

Mảng trong PHP

```
// Tạo mảng
$a = array();

// Tạo mảng có giá trị ban đầu
$a = array(1, 2, 3);
// or
$a = array("a" => 1, "b" => 2);

// Thêm phần tử vào mảng
$a[] = "something";
// or
$a["a"] = 1;
$a["b"] = 2;

// Lấy giá trị tại một index
$a[0]

// Vòng lặp Foreach
foreach($myArray as $key => $value) {
    // key là index
    // value là giá trị tại index đó
}

// Mảng đa chiều
$student = array();
$student["001"] = array("name" => "John", "sex" => 1, "score" => array(4, 5, 6));

// Lấy các value từ mảng a gán qua mảng b
$b = array_values($a);

// Lấy các index (key) từ mảng a gán qua mảng b
$b = array_keys($a);

// Kiểm tra mảng rỗng
empty($a)

// Đếm số phần tử xuất hiện trong mảng
count($a)

// Loại bỏ phần tử đầu tiên của mảng và trả về giá trị đó
$a = array("a" => 1, "b" => 2);
$head = array_shift($a);

// Loại bỏ phần tử cuối cùng của mảng và trả về giá trị đó
$a = array("a" => 1, "b" => 2);
$tail = array_pop($a);

// Loại bỏ những phần tử trùng nhau trong mảng và trả về mảng mới
$a = array(1, 2, 1, 1, 3);
$newArray = array_unique($a);
```

```
// Xóa phần tử tại một index trong mảng
$a = array(1, 2, 1, 1, 3);
unset($a[2]); // xóa giá trị tại index = 2
unset($a[0], $a[1]); // xóa tại 2 index 0 và 1
unset($a); // xóa hết mảng

// Thêm phần tử vào cuối mảng, trả về số lượng phần tử của mảng mới
$a = array(1, 2, 1, 1, 3);
$leng = array_push($a, 4, 5, 6);

// Thêm phần tử vào đầu mảng, trả về số lượng phần tử của mảng mới
$a = array(1, 2, 1, 1, 3);
$leng = array_unshift($a, 4, 5, 6);

// Đảo ngược mảng sau đó trả về một mảng mới
$a = array(1, 2, 3);
$newArr = array_reverse($a);

// Swap key và value của mảng sau đó trả về mảng mới, vd (key: 1, value: 2) => (key: 2, value: 1)
$newArr = array_flip($a);

// Tổng mảng
$sum = array_sum($a);

// Max/min mảng
$max = max($a)/min($a);
```

Chuỗi trong PHP

```

// Chuyển số thành chuỗi
$b = strval(123);

// Chuỗi thành số
$a = (int)"123";

// Chiều dài chuỗi
$len = strlen($string);
// chuỗi có utf-8 sài
mb_strlen();

// Đảo ngược chuỗi
$s = "I love you"
$res = strrev($s);

// Lấy các kí tự từ vị trí thứ x và lấy y kí tự
$res = substr($s, x, y);

// Lấy 3 kí tự cuối chuỗi
$res = substr($s, -3, 3); // lấy ra dc chữ you

// Xóa các khoảng trắng, \n thừa bên trái chuỗi và trả về chuỗi mới
$res = ltrim("  sadd");

// Xóa chuỗi xuất hiện lần đầu tiên từ bên trái chuỗi từ kí tự 0
$res = ltrim("you you you", "you") // output: you you
$res = ltrim("I you you", "you") // Output: I you you

// Xóa chuỗi xuất hiện lần đầu tiên từ bên phải chuỗi
rtrim() // tương tự ltrim

// Xóa bỏ các kí tự bên phải và trái chuỗi
trim(chuỗi, chuỗi muốn xóa) // tương tự ltrim và rtrim kết hợp vs nhau

// Chuyển các từ trong chuỗi thành mảng cách nhau bởi kí tự x
$s = "I love you";
$res = explode(x, $s);
// $res = {"I", " ", "love", " ", " ", "you"}

// Chuyển mảng thành chuỗi cách nhau bởi kí tự x
$res = implode(x, $mảngChứaCácTừ);

// Lặp chuỗi
$res = str_repeat(chuỗi cần lặp, số lần);

// trả về mã ascii của kí tự truyền vào
ord("a");

// trả về kí tự dựa vào mã ascii
chr(45);

```

```
// Phân tích chuỗi URL
$str = "name=Peter&age=23";
parse_str($str);
echo $name; // output là peter
// nếu chuyển vào mảng vs format là key: value
parse_str($str, mảng);
```

Session

Nhớ đặt hàm **session start** trước thẻ `<html>`

```
// Tại file 01.php
<?php
    session_start();

    // Lưu giá trị vào session
    $_SESSION['name'] = "Nguyễn Văn A";
?>

// Tại file 02.php
<?php
    session_start();

    // Đọc giá trị từ session
    $name = $_SESSION['name'];
    echo $name;

    // output: Nguyễn Văn A
?>
```

