HTML

□□php css js bootscript□□

- □□□□
- SEO□□
- SVG□□
- php□□□□

快取指令

php.ini設定備註

php指令

檢視開啟的php服務

```
php -m
```

```
<?php
?>
```

四捨五入

```
round(x,prec)
```

x:要處理的值, prec要處理的位數

向上取整數(無條件進位)

```
ceil(x)
```

向下取整數(無條件捨去)

```
floor(x)
```

上傳大小限制

```
; 上傳檔案大小上限（單一檔案大小）
upload_max_filesize = 50M

; POST 大小上限（所有檔案大小加總）
post_max_size = 200M

; 記憶體用量上限
memory_limit = 512M
```

記憶體用量上限 > POST 大小上限 > 上傳檔案大小上限

```
; PHP 指令稿執行時間上限（秒）
max_execution_time = 600

; PHP 指令稿解析輸入資料時間上限（秒）
max_input_time = 600
```

```
; MySQL 資料庫連線逾時（秒，-1 代表永不斷線）
mysql.connect_timeout = -1
```

php config設定優化

```
;OPcache打開/關閉開關，當設置為0時，會關閉Opcache，代碼沒有被優化和緩存。
opcache.enable = 1

;OPcache共享內存存儲大小，用於存儲預編譯的opcode（以MB為單位）
opcache.memory_consumption = 256

;這是一個很有用的選項，但是似乎完全沒有文檔說明
;PHP使用了一種叫做字符串駐留（string interning）的技術來改善性能。
;例如，如果你在代碼中使用了1000次字符串"foobar"，在PHP內部只會在第一使用
;這個字符串的時候分配一個不可變的內存區域來存儲這個字符串，其他的999次使用
;都會直接指向這個內存區域。這個選項則會把這個特性提升一個層次——默認情況下這
;個不可變的內存區域只會存在於單個php-fpm的進程中，如果設置了這個選項，那麼它
;將會在所有的php-fpm進程中共享。在比較大的應用中，這可以非常有效地節約內存，
;提高應用的性能。
;這個選項的值是以兆字節（megabytes）作為單位，如果把它設置為16，
;則表示16MB，默認是4MB，這是一個比較低的值。
;(default "4")
opcache.interned_strings_buffer = 8

;這個選項用於控制內存中最多可以緩存多少個PHP文件。
;這個選項必須得設置得足夠大，大於你的項目中的所有PHP文件的總和。
;設置值取值範圍最小值是 200，最大值在 PHP 5.5.6 之前是 100000，
;PHP 5.5.6 及之後是 1000000。也就是說在200到1000000之間。
;(default "2000")
opcache.max_accelerated_files = 4000

;如果啟用，OPcache會在 opcache.revalidate_freq 設置的秒數去檢測文件的timestamp
;檢查腳本是否更新。
;建議在開發/測試環境設為"1"，而正式環境設為"0"
```

```
;(default "1")
opcache.validate_timestamps = 1

;這個選項用於設置緩存的過期時間（單位是秒），
;當這個時間達到後，opcache會檢查你的代碼是否改變，
;如果改變了PHP會重新編譯它，生成新的opcode，並且更新緩存。
;值為"0"表示每次請求都會檢查你的PHP代碼是否更新（這意味著會
;增加很多次stat系統調用，譯註：stat系統調用是讀取文件的狀態，
;這裡主要是獲取最近修改時間，這個系統調用會發生磁盤I/O，
;所以必然會消耗一些CPU時間，當然系統調用本身也會消耗一些CPU時間）。
;建議可以在開發環境中把它設置為"0"，正式環境下不用管。
;(default "2")
opcache.revalidate_freq = 60

;如果啟用，則會使用快速停止續發事件。所謂快速停止續發事件是指依賴
;Zend 引擎的內存管理模塊一次釋放全部請求變量的內存，而不是依次釋放
;每一個已分配的內存塊。
;該指令已在PHP 7.2.0中被刪除。快速關機序列的一個變種已經被集成到PHP中，並且如果可能的話將被自動使用。
;(default "0")
opcache.fast_shutdown = 1

;CLI環境下，PHP啟用OPcache。這主要是為了測試和調試。從 PHP 7.1.2 開始，默認啟用。
opcache.enable_cli = 1

;如果啟用，OPcache將在哈希表的腳本鍵之後附加改腳本的工作目錄，
;以避免同名腳本衝突的問題。禁用此選項可以提高性能，但是可能會導致應用崩潰
;(default "1")
opcache.use_cwd = 0

opcache.max_file_size = 0
```

