



THỰC HỌC – THỰC NGHIỆP



Conceive Design Implement Operate

LẬP TRÌNH PHP1

SESSION & COOKIE

- ⦿ Cookie
- ⦿ Session
- ⦿ PHP & PDO





PHẦN 1

- ☐ Cookie được sử dụng để lưu trữ thông tin ở phía máy tính của client
- ☐ Cookie là một tập tin nhỏ (4KB) mà webserver chỉ định lưu trữ trên client
- ☐ Cookie không bị mất đi khi đóng ứng dụng. Cookie phụ thuộc vào thời gian sống được thiết lập
- ☐ Cookie thường được thiết lập trên HTTP header

Thiết lập cookie với PHP

- ☐ Dùng hàm ***setcookie***
- ☐ Setcookie phải được gọi trước bất kỳ output nào được gửi đến trình duyệt.
- ☐ Setcookie có tổng 5 tham số nhưng thường được chứa 3 tham số chính là ***name***, ***value*** và ***expiration date***
- ☐ Cú pháp (thường sử dụng)

```
setcookie("lastvisit", date("H:i:s"), time() + 60*60);
```

❑ Cú pháp (đầy đủ) \longrightarrow `setcookie("lastvisit",
date("H:i:s"),
time() + 60*60,
'/foo/',
'fr.example.com');`

❑ Giải thích các tham số

- ❖ Name: tên cookie
- ❖ Value: được thiết lập bằng hàm date
- ❖ Expiration data (ngày hết hạn) được đo bằng giây và thường được thiết lập tương đối so với thời gian hiện tại tính bằng giây, được truy xuất thông qua hàm time
- ❖ Có thể cung cấp thêm 2 đối số tùy chọn: path và domain

□ Ví dụ

 index.php

```
1  <?php
2  $cookie_name = "user";
3  $cookie_value = "John Doe";
4  setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
5  ?>
6  <html>
7  <body>
8
9  <?php
10 if(!isset($_COOKIE[$cookie_name])) {
11     echo "Cookie named '" . $cookie_name . "' is not set!";
12 } else {
13     echo "Cookie '" . $cookie_name . "' is set!<br>";
14     echo "Value is: " . $_COOKIE[$cookie_name];
15 }
16 ?>
17
18 </body>
19 </html>
20
```

- ❑ Để sửa đổi cookie chỉ cần đặt lại cookie bằng hàm `setcookie()`
- ❑ Có thể xóa cookie theo cách thủ công bằng cách tạo lại cookie đó với ngày hết hạn cũ. Sau đó, nó sẽ bị gỡ bỏ khi đóng trình duyệt.

```
setcookie("lastvisit", 0, 0);
```

Hoặc

```
// set the expiration date to one hour ago  
setcookie("user", "", time() - 3600);
```

- ❑ Kiểm tra cookie có được bật hay không: `count($_COOKIE) > 0`

Cookie array

- ❑ Có thể đặt cookie cho người dùng (user). cookie này sẽ được gửi cùng lần tiếp theo khi người dùng đó xem một trang trên cùng domain. Sau đó, nó có thể được truy cập thông qua mảng `$_COOKIE`.

```
if (isset($_COOKIE['lastvisit']))  
    echo "Last visit: " . $_COOKIE['lastvisit'];
```

demo



PHẦN 2

- ❑ Session là cách để lưu trữ thông tin (trong các biến) trên nhiều pages
- ❑ Ví dụ:
 - ❖ 1 session: mở ứng dụng, thực hiện 1 số thay đổi, đóng ứng dụng
 - ❖ máy tính biết bạn mở/đóng ứng dụng lúc nào, làm gì nhưng server thì không
- ❑ Biến session lưu trữ thông tin người dùng được sử dụng trên nhiều page (username, shopping cart,...)
- ❑ Mặc định, biến session kéo dài cho đến khi người dùng đóng trình duyệt.

Bắt đầu session

- ❑ Để bắt đầu session, sử dụng hàm `session_start`. Hàm này phải xuất hiện trước bất kỳ output nào được gửi đến web

```
<?php session_start(); ?>
```

- ❑ Hàm `session_start` đặt trên máy tính của khách hàng, chứa một id được sử dụng để liên kết máy khách với phiên. Nếu máy khách đã có một phiên đang diễn ra, chức năng sẽ tiếp tục phiên đó thay vì bắt đầu một phiên mới.

