



CHƯƠNG 3: LẬP TRÌNH WEB VỚI PHP

ThS. Lê Tuấn Anh

Tel: 034.619.6934

Email: anhlt@dainam.edu.vn

CHƯƠNG 3: LẬP TRÌNH WEB VỚI PHP

1

Giới thiệu ngôn ngữ PHP

2

Cú pháp PHP

3

Hàm và đối tượng

4

Mảng

5

Xử lý form trong PHP

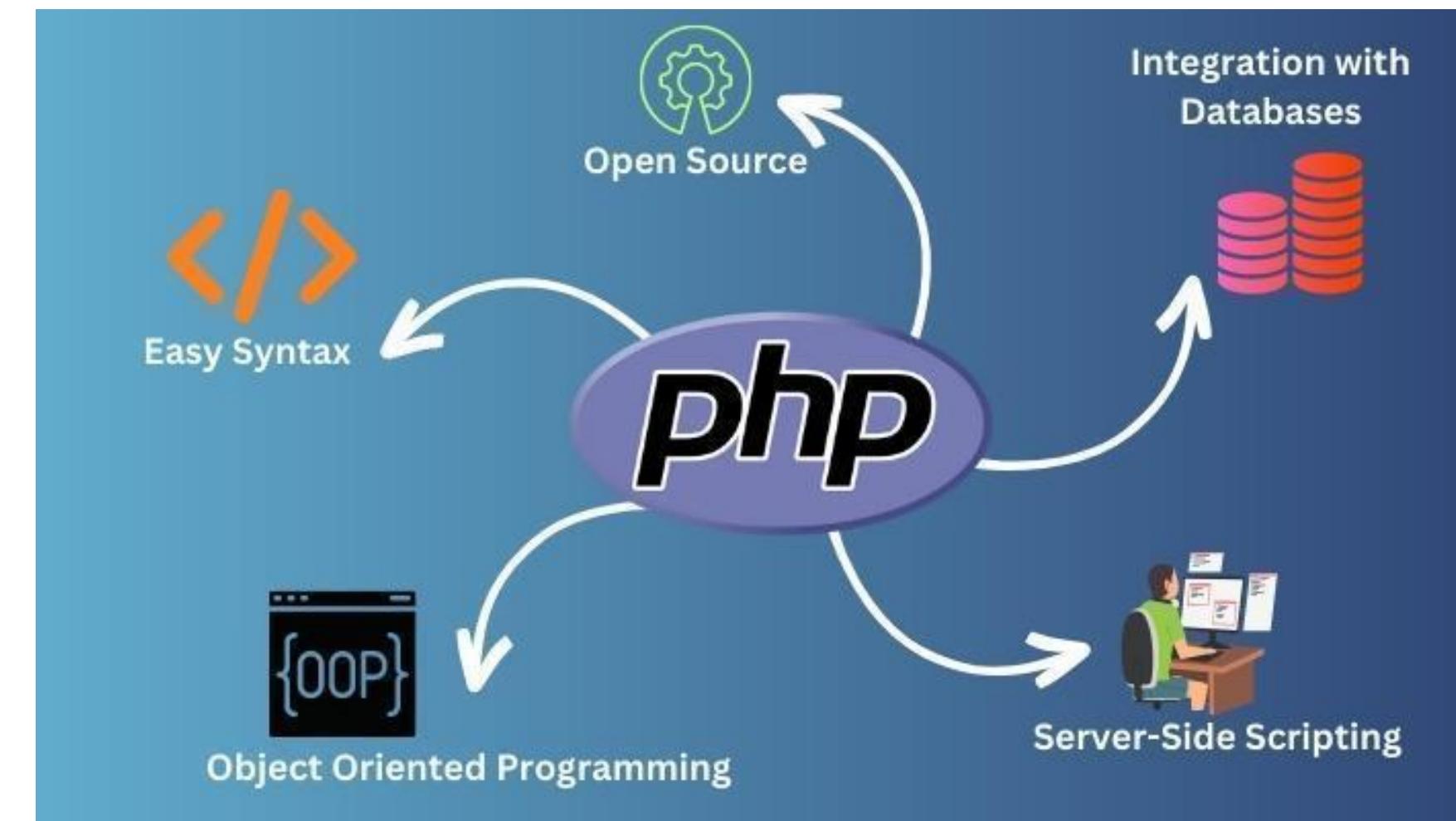
6

Chèn tập tin trong PHP

7

SESSION và COOKIE

PHP: **Hypertext Preprocessor**, thường được viết tắt thành PHP là một ngôn ngữ lập hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở. Nó rất thích hợp với web và có thể dễ dàng nhúng vào trang HTML.



