

PHP



LẬP TRÌNH PHP 1

BÀI 6: COOKIE VÀ SESSION

PHẦN 1

Nội dung bài học

- Làm việc với Cookie
- Làm việc với Session



1. Làm việc với cookie

Trong phần này có các nội dung:

- 1.1. Giới thiệu về cookie
- 1.2. Thiết lập và truy xuất cookie
- 1.3. Bật và tắt cookie

1. Giới thiệu cookie

❑ Cookie là cặp tên/giá trị được lưu trên trình duyệt máy khách

❑ Ví dụ một cookie:

```
PHPSESSID=D1F15245171203E8670487F020544490  
user_id=87  
email=j_smi_thehotmail.com userName=jsmith  
passwordCookie=opensesame
```

❑ Những cách dùng thường gặp của cookie:

- Cho phép người dùng không cần đăng nhập và điền form đăng ký
- Thu thập những dữ liệu như tên người dùng, mật khẩu, địa chỉ hoặc thông tin thẻ tín dụng
- Tùy chỉnh trang hiển thị các thông tin như dự báo thời tiết, tỷ số các trận đấu thể thao và giá cổ phiếu
- Quảng cáo trúng đích: Các banner quảng cáo nhắm vào khách hàng mục tiêu quan tâm đến quảng cáo

- ❑ Trên máy chủ, ứng dụng web tạo cookie và gửi nó tới trình duyệt. Trên máy khách, trình duyệt lưu cookie và gửi nó trở lại máy chủ mỗi khi truy cập trang từ máy chủ đó
- ❑ Mặc định, cookie chỉ có hiệu lực cho đến khi người dùng đóng trình duyệt. Tuy nhiên, có thể thiết lập để cookie tồn tại trong trình duyệt của người dùng với thời gian lên đến ba năm
- ❑ Mỗi cookie được giới hạn 4Kb dung lượng

1.2. Thiết lập và truy xuất cookie

❑ Sử dụng hàm `setcookie` với cú pháp:

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

❑ Các tham số cho hàm `setcookie`:

Tham số	Mô tả
<code>\$name</code>	Tên cookie.
<code>\$value</code>	Giá trị của cookie. Mặc định là chuỗi rỗng.
<code>\$expire</code>	Ngày hết hạn của cookie là nhãn thời gian. Nếu được thiết lập là 0, cookie sẽ hết hạn khi người dùng tắt trình duyệt. Giá trị mặc định là 0.
<code>\$path</code>	Đường dẫn trên máy chủ mà cookie có hiệu lực. Nếu được thiết lập là 7, cookie sẽ có hiệu lực với tất cả các đường dẫn trên máy chủ hiện tại. Đường dẫn mặc định là đường dẫn cho file thiết lập cookie.
<code>\$domain</code>	Tên miền mà cookie có hiệu lực. Mặc định là tên của máy chủ thiết lập cookie.
<code>\$secure</code>	Nếu là TRUE, cookie sẽ chỉ có hiệu lực nếu được gửi qua HTTPS. Mặc định là FALSE.
<code>\$httponly</code>	Nếu là TRUE, cookie chỉ có hiệu lực qua giao thức HTTP chứ không qua các ngôn ngữ kịch bản phía máy khách như JavaScript. Mặc định là FALSE.

□ Ví dụ:

```
//Thiết lập cookie trong trình duyệt
$name = 'userid';
$value = 'rharris';
$expire = strtotime('+1 year');
$path = ' / ';
setcookie($name, $value, $expire, $path);

//Lấy giá trị của cookie từ trình duyệt
$userid = $COOKIE['userid']; // $userid là 'rharris'

//Xóa cookie khỏi máy chủ
$expire = strtotime('-1 year'); setcookie('userid', '', $expire, '/');
```

1.3. Bật và tắt cookie

- ❑ Để kiểm thử hoạt động của ứng dụng trong trường hợp người dùng tắt cookie, bạn có thể tắt cookie trên trình duyệt của mình
- ❑ Để kiểm thử ứng dụng trong điều kiện bình thường, bật cookie trên trình duyệt
- ❑ **Hướng dẫn bật và tắt cookie trong trình duyệt Firefox 3.6:**
 - Mở menu Tools và chọn lệnh Options.
 - Nhấn vào tab Privacy.
 - Sử dụng check box “Accept cookies from sites” để bật hoặc tắt cookie.
- ❑ **Hướng dẫn bật và tắt cookie trong Internet Explorer 8:**
 - Mở menu Tools và chọn lệnh Options.
 - Nhấn vào tab Privacy.
 - Sử dụng thanh trượt điều khiển để bật hoặc tắt cookie. Để tắt cookie, thiết lập cấp bảo mật là “Block All Cookies”. Để bật cookie, nhấn vào nút Default nhằm quay về thiết lập bảo mật mặc định.