FORM

VÍ DỤ 1

```
# index.html

<!DOCTYPE html>

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <title>Login Form</title>
  <link rel="stylesheet" href="css/style.css">
</head>

<body>
  <section class="container">
    <div class="login">
      <h1>Login to Web App</h1>
      <form method="post" action="process.php" name="main-form">
        <p><input type="text" name="login" value="" placeholder="Username or Email"></p>
        <p><input type="password" name="password" value="" placeholder="Password"></p>
        <p class="remember_me">
          <label>
            <input type="checkbox" name="remember_me" id="remember_me">
            Remember me on this computer
          </label>
        </p>
        <p class="submit"><input type="submit" name="commit" value="Login"></p>
      </form>
    </div>

    <div class="login-help">
      <p>Forgot your password? <a href="index.html">Click here to reset it</a>.</p>
    </div>
  </section>
</body>

</html>
```

```
# process.php
```

```
<?php
```

```
    /*
    // trường hợp với method là get
    echo $_GET["login"]."<br>";
    echo $_GET["password"]."<br>";
    */
    /*
    // trường hợp với method là post
    echo $_POST["login"]."<br>";
    echo $_POST["password"]."<br>";
    */
    /*
    // nếu dùng Request thì áp dụng dc cho cả 2 method là get và post
    echo $_REQUEST["login"]."<br>";
    echo $_REQUEST["password"]."<br>";
    */
    $userName = $_POST["login"];
    $passWord = $_POST["password"];
    /*
    // kiểm tra coi lấy dc thông tin user hay ko
    echo $userName. " " . $passWord;
    */
    $res = ($userName == "admin" && $passWord == "123456" ? "Đăng nhập thành công." : "Đăng nhập t
    echo $res;
```

VÍ DỤ 2

```
# pagination.php
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>Pagination TNT style</title>
```

```
<style type="text/css">
```

```
    #tnt_pagination {
        display: inline-block;
        text-align: left;
        height: 22px;
        line-height: 21px;
        clear: both;
        padding-top: 3px;
        font-family: Arial, Helvetica, sans-serif;
        font-size: 12px;
        font-weight: normal;
```

```
    }
```

```
    #tnt_pagination a:link,
```

```
    #tnt_pagination a:visited {
        padding: 7px;
        padding-top: 2px;
        padding-bottom: 2px;
        border: 1px solid #EBEBEB;
        margin-left: 2px;
        text-decoration: none;
        background-color: #F5F5F5;
        color: #0072bc;
        width: 22px;
        font-weight: normal;
```

```
    }
```

```
    #tnt_pagination a:hover {
        background-color: #DDEEFF;
        border: 1px solid #BBDDFF;
        color: #0072BC;
```

```
    }
```

```
    #tnt_pagination .active_tnt_link {
        padding: 7px;
        padding-top: 2px;
        padding-bottom: 2px;
        border: 1px solid #BBDDFF;
        margin-left: 2px;
        text-decoration: none;
        background-color: #DDEEFF;
        color: #0072BC;
        cursor: default;
```

```
    }
```

```
    #tnt_pagination .disabled_tnt_pagination {
        padding: 7px;
```



```

padding-top: 2px;
padding-bottom: 2px;
border: 1px solid #EBEBEB;
margin-left: 10px;
text-decoration: none;
background-color: #F5F5F5;
color: #D7D7D7;
cursor: default;
}
.content {
margin: 10px auto;
width: 800px;
border: 1px solid #dfd;
padding: 20px;
text-align: center;
}
</style>
</head>

<body>
<div class="content">
<div id="tnt_pagination">
<span class="disabled_tnt_pagination"><<Prev</span>
<a href="pagination.php?page=1">1</a>
<a href="pagination.php?page=2">2</a>
<a href="pagination.php?page=3">3</a>
<a href="pagination.php?page=4">4</a>
<a href="pagination.php?page=5">5</a>
<a href="pagination.php?page=6">6</a>
<a href="pagination.php?page=7">7</a>
<a href="pagination.php?page=8">8</a>
<a href="pagination.php?page=9">9</a>
<a href="pagination.php?page=10">10</a>
<a href="#next">Next >></a>
</div>

<?php
// kiểm tra nếu tồn tại var page trên URL thì mới thực hiện
$page = (isset($_GET["page"]) == true ? $_GET["page"] : 0);
?>

<p class="result">Bạn đang ở trang thứ <?php echo $page?></p>
</div>
</body>

</html>