如何增加 PHP 記憶體限制 (memory_limit)

通常 PHP 會有三種修改 memory_limit 的方法:

修改 php.ini 檔案

```
memory_limit = 256M
```

修改 .htaccess 檔案

```
php_value memory_limit 512M
```

應用程式動態調整

第一步: 確認現在的 memory_limit 數值

```
$memory_limit = ini_get('memory_limit');
echo "目前的 Memory Limit 為: " . $memory_limit . "\n";
```

第二步: 設定 memory_limit 數值

```
ini_set('memory_limit', '4GB');
```

第三步: 確認 memory limit 數值

```
$memory_limit = ini_get('memory_limit');
echo "新的 Memory Limit 為: " . $memory_limit . "\n";
```

或是使用 phpinfo 頁面做確認

```
<?php
phpinfo();
```

php註解

```
<?php
// 這裡是單行 PHP 註解 ...
// 這裡是單行 PHP 註解 ...
?>



<?php
/*
...這裡是多行 PHP 註解
...這裡是多行 PHP 註解
...這裡是多行 PHP 註解
*/
?>
```

```
<font size="1" color="blue">這是大小為 1 的文字</font><br>
```

```
<a href="/html/image-img-tag.html">這個連結</a>會連到 Fooish 圖片標籤教學頁面
<a href="#" style="color:red;" target="屬性" rel="屬性">修改超連結顏色為紅色</a>
```

修改超連結顏色方法二、另外寫 CSS 來修改，全網頁

```
<style>
a {
    color:green;
}
</style>
<a href="#">修改超連結顏色為綠色</a>
```

```
<table backgroun="" border"1" width="" height="" align="center" valign=""><tr><td align="" colspan=""
rowspan="">
```

```
<DIV style="background-color:#C7FF91;width:300px;height:100px;text-align:center;line-height:100px;">
測試文字垂直置中
</DIV>
```

<a> 標籤的屬性 (attributes)

href
指定一個 URL 看連結要連到哪邊，亦可以用target指向相對位址。
target
target 有下面這些屬性值:
- _self: 預設值，在當前視窗開啟
- _blank: 在新視窗開啟
- _parent: 在上一層父視窗開啟
- _top: 在最頂層父視窗開啟
- framename: 指定在哪個框架中開啟

錨定連結

rel屬性
- nofollow: 不要追蹤跟索引
- external: 等效於target="_blank"，但_blank在xhtml會失效
- noopener: 等效於target="_blank"，但可以避免window.opener.location = newURL的攻擊
- noreferrer: 安全設定，會跟noopener一起使用。ex: <a href=url" rel ="noreferrer noopener" target="_blank"></a

備註: noreferrer和noopener屬性值針對瀏覽器，對SEO優化沒有任何影響

跳往同頁面不同區塊的位置

使用方法:

```
<a href="#some-section-id">連結文字</a>
```

上面的連結點擊後會跳去 id="some-section-id" 的 HTML 元素區塊。

例如跳到這邊:

```
<div id="some-section-id">
  hello
</div>
```

圖片連結

```
<a href="https://example.com/" target="_blank">
  <img src="https://source.unsplash.com/WLUHO9A_xik/600x400">
</a>
```

