

Buổi 24

Các hàm dùng xử lý trên chuỗi

`$array = explode ($delimiter , $string);` //Cắt chuỗi thành array, chỗ xác định cắt là giá trị `$delimiter`

`$str = implode($delimiter, $array);` //Nối các phần tử trong array lại thành chuỗi bằng giá trị `$delimiter`

`$length = strlen($string);` //Tìm chiều dài của chuỗi

`$count = str_word_count($str);` //Đếm số từ trong chuỗi

`$new_str = str_repeat($str, $n);` //Lập lại chuỗi `$str` lại `$n` lần

`$new_str = str_replace($search, $replace, $str);` //Thay thế giá trị `$search` bằng giá trị `$replace` trong chuỗi `$str`

`$new_str = substr($string, $start, $length);` //Tìm chuỗi con trong chuỗi

`$new_str = strstr($string, $substr);` // Tách một chuỗi bắt đầu từ `$ky_tu_cho_truoc` cho đến hết chuỗi.

`$position = strpos($str, $substr);` //Tìm vị trí của chuỗi con trong chuỗi

`$new_str = strtolower($str);` //Chuyển thành chữ thường

`$new_str = strtoupper($string);` //Chuyển thành chữ hoa

`ucfirst($string);` // Chuyển thành chữ hoa ký tự đầu tiên của chuỗi

`lcfirst($string);` // Chuyển thành chữ thường ký tự đầu tiên của chuỗi

`ucwords($string);` // Chuyển chữ hoa ký tự đầu tiên của mỗi từ trong chuỗi

`trim($string, $char);` //Cắt khoảng trắng đầu và cuối chuỗi

`ltrim($string, $char);` //Cắt khoảng trắng đầu chuỗi

`rtrim($string, $char);` //Cắt khoảng trắng cuối chuỗi

Các hàm kiểm tra dữ liệu thường dùng

1. `isset($var)`

Dùng để kiểm tra biến \$var có tồn tại hay không. Thông thường bạn hay dùng để kiểm tra một biến trước khi xử lý thao tác đến nó.

2. *empty(\$var)*

Kiểm tra biến \$var có phải giá trị trống hay không. Tất cả các giá trị như: số 0, giá trị null, giá trị rỗng, giá trị false đều được quy về là empty. Nếu biến \$var không tồn tại thì hàm này vẫn không báo lỗi và sẽ trả kết quả là true

3. *is_array(\$var)*

Kiểm tra biến \$var có phải kiểu mảng hay không

4. *is_string(\$var)*

Kiểm tra biến \$var có phải kiểu chuỗi hay không.

5. *is_numeric(\$var)*

Kiểm tra biến \$var có phải kiểu số hay không.

6. *is_null(\$var)*

Kiểm tra biến \$var có phải giá trị null không

7. *in_array(\$value, \$array)*

Kiểm tra giá trị \$value có trong mảng \$array hay không

8. *array_key_exists(\$key, \$array)*

Kiểm tra key \$key có trong mảng \$array hay không

9. *is_object(\$var)*

Kiểm tra biến \$var có phải giá trị object không

10. *is_bool(\$var)*

Kiểm tra biến \$var có phải giá trị boolean không

GET/POST

Truyền dữ liệu từ client lên server có 2 method: GET hoặc POST

- GET: Phương thức GET gửi dữ liệu tường minh thông qua url hay gọi là link, liên kết.

Ví dụ: <http://localhost/backend/buoi19/transferDataGet.php?param1=value1¶m2=value2...>

Tất cả dữ liệu gửi lên server theo phương thức GET đều được server bắt lấy và lưu trong biến toàn cục `$_GET`. Trên server dùng `$_GET` để lấy dữ liệu:
`$_GET["param1"], $_GET["param2"]`

- POST: Phương thức POST được truyền ngầm, nhằm bảo mật thông tin, người dùng hay người tò mò sẽ không thấy dữ liệu trên address

Tất cả dữ liệu gửi lên server theo phương thức POST đều được server bắt lấy và lưu trong biến toàn cục `$_POST`. Trên server dùng `$_POST` để lấy dữ liệu:
`$_POST["param1"], $_POST["param2"]`

