the pointer is on

isset($var)
```

Code that stores 10 random numbers in an array

```
$numbers = array();
for ($i = 0; $i < 10; $i++) {
    $numbers[] = mt_rand(1, 100);
}
```

Code that displays the elements of an array

```
$numbers_string = '';
for ($i = 0; $i < count($numbers); $i++) {
    $numbers_string .= $numbers[$i] . ' ';
}
echo $numbers_string;
```

How to skip gaps in an array

```
$numbers = array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
unset($numbers[2], $numbers[6]);
end($numbers);
$last = key($numbers);
$numbers_string = '';
for($i = 0; $i <= $last; $i++) {
    if (isset($numbers[$i])) {
        $numbers_string .= $numbers[$i] . ' ';
    }
}
echo $numbers_string;       // Displays: 1 2 4 5 6 8 9 10
```

An associative array is one that uses strings for its index values. These are often called keys.

The syntax for creating an associative array

```
array([key1 => value1, key2 => value2, ... ])
```

How to create an associative array of tax rates

With one statement

```
$tax_rates =
    array('NC' => 7.75, 'CA' => 8.25, 'NY' => 8.875);
```

With multiple statements

```
$tax_rates = array();
$tax_rates['NC'] = 7.75;
$tax_rates['CA'] = 8.25;
$tax_rates['NY'] = 8.875;
```

How to create an associative array of codes

With one statement

```
$country_codes =
    array('DEU' => 'Germany',
          'JPN' => 'Japan',
          'ARG' => 'Argentina',
          'USA' => 'United States');
```

With multiple statements

```
$country_codes = array();
$country_codes['DEU'] = 'Germany';
$country_codes['JPN'] = 'Japan';
$country_codes['ARG'] = 'Argentina';
$country_codes['USA'] = 'United States';
```

How to create an associative array of extensions

```
$ext = array();
$ext[10] = 'Sales';
$ext[13] = 'Customer Service';
$ext[16] = 'Returns';
$ext[18] = 'Warehouse';
```

An array that contains integer and string indexes

```
$employees = array();
$employees[0] = 'Mike';
$employees[1] = 'Anne';
$employees[2] = 'Judy';
$employees['senior'] = 'Mike';
$employees['newest'] = 'Pren';
```

How to set a value with a specific key

```
$name = array('first' => 'Ray', 'last' => 'Harris');
$name['middle'] = 'Thomas';
```

What happens when you omit the key when adding a value

```
$name = array('first' => 'Ray', 'last' => 'Harris');
$name[] = 'Thomas';                // key is 0
```

How to get a value at a specified key

```
$name = array('first' => 'Ray', 'last' => 'Harris');
$first_name = $name['first'];
$last_name = $name['last'];
```

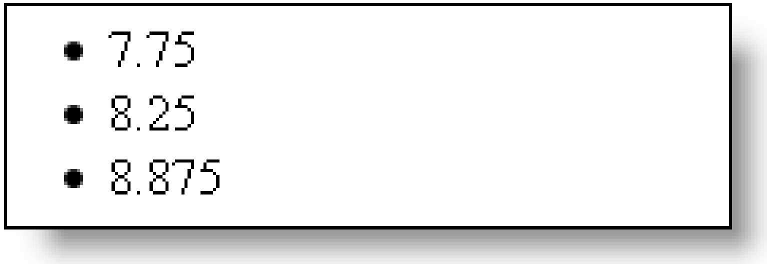
The syntax of a foreach loop

```
foreach ($array_name as [ $key => ] $value) {
    // Statements that use $key and $value
}
```

A loop that displays the values in an array

```
$tax_rates = array('NC' => 7.75, 'CA' => 8.25,
                   'NY' => 8.875);
echo '<ul>';
foreach ($tax_rates as $rate) {
    echo "<li>$rate</li>";
}
echo '</ul>';
```

The result displayed in a browser

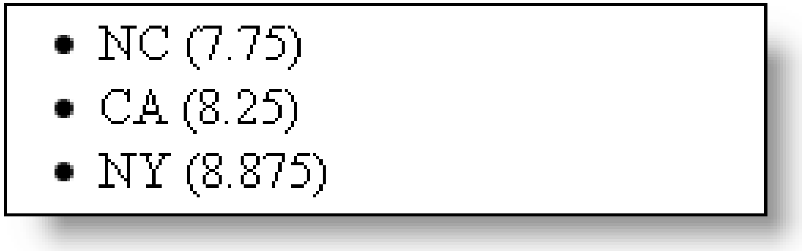