電子郵件超連結 mailto:

```
<a href="mailto:電子郵件信箱">連絡信箱</a>
```

電話號碼連結 tel: **遵循 RFC 3966 標準格式**

```
<a href="#" style="color:red;">修改超連結顏色為紅色</a>
<a href="tel:電話號碼">連結文字</a>
```

```
<marquee>跑馬燈</marquee>
```

direction="參數值": 控制跑的方向，上下左右分別為 up、dun、left、right
align="參數值": 控制對齊方式，向上對齊 top、垂直至中 midden、向下對齊 botton
scrollamount="參數值": 跑馬燈的速度，數字越大表示跑得越快
height="參數值": 區域高度
width="參數值": 區域寬度
behavior="參數值": 跑馬燈的行為模式，來回跑用 alternate、跑入就停止用 slide
bgcolor="參數值": 設定跑馬燈的背影顏色，預設為透明
Loop="輪播次數"
Behavior"輪播方式" 設 "Scroll"(內定值)、"Slide" 當文字碰到左邊就會停止、"Alternate" 在左右兩邊彈來彈去。
"
scrolldelay="0" 振動效果

 (空白鍵字符) （半中文字型空白鍵字符） （一中文字型空白鍵字符）
onMouseOver="this.stop()" onMouseOut="this.start()" 滑鼠放在上方就停止
hspace="" 左右外框
vspace="" 上下外框

半形的不換行空格，就是一般鍵盤上的空白鍵(space key)產生的空格

□□□□

□□ date() □□□□ PHP □□□□□
□□ strftime() □□□□ PHP □□□□□
□□ DateTime □□□□ PHP □□□□□

```
<?php
$Date = date("d-m-Y");
echo "The current date is $Date.";
echo "\n";
$Year = date("Y");
echo "The current year is $Year.";
echo "\n";
$Year2 = date("y");
echo "The current year in two digits is $Year2.";
?>
```

The current date is 20-04-2020.
The current year is 2020.
The current year in two digits is 20.

```
<?php
$Date = strftime("%d-%m-%Y");
echo "The current date is $Date.";
echo "\n";
$Year = strftime("%Y");
echo "The current year is $Year.";
echo "\n";
$Year2 = strftime("%y");
echo "The current year in two digits is $Year2.";
?>
```

The current date is 20-04-2020.
The current year is 2020.
The current year in two digits is 20.

```
<?php
$Object = new DateTime();
$Date = $Object->format("d-m-Y");
echo "The current date is $Date.";
echo "\n";
$Year = $Object->format("Y");
echo "The current year is $Year.";
echo "\n";
$Year2 = $Object->format("y");
echo "The current year in two digits is $Year2.";
?>
```

The current date is 20-04-2020.
The current year is 2020.
The current year in two digits is 20.

□□□□□□□□□□

```
$L = new DateTime( '2020-02-01' );
echo $L->format( 'Y-m-t' );
```

□□ PHP 5.3 □□□□□□□ DateTime() □ DateInterval() □□□

□□□2□□□

PHP >5.3

```
$firstDate  = new DateTime("2019-01-01");
$secondDate = new DateTime("2020-03-04");
$intvl = $firstDate->diff($secondDate);
```

echo $intvl->y . " year, " . $intvl->m." months and ".$intvl->d." day";
echo "<br>";
// Total amount of days

```
echo $intvl->days . " days ";
```

//output: 1 year, 2 months and 1 day
//   428 days

□□□□□□□□□□□□□□□

date($format, $timestamp);

```
<?php
$date = date('d-m-Y H:i:s', 1565600000);
echo "The date is $date.";
?>
```

□□□□□□□□ d-m-Y - □-□-□□□□□□□ H:i:s - □□□□□□□□
□□□The date and time are 12-08-2019 08:53:20.

