

BÀI 3:

Session & Cookies trong PHP

Xử lý form nhập với PHP



Mục tiêu bài học

- Cookies trong PHP trong PHP
- Session trong PHP
- Xử lý request với method POST/GET trong PHP

- Cookie là một file văn bản có kích thước nhỏ, do web server tạo ra và được nhúng tại máy khách.
- Cookies thường được dùng để nhận diện người dùng. Mỗi khi cùng một máy tính request một trang web tại trình duyệt, request sẽ được gửi kèm cookie để nhận diện máy khách.
- PHP cho phép tạo và lấy về giá trị của cookie.
- Để tạo một cookie, ta gọi hàm `setcookie()`.
 - Chú ý: Chỉ có tham số `name` là bắt buộc, các tham số khác là tùy chọn.
- Cú pháp:

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

➤ Ví dụ

```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>

<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```

- Để cập nhật cookie, ta chỉ cần thiết lập lại nội dung cookie bằng cách gọi hàm `setcookie()`

```
<?php
$cookie_name = "user";
$cookie_value = "Alex Porter";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
?>

<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```


- Để xóa cookie, ta sử dụng hàm `setcookie()` với ngày hết hạn là một thời điểm trong quá khứ.

```
<?php
// set the expiration date to one hour ago
setcookie("user", "", time() - 3600);
?>
<html>
<body>

<?php
echo "Cookie 'user' is deleted.";
?>

</body>
</html>
```

- Session là một phương pháp lưu trữ thông tin thông qua các biến để sử dụng trong suốt một phiên làm việc của người dùng.
- Các biến session có thể được truy cập qua nhiều trang web. Không giống với cookie, thông tin trong session không được lưu trữ trên máy khách.
- Session là khoảng thời gian từ lúc người dùng đăng nhập vào hệ thống cho đến lúc người dùng đăng xuất khỏi hệ thống.
- Trên môi trường internet sẽ có một vấn đề phát sinh: web server không biết bạn là ai, bạn làm gì, bởi lẽ giao thức HTTP không duy trì trạng thái của máy khách.
- Các biến session sẽ giải quyết vấn đề này bằng cách lưu trữ thông tin người dùng qua nhiều trang (VD: username, password v.v...). Mặc định các biến session sẽ tồn tại cho đến khi người dùng đóng trình duyệt.
- Các biến session lưu trữ thông tin về một người dùng duy nhất variables hold information about one single user, và tồn tại trong tất cả các trang web của website.

- Một session sẽ được bắt đầu với hàm `session_start()`.
- Để truy cập đến các biến session, ta sử dụng một biến toàn cục của PHP: `$_SESSION`.
- **Chú ý:** Hàm `session_start()` phải xuất hiện ở dòng lện đầu tiên trong trang web, trước các thẻ HTML khác.
- Ví dụ:

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```


- Để lấy giá trị của các biến session, ta sử dụng biến toàn cục \$_SESSION.
- Ví dụ:

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . "<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>

</body>
</html>
```

- Để xóa tất cả các biến session, ta gọi hàm `session_unset()`
- Để hủy session, ta gọi hàm `session_destroy()`
- Ví dụ:

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// remove all session variables
session_unset();

// destroy the session
session_destroy();
?>

</body>
</html>
```

- Giảng viên demo về session/cookie.

- Có 2 phương thức request để trình duyệt ở máy khách có thể gửi thông tin lên web server.
 - GET Method
 - POST Method
- Trước khi trình duyệt gửi thông tin, trình duyệt sẽ encodes thông tin bằng cách sử dụng một scheme được gọi là URL encoding. Trong scheme này, dữ liệu sẽ được gửi lên server bằng cách cặp name/value theo định dạng name=value.
- Mỗi cặp dữ liệu sẽ được phân tách với nhau bởi ký tự (&).

- Phương thức GET sẽ gửi thông tin người dùng lên cùng với URL của trang. Thông tin và tên trang sẽ được phân tách bởi dấu (?).
- Phương thức GET có một hạn chế là chỉ giới hạn tối đa là 1024 ký tự.
- Chú ý: Đừng sử dụng phương thức GET nếu ta muốn gửi thông tin mật khẩu hay những thông tin nhạy cảm khác lên server.
- Phương thức GET không thể được dùng để gửi dữ liệu nhị phân (binary data), chẳng hạn như ảnh hay các văn bản word lên server.
- PHP cung cấp mảng associative **\$_GET** để truy cập đến thông tin được gửi lên server với phương thức GET.

```
http://www.test.com/index.htm?name1=value1&name2=value2
```


GET Method

➤ Ví dụ:

```
<?php
    if( $_GET["name"] || $_GET["age"] ) {
        echo "Welcome ". $_GET['name']. "<br />";
        echo "You are ". $_GET['age']. " years old.";

        exit();
    }
?>
<html>
    <body>

        <form action = "<?php $_PHP_SELF ?>" method = "GET">
            Name: <input type = "text" name = "name" />
            Age: <input type = "text" name = "age" />
            <input type = "submit" />
        </form>

    </body>
</html>
```

- Phương thức POST sẽ truyền thông tin thông qua HTTP headers. Thông tin sẽ được encoded và được đặt vào trong một header được gọi là QUERY_STRING.
 - Phương thức POST không hạn chế kích thước dữ liệu được gửi lên server.
 - Phương thức POST có thể được sử dụng để gửi mã ASCII cũng như dữ liệu nhị phân.
 - PHP cung cấp mảng associative **\$_POST** để truy cập đến thông tin được gửi lên server với phương thức POST.