```

VÍ DỤ 3

```
# index.php
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link type="text/css" href="style.css" rel="stylesheet" media="screen">
    <title>Mô phỏng máy tính điện tử</title>
</head>
<body>
<?php
    $n1 = $n2 = $ope = $res = "";
    if (isset($_POST["number1"]) && isset($_POST["number2"]) && isset($_POST["calculate"])) {
        $n1 = $_POST["number1"];
        $n2 = $_POST["number2"];
        $ope = $_POST["calculate"];
        $flag = true;
        if (is_numeric($n1) && is_numeric($n2)) {
            switch ($ope) {
                case "+":
                    $res = $n1 + $n2;
                    break;
                case "-":
                    $res = $n1 - $n2;
                    break;
                case "x":
                case "*":
                    $res = $n1 * $n2;
                    break;
                case ":":
                case "/":
                    $res = $n1 / $n2;
                    break;
                case "%":
                    $res = $n1 % $n2;
                    break;
                default:
                    $res = "Toán tử không hợp lệ.";
                    $flag = false;
            }
        }
        else {
            $res = "Giá trị nhập vào không hợp lệ";
            $flag = false;
        }
    }
?>

<div class="content">
    <h1>Mô phỏng máy tính điện tử</h1>
```

```
<form action="#" method="post" name="main-form">
  <div class="row">
    <span>Số thứ nhất: </span>
    <input type="text" name="number1" value="<?php echo $n1;?>">
  </div>
  <div class="row">
    <span>Phép toán: </span>
    <input type="text" name="calculate" value="<?php echo $ope;?>">
  </div>
  <div class="row">
    <span>Số thứ hai: </span>
    <input type="text" name="number2" value="<?php echo $n2;?>">
  </div>
  <div class="row">
    <input type="submit" name="submit" value="Xem kết quả">
  </div>
  <div class="row">
    <?php
      echo "<p>";
      if ($flag == true){
        echo "Kết quả: " . $n1 . " " . $ope , " " . $n2 . " = ";
      }
      echo $res . "</p>";
    ?>
  </div>
</form>
</div>
</body>
</html>
```

```

# style.css

* {
    margin: 0px;
    padding: 0px;
}

.content {
    margin: 20px auto;
    width: 600px;
    border: 1px solid #999;
    padding: 10px;
}

.content h1 {
    color: red;
    text-align: center;
}

.content div.row {
    margin-top: 20px;
}

.content div.row span {
    display: inline-block;
    width: 255px;
    text-align: right;
}

.content div.row input[type=text] {
    padding: 3px 5px;
}

.content div.row input[type=submit] {
    padding: 3px 5px;
    display: block;
    margin: 0px auto 20px auto;
}

.content div.row p {
    font-weight: bold;
    font-size: 20px;
    text-align: center;
}

```

VÍ DỤ 4

```
# index.php
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link type="text/css" href="style.css" rel="stylesheet" media="screen">
    <title>Lấy chòm sao</title>
</head>
<body>
<?php
    $day = $month = "";
    $image = $name = $time = "";
    $flag = true;
    if (isset($_POST["day"]) && isset($_POST["month"])) {
        $day = $_POST["day"];
        $month = $_POST["month"];
        if (is_numeric($day) && is_numeric($month)) {
            switch ($month) {
                case 1:
                    if ($day <= 19) {
                        $image = "capricorn";
                        $name = "Ma Kết";
                        $time = "23/12 - 19/01";
                    }
                    if ($day >= 20) {
                        $image = "aquarius";
                        $name = "Bảo Bình";
                        $time = "20/01 - 19/02q";
                    }
                    break;
                case 2:
                    if ($day <= 19) {
                        $image = "aquarius";
                        $name = "Bảo Bình";
                        $time = "20/01 - 19/02q";
                    }
                    if ($day >= 20) {
                        $image = "pisces";
                        $name = "Song Ngư";
                        $time = "20/02 - 21/03";
                    }
                    break;
                // còn lại tương tự cho các case tử [3, 12]
                default:
                    $flag = false;
            }
        }
    }
}
```

```

?>
<div class="content">
    <h1>Bạn thuộc chòm sao gì?</h1>

    <form action="#" method="post" name="main-form">
        <div class="row">
            <span>Ngày sinh: </span>
            <input type="text" name="day" value="<?php echo $day;?>">
        </div>
        <div class="row">
            <span>Tháng sinh: </span>
            <input type="text" name="month" value="<?php echo $month;?>">
        </div>

        <div class="row">
            <input type="submit" name="submit" value="Lấy chòm sao">
        </div>
    </form>
<?php
    // CÁCH 1
    // ucfirst(): dùng để in hoa kí tự đầu tiên của một chuỗi
    if ($flag == true) {
        $res = '<div class="result">
            
            <p>Cung ' . $name . ' <span>(' . ucfirst($image) . ': ' . $time . ')</span></p>
        </div>';
    }
    else {
        $res = "Dữ liệu không hợp lệ.";
    }

    echo $res;
?>

<!-- CÁCH 2 -->
<!-- <div class="result">
    ">
    <p>Cung <?php echo $name;?> <span>(<?php echo $image;?>: <?php echo $time;?>)</span></p>
</div> -->
</div>
</body>
</html>