$datetimeObject->setTimestamp($timestamp);

□□□□□□

```
<?php
$date = new DateTime();
$date->setTimestamp(1565600000);
$variable = $date->format('U = d-m-Y H:i:s');
echo "The date and time is $variable.";
?>
```

□□□The date and time are 1565600000 = 12-08-2019 08:53:20.

DateTime::createFromFormat($format, $time, $timezone);

□□ $format □□□□□□□□□ $time □□□□□□□□□□□□ $timezone □□□□□□□□□□□□□□□□□

```
<?php
// Calling the createFromFormat() function
$datetime = DateTime::createFromFormat('U', '1565600000');
// Getting the new formatted datetime
$date= $datetime->format('d-m-Y H:i:s');
echo "The date and time is $date.";
?>
```

□□ d-m-Y H□i□s □□□□□□□□

□□□The date and time are 12-08-2019 08:53:20.

PHP sprintf() □□

sprintf(format,arg1,arg2,arg++)

%% - □□□□□□□
%b - □□□□
%c - □□ ASCII □□□□
%d - □□□□□□□
%e - □□□□□□□□ 1.5e+3□
%u - □□□□□□□
%f - □□□(local settings aware)
%F - □□□(not local settings aware)
%o - □□□□
%s - □□□
%x - □□□□□□□□□□□
%X - □□□□□□□□□□□

□□ 1

```
<?php
$str = "Hello";
$number = 123;
$txt = sprintf("%s world. Day number %u",$str,$number);
echo $txt;
 ?>
```

□□□Hello world. Day number 123

□□ 2

```
<?php
$number = 123;
$txt = sprintf("%f",$number);
echo $txt;
 ?>
```

□□□123.000000

□□ 3

```
<?php
$number = 123;
$txt = sprintf("With 2 decimals: %1$.2f<br />With no decimals: %1$u",$number);
echo $txt;
 ?>
```

輸出: With 2 decimals: 123.00
With no decimals: 123

改為偵錯模式

到php.ini 將display_errors = on

span用法

```
<p align="center"></p>
```

```
<p>My mother has <span class="highlight">blue</span> eyes.</p>
```

align="參數值": right left center

```
<table style="border:3px #cccccc solid;" cellpadding="10" border='1'>
<tr><td>這是表格欄位</td><td>這是表格欄位</td></tr>
<tr><td>這是表格欄位</td><td>這是表格欄位</td></tr>
</table>

<table style="border:3px #FFD382 dashed;" cellpadding="10" border='1'>
<tr><td>這是表格欄位</td><td>這是表格欄位</td></tr>
<tr><td>這是表格欄位</td><td>這是表格欄位</td></tr>
</table>

<table style="border:8px #FFD382 groove;" cellpadding="10" border='0'>
<tr><td>這是表格欄位</td><td>這是表格欄位</td></tr>
<tr><td>這是表格欄位</td><td>這是表格欄位</td></tr>
</table>
```

border-top ← 用來設計上邊框。
border-right ← 用來設計右邊框。
border-bottom ← 用來設計下邊框。
border-left ← 用來設計左邊框。
border-color:上 右 下 左
border-radius: 10px 20px 30px 40px;
border-style: 可改變框線的style。可改變的值有

none(預設)
dotted
dashed

solid
double
groove
ridge
inset

□□□java□□head□□□□□□□□□□□□□□□□□□

```
<script type="text/javascript">
function parseLink(){
  var tagA = document.getElementsByTagName( "a" );
  re = new RegExp( "^(http://" + document.domain + ")|(javascript:)", "i" );
  for( var i=0; i < tagA.length; i++ ){
    if( !tagA[i].href.match( re ) ){ tagA[i].target = "_blank"; }
  }
}
</script>
```

□□□body□□onload

```
<body onload="parseLink();">
```

□□□java□□head□□□□□□□□□□□□□□□□□□2

```
<script type='text/javascript'>
  window.onload = function() {
    var links = document.links;
    for (var i = 0, linksLength = links.length; i < linksLength; i++) {
      if (links[i].hostname != window.location.hostname) {
        links[i].target = '_blank';
      }
    }
  }
</script>
```