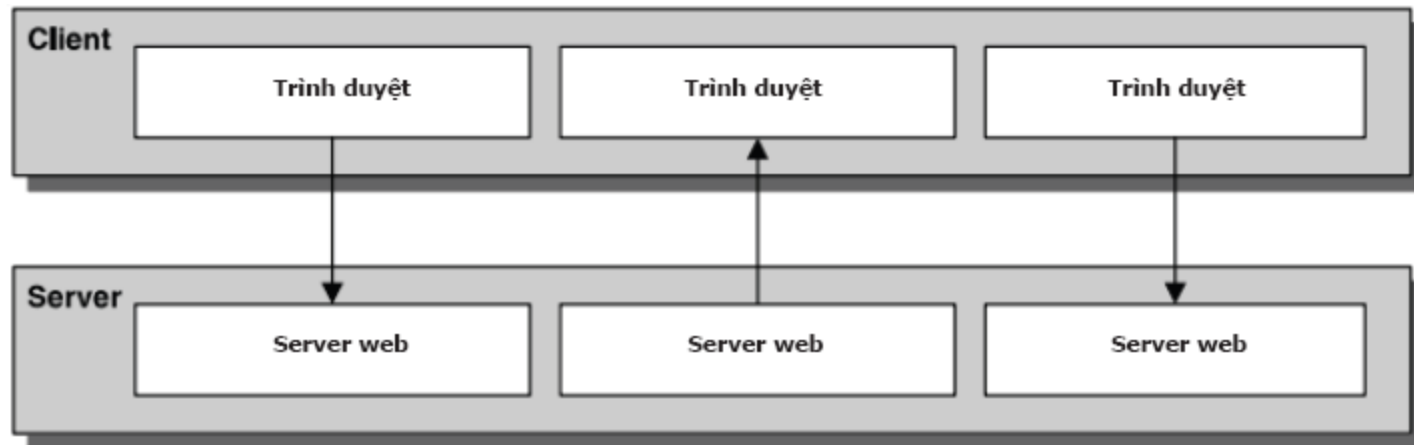
Trong phần này có các nội dung:

- 2.1. Nguyên nhân khó sử dụng session với HTTP
- 2.2. Cách thức hoạt động của session trong PHP
- 2.3. Bắt đầu một session
- 2.4. Gán và lấy giá trị của biến session
- 2.5. Kết thúc một session
- 2.6. Quản lý session

2.1. Nguyên nhân khó sử dụng session với HTTP

- ❑ Session (phiên): sau khi trình duyệt gửi yêu cầu thì HTTP ngắt kết nối tới máy chủ. Để duy trì trạng thái, ứng dụng web phải thực hiện session
- ❑ Mặc định, PHP sử dụng cookie để lưu session ID trên mỗi trình duyệt. Khi đó, trình duyệt chuyển cookie tới máy chủ với mỗi yêu cầu
- ❑ Để theo dõi session khi cookie bị tắt, có thể sử dụng cách mã hóa URL để lưu session ID trong URL cho mỗi trang của ứng dụng

Nguyên nhân khó sử dụng session với HTTP



Yêu cầu HTTP đầu tiên:
Trình duyệt yêu cầu xem trang.

Phản hồi HTTP đầu tiên:
Server trả về trang được yêu cầu và ngắt kết nối.

Các yêu cầu HTTP tiếp theo:
Trình duyệt gửi yêu cầu xem trang.
Server web không thể liên kết trình duyệt với yêu cầu trước đó.

2.2. Cách thức hoạt động của session trong PHP

- ❑ **Bước 1:** trình duyệt trên máy khách gửi yêu cầu xem trang PHP tới máy chủ web
- ❑ **Bước 2:** PHP kiểm tra liệu yêu cầu đã bao gồm session ID (định danh phiên) chưa. Nếu chưa, PHP sẽ tạo một session mới trên máy chủ và gán cho nó một session ID duy nhất. Lúc này, ứng dụng có thể lưu dữ liệu vào session
- ❑ **Bước 3:** session ID sẽ được gửi trả lại trình duyệt như cookie trong phản hồi
- ❑ Khi trình duyệt gửi các yêu cầu sau đó, cookie session ID được gộp trong yêu cầu. PHP cũng kiểm tra liệu yêu cầu đã bao gồm session ID chưa. Vì lần này session ID đã có, nên PHP sử dụng session ID để truy cập, điều chỉnh hoặc thêm dữ liệu khi cần

