How to set and get a cookie

Figure 12-2 shows you how to set a cookie in the user's browser and how to get data from that cookie. To create a cookie and set it in the browser, you can use the setcookie() function. To get the value of a cookie, you can use the superglobal $_COOKIE variable. This variable is an associative array.

However, since the data in the $_COOKIE array can be manipulated by the user, it's a better practice to use the filter_input() function when getting a value from it. The figure shows how to use the filter_input() function to get a cookie that stores an integer value, but you can use other validation or sanitation filters to work with cookies that store other types of values.

When using the setcookie() function, the $name parameter is the only required parameter. However, in most cases, you need to provide values for the first four parameters. That way, you can use the $name and $value parameters to specify the name/value pair for the cookie, you can use the $expire parameter to specify an expiration date for the cookie, and you can use the $path parameter to make the cookie available to all pages for the current web application.

If you set the $expire parameter to 0, the cookie only exists until the user closes the browser. This is called a *session cookie*. However, if you want, you can set the $expire parameter to a date in the future. In that case, the cookie stays in the browser until the expiration date. This is called a *persistent cookie*.

The $path parameter should almost always be set to the root of your website. That way, every page in your application can access the cookie. If you use the default value of the current directory, a cookie created here:

```
http://www.example.com/login/index.php
```

isn't available here:

```
http://www.example.com/checkout/index.php.
```

The first example shows how to create a cookie with a name of "userid" and a value of "87". This cookie expires after one year and is available to any page in any directory on the web server. To create the timestamp for the expiration date, this code uses the strtotime() function to get a timestamp that's one year from the current date.

The second example uses the filter_input() function to get the value of the cookie named userid from the $_COOKIE variable. As a result, if the browser has a cookie named userid, this code returns the value. Otherwise, it returns a NULL value.

The third example shows how to delete a cookie. To do that, you can use the setcookie() function to set the $expire parameter of the cookie to a date in the past. Here, the code sets the $expire parameter to one year in the past. In addition, you must set the $value parameter to an empty string, and you must set all remaining parameters to the same values that were used when the cookie was created.

The syntax of the setcookie() function

```
setcookie($name, $value, $expire, $path, $domain, $secure, $httponly)
```

The parameters of the setcookie() function

Parameter	Description
`$name`	The name of the cookie.
`$value`	The value of the cookie. The default is an empty string.
`$expire`	The expiration date of the cookie as a timestamp. If set to 0, the cookie expires when the user closes the browser. The default is 0.
`$path`	The path on the server that the cookie is available to. If set to '/', the cookie is available to all directories on the current server. The default is the directory of the PHP file that's setting the cookie.
`$domain`	The domain that the cookie is available to. The default is the name of the server that's setting the cookie.
`$secure`	If TRUE, the cookie is available only if it is sent using HTTPS. The default is FALSE.
`$httponly`	If TRUE, the cookie is only made available through the HTTP protocol and not through client-side scripting languages such as JavaScript. The default is FALSE.

Set a cookie in the browser

```
$name = 'userid';
$value = '87';
$expire = strtotime('+1 year');
$path = '/';
setcookie($name, $value, $expire, $path);
```

Get the value of a cookie from the browser

```
$userid = filter_input(INPUT_COOKIE, 'userid', FILTER_VALIDATE_INT);   // 87
```

Delete a cookie from the browser

```
$expire = strtotime('-1 year');
setcookie('userid', '', $expire, '/');
```

Description

- To set a cookie in the user's browser, you can use the setcookie() function. This function must be called before any HTML output is sent from your application.
- A *session cookie* expires when the user closes the browser. A *persistent cookie* doesn't expire until the specified expiration date.
- Once a cookie has been set, you can get it the next time the browser requests a page. To do that, you can use the filter_input() function to get it from the superglobal $_COOKIE variable. This variable is an associative array where the cookie name is the key and the cookie value is the value.
- To delete a cookie from a browser, set the value to an empty string and the expiration date to a time in the past. Any remaining parameters must have the same values as when the cookie was originally created.

Figure 12-2 How to set and get a cookie