```
• 7.75
• 8.25
• 8.875
```

A foreach loop that displays the keys and values

```
$tax_rates = array('NC' => 7.75, 'CA' => 8.25,
                   'NY' => 8.875);
echo '<ul>';
foreach ($tax_rates as $state => $rate) {
    echo "<li>$state ($rate)</li>";
}
echo '</ul>';
```

The result displayed in a browser



- NC (7.75)
- CA (8.25)
- NY (8.875)

A foreach loop that displays the values in a regular array

```
$numbers = array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
unset($numbers[2], $numbers[6]);
$numbers_string = '';
foreach($numbers as $number) {
    $numbers_string .= $number . ' ';
}
echo $numbers_string;      // Displays: 1 2 4 5 6 8 9 10
```

How to use the range function to create an array

```
$numbers = range(1, 4);              // 1, 2, 3, 4
$numbers = range(10, 22, 4);         // 10, 14, 18, 22
```

How to use the array_fill and array_pad functions

```
$numbers = array_fill(0, 5, 1);      // 1, 1, 1, 1, 1
$numbers = array_pad($numbers, 10, 0);
// 1, 1, 1, 1, 1, 0, 0, 0, 0, 0
```

How to use the array_merge function

```
$employees = array('Mike', 'Anne');
$new_hires = array('Ray', 'Pren');
$employees = array_merge($employees, $new_hires);
echo implode(', ', $employees);
// Mike, Anne, Ray, Pren
```

How to use the array_sum function

```
$prices = array(141.95, 212.95, 411, 10.95);
$sum = array_sum($prices);                // 776.85
```

How to use functions to search an array

```
$tax_rates = array('NC' => 7.75,
                   'CA' => 8.25, 'NY' => 8.875);
$is_found = in_array(7.75, $tax_rates);       // TRUE
$is_found = in_array('7.75', $tax_rates);     // TRUE
$is_found = in_array('7.75', $tax_rates, true);
$key_exists = array_key_exists('CA', $tax_rates);
$key = array_search(7.75, $tax_rates);        // 'NC'
```

How to count the number of occurrences of a value in an array

```
$names = array('Mike', 'Mike', 'Mike',
               'Anne', 'Joel', 'Joel');
$occurences = array_count_values($names );
echo $occurences['Mike'];            // 3
echo $occurences['Anne'];            // 1
echo $occurences['Joel'];            // 2
```

How to sort strings in ascending order

```
$names = array('Mike', 'Anne', 'Joel', 'Ray', 'Pren');
sort($names);           // Anne, Joel, Mike, Pren, Ray
```

How to sort numbers in ascending order

```
$numbers = array(520, '33', 9, '199');
sort($numbers, SORT_NUMERIC);       // 9, 33, 199, 520
```

How to sort in descending order