Session array

- ❑ Khi phiên bắt đầu, mảng `$_SESSION` được sử dụng để lưu trữ dữ liệu phiên cũng như truy xuất dữ liệu đó.
- ❑ Ví dụ: số lượt xem trang được lưu trữ bằng mã sau. Lần đầu tiên trang được xem, phần tử phiên được khởi tạo thành một phần tử.

```
if(isset($_SESSION['views']))  
    $_SESSION['views'] += 1;  
else  
    $_SESSION['views'] = 1;
```

- ❑ Phần tử này hiện có thể được truy xuất từ bất kỳ trang nào trên miền miền là `session_start` được gọi trên đầu trang đó.

```
echo 'Views: ' . $_SESSION['views'];
```

Xoá session

- ❑ Một session được đảm bảo sẽ kéo dài cho đến khi người dùng rời khỏi trang web. Sau đó, trình thu gom rác có quyền xóa session đó.
- ❑ Để loại bỏ một biến session theo cách thủ công, có thể sử dụng hàm ***unset***. Để loại bỏ tất cả các biến phiên, có hàm ***session_destroy***.

```
unset($_SESSION['views']); // destroy session variable  
session_destroy(); // destroy session
```

□ Ví dụ: lấy giá trị biến session.

 getsession.php

```
1  <?php
2  session_start();
3  ?>
4  <!DOCTYPE html>
5  <html>
6  <body>
7
8  <?php
9  // Echo session variables that were set on previous page
10 echo "Favorite color is " . $_SESSION["favcolor"] . "<br>";
11 echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
12 ?>
13
14 </body>
15 </html>
16
```


❑ Xoá tất cả các biến global session và huỷ session: `session_unset()` và `session_destroy()`

 getsession.php

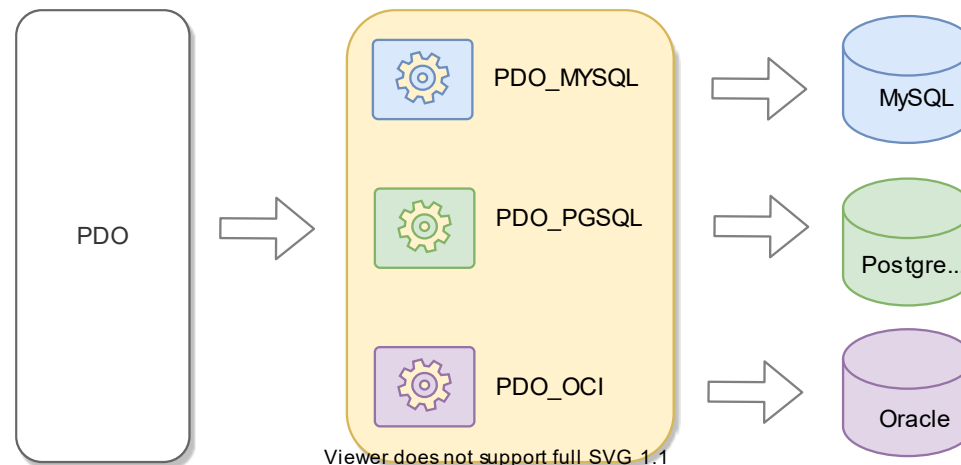
```
1  <?php
2  session_start();
3  ?>
4  <!DOCTYPE html>
5  <html>
6  <body>
7
8  <?php
9  // remove all session variables
10 session_unset();
11
12 // destroy the session
13 session_destroy();
14 ?>
15
16 </body>
17 </html>
18
```