Phương thức POST gửi dữ liệu ngầm thông qua form với nút submit

- So sánh GET & POST
 - POST an toàn hơn GET
 - GET nhanh hơn POST vì GET có lưu cache
 - GET tốt cho SEO

Xử lý file

- Mở file
 - `$fp = @fopen('demo.txt', "r");`

- Đọc file

- Đọc file từng ký tự

```
while(!feof($fp))  
{  
    echo fgetc($fp);  
}
```

- Đọc file từng dòng

```
while(!feof($fp))  
{  
    echo fgets($fp);  
}
```

- Đọc toàn bộ file

Cách 1:

```
$data = fread($fp, filesize('demo.txt'));  
echo $data;
```

Cách 2:

```
$data = file_get_contents('demo.txt');  
echo $data;
```

- Viết file

- Cách 1:

```
fwrite($fp, 'noi dung');
```

- Cách 2:

```
file_put_contents('demo.txt', 'noi dung');
```

- Đóng file

```
fclose($fp);
```

Lưu ý: Chỉ sử dụng fclose trong trường hợp dùng fopen.
file_get_contents và file_put_contents không cần dùng fclose

Upload file

Upload hình từ trình duyệt lên server. Dưới đây là các bước để thực hiện:

Step1: Tạo form upload (lưu ý rằng phải có thuộc tính enctype mới có thể upload file)

```
<form method="post" action="processUploadFile.php"  
enctype="multipart/form-data">  
  
  <input type="file" name="avatar"/>  
  
  <input type="submit" name="upload" value="Upload"/>  
  
</form>
```

Step2: Tạo file processUploadFile.php để lưu file trên server

Sử dụng hàm `move_uploaded_file` để di chuyển file từ thư mục tạm trên server sang thư mục của project

```
<?php

// Xử Lý Upload

// Nếu người dùng click nút Upload
if (isset($_POST["upload"]))
{
    // Nếu người dùng có chọn file để upload
    if (isset($_FILES['avatar']))
    {
        // Nếu file upload bị lỗi
        if ($_FILES['avatar']['error'] > 0)
        {
            echo 'File Upload Bị Lỗi';
        }
        else{// Nếu file upload không bị lỗi
            // Upload file
            move_uploaded_file($_FILES['avatar']['tmp_name'],
                './images/'.$_FILES['avatar']['name']);
            echo 'File Uploaded';
        }
    }
    else{
        echo 'Bạn chưa chọn file upload';
    }
}

?>
```

include/include_once/require/require_once

- Include
 - Import file php, nếu file không tồn tại chỉ warning (cảnh báo) và chương trình vẫn tiếp tục
 - Nếu file này đã import thì nó vẫn tiếp tục import
- include_once
 - Import file php, nếu file không tồn tại chỉ warning (cảnh báo) và chương trình vẫn tiếp tục
 - Nếu file này đã import thì nó không import nữa
- require
 - Import file php, nếu file không tồn tại thì sẽ fatal error và chương trình dừng lại
 - Nếu file này đã import thì nó vẫn tiếp tục import
- require_once
 - Import file php, nếu file không tồn tại thì sẽ fatal error và chương trình dừng lại
 - Nếu file này đã import thì nó không import nữa

Bài tập

- Bài 1:
 - Viết hàm listSortStudents(\$studentList); //hàm này trả về danh sách sinh viên đã được sắp xếp theo thứ tự alphabet
- Bài 2:
 - Viết hàm payPost(\$post); //Hàm này trả về số tiền cần trả cho bài viết theo tiêu chí sau:
 - chiều dài từ ≤ 4 thì giá là 50đ
 - chiều dài từ >4 thì giá là 100đ
- Bài 3:
 - Tạo form đăng nhập gồm các thành phần sau:
 - Username
 - Password
 - Button Login
 - Hiện thực theo method POST và hiển thị thông tin người dùng đăng nhập trên server
- Bài 4:
 - Tạo hình sản phẩm, khi click vào sản phẩm sẽ link đến trang sanpham.php?id=10 và hiển thị giá trị id này trên server (lấy thông qua \$_GET)