❖ Các kiểu dữ liệu:

```
<?php

$student_id = 123;                      // integer
$student_name = "Nguyen Van A";           // string
$gpa = 3.5;                             // float
$is_active = true;                        // boolean
$courses = ["Math", "Physics"];           // array
$student = ["id" => 1, "name" => "A", "gpa" => 3.9]; // associative array

?>
```

❖ Biến, Hằng, Toán tử:

```
<?php  
  
$student_age = 20; // Biến  
define('MAX_GPA', 4.0); // Hằng  
  
// Toán tử  
  
$gpa1 = 3.5;  
$gpa2 = 3.8;  
$average = ($gpa1 + $gpa2) / 2; // Toán tử số học  
?>
```

❖ Cấu trúc điều khiển:

```
<?php

$student = ["id" => 1, "name" => "A", "gpa" => 3.9];

// Duyệt qua danh sách sinh viên

if ($student["gpa"] >= 3.2) {

    echo $student["name"] . " là sinh viên giỏi<br>";

} else {

    echo $student["name"] . " chưa đạt sinh viên giỏi<br>";

}

?>
```

❖ Hàm:

```
<?php  
  
function myMessage() {  
    echo "Hello world!";  
}  
  
myMessage();  
  
?>
```

```
<?php  
  
function showStudentInfo($name, $age) {  
    echo "Tên sinh viên: $name<br>";  
    echo "Tuổi: $age";  
}  
  
// Sử dụng hàm  
showStudentInfo("Nguyen Van An", 20);  
  
?>
```

Đối tượng:

```
<?php
class Student {
    // Thuộc tính
    public $id;
    public $name;
    public $gpa;
    // Phương thức khởi tạo
    public function __construct($id, $name, $gpa) {
        $this->id = $id;
        $this->name = $name;
        $this->gpa = $gpa;
    }
    // Phương thức kiểm tra sinh viên giỏi
    public function isExcellent() {
        return $this->gpa >= 3.2;
    }
    // Phương thức hiển thị thông tin
    public function display() {
        echo "ID: $this->id - Tên: $this->name - GPA: $this->gpa";
        if ($this->isExcellent()) {
            echo " (Sinh viên giỏi)";
        }
        echo "<br>";
    }
}
```

```
// Tạo đối tượng sinh viên
$student1 = new Student(1, "Nguyen Van A", 3.5);
$student2 = new Student(2, "Le Thi B", 2.7);
// Sử dụng phương thức đối tượng
$student1->display();
$student2->display();
?>
```

MẢNG

Mảng dùng chỉ số (Indexed Array)

```
<?php  
  
$studentNames = ["An", "Bình", "Cường"];  
  
echo $studentNames[0];  
  
// Kết quả: An  
  
?>
```

Mảng kết hợp (Associative Array)

```
<?php  
  
$student = [  
    "id" => 1,  
    "name" => "Nguyen Van An",  
    "age" => 20  
];  
  
echo $student["name"];  
  
// Kết quả: Nguyen Van An  
  
?>
```

Duyệt mảng

```
<?php  
  
$studentNames = ["An", "Bình", "Cường"];  
  
foreach ($studentNames as $name) {  
  
    echo $name . "<br>";  
  
}  
  
?>
```

```
<?php  
  
$names = ["An", "Bình", "Cường"];  
  
for ($i = 0; $i < count($names); $i++) {  
  
    echo $names[$i] . "<br>";  
  
}  
  
?>
```

Duyệt mảng kết hợp

```
<?php  
  
$student = [  
  
    "id" => 1,  
  
    "name" => "Nguyen Van An",  
  
    "age" => 20  
  
];  
  
foreach ($student as $key => $value) {  
  
    echo $key . ":" . $value . "<br>";  
  
}  
  
?>
```

