Array PHP

|  |  |
| --- | --- |
| Khởi tạo array | $arr = array() |
| Chuyển arr sang json, string | Implode(‘ky tu’, $ten\_arr);  Json\_encode($ten\_arr); |
| Thêm vào đầu arr | Array\_unshift($ten\_arr, ‘value’); |
| Thêm vào cuối arr | Array\_push($ten\_arr, ‘value’); |
| Lấy giá trị đầu | Reset($ten\_arrr); |
| Lấy giá trị đầu và xóa nó | Array\_shift($ten\_arr) |
| Lấy giá trị cuối | End($ten\_arr) |
| Lấy giá trị cuối và xóa nó | Array\_pop($ten\_arr) |
| Sắp xếp tang | Sort($ten\_arr) |
| Sắp xếp tang theo key | Ksort($ten\_arr) |
| Đảo ngược thứ tự arr | Array\_reverse($ten\_arr) |
| Xóa trùng lặp trong arr | Array\_unique($ten\_arr) |
| Đảo key với value | Array\_flip($arr) |
| Chuyển string sang aray | Explode(‘ky tu’, $chuoi) |
| Lấy mảng keu trong arr | Array\_key($ten\_arr) |
| Kiểm tra giá trị trong arr (true/false) | In\_array(‘value’, $ten\_arr) |
| Tìm kiếm trong arr (value or false) | Array\_search(‘value’, $ten\_arr) |
| Kiểm tra key tồn tại (true/ false) | Array\_key\_exists(‘value’, $arr) |
| Lấy giá trị chung của 2 array | Array\_intersect($arr1, $arr2) |
| Lấy key chung của 2 array | Array\_intersect\_key($arr1, $arr2) |
| Lấy giá trị khác của 2 array | Array\_diff($arr1, $arr2) |
| Lấy key khác của 2 array | Array\_diff\_key($arr1, $arr2) |
| Cộng arr //các thứ lien quan google | Array\_merge($arr1, $arr2) => lấy tất cả giá trị cả 2 arr lại, nếu key dạng str thì trùng key sẽ lấy value ở arr1.  $arr1+$arr2 => lấy độ dài arr lớn nhất, sau đó điền value arr1 vào, nếu thiếu thì lấy arr2 bù vào.  Array\_merge\_recursive($arr1, $arr2) => (như màu xanh dương) khác nhau là nếu trùng key thì key đó sẽ trỏ tới 1 arr mới chứa cả 2 value |
| Lọc và bỏ empty trong arr | Array\_filter($ten\_arr) |
| Lọc và bỏ trong arr có điều kiện | Array\_filter($ten\_arr, function()) |
| Kiểm tra có là array không | Is\_array($ten\_arr) |
| Đếm số lượng value(length) có trong chuỗi | Count($ten\_arr) |
| Xuất giá trị trong arr (hỗ trợ thu gọn mảng khi dung unique) | Array\_values($ten\_arr) |
| Gọi tới function và truyền các giá trị (only value) trong arr tới biến trong function. | Array\_map(‘fuction callback’, $arr)  VD: |
| Gọi tới function và truyền các giá trị (key and value) trong arr tới biến trong function. | Array\_walk($arr, function)  VD: |

**String PHP**

|  |  |
| --- | --- |
| Độ dài chuỗi | Strlen($str) |
| Số từ trong chuỗi //return number | Str\_word\_count($str) |
| Lặp chuỗi (ít dùng) | Str\_repeat($str, so\_lan) |
| Tìm và thay thế | Str\_replace($str\_tim, $str\_thaythe, $str\_gốc)  Mb\_str\_replace() //bỏ qua dấu |
| Mã hóa chuỗi md5 | Md5($str) |
| Mã hóa chuỗi sha1 | Sha1($str) |
| Hủy tính năng html | Htmlentities(‘chuỗi html’)  Vd:htmlentities('<b>freetuts.net</b>'); |
| Hủy tính năng html với tên thẻ xác định | Vd: Strip\_tags(‘chào<br>’, ’br’) |
| Cắt chuỗi từ vị trí, theo độ dài | Substr($str, to, length);  Mb\_substr() //bỏ qua dấu  Vd: substr($str, 0, 8); |
| Cắt chuỗi từ vị trí đến hết | Strstr($str, vi\_tri) |
| Trả về vị trí chuỗi cần tim | Strpos($str, $chuoi\_tim) |
| Chuyển sang chữ thường | Strtolower($str) |
| Chuyển snag chữ hoa | Strtoupper($str) |
| Chuyển ký tự đầu chuỗi sang hoa | Ucfirst($str) |
| Chuyển ký tự đầu của từng ký tự sang hoa | Ucwords($str) |
| Xóa ký tự trong chuỗi | Trim($str, ky\_tu\_xoa)  Ltrim or rtrim => chỉ xóa trái hoặc phải |
| Chuyển chuỗi dạng json sang mảng or object | Json\_decode($json, $is\_array) |
| Chuyển mảng or object sang json | Json\_encode($arr\_or\_obj) |
| Đếm byte không tính dấu | Mb\_strlen($str) |
| Đếm byte bao gồm cả dấu | Strlen($str) |
| Đảo chuỗi | Strrev($str) |
| Kiểm tra ngày tháng | Checkdate(tháng, ngày, năm) |