1□□□□□□□□□□□html□□□form□□□□□□□□□

```
enctype='multipart/form-data'
```

2□□□□□□□□□□□□□form□□□□□□file□□□□□□□□□

```
<input type=hidden name='MAX_FILE_SIZE' value='value'>
```

檔案上傳錯誤程式碼:

1、預定義變數$_FILES陣列有5個內容:

$_FILES['userfile']['name']——用戶端機器檔案的原名稱

$_FILES['userfile']['type']——檔案的 MIME 型別

$_FILES['userfile']['size']——已上傳檔案的大小，單位爲位元組

$_FILES['userfile']['tmp_name']——檔案被上傳後在伺服器端儲存的臨時檔名

$_FILES['userfile']['error']——和該檔案上傳相關的錯誤程式碼

2、其中$_FILES['userfile']['error']的可以有下列取值和意義:

0——沒有錯誤發生，檔案上傳成功。不一定真的有檔案上傳了，有可能你檢視發現size是0。

1——上傳的檔案超過了 php.ini 中 upload_max_filesize 選項限制的值。

2——上傳檔案的大小超過了 HTML 表單中 MAX_FILE_SIZE 選項指定的值。

3——檔案只有部分被上傳。

4——沒有檔案被上傳。是指表單的file域沒有內容，是空字串。

上傳檔案類型限制

```
<input accept=".pdf,.xls" type="file" />  #指定副檔名
<input accept="image/*" type="file" />    #指定類型，此例為jpg、gif、png
<input accept="text/html" type="file" />  #只接受網頁格式htm html
<input accept="video/*" type="file" />    #只接受影片檔（包含 .avi、.mp4、.mpg 等影片檔）
<input accept="audio/*" type="file" />    #只接受聲音檔（包含 .mp3、.wav 等聲音檔）
<input accept="image/*,.pdf,.xls" type="file" />  #複合方式
```

寫上傳資料用的程式

操作介面（方式1）

```
 <!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title></title>
</head>

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

```
    </form>
</body>
</html>
```

□□□□upload.php□

```
<?php
  if ($_FILES["file"]["error"] == UPLOAD_ERR_OK) {

    $html = "Upload(name): " . $_FILES["file"]["name"] . "<br />";
    $html .= "Type: " . $_FILES["file"]["type"] . "<br />";
    $html .= "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
    $html .= "Stored in: " . $_FILES["file"]["tmp_name"] . "<br />";

    //□□□□□□□□□□□□□□□□
    $moveRes = move_uploaded_file($_FILES["file"]["tmp_name"],
    "/path_for_uploaded_file");

    $html .= "Uploaded file is moved to /path_for_uploaded_file". "<br />";
    echo $html;
  }
  else {
    echo "Error: " . $_FILES["file"]["error"] . "<br />";
  }

?>
```

□□□□□□□2□

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>□□ PHP □□□□ - □□□ function</title>
</head>
<body>

<form action="doAction.php" method="post" enctype="multipart/form-data">
    <!-- □□□□□□□□□□ -->
    <input type="hidden" name="MAX_FILE_SIZE" value="2097152">
```

```
    <!-- accept 限制上傳檔案類型 -->
    <input type="file" name="myFile" accept="image/jpeg,image/jpg,image/gif,image/png">

    <input type="submit" value="上傳檔案">
</form>

</body>
</html>
```

運作介面 (doAction.php)

```
<?php
/**
 * 表單接收頁面
 */
// 網頁編碼宣告 (防止產生亂碼)
header('content-type:text/html;charset=utf-8');
// 封裝好的單一 PHP 檔案上傳 function
include_once 'upload.func.php';
// 取得 HTTP 文件上傳變數
$fileInfo = $_FILES['myFile'];
// 呼叫封將好的 function
$newName = uploadFile($fileInfo);

print_r($newName);
```