```
$names = array('Mike', 'Anne', 'Joel', 'Ray', 'Pren');
rsort($names);          // Ray, Pren, Mike, Joel, Anne
```

How to sort an associative array

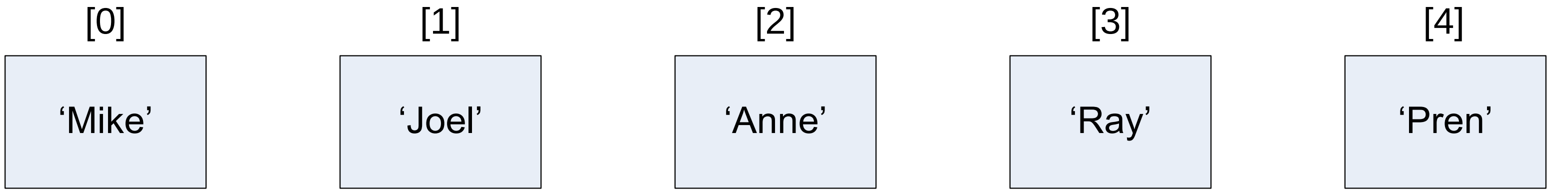
```
$tax_rates = array('NC' => 7.75,
                   'NY' => 8.875, 'CA' => 8.25);
asort($tax_rates);     // sorts by value (ascending)
ksort($tax_rates);     // sorts by key (ascending)
arsort($tax_rates);    // sorts by value (descending)
krsort($tax_rates);    // sorts by key (descending)
```

Examples for study

How to shuffle and deal a deck of cards

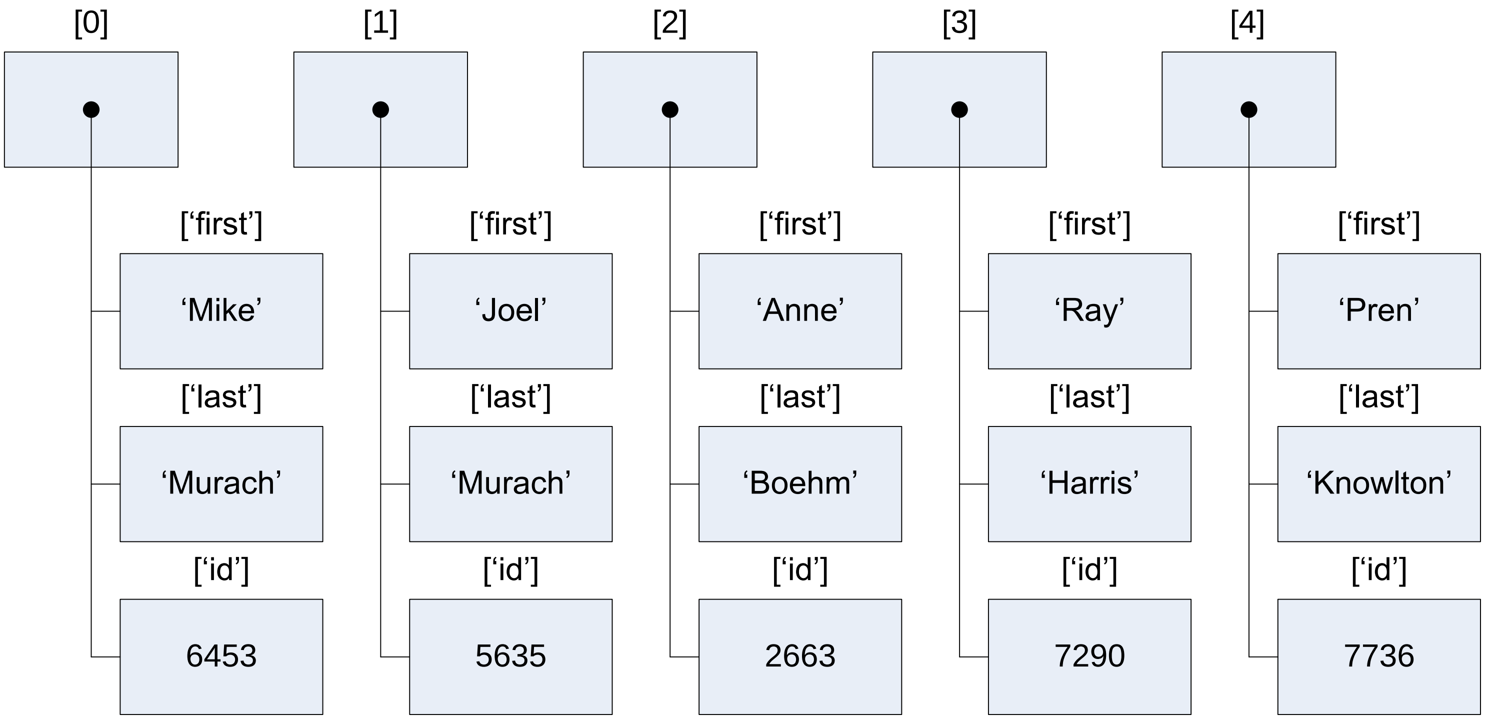
```
// Create the deck of cards
$faces = array('2', '3', '4', '5', '6', '7', '8',
               '9', 'T', 'J', 'Q', 'K', 'A');
$suits = array('h', 'd', 'c', 's');
$cards = array();
foreach($faces as $face) {
    foreach($suits as $suit) {
        $cards[] = $face . $suit;
    }
}