***Mail PHP:***

Mail($to, $subject, $message, $headers)

$to: địa chỉ người nhận

$subject: tiêu đề email

$message: nội dung

$header: quản lý việc gửi mail.

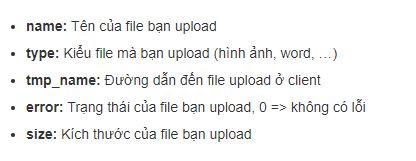
**File & Upload File:**

[Link các lỗi khi gặp lúc upload file](http://php.net/manual/en/features.file-upload.errors.php)

[Upload file](http://dungbui.top/danh-muc/lap-trinh/upload-file-bang-ajax-trong-php.html)

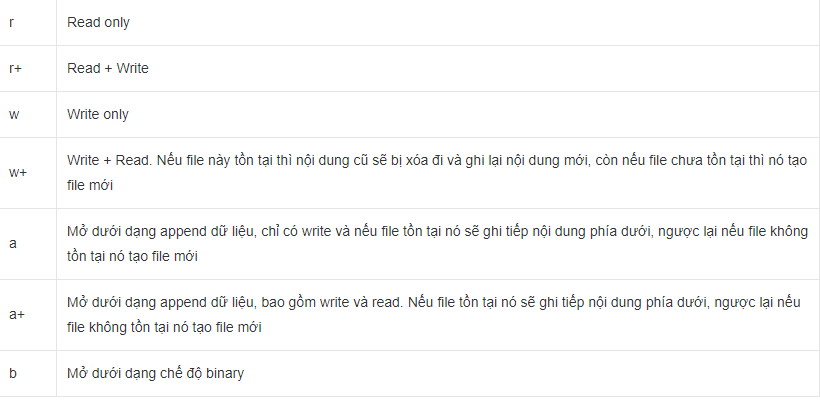
$\_FILES[‘input\_name’][‘properties’]

Properties:



$fopen = @fopen(‘đường\_dẫn’, ‘option’) //A@ để khi sai đường dẫn k bung lỗi ra browser

Option:



Read & write

|  |  |
| --- | --- |
| Khi đọc xong (char, string, all) thì đọc tiếp theo | While(Feof($fopen)){} |
| Đọc theo dòng | Fgets($fopen) |
| Đọc theo ký tự | Fgetc($fopen) |
| Đọc hết file, 2 param là $fopen và đường dẫn tới file | Fread($fopen, filesize(‘path’)) |
| Viết file, nếu chưa có thì tạo, có rồi thì viết đè | Fwrite($fopen, $data) |
|  | Tempnam($dir, $prefix)  $dir là thư mục chứa file  $prefix tên file, lấy 3 ký tự đầu, ký tự sau random |

SplFileObject:

[Link tổng hợp](http://php.net/manual/en/splfileobject.fgetss.php)

$file = new SplFileObject('demo.txt');

// đọc file chuỗi string

function readFiles() {

$file = new SplFileObject("demo.txt");

while ( ! $file->eof()) {// xác định xem đã tới cuối file chưa

echo $file->fgets();// đọc theo string

}

}

// đọc theo theo ký tự

function readFileWithChar() {

$file = new SplFileObject('demo.txt');

while (false !== ($char = $file->fgetc())) {

echo "$char";

}

}

// write file

function writeFile() {

$file = new SplFileObject('demo.txt', 'r+');

$file->fwrite("Foo"); // nội dung viết

$file->ftruncate($file->ftell()); // thực hành

}

// đọc file, mỗi dòng tương ứng 1 mảng.

function showArr() {

$file = new SplFileObject("demo.txt");

while (!$file->eof()) {

var\_dump($file->fgetcsv());

}

}

// tạo file và viết

function createAndWrite() {

$file = new SplFileObject("fwrite.txt", "w");

$written = $file->fwrite("12345");

echo "Wrote $written bytes to file";

}

Ajax:

+ Lấy giá trị được echo từ server:

+ áp dụng json\_encode và decode.

