How to start a session

Figure 12-5 shows how to start a session. Although session tracking is built into PHP, it isn't automatically available. To make it available, you can start a new session or resume a previous session by calling the session_start() function at the beginning of each page in your application that needs to access the session data.

The session_start() function prompts PHP to check for a session ID in the request and to create a new session ID and session cookie if one isn't found. Since the session_start() function may set a cookie, it must be called before any HTML content is sent to the browser. By default, PHP uses a session cookie to store the session ID in the user's browser. As a result, when the user closes the browser, the session ends.

However, older versions of some browsers, like Chrome and Firefox, allow you to save tabs when you close the browser. In this case, the session doesn't end. Instead, the browsers save the session cookie and use it the next time they restart. As a result, to end a session with one of these browsers, you must make sure that you don't save the open tabs.

The cookie for a session works like the cookies described at the beginning of this chapter. To control how the session cookie works, you can use the session_set_cookie_params() function to set the parameters of the session cookie. In this figure, the second example creates a session cookie that persists in the user's browser for one year and is available to all pages of the application. Note that you need to call the session_set_cookie_params() function before the session_start() function.

For the session_set_cookie_params() function, the $lifetime parameter is the only required parameter. This parameter specifies the number of seconds for the session cookie to be stored in the browser. When PHP creates the session cookie, it adds this number of seconds to the current time to get the expiration date for the session cookie. In addition to the $lifetime parameter, you usually want to set the $path parameter to allow all pages in the application to access the session cookie. And you may occasionally want to set one of the other four parameters described in this figure.

A function to start a session

Function	Description
`session_start()`	Starts a new session or resumes a previous session. Returns TRUE if successful and FALSE otherwise. This function must be called before the page sends any HTML output to your application.

Start a session with the default cookie parameters

```
session_start();
```

The syntax of the session_set_cookie_params() function

```
session_set_cookie_params($lifetime, $path, $domain, $secure, $httponly)
```

The parameters of the session_set_cookie_params() function

Parameter	Description
`$lifetime`	The lifetime of the session cookie in seconds. The default is 0.
`$path`	The path on the server the session cookie is available to. The default is the current directory of the script that is setting the cookie.
`$domain`	The domain that the cookie is available to. The default is the name of the server that is setting the cookie.
`$secure`	If TRUE, the cookie is available only if it is sent using a secure HTTP connection (an HTTPS connection). The default is FALSE.
`$httponly`	If TRUE, the cookie is only available through the HTTP protocol and not through client-side scripting languages such as JavaScript. The default is FALSE.

Start a session with custom cookie parameters

```
$lifetime = 60 * 60 * 24 * 365;    // 1 year in seconds
session_set_cookie_params($lifetime, '/');
session_start();
```

Description

- A *session* starts when a browser makes its first request to a page that includes the session_start() function.
- By default, a session uses a session cookie to associate a browser with the data for its session. However, you can use the session_set_cookie_params() function to customize the cookie for the session.
- In the session_set_cookie_params() function, the $lifetime parameter is the only required parameter. The other parameters are optional.
- The session_set_cookie_params() function must be called before the session_start() function.