```

```
# style.css

* {
    margin: 0px;
    padding: 0px;
}

.content {
    margin: 20px auto;
    width: 600px;
    border: 1px solid #999;
    padding: 10px;
}

.content h1 {
    color: red;
    text-align: center;
}

.content div.row {
    margin-top: 20px;
}

.content div.row span {
    display: inline-block;
    width: 255px;
    text-align: right;
}

.content div.row input[type=text] {
    padding: 3px 5px;
}

.content div.row input[type=submit] {
    padding: 3px 5px;
    display: block;
    margin: 0px auto 20px auto;
}

.content div.row p {
    font-weight: bold;
    font-size: 20px;
    text-align: center;
}

.content div.result {
    margin-left: 50px;
}

.content div.result img {
    width: auto;
}
```

```
height: 100px;
overflow: hidden;
float: left;
}

.content div.result p {
padding-left: 120px;
text-align: justify;
font-size: 22px;
height: 100px;
line-height: 100px;
}

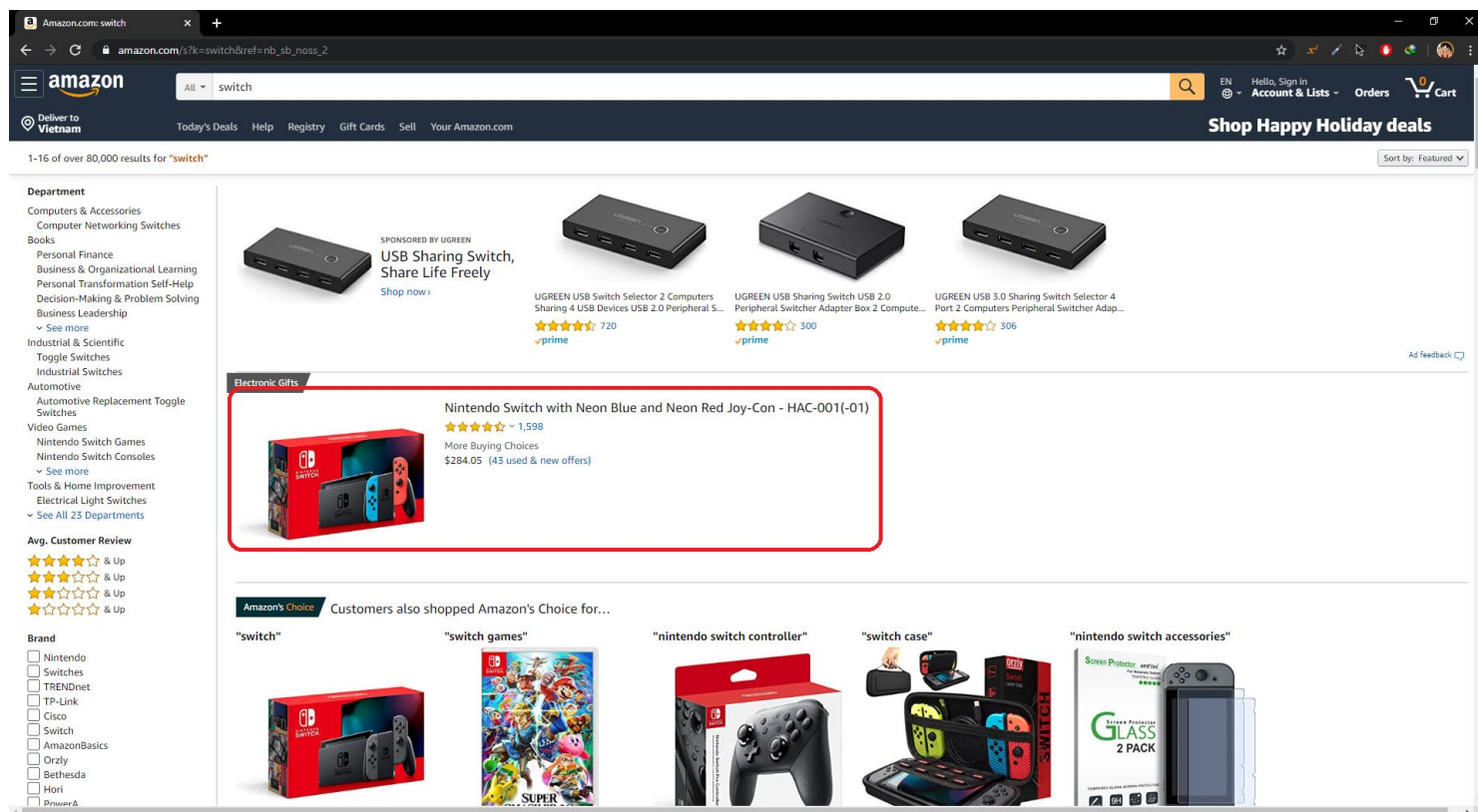
.content div.result p span {
font-style: italic;
}
```