// Shuffle the deck and deal the cards
shuffle($cards);
$hand = array();
for ($i = 0; $i < 5; $i++) {
    $hand[] = array_pop($cards);
}
echo implode(', ', $hand);
// 3c, Js, Qs, Jc, Qc (for example)
```

A simple array



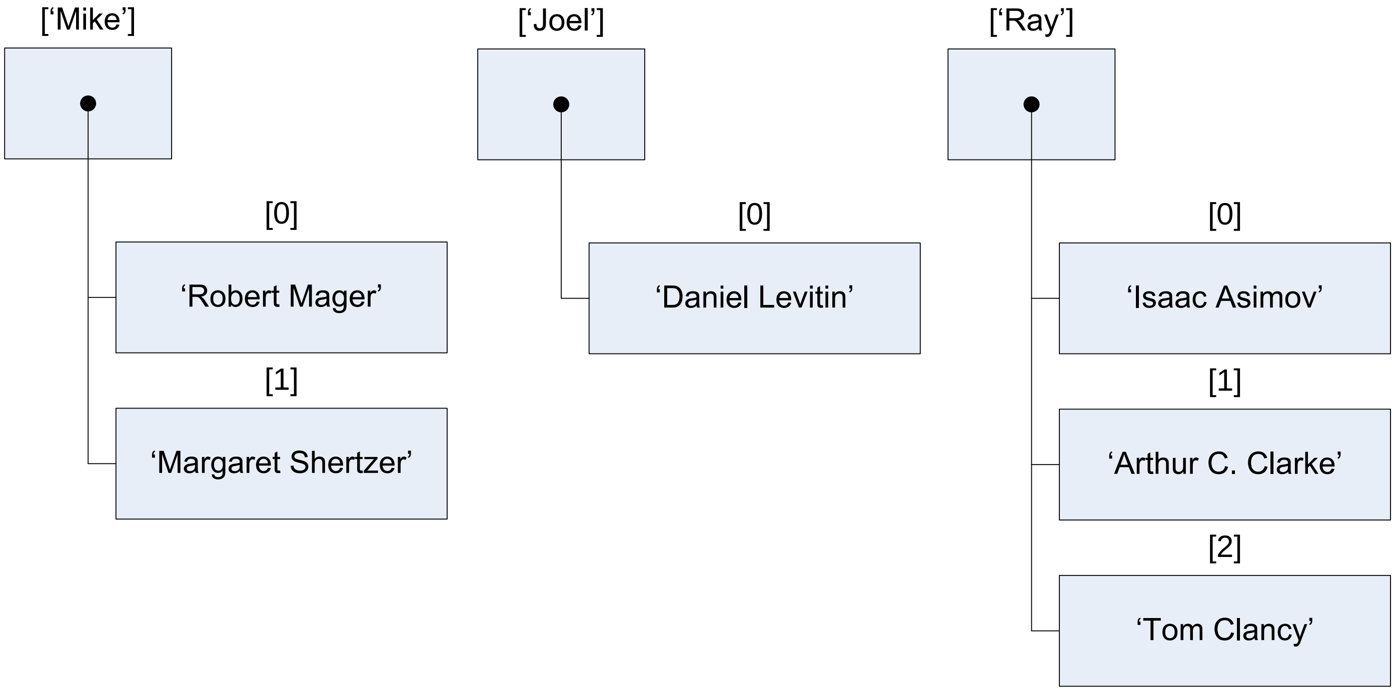
[0]	[1]	[2]	[3]	[4]
‘Mike’	‘Joel’	‘Anne’	‘Ray’	‘Pren’

A rectangular array of arrays



	[0]	[1]	[2]	[3]	[4]
[‘first’]	‘Mike’	‘Joel’	‘Anne’	‘Ray’	‘Pren’
[‘last’]	‘Murach’	‘Murach’	‘Boehm’	‘Harris’	‘Knowlton’
[‘id’]	6453	5635	2663	7290	7736

A jagged array of arrays


['Mike']

[0]

'Robert Mager'

[1]

'Margaret Shertzer'

['Joel']

[0]

'Daniel Levitin'

['Ray']

[0]

'Isaac Asimov'

[1]

'Arthur C. Clarke'

[2]

'Tom Clancy'

Code that creates an array of arrays

```
$times_table = array();
for ($i = 0; $i <= 12; $i++) {    // add 13 elements that
    $times_table[$i] = array();   // contain empty arrays
}
```

Code that adds values to the array of arrays