SO SÁNH GIỮA COOKIE VÀ SESSION

Cookie	Session
Cookie được lưu trữ trên trình duyệt của người dùng.	Session không được lưu trữ trên trình duyệt.
Dữ liệu cookie được lưu trữ ở phía client.	Dữ liệu session được lưu trữ ở phía server.
Dữ liệu cookie dễ dàng sửa đổi hoặc đánh cắp khi chúng được lưu trữ ở phía client.	Dữ liệu session không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ.
Dữ liệu cookie có sẵn trong trình duyệt đến khi expired.	Sau khi đóng trình duyệt sẽ hết phiên làm việc (session)

demo

- ❑ PHP Data Objects (PDO) là một lớp truy xuất cơ sở dữ liệu cung cấp một phương pháp thống nhất để làm việc với nhiều loại cơ sở dữ liệu khác nhau.
- ❑ Khi làm việc với PDO bạn sẽ không cần phải viết các câu lệnh SQL cụ thể mà chỉ sử dụng các phương thức mà PDO cung cấp.



```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
} catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>
```

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
    $conn = new PDO("mysql:host=$servername", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "CREATE DATABASE myDBPDO";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Database created successfully<br>";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;

?>
```

```
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // sql to create table
    $sql = "CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
    )";

    // use exec() because no results are returned
    $conn->exec($sql);

    echo "Table MyGuests created successfully";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}
```

```
try {  
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);  
    // set the PDO error mode to exception  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
    $sql = "INSERT INTO MyGuests (firstname, lastname, email)  
VALUES ('John', 'Doe', 'john@example.com')";  
    // use exec() because no results are returned  
    $conn->exec($sql);  
    echo "New record created successfully";  
} catch(PDOException $e) {  
    echo $sql . "<br>" . $e->getMessage();  
}
```


PHP & PDO LẤY ID CỦA RECORD CUỐI CÙNG

```
try {  
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);  
    // set the PDO error mode to exception  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
    $sql = "INSERT INTO MyGuests (firstname, lastname, email)  
VALUES ('John', 'Doe', 'john@example.com')";  
    // use exec() because no results are returned  
    $conn->exec($sql);  
    $last_id = $conn->lastInsertId();  
    echo "New record created successfully. Last inserted ID is: " . $last_id;  
} catch(PDOException $e) {  
    echo $sql . "<br>" . $e->getMessage();  
}
```

```
try {  
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);  
    // set the PDO error mode to exception  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
  
    // prepare sql and bind parameters  
    $stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)  
VALUES (:firstname, :lastname, :email)");  
    $stmt->bindParam(':firstname', $firstname);  
    $stmt->bindParam(':lastname', $lastname);  
    $stmt->bindParam(':email', $email);
```

```
// insert a row
$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
$stmt->execute();

// insert another row
$firstname = "Mary";
$lastname = "Moe";
$email = "mary@example.com";
$stmt->execute();

// insert another row
$firstname = "Julie";
$lastname = "Dooley";
$email = "julie@example.com";
$stmt->execute();

echo "New records created successfully";
} catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
```

Câu
truy vấn

```
<?php
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests");
    $stmt->execute();

    // set the resulting array to associative
    $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
    foreach($stmt->fetchAll() as $k=>$v) {
        echo $v;
    }
} catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
?>
```

Câu
truy vấn

```
<?php
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests WHERE lastname='Doe'");
    $stmt->execute();

    // set the resulting array to associative
    $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
    foreach($stmt->fetchAll() as $k=>$v) {
        echo $v;
    }
}
catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
?>
```

Câu
truy vấn

```
<?php
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $conn->prepare("SELECT id, firstname, lastname FROM MyGuests ORDER BY lastname");
    $stmt->execute();

    // set the resulting array to associative
    $result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
    foreach($stmt->fetchAll() as $k=>$v) {
        echo $v;
    }
} catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
?>
```

```
<?php
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // sql to delete a record
    $sql = "DELETE FROM MyGuests WHERE id=3";

    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Record deleted successfully";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

?>
```

```
<?php
try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

    // Prepare statement
    $stmt = $conn->prepare($sql);

    // execute the query
    $stmt->execute();

    // echo a message to say the UPDATE succeeded
    echo $stmt->rowCount() . " records UPDATED successfully";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}
```

```
?>
```


- ✓ PHP & MySQLi INSERT
- ✓ PHP & MySQLi UPDATE
- ✓ PHP & MySQLi DELETE
- ✓ PHP & MySQLi Tìm kiếm sản phẩm
- ✓ PHP & MySQLi prepare statements



thank
you!