1. Cookies & Sessions

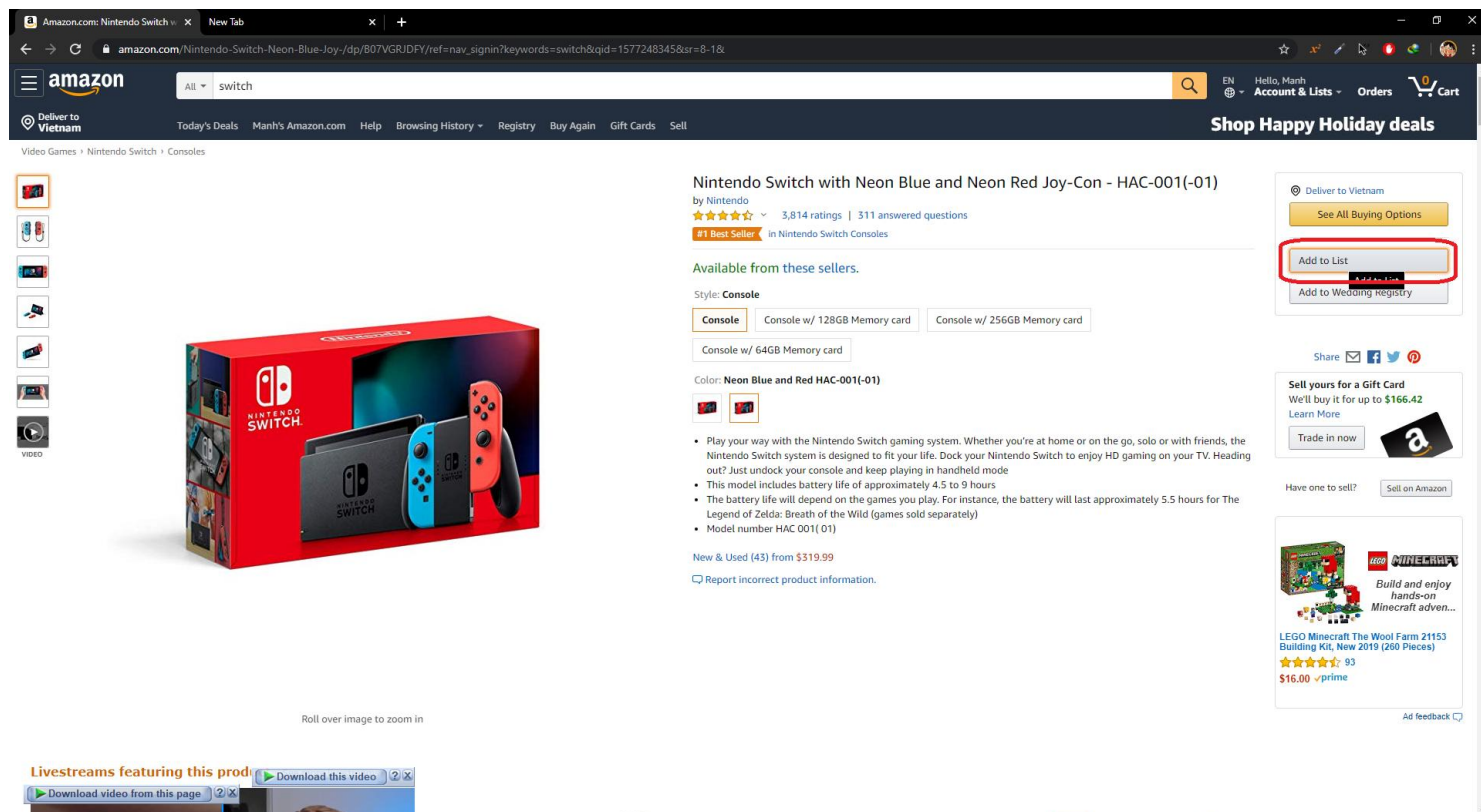
1.1. Cookies

VÍ DỤ

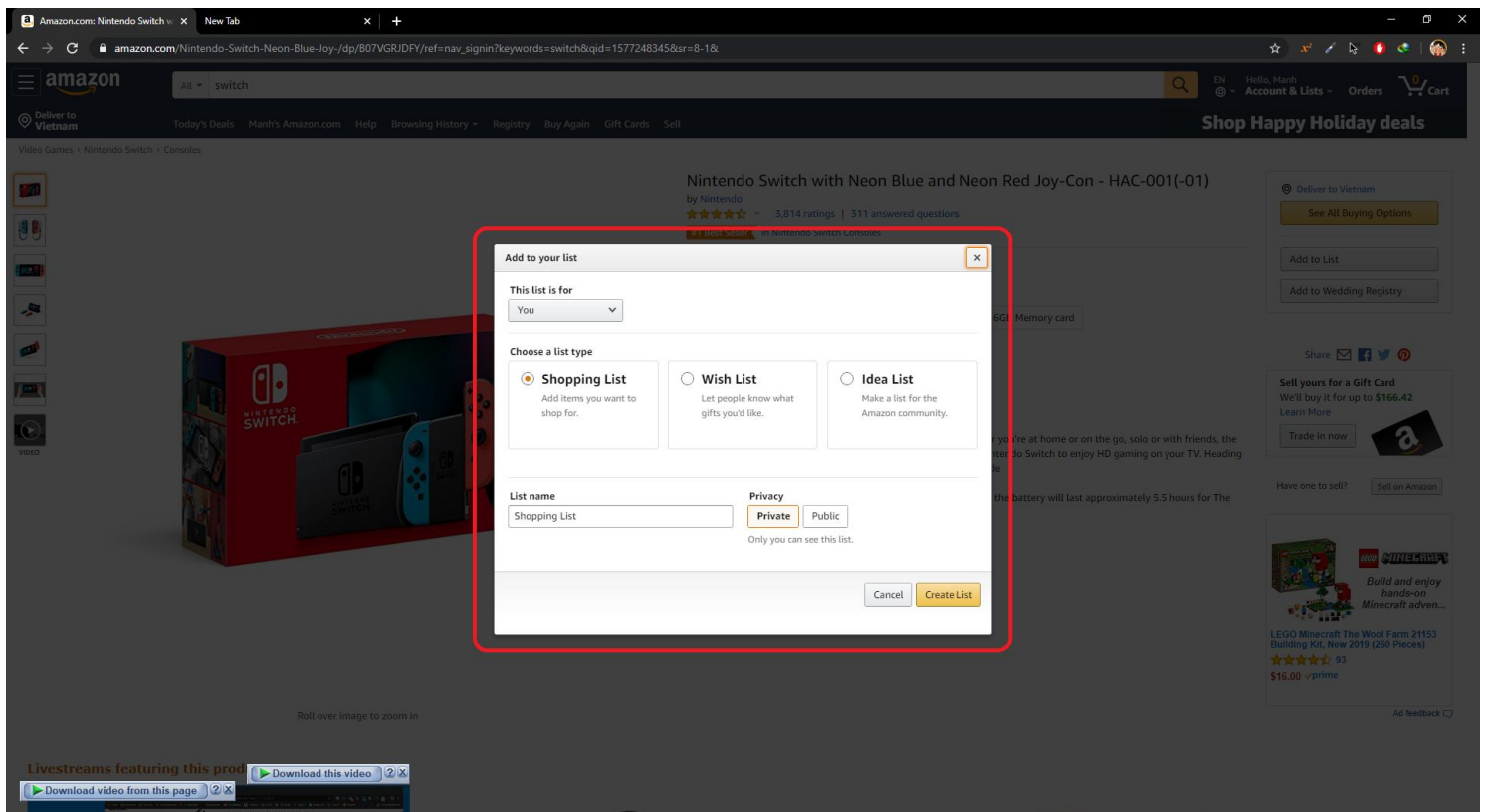
Ta vào trang Amazon, click vào thanh tìm kiếm và gõ "switch". Chúng ta thử click vào một mặt hàng nào đó (tại đây tôi sẽ click vào mặt hàng khoan lỗ trong hình bên dưới).