運作介面2(upload.func.php)

```
<?php
/**
 * string uploadFile(array $files, array $allowExt, number $maxSize, boolean $flag, string $uploadPath) 單一 PHP 檔案上
 *
 * @param files 透過 $_FILES 取得的 HTTP 檔案上傳的項目陣列
 * @param allowExt 允許上傳檔案的擴展名, 預設 'jpeg', 'jpg', 'gif', 'png'
 * @param maxsize 上傳檔案容量大小限制, 預設 2097152 (2M * 1024 * 1024 = 2097152byte)
 * @param flag 檢查是否為真實的圖片類型 (只允許上傳圖片的話), true (預設) 檢查; false 不檢查
 * @param uploadPath 存放檔案的目錄, 預設 uploads
 *
 * @return 回傳存放目錄 + md5 產生的檔案名稱 + 擴展名
```

```
 */
function uploadFile($fileInfo, $allowExt = array('jpeg', 'jpg', 'gif', 'png'), $maxSize = 2097152, $flag = true,
$uploadPath = 'uploads') {
    // 存放錯誤訊息
    $mes = '';

    // 取得上傳檔案的擴展名
    $ext = pathinfo($fileInfo['name'], PATHINFO_EXTENSION);

    // 確保檔案名稱唯一, 防止重覆名稱產生覆蓋
    $uniName = md5(uniqid(microtime(true), true)) . '.' . $ext;
    $destination = $uploadPath . '/' . $uniName;

    // 判斷是否有錯誤
    if ($fileInfo['error'] > 0) {
        // 匹配的錯誤代碼
        switch ($fileInfo['error']) {
            case 1:
                $mes = '上傳的檔案超過了 php.ini 中 upload_max_filesize 允許上傳檔案容量的最大值';
                break;
            case 2:
                $mes = '上傳檔案的大小超過了 HTML 表單中 MAX_FILE_SIZE 選項指定的值';
                break;
            case 3:
                $mes = '檔案只有部分被上傳';
                break;
            case 4:
                $mes = '沒有檔案被上傳（沒有選擇上傳檔案就送出表單）';
                break;
            case 6:
                $mes = '找不到臨時目錄';
                break;
            case 7:
                $mes = '檔案寫入失敗';
                break;
            case 8:
                $mes = '上傳的文件被 PHP 擴展程式中斷';
                break;
        }
```

```
        exit($mes);
    }

    // 檢查檔案是否是通過 HTTP POST 上傳的
    if (!is_uploaded_file($fileInfo['tmp_name']))
        exit('檔案不是通過 HTTP POST 方式上傳的');

    // 檢查上傳檔案是否為允許的擴展名
    if (!is_array($allowExt))  // 判斷參數是否為陣列
        exit('檔案類型型態必須為 array');
    else {
        if (!in_array($ext, $allowExt))  // 檢查陣列中是否有允許的擴展名
            exit('非法檔案類型');
    }

    // 檢查上傳檔案的容量大小是否符合規範
    if ($fileInfo['size'] > $maxSize)
        exit('上傳檔案容量超過限制');

    // 檢查是否為真實的圖片類型
    if ($flag && !@getimagesize($fileInfo['tmp_name']))
        exit('不是真正的圖片類型');

    // 檢查指定目錄是否存在，不存在就建立目錄
    if (!file_exists($uploadPath))
        mkdir($uploadPath, 0777, true);  // 建立目錄

    // 將檔案從臨時目錄移至指定目錄
    if (!@move_uploaded_file($fileInfo['tmp_name'], $destination))  // 如果移動檔案失敗
        exit('檔案移動失敗');

    return $destination;
}
```