POST Method

➤ Ví dụ:

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
    Name: <input type="text" name="fname">
    <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = $_POST['fname'];
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>

</body>
</html>
```

POST Method

➤ Ví dụ:

```
<!DOCTYPE HTML>
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

```
<html>
<body>

Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>

</body>
</html>
```

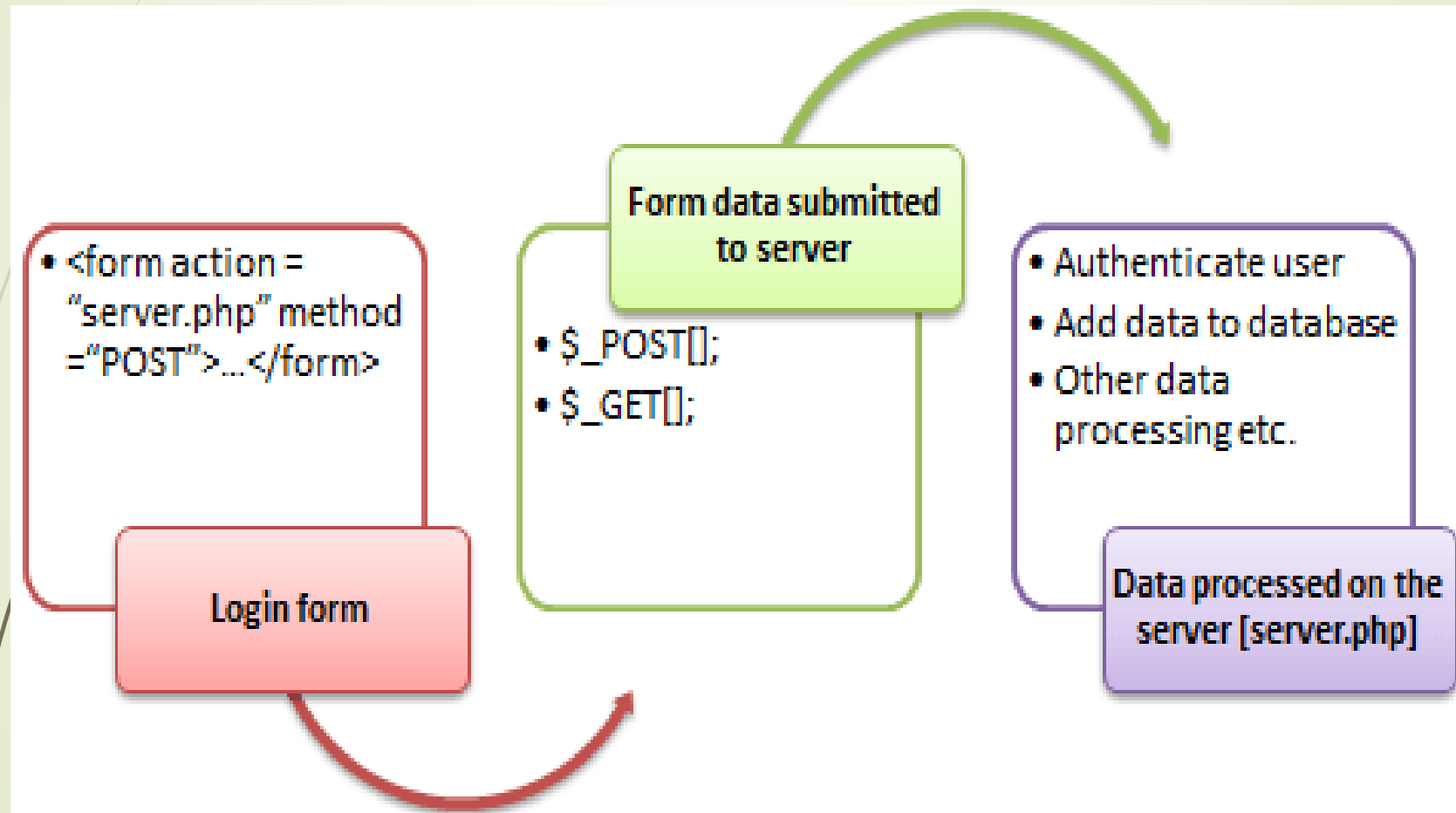
➤ Kết quả:

Name: Tony Nguyen
E-mail: tony@gmail.com
Submit

Welcome Tony Nguyen
Your email address is: tony@gmail.com

POST Method

➤ Minh họa phương thức POST/GET



PHP \$_REQUEST

- PHP cung cấp biến toàn cục \$_REQUEST để lấy dữ liệu của người dùng, được sử dụng với cả 2 phương thức POST và GET.

- Ví dụ:

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
    Name: <input type="text" name="fname">
    <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // collect value of input field
    $name = $_REQUEST['fname'];
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>

</body>
</html>
```

- \$GLOBALS là một biến toàn cục đặc biệt của PHP được dùng để truy cập đến tất cả các biến toàn cục từ bất cứ đâu trong các đoạn mã kịch bản PHP.
- PHP lưu trữ tất cả các biến toàn cục trong một mảng \$GLOBALS[index]. Chỉ số *index* sẽ lưu trữ tên biến.
- Ví dụ:

```
<?php
$x = 75;
$y = 25;

function addition() {
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

addition();
echo $z;
?>
```

- \$_SERVER là một biến toàn cục của PHP chứa thông tin về headers, paths, và vị trí của script.
- Ví dụ:

```
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

- Giáo viên demo phương thức POST/GET cho sinh viên.

Tổng kết

- Cookies trong PHP trong PHP
- Session trong PHP
- Xử lý request với method POST/GET trong PHP