```
for ($i = 0; $i <= 12; $i++) {
    for ($j = 0; $j <= 12; $j++) {
        $times_table[$i][$j] = $i * $j;
    }
}
```

Code that refers to elements in the array of arrays

```
echo $times_table[4][3];          // displays 12
echo $times_table[7][6];          // displays 42
```

Code that creates a cart array

```
$cart = array();
```

Adding an associative array to the cart array

```
$item = array();       // create an empty item array
$item['itemCode'] = 123;
$item['itemName'] = "Visual Basic 2010";
$item['itemCost'] = 52.5;
$item['itemQuantity'] = 5;
$cart[] = $item;       // add item array to cart array
```

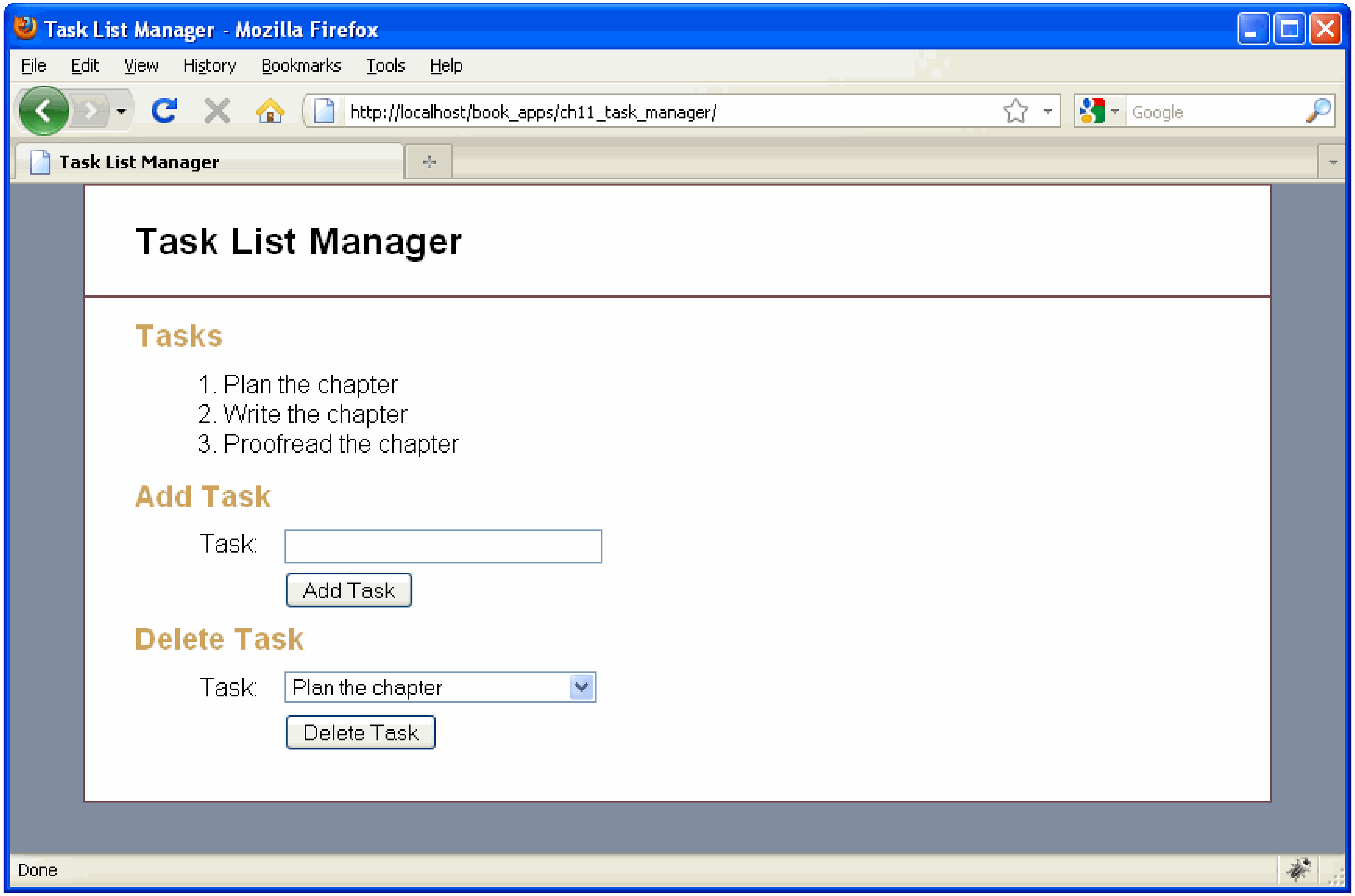
Adding another associative array to the cart array