操作介面(多檔傳送)

```
<!DOCTYPE html>
<html>
```

```
<head>
    <meta charset="UTF-8">
    <title>PHP 多檔案上傳 - 封裝成 function</title>
</head>
<body>

<form action="doAction.php" method="post" enctype="multipart/form-data">
    <!-- 限制上傳檔案的最大值 -->
    <input type="hidden" name="MAX_FILE_SIZE" value="2097152">

    <!-- accept 限制上傳檔案類型。多檔案上傳 name 的屬性值須定義為 array -->
    <input type="file" name="myFile[]" accept="image/jpeg,image/jpg,image/gif,image/png" style="display:
block;margin-bottom: 5px;">
    <input type="file" name="myFile[]" accept="image/jpeg,image/jpg,image/gif,image/png" style="display:
block;margin-bottom: 5px;">
    <input type="file" name="myFile[]" accept="image/jpeg,image/jpg,image/gif,image/png" style="display:
block;margin-bottom: 5px;">
    <input type="file" name="myFile[]" accept="image/jpeg,image/jpg,image/gif,image/png" style="display:
block;margin-bottom: 5px;">

    <!-- 使用 html 5 實現單一上傳框可多選檔案方式，須新增 multiple 元素 -->
    <!-- <input type="file" name="myFile[]" id="" accept="image/jpeg,image/jpg,image/gif,image/png"
multiple> -->

    <input type="submit" value="上傳檔案">
</form>

</body>
</html>
```

運作介面 (doAction.php)

```
<?php
/**
 * 表單接收頁面
 */
// 網頁編碼宣告 (防止產生亂碼)
header('content-type:text/html;charset=utf-8');
// 封裝好的 PHP 多檔案上傳 function
include_once 'upload.func.php';
```

```
// □□□□□□□□ array □□
$files = getFiles();

// □□□□□□□□
foreach ($files as $fileInfo) {
    // □□□□□□ function
    $res = uploadFile($fileInfo);

    // □□□□□□□□
    echo $res['mes'] . '<br>';

    // □□□□□□□□□□□□□□ array□□□□□□□□□
    if (!empty($res['dest'])) {
        $uploadFiles[] = $res['dest'];
    }
}

print_r($uploadFiles);
```