Và sau đó chọn thêm nó vào giỏ hàng.



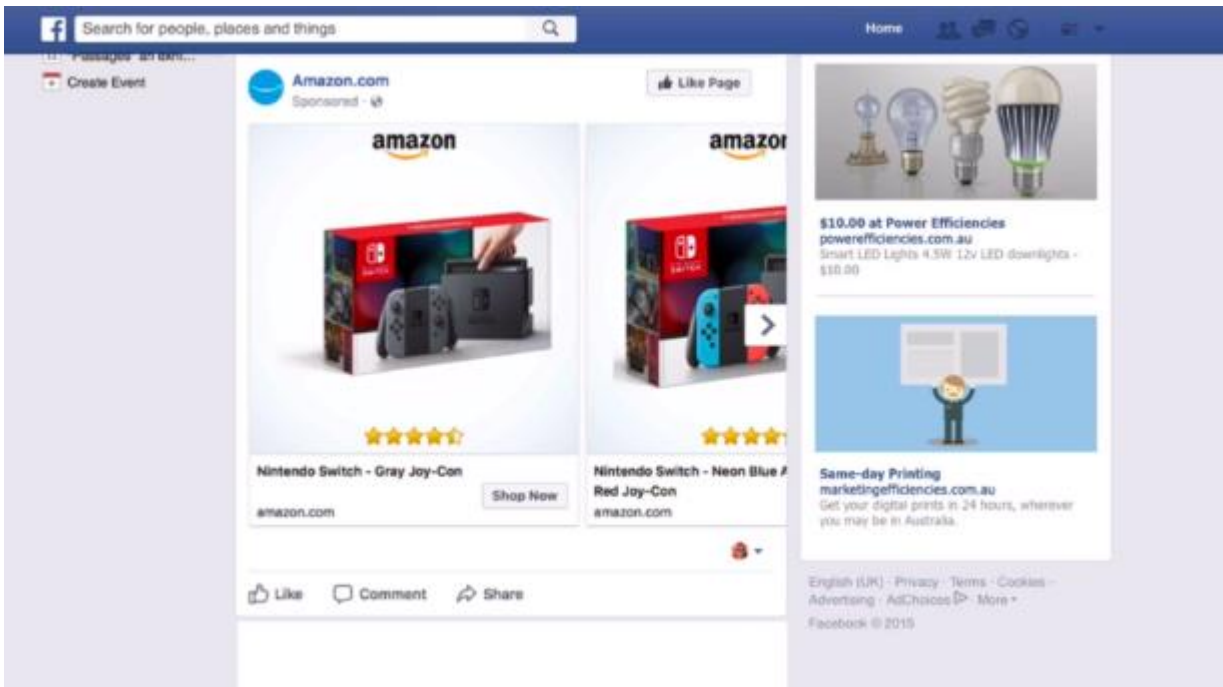
Và bây giờ nó hiện lên một thông báo hỏi xem chúng ta có xác định mua hàng này không, ta chọn **CREATE LIST**.



Và bây giờ, nếu ta có tắt đi trình duyệt và hôm sau mở lại và vào Amazon, chúng ta vẫn sẽ thấy item này nằm trong giỏ hàng.

Và khi Amazon đã gắn được Cookies vào máy tính cá nhân, điều đó có nghĩa rằng khi user đăng nhập vào một trang web nào khác sau đó, chẳng hạn như Facebook thì tức sau đó, Amazon đã biết user này là ai và đâu là những thứ mà user này muốn mua trên trang e-commerce của họ và từ đó Amazon sẽ cố gắng nhắc nhở cho user nhớ rằng "có vài thứ bạn muốn mua mà chưa mua" trên Amazon, hay còn gọi là **CHẠY QUẢNG CÁO**. Và đây là một trong những cách phổ biến mà ứng dụng quảng cáo ngày nay sử dụng.