<?php

if (!empty($\_POST['submit']) && $countErr == 0) {

ob\_start(); // begin collecting output

include 'server.php';

$result = ob\_get\_clean(); // retrieve output from myfile.php, stop buffering

$result = json\_decode($result);

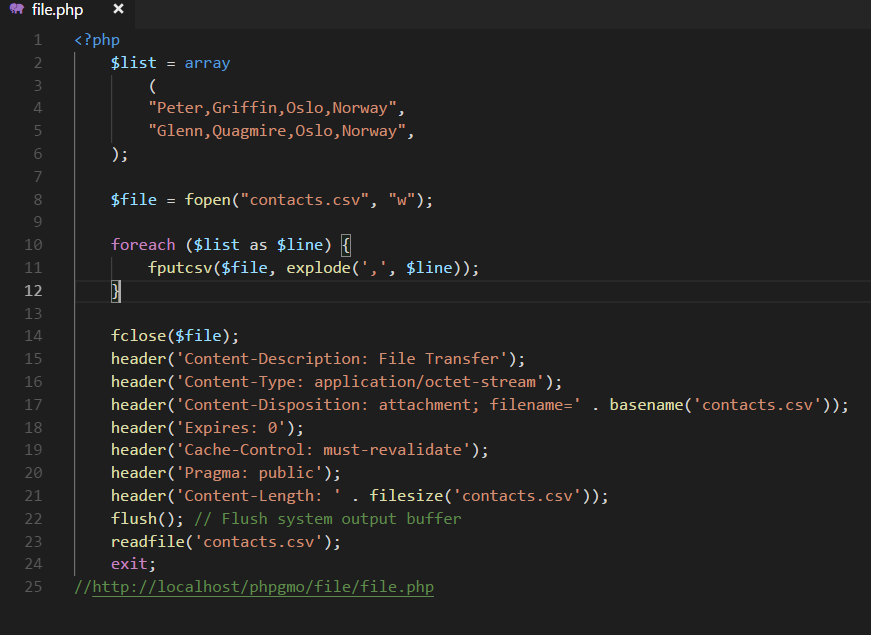
}

?>

Upload:

|  |  |
| --- | --- |
| Kiểm tra file tồn tại với path là đường dẫn server | File\_exists($path) |
| Kiểm tra file có upload thành công hay ko | Is\_uploaded\_file($\_FILES[‘input\_name][‘tmp\_name’]) |
| Chuyển file từ client lên server với param là 2 đường dẫn lần lượt là client và server. | Move\_uploaded\_file($\_FILES[‘image’][‘tmp\_name’],‘./folder\_chứa/’.$\_FILES[‘input\_name’][‘name’]) |
|  |  |

Dowload:



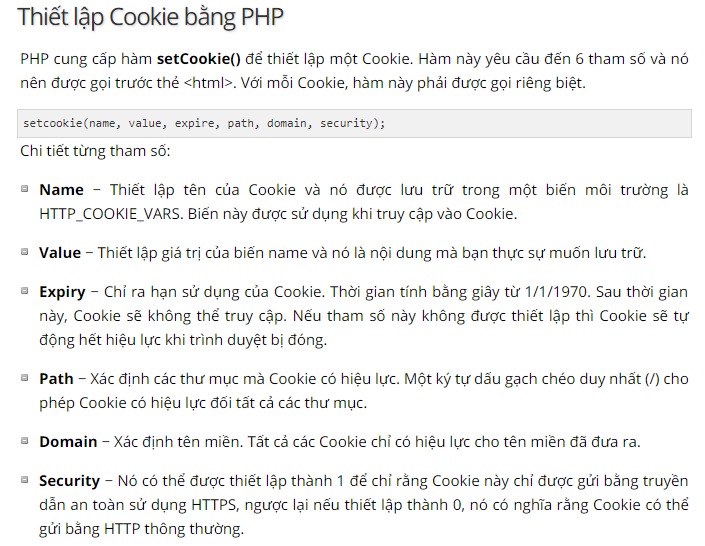
***Cookie***

Cookie là gì? Hoạt động ở đâu?

Cookie thường được dung trong trường hợp nào?

Cookie được khởi tạo ở đâu? Và khi nào mất?

* Cookie là 1 file được server chỉ định lưu trữ trên client và php có thể truy xuất tới được.
* Chứa thao tác người dung (quảng cáo), tần xuất ghé thăm, thời gian truy cập
* Cookie được tạo ở server sau đó gửi và lưu trữ trên client. Thời gian sống do dev quyết định.