Mảng hai chiều

```
<?php  
  
$students = [  
    ["id" => 1, "name" => "An", "age" => 20],  
    ["id" => 2, "name" => "Bình", "age" => 21],  
    ["id" => 3, "name" => "Cường", "age" => 22]  
];  
  
// Lấy tên sinh viên thứ hai  
echo $students[1]["name"]; // Kết quả: Bình  
?>
```

```
foreach ($students as $sinhvien) {  
    echo "ID: " . $sinhvien["id"] . ",  
    Tên: " . $sinhvien["name"] . "<br>";  
}
```

Duyệt mảng

Các hàm thao tác với mảng

```
<?php  
  
// Khởi tạo mảng sinh viên ban đầu  
$students = ["An", "Bình"];  
  
// 1. Thêm phần tử vào cuối mảng  
array_push($students, "Cường");  
  
// 2. Xóa phần tử cuối và lấy ra giá trị  
$last = array_pop($students);  
  
// 3. Xóa phần tử đầu tiên và lấy ra giá trị  
$first = array_shift($students);  
  
// 4. Thêm phần tử vào đầu mảng  
array_unshift($students, "Dũng");  
  
echo "Phần tử cuối đã xóa: $last<br>";  
echo "Phần tử đầu đã xóa: $first<br>";  
echo "Danh sách hiện tại: ";  
print_r($students); // In ra toàn bộ mảng  
?>
```

```
<?php  
  
// 5. Gộp 2 mảng sinh viên lại thành 1  
$a1 = ["An", "Bình"];  
$a2 = ["Cường", "Nam"];  
$all_students = array_merge($a1, $a2);  
  
// 6. Lấy ra 2 phần tử bắt đầu từ vị trí 1  
$subset = array_slice($all_students, 1, 2);  
  
// 7. Tìm vị trí của phần tử trong mảng  
$index = array_search("Cường", $all_students);  
  
// 8. Kiểm tra xem 1 phần tử có trong mảng không  
$has_dung = in_array("Nam", $all_students);  
  
echo "<br>Lấy ra 2 phần tử (từ vị trí 1): ";  
print_r($subset);  
echo "<br>Vị trí của 'Cường' là: $index";  
echo "<br>Có 'Nam' trong mảng không? " . ($has_dung ? "Có" : "Không");  
?>
```

Các hàm thao tác với mảng

```
<?php
$students = ["Cường", "An", "Bình"];
sort($students);
print_r($students);
// Kết quả: Array ( [0] => An [1] =>
Bình [2] => Cường )
?>
```

Sắp xếp mảng tăng dần

```
<?php
$students = ["Cường", "An", "Bình"];
$list = implode(", ", $students);
echo $list;
// Kết quả: Cường, An, Bình
?>
```

Chuyển mảng thành chuỗi

XỬ LÝ FORM TRONG PHP

```
<form method="POST" action="">  
  
Tên sinh viên: <input type="text" name="name" required><br>  
Tuổi: <input type="number" name="age" required><br>  
<input type="submit" value="Gửi">  
</form>
```

```
<?php  
  
// Xử lý form  
  
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
    // Lấy dữ liệu từ form  
    $name = $_POST['name'] ?? '';  
    $age = $_POST['age'] ?? '';  
    echo "<h3>Kết quả nhận được:</h3>";  
    echo "Tên sinh viên: <b>$name</b><br>";  
    echo "Tuổi: <b>$age</b><br>";  
    echo '<a href="' .  
        htmlspecialchars($_SERVER["PHP_SELF"]) . '">Quay  
lại nhập mới</a>';  
}  
?>
```

index.php

```
<?php  
if ($_SERVER['REQUEST_METHOD'] === 'POST') {  
    header("Location: chuyen_huong.php");  
    exit();  
}  
?  
  
<form method="POST" action="">  
    <button type="submit">Nhấn vào đây để chuyển  
    hướng</button>  
</form>
```

chuyen_huong.php

```
<?php  
echo "Chuyển hướng thành công!";  
?>
```

trang1.php

```
<a href="trang2.php?name=An&age=20">  
Xem thông tin sinh viên  
</a>
```

trang2.php

```
<?php  
$name = $_GET['name'];  
$age = $_GET['age'];  
echo "Tên: $name<br>Tuổi: $age";  
?>
```