```
$item = array();       // create an empty item array
$item["itemCode"] = 456;
$item["itemName"] = "C++ 2010";
$item["itemCost"] = 52.5;
$item["itemQuantity"] = 2;
$cart[]  = $item;      // add item array to cart array
```

Referring to the elements in the array of arrays

```
echo $cart[0]["itemCode"];          // displays 123
echo $cart[1]["itemName"];          // displays C++ 2010
```

The Task List Manager application



Task List Manager - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/book_apps/ch11_task_manager/
Google
Task List Manager
Done

Task List Manager

Tasks

1. Plan the chapter
2. Write the chapter
3. Proofread the chapter

Add Task

Task:

Add Task

Delete Task

Task: Plan the chapter

Delete Task

The index.php file

```
<?php
if (isset($_POST['tasklist'])) {
    $task_list = $_POST['tasklist'];
} else {
    $task_list = array();
}

$errors = array();
```

The index.php file (continued)

```
switch( $_POST['action'] ) {
    case 'add':
        $new_task = $_POST['newtask'];
        if (empty($new_task)) {
            $errors[] = 'The new task cannot be empty.';
        } else {
            $task_list[] = $new_task;
        }
        break;
    case 'delete':
        $task_index = $_POST['taskid'];
        unset($task_list[$task_index]);
        $task_list = array_values($task_list);
        break;
}

include('task_list.php');
?>
```

The task_list.php file

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 ...>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Task List Manager</title>
    <link rel="stylesheet" type="text/css"
          href="main.css"/>
</head>
<body>
    <div id="page">
    <div id="header">
        <h1>Task List Manager</h1>
    </div>
    <div id="main">
```

The task_list.php file (continued)

```
<!-- part 1: the errors -->
<?php if (count($errors) > 0) : ?>
<h2>Errors</h2>
<ul>
    <?php foreach($errors as $error) : ?>
        <li><?php echo $error; ?></li>
    <?php endforeach; ?>
</ul>
<?php endif; ?>

<!-- part 2: the tasks -->
<h2>Tasks</h2>
<?php if (count($task_list) == 0) : ?>
    <p>There are no tasks in the task list.</p>
<?php else: ?>
    <ul id="task">
    <?php foreach( $task_list as $id => $task ) : ?>
        <li><?php echo $id + 1 . '. ' . $task; ?></li>
    <?php endforeach; ?>
    </ul>
<?php endif; ?>
```

The task_list.php file (continued)

```
<!-- part 3: the add form -->
<h2>Add Task</h2>
<form action="." method="post" >
   <?php foreach( $task_list as $task ) : ?>
     <input type="hidden" name="tasklist[]"
            value="<?php echo $task; ?>"/>
   <?php endforeach; ?>
   <input type="hidden" name="action" value="add"/>
   <label>Task:</label>
   <input type="text" name="newtask"
          id="newtask"/><br />
   <label>&nbsp;</label>
   <input type="submit" value="Add Task"/><br />
</form>
```

The task_list.php file (continued)

```
<!-- part 4: the delete form -->
<?php if (count($task_list) > 0) : ?>
<h2>Delete Task</h2>
<form action="." method="post" >
   <?php foreach( $task_list as $task ) : ?>
     <input type="hidden" name="tasklist[]"
            value="<?php echo $task; ?>"/>
   <?php endforeach; ?>
   <input type="hidden" name="action" value="delete"/>
   <label>Task:</label>
   <select name="taskid">
       <?php foreach( $task_list as $id => $task ) : ?>
           <option value="<?php echo $id; ?>">
               <?php echo $task; ?>
           </option>
       <?php endforeach; ?>
   </select><br />
   <label>&nbsp;</label>
   <input type="submit" value="Delete Task"/>
</form>
<?php endif; ?>
```

The task_list.php file (continued)

```
    </div><!-- end main -->
    </div><!-- end page -->
</body>
</html>
```