PHP

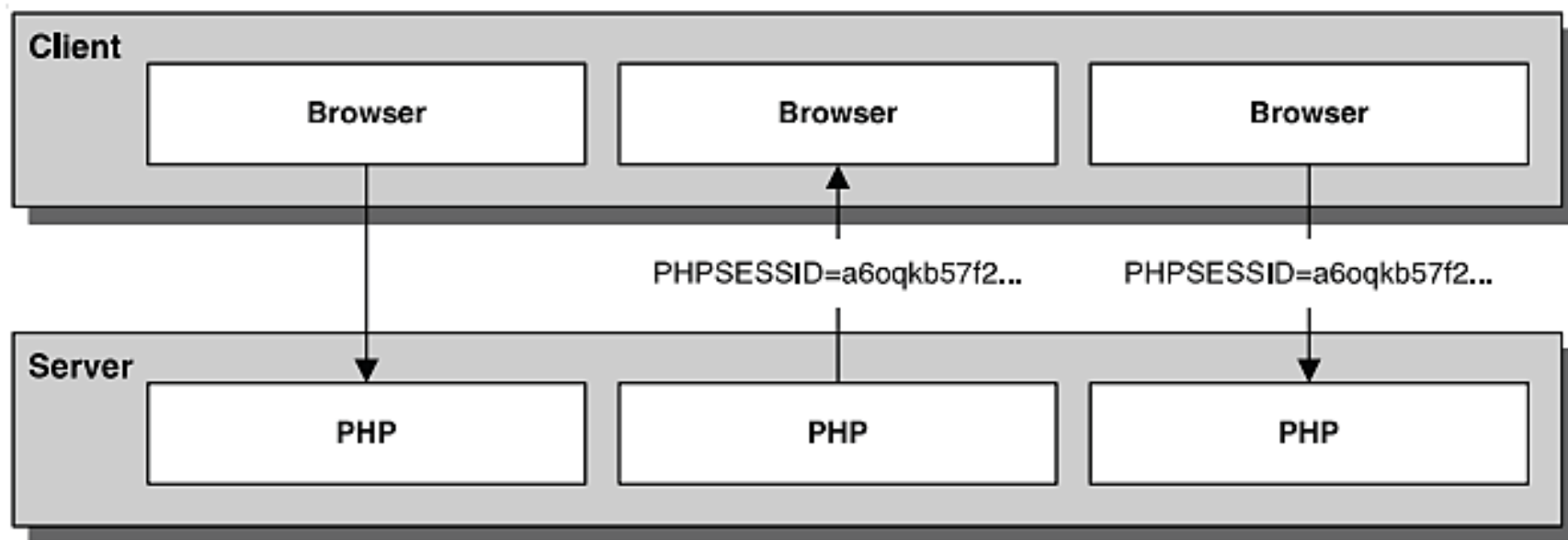


LẬP TRÌNH PHP 1

BÀI 6: COOKIE VÀ SESSION

PHẦN 2

Cách thức hoạt động của session trong PHP



First HTTP Request:
The browser requests a PHP page. PHP creates a session and assigns it an ID.

First HTTP Response:
The server returns the requested page and the ID for the session as a cookie.

Following HTTP Requests:
The browser requests a PHP page and sends the session ID cookie. PHP uses the session ID to associate the browser with its session.

2.3. Bắt đầu một session

- ❑ Session bắt đầu khi trình duyệt gửi yêu cầu xem trang đầu tiên với hàm `session_start`

Hàm	Mô tả
<code>Sessionstart ()</code>	Bắt đầu phiên làm việc mới hoặc tiếp tục phiên cũ. Trả về TRUE nếu thành công và FALSE nếu ngược lại. Hàm này phải được gọi trước khi trang gửi bất kỳ dữ liệu HTML nào tới ứng dụng.

- ❑ Mặc định, session làm việc sử dụng cookie theo phiên để liên kết trình duyệt với dữ liệu. Tuy nhiên, có thể sử dụng hàm `session_set_cookie_params` để tùy chỉnh cookie của phiên
- ❑ Hàm `session_set_cookie_params` phải được gọi trước hàm `session_start`

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

❑ Cú pháp hàm `session_set_cookie_params`.

Tham số cho hàm session_set_cookie_params:

Tham số	Mô tả
\$lifetime	Vòng đời của cookie theo phiên tính theo giây. Mặc định là 0.
\$path	Đường dẫn trên máy chủ mà cookie theo phiên có hiệu lực. Mặc định là đường dẫn hiện tại của đoạn mã thiết lập cookie.
\$domain	Tên miền mà cookie có hiệu lực. Mặc định là tên máy chủ thiết lập cookies.
\$secure	Nếu TRUE, cookie chỉ sẵn dùng nếu được gửi qua kết nối bảo mật HTTP (kết nối HTTPS). Mặc định là FALSE.
\$httponly	Nếu TRUE, cookie chỉ sẵn dùng qua giao thức HTTP, chứ không qua các ngôn ngữ kịch bản phía khách như JavaScript. Mặc định là FALSE.

```
$lifetime = 60 * 60 * 24 * 365; // 1 năm theo giây
```

```
session_set_cookie_params($lifetime, '/');  
sessionstart();
```