Và tất cả các thứ trên, đều được thông qua **Cookies & Sessions**.



VẬY TÁC DỤNG CỦA COOKIES LÀ GÌ?

- Ngày nay, Cookies được sử dụng rộng rãi trên Internet để lưu lại các phiên duyệt web của user.
- Ngoài ra còn giúp giảm thiểu các thao tác, hành động lặp đi lặp lại, hay không quá cần thiết của người dùng trên web (ví dụ như khi ta đăng nhập vào Facebook một lần, thì những lần sau chúng ta chỉ cần ghi URL facebook.com là có thể đăng nhập ngay vào trang cá nhân mà không cần gõ lại account hay password).
- Cookie cũng được dùng để lưu những thông tin tạm thời.
- Tập tin cookie sẽ được truyền từ server tới browser và được lưu trữ trên máy tính của bạn khi bạn truy cập vào ứng dụng.

VẤN ĐỀ BẢO MẬT

Do Cookies thường được sử dụng để quản lý trạng thái của người dùng (ví dụ trạng thái đăng nhập), do đó có thể có một số vấn đề liên quan đến bảo mật khi sử dụng Cookies. Ví dụ như Cookies lưu trạng thái đăng nhập của người dùng bị đánh cắp, hacker có thể dùng Cookies này để giả mạo làm người dùng và thực hiện tương tác với trang web mà không cần đăng nhập. Một số extension của trình duyệt cho phép share Cookies để những người dùng khác nhau cùng truy cập vào 1 tài khoản mà không cần biết thông tin truy cập như username và password.

1.2. Sessions

Có rất nhiều loại Cookies khác nhau nhưng những cái mà ta thấy được trên Chrome là những Cookies dùng để thiết lập và duy trì một **Session** (hay còn gọi là một **PHIÊN LÀM VIỆC**).

VẬY SESSION LÀ GÌ?

- Session là khoảng thời gian browser có thể tương tác với server.
- Được lưu trên server.
- Chứa dữ liệu từ user.
- Session bắt đầu khi client gửi request đến server và kết thúc khi hết thời gian timeout (thời gian cho phép client có thể tương tác với server, (hãy nhớ về hệ thống portal của HCMUS mỗi khi đăng kí học phần, thời gian timeout là 10 phút, sau đó bắt đăng nhập lại)) hoặc khi bạn đăng xuất khỏi trang web đó.
- Do session lưu trên server, nên nếu sử dụng vô tội vạ với các hệ thống lớn số lượng client có thể lên đến hàng triệu thì sẽ đè nặng lên server phải lưu trữ nhiều, nên session thường chỉ lưu những thông tin đơn giản như thông tin đăng nhập, thông tin chi tiết giỏ hàng,...
- Mỗi session sẽ được cấp một ID duy nhất trên mỗi phiên làm việc, ID này sẽ được thay đổi qua từng phiên làm việc khác nhau.

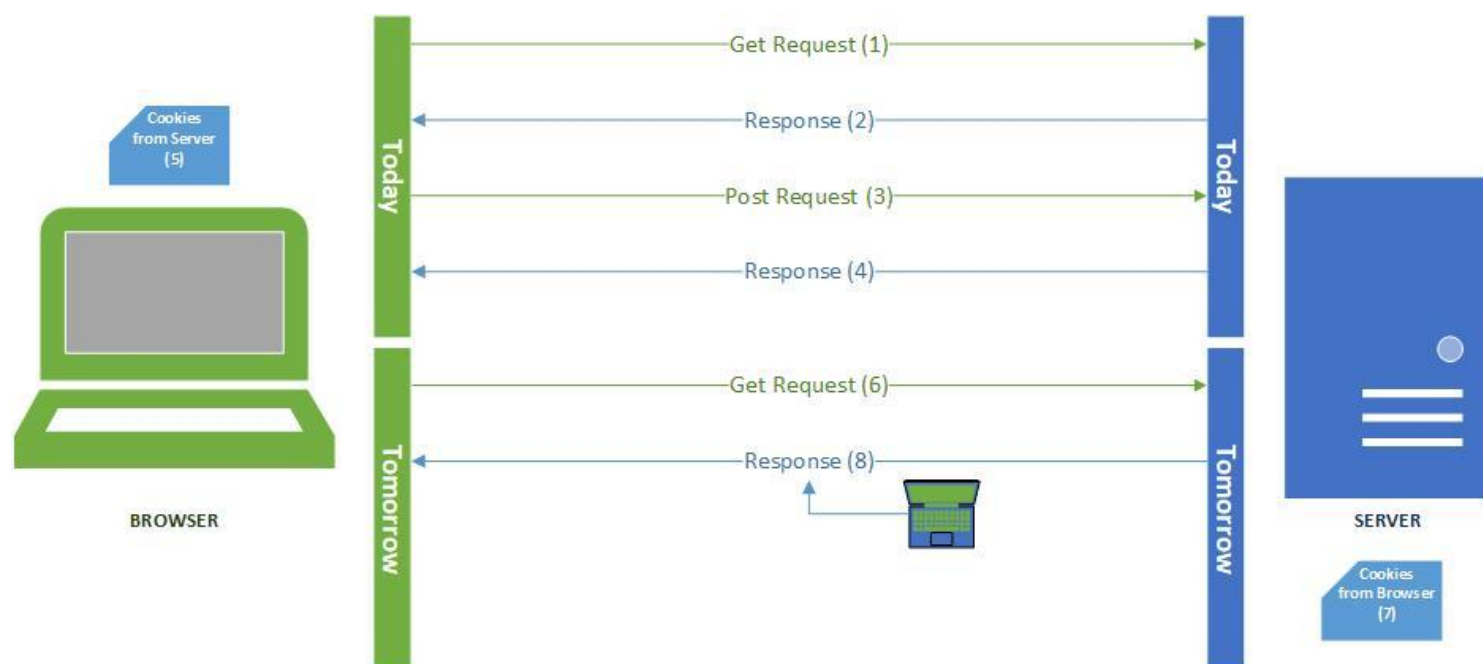
Vậy nên thông thường khi bạn đăng nhập vào một trang web, tức lúc này là Session của bạn đã được bắt đầu và cũng là lúc Cookies của bạn đã được tạo ra. Và bên trong Cookies của bạn sẽ chứa thông tin của bạn rằng: user này đã đăng nhập và xác thực thành công. Và từ đây, nếu bạn có tiếp tục duyệt web thì nó cũng sẽ không yêu cầu bạn đăng nhập lại lần nữa cho dù bạn có cố gắng truy cập lại vào chính trang đăng nhập bằng địa chỉ URL, bởi vì server luôn có thể kiểm tra các Cookies khả dụng mà bạn có trên chính browser của bạn và điều này giúp duy trì tính xác thực của bạn cho Session hiện tại cho tới khi bạn đăng xuất ra thì lúc này Session sẽ kết thúc và mọi Cookies liên quan trực tiếp đến Session này sẽ đều bị phá hủy.