Truyền qua phương thức GET

trang1.php

```
<form action="trang2.php" method="POST">  
    Tên: <input type="text" name="name"><br>  
    Tuổi: <input type="number" name="age"><br>  
    <input type="submit" value="Gửi">  
</form>
```

trang2.php

```
<?php  
$name = $_POST['name'] ?? '';  
$age = $_POST['age'] ?? '';  
echo "Tên: $name<br>Tuổi: $age";  
?>
```

Ví dụ: form gửi từ trang 1 sang trang 2 sử dụng POST

CHÈN TẬP TIN TRONG PHP

- Tính nhất quán: nên tách các phần dùng chung như kết nối CSDL, header, menu... ra file riêng để dễ tái sử dụng, bảo trì.

```
<?php
// Bắt buộc phải có, lỗi dừng chương trình nếu
// không tìm thấy file
require "db_connect.php";
echo "Kết nối CSDL thành công!";
?>
```

require()

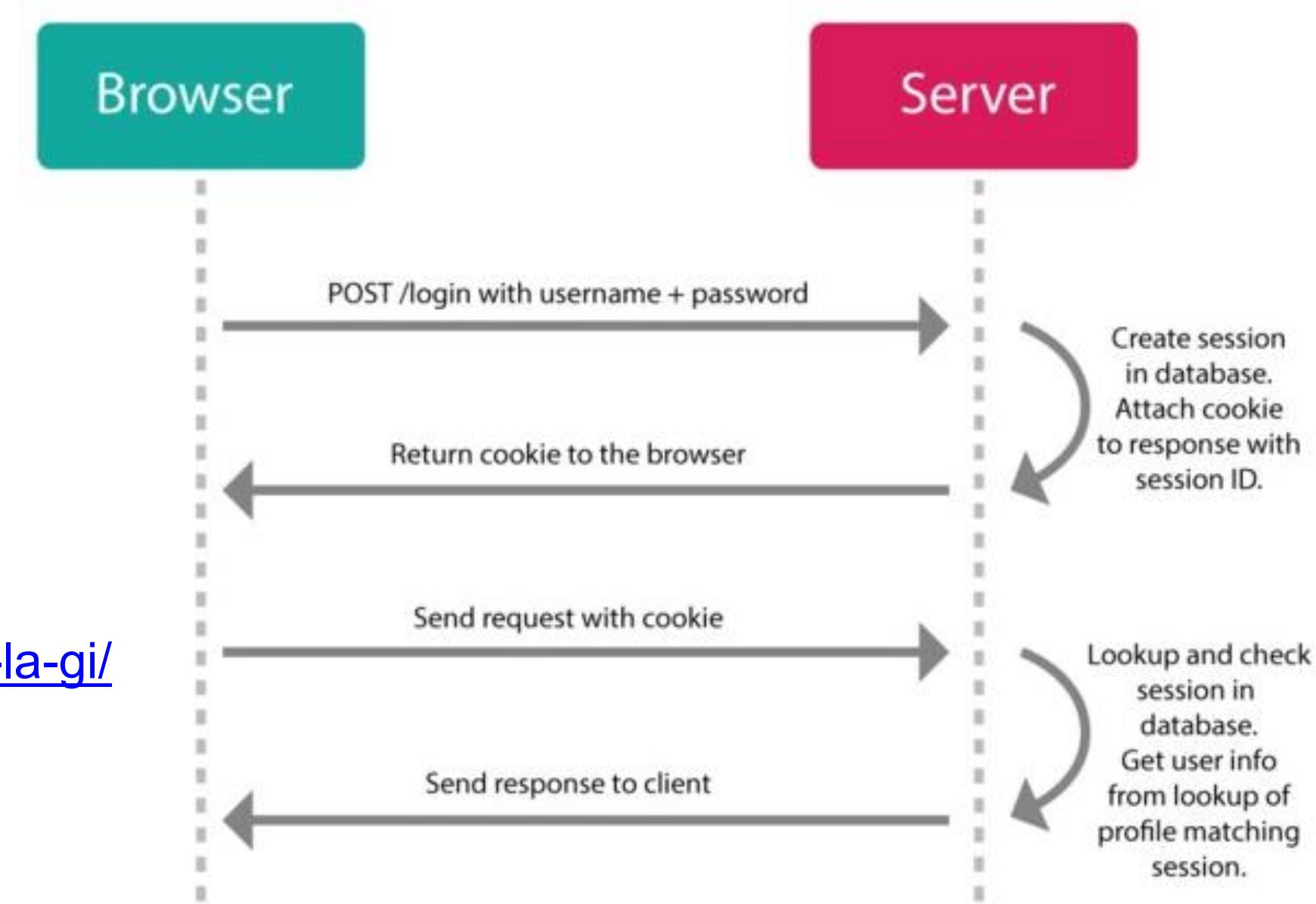
```
<?php
// Nếu không có file chỉ cảnh báo, chương trình
// vẫn chạy tiếp
include "menu.php";
echo "Chào mừng đến hệ thống quản lý sinh viên!";
?>
```