2.4. Gán và lấy giá trị của biến session

- ❑ Khi khởi tạo phiên, sử dụng biến toàn cục tự động `$_SESSION` để thiết lập và lấy dữ liệu của người dùng cho phiên. Biến này là một mảng liên kết
- ❑ Sử dụng hàm `isset` để kiểm tra sự tồn tại của một phần tử trong mảng `$_SESSION`

```
//Thiết lập biến vô hướng trong phiên : lập thành  
$_SESSION['product_code'] = 'MBT-1753';  
  
//Lấy biến từ phiên ởng:  
$product_code = $_SESSION['product_code'];
```

❑ Hướng dẫn thiết lập và lấy mảng:

```
//Thiết lập mảng trong phiên
if (!isset($_SESSION['cart'])) {
    $_SESSION['cart'] = array();
}
//Thêm phần tử vào mảng được lưu trong phiên
$_SESSION['cart']['key1'] = 'value1';
$_SESSION['cart']['key2'] = 'value2';

//Lấy và sử dụng mảng được lưu trong phiên
$cart = $_SESSION['cart'];
foreach ($cart as $item) { echo $item; }
```

```
//Xóa một biến
```

❑ unset(\$_SESSION['cart']); khởi phiên:

```
//Xóa tất cả các biến
$_SESSION = array();
```

2.5. Kết thúc một session

❑ Sử dụng hàm `session_destroy`:

Hàm	Mô tả
<code>session_destroy()</code>	Kết thúc phiên. Trả về TRUE nếu thành công và FALSE nếu ngược lại.

❑ Session kết thúc khi:

- Người dùng tắt trình duyệt, một khoảng thời gian nhất định trôi qua mà không có yêu cầu nào
- Đoạn mã gọi hàm `session_destroy`

❑ Xóa tất cả dữ liệu liên quan đến phiên làm việc khỏi máy khách và máy chủ:

- Xóa dữ liệu phiên khỏi bộ nhớ
- Gọi hàm `session_destroy`
- Sử dụng hàm `setcookie`

❑ Hàm `session_name` lấy tên của cookie theo phiên. Mặc định, cookie theo phiên có tên là “PHPSESSID”

❑ Hàm `session_get_cookie_params` lấy mảng liên

```
//Kết thúc session
$_SESSION = array(); //xóa dữ liệu về session khỏi bộ nhớ
Session_destroy(); //xóa session ID

//Xóa cookie session khỏi trình duyệt
$name = session_name(); //lấy tên của cookie theo session
$expire = strtotime('-1 year'); //tạo ngày hết hạn trong quá khứ
$params = session_get_cookie_params(); //lấy tham số session
$path = $params['path'];
$domain = $params['domain'];
$secure = $params['secure'];
$httponly = $params['httponly'];
setcookie($name, '', $expire, $path, $domain, $secure, $httponly);
```

❑ Sử dụng các hàm quản lý:

Hàm	Mô tả
<code>session_name()</code>	Lấy tên của cookie theo phiên. Mặc định là PHPSESSID.
<code>session_id([\$id])</code>	Nếu tham số không được chỉ định, hàm sẽ lấy session ID hiện thời. Nếu không có phiên nào tồn tại, hàm sẽ lấy chuỗi rỗng. Nếu tham số được chỉ định, hàm sẽ gán một giá trị của thẻ cho session ID .
<code>session_write_close()</code>	Kết thúc phiên hiện thời và lưu dữ liệu của phiên. Hàm này chỉ cần trong những trường hợp đặc biệt như chuyển hướng.
<code>session_regenerate_id()</code>	Tạo session ID mới cho phiên làm việc hiện thời. Trả về TRUE nếu thành công và FALSE nếu ngược lại. Hàm này có thể được sử dụng để tránh hiện tượng chiếm đoạt phiên.

- ❑ Nếu chuyển hướng trình duyệt bằng cách dùng hàm header theo sau là hàm exit, không phải lúc nào PHP cũng lưu dữ liệu về session. Vì vậy, cần gọi hàm `session_write_close` để buộc PHP lưu dữ liệu về session

□ Ví dụ:

```
//Lấy tên của cookie theo phiên  
$name = session_name(); //mặc định là FHFSESSID  
  
//Lấy giá trị của session ID  
$id = session_id(); //ví dụ, 1ljeflfoilg8u6qnui4f8b6el4  
  
//Thiết lập session ID  
session_id('abc123');
```

Tổng kết bài học

- Làm việc với Cookie
- Làm việc với Session





KẾT THÚC