1.3. Mô hình hoạt động

Như vậy, ví dụ trên đã cho ta hiểu sơ sơ về **Cookies & Sessions** là gì, cũng như ứng dụng của nó. Vậy khi ta nhìn nó dưới khía cạnh của một Web Development thì như thế nào?

Giả sử vào một ngày tôi truy cập vào trang [Amazon.com](https://www.amazon.com), lúc này Browser của tôi sẽ tạo một Get Request và gửi đến server của Amazon **(1)**. Sau đó server sẽ hồi đáp lại (hay còn gọi là Response) request của tôi và sẽ tiến hành gửi các file HTML, CSS, Javascript cần thiết cho browser của tôi để tiến hành render ra trang Amazon **(2)**. Và giả sử bây giờ, tôi thêm một cái laptop vào giỏ hàng của tôi và điều này tương đương với việc tôi đang thực hiện một cái Post Request, nói với server Amazon rằng là tôi đang muốn mua một cái Laptop **(3)**. Và sau đó, server Amazon sẽ tạo ra một cái Cookies chứa các tập dữ liệu về giao dịch của tôi rằng "User này muốn mua một cái laptop" và khi server tiến hành Response lại cho Post Request này (đã tiến hành ở bước (3)), cái Cookies được tạo ra sẽ được gửi theo cùng đồng thời nói cho Browser biết rằng "Hãy lưu cái Cookies này lại" **(4)**. Vậy nên nếu như bây giờ tôi tắt trang Amazon đi và vào một số trang khác như Facebook, Twitter hay bất cứ trang web nào

khác và vào lại trang Amazon vào ngày hôm sau. Cái Cookies này vẫn còn nằm ở đó trong Browser của tôi do nó đã được Browser lưu lại rồi **(5)**. Và ngày hôm sau khi tôi cũng gửi một cái Get Request này đến server Amazon, cái Cookies của ngày hôm trước sẽ được đính kèm và gửi theo cùng để cho server có thể xác định được "tôi là ai", "tôi đã truy cập vào trang Amazon trước đây chưa" **(6)**. Và từ đây, tương đương Cookies đã được server tiếp nhận và tiến hành mở ra và biết được giao dịch này của bạn là gì, những gì bạn định mua,... Trong trường hợp này là chiếc laptop **(7)**. Sau đó server sẽ tiến hành Response lại bao gồm các file HTML, CSS, Javascript và cũng bao gồm luôn giỏ hàng của user với chiếc laptop mà hôm trước chúng ta xem đã được thêm sẵn vào **(8)**.



1.4. Điểm khác biệt giữa Cookies và Sessions

Cookie	Session
Cookie được lưu trữ trên trình duyệt của người dùng.	Session không được lưu trữ trong trình duyệt của người dùng.
Dữ liệu cookie được lưu trữ ở phía máy khách.	Dữ liệu session được lưu trữ ở phía máy chủ.
Dữ liệu cookie dễ dàng sửa đổi khi chúng được lưu trữ ở phía khách hàng.	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 của chúng ta đến khi hết hạn.	Dữ liệu session có sẵn cho trình duyệt chạy. Sau khi đóng trình duyệt sẽ mất thông tin session.