include()

- require_once: Chèn nội dung file chỉ một lần duy nhất, nếu đã chèn rồi sẽ không chèn lại lần nữa
- include_once: Chèn nội dung của một file PHP vào file hiện tại, chỉ chèn một lần duy nhất (nếu đã được chèn trước đó thì sẽ bỏ qua)

SESSION VÀ COOKIE

- Session là một phiên làm việc tạm thời giữa người dùng và server, dùng để lưu trữ dữ liệu riêng cho từng người dùng (như tên đăng nhập, quyền truy cập,...)
- Dữ liệu session được lưu trên máy chủ, không bị lộ cho người dùng.



<https://topdev.vn/blog/session-la-gi-cookie-la-gi/>

❖ Chu kỳ sống của session:

- Một session bắt đầu khi client gửi request đến server, nó tồn tại xuyên suốt từ trang này đến trang khác trong ứng dụng web và chỉ kết thúc khi hết thời gian timeout hoặc bạn đóng ứng dụng. Giá trị của session sẽ được lưu trong một file trên server
- Có thể xóa thủ công bằng hàm `session_destroy()`.

❖ Khai báo khởi tạo Session:

```
<?php  
session_start(); // Bắt đầu session (luôn đặt ở đầu file)  
?>
```

Gán giá trị cho biến Session

```
<?php  
  
$_SESSION[ 'name' ] = "Nguyễn Văn A";  
  
$_SESSION[ 'role' ] = "Sinh viên";  
  
?>
```

Cập nhật giá trị Session

```
<?php  
  
$_SESSION[ 'name' ] = "Trần Thị B";  
  
?>
```

Lấy giá trị từ Session

```
<?php  
  
echo $_SESSION[ 'name' ];  
  
?>
```

Xoá một biến Session

```
<?php  
  
unset($_SESSION[ 'name' ]);  
  
?>
```

❖ Xoá toàn bộ Session:

```
session_unset();      // Xóa tất cả biến session  
session_destroy();   // Hủy phiên làm việc hoàn toàn
```

❖ Ứng dụng thực tế của Session:

- Lưu thông tin người dùng sau khi đăng nhập.
- Giao diện theo vai trò: quản trị viên, sinh viên...
- Duy trì giỏ hàng trong website thương mại điện tử.
- Quản lý trạng thái học kỳ, trạng thái đăng ký tín chỉ...