**-Session**

**Lưu ý: Khai báo SESSIONSTART**

|  |  |
| --- | --- |
| Khai báo session | $\_SESSION[‘name’] = value; |
| Xóa session | Unset($\_SESSION[‘name’]) |
| Xóa tất cả session | Session\_destroy() |

Session DB:

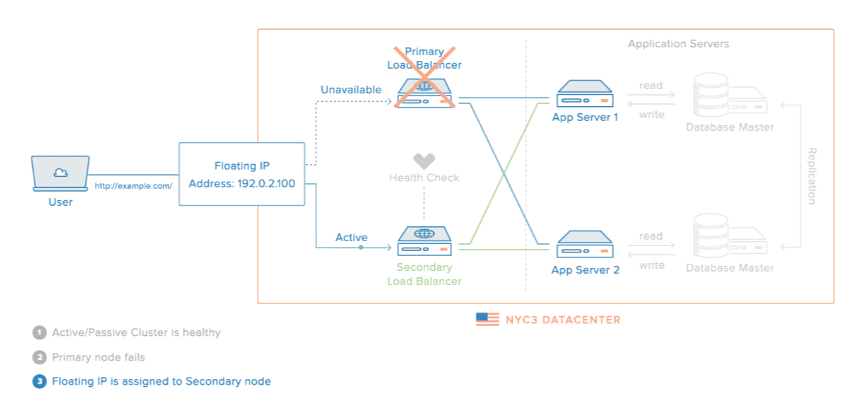
[Link tham khao](http://www.hackingwithphp.com/10/3/7/files-vs-databases)

+ Session DB là session được lưu trữ trong database.

+ Sử dụng session db trong trường hợp lưu thông tin đăng nhập của khách hàng, hay những thao tác đòi hỏi tính bảo mật, nhưng vẫn đảm bảo tính tiện lợi dành cho khách hàng.

+ Cân bằng tải webserver là một hệ thống hỗ trợ bổ sung trên backend, có tác dụng chuyển tiếp yêu cầu của người dùng đến server phụ trợ và đáp ứng yêu cầu đó.

Mô hình cân bằng tải :



Virtuahost localhost : [Link tim hiểu](https://kipalog.com/posts/Cau-hinh-Virtual-Host-trong-XAMPP)

C:\xampp\apache\conf\extra\httpd-vhosts

%windir%/system32/drivers/etc/hosts

OOP

|  |  |
| --- | --- |
| Tự động gọi tới hàm này mỗi khi class được khởi tạo | \_\_contruct |
| Hủy hàm khởi tạo, thường dùng để giải phóng bộ nhớ. | \_\_destruct |
| Viết lại hàm session cho php | Session\_set\_save  -handler(‘open’, ‘close’, ‘read’, ‘write’, ‘destroy’, ‘gc’) |

//1

function sess\_open($sess\_path, $sess\_name) {

print "Session opened.\n";

print "Sess\_path: $sess\_path\n";

print "Sess\_name: $sess\_name\n\n";

return true;

}

//4

function sess\_close() {

print "Session closed.\n";

return true;

}

//2

function sess\_read($sess\_id) {

print "Session read.\n";

print "Sess\_ID: $sess\_id\n";

return '';

}

//3

function sess\_write($sess\_id, $data) {

print "Session value written.\n";

print "Sess\_ID: $sess\_id\n";

print "Data: $data\n\n";

return true;

}

function sess\_destroy($sess\_id) {

print "Session destroy called.\n";

return true;

}

function sess\_gc($sess\_maxlifetime) {

print "Session garbage collection called.\n";

print "Sess\_maxlifetime: $sess\_maxlifetime\n";

return true;

}

session\_set\_save\_handler("sess\_open", "sess\_close", "sess\_read", "sess\_write", "sess\_destroy", "sess\_gc");

session\_start();

Upload big database mysql:

mysql -u username -p -h hostname databasename < dump.sql

Xdebug:

[XDebug]

zend\_extension = "C:\XAMPP\php\ext\php\_xdebug.dll"

xdebug.remote\_autostart = 1

xdebug.profiler\_append = 0

xdebug.profiler\_enable = 0

xdebug.profiler\_enable\_trigger = 0

xdebug.profiler\_output\_dir = "c:\xampp\tmp"

;xdebug.profiler\_output\_name = "cachegrind.out.%t-%s"

xdebug.remote\_enable = 1

xdebug.remote\_handler = "dbgp"

xdebug.remote\_host = "127.0.0.1"

xdebug.remote\_log="c:\xampp\tmp\xdebug.txt"

xdebug.remote\_port = 9000

xdebug.trace\_output\_dir = "c:\xampp\tmp"

; 3600 (1 hour), 36000 = 10h

xdebug.remote\_cookie\_expire\_time = 36000