login.php

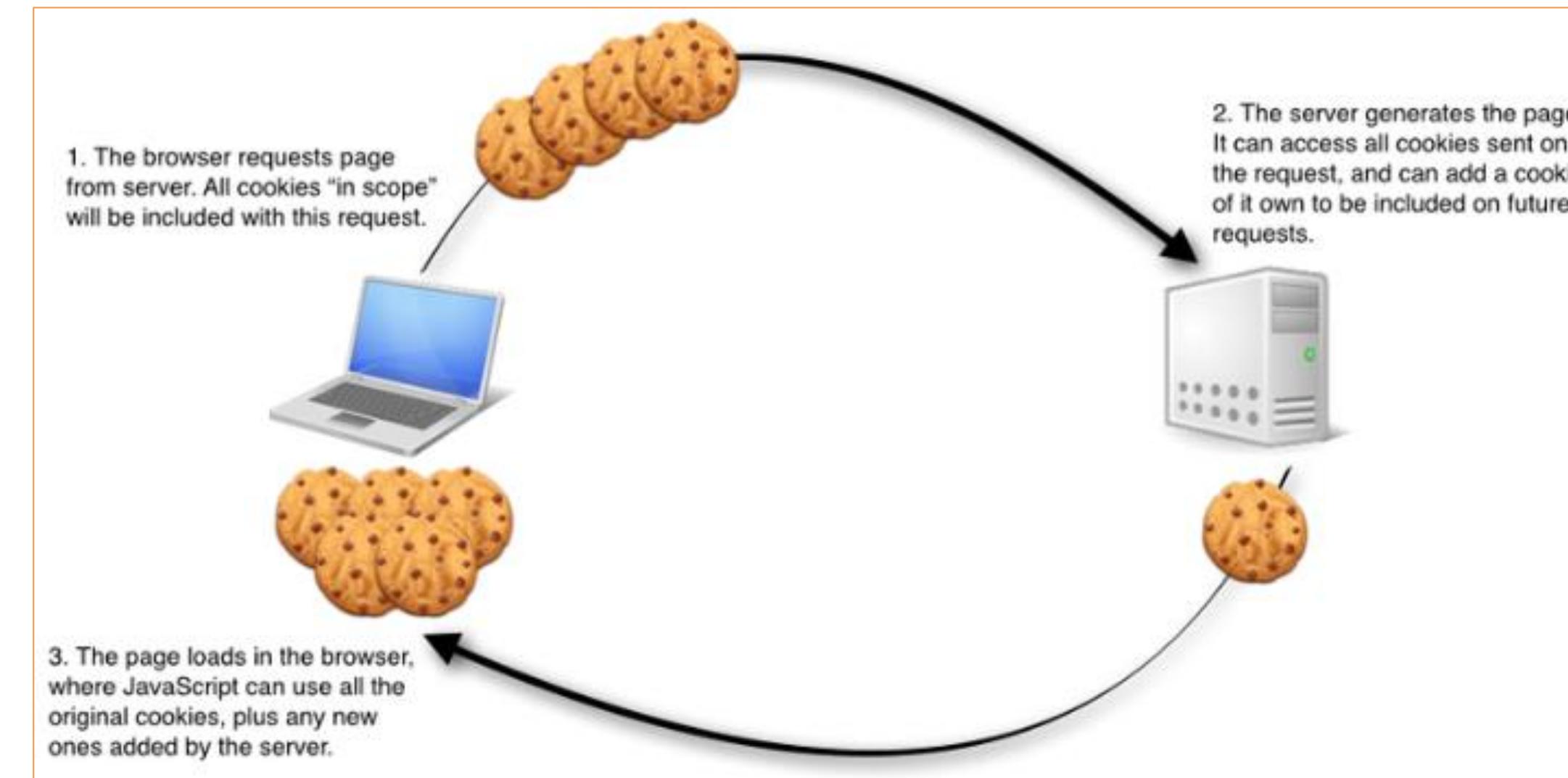
```
<?php  
session_start();  
$_SESSION['user'] = "sinhvien01";  
header("Location: welcome.php");  
exit();  
?>
```

welcome.php

```
<?php  
session_start();  
echo "Xin chào: " . $_SESSION['user'];  
?>
```

Ví dụ: Session đăng nhập đơn giản

- Cookie là cách lưu trữ dữ liệu trên trình duyệt của người dùng (client)
- Có thể dùng để ghi nhớ đăng nhập, lựa chọn ngôn ngữ, theme...



<https://topdev.vn/blog/session-la-gi-cookie-la-gi/>

- Tạo Cookie: setcookie("username", "sinhvien01", time() + 3600); // tồn tại 1 tiếng
- Lấy giá trị Cookie: echo \$_COOKIE["username"];
- Xoá Cookie: setcookie("username", "", time() - 3600);
// Đặt thời gian hết hạn về quá khứ

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)







Thank
you