□□□□□upload.func.php□

```
<?php
/**
 * array getFiles() □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ array □□
 *
 * @return □□□□□□□□ array □□
 */function getFiles() {
    $i = 0;  // □□ array □□

    foreach ($_FILES as $file) {
        // string □□□□□□□□□□□
        if (is_string($file['name'])) {
            $files[$i] = $file;
            $i++;
        }
        // array □□□□□□□□□□□
        elseif (is_array($file['name'])) {
            foreach ($file['name'] as $key => $value) {
                $files[$i]['name'] = $file['name'][$key];
                $files[$i]['type'] = $file['type'][$key];
```

```
            $files[$i]['tmp_name'] = $file['tmp_name'][$key];
            $files[$i]['error'] = $file['error'][$key];
            $files[$i]['size'] = $file['size'][$key];
            $i++;
        }
    }
  }

  return $files;
}

/**
 * string uploadFile(array $files, array $allowExt, number $maxSize, boolean $flag, string $uploadPath) PHP □□□□□
 *
 * @param files □□ $_FILES □□□ HTTP □□□□□□□□□
 * @param allowExt □□□□□□□□□□□□□ 'jpeg', 'jpg', 'gif', 'png'
 * @param maxsize □□□□□□□□□□□□□ 2097152□2M * 1024 * 1024 = 2097152byte□
 * @param flag □□□□□□□□□□□□□□□□□□□□□□□□□□□true□□□□□□□false □□□
 * @param uploadPath □□□□□□□□□□ uploads
 *
 * @return □□□□□□ + md5 □□□□□□□ + □□□
 */function uploadFile($fileInfo, $allowExt = array('jpeg', 'jpg', 'gif', 'png'), $maxSize = 2097152, $flag = true,
$uploadPath = 'uploads') {
  // □□□□□□
  $res = array();

  // □□□□□□□□□□
  $ext = pathinfo($fileInfo['name'], PATHINFO_EXTENSION);

  // □□□□□□□□□□□□□□□□□□□□□
  $uniName = md5(uniqid(microtime(true), true)) . '.' . $ext;
  $destination = $uploadPath . '/' . $uniName;

  // □□□□□□□
  if ($fileInfo['error'] > 0) {
    // □□□□□□□
    switch ($fileInfo['error']) {
      case 1:
        $res['mes'] = $fileInfo['name'] . ' □□□□□□□□ php.ini □ upload_max_filesize □□□□□□□□□□□□';
        break;
```

```
        case 2:
            $res['mes'] = $fileInfo['name'] . ' 上傳檔案的大小超過了 HTML 表單中 MAX_FILE_SIZE 選項指定的值';
            break;
        case 3:
            $res['mes'] = $fileInfo['name'] . ' 檔案只有部分被上傳';
            break;
        case 4:
            $res['mes'] = $fileInfo['name'] . ' 沒有檔案被上傳（沒有選擇上傳檔案就送出表單）';
            break;
        case 6:
            $res['mes'] = $fileInfo['name'] . ' 找不到臨時目錄';
            break;
        case 7:
            $res['mes'] = $fileInfo['name'] . ' 檔案寫入失敗';
            break;
        case 8:
            $res['mes'] = $fileInfo['name'] . ' 上傳的文件被 PHP 擴展程式中斷';
            break;
    }

    // 直接 return 無需在往下執行
    return $res;
}

// 檢查檔案是否是通過 HTTP POST 上傳的
if (!is_uploaded_file($fileInfo['tmp_name']))
    $res['mes'] = $fileInfo['name'] . ' 檔案不是通過 HTTP POST 方式上傳的';

// 檢查上傳檔案是否為允許的擴展名
if (!is_array($allowExt))  // 判斷參數是否為陣列
    $res['mes'] = $fileInfo['name'] . ' 檔案類型型態必須為 array';
else {
    if (!in_array($ext, $allowExt))  // 檢查陣列中是否有允許的擴展名
        $res['mes'] = $fileInfo['name'] . ' 非法檔案類型';
}

// 檢查上傳檔案的容量大小是否符合規範
if ($fileInfo['size'] > $maxSize)
    $res['mes'] = $fileInfo['name'] . ' 上傳檔案容量超過限制';
```

```
    // 檢查是否為真實的圖片類型
    if ($flag && !@getimagesize($fileInfo['tmp_name']))
        $res['mes'] = $fileInfo['name'] . ' 不是真正的圖片類型';

    // array 有值表示上述其中一項檢查有誤，直接 return 無需在往下執行
    if (!empty($res))
        return $res;
    else {
        // 檢查指定目錄是否存在，不存在就建立目錄
        if (!file_exists($uploadPath))
            mkdir($uploadPath, 0777, true);

        // 將檔案從臨時目錄移至指定目錄
        if (!@move_uploaded_file($fileInfo['tmp_name'], $destination))  // 如果移動檔案失敗
            $res['mes'] = $fileInfo['name'] . ' 檔案移動失敗';

        $res['mes'] = $fileInfo['name'] . ' 上傳成功';
        $res['dest'] = $destination;

        return $res;
    }
}
```

操作介面（多檔案）

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>PHP 多檔案上傳 - 封裝成 class</title>
</head>
<body>

<form action="doAction.php" method="post" enctype="multipart/form-data">
    <!-- 限制上傳檔案的最大值 -->
    <input type="hidden" name="MAX_FILE_SIZE" value="2097152">

    <!-- accept 限制上傳檔案類型。多檔案上傳 name 的屬性值須定義為 array -->
    <input type="file" name="myFile[]" accept="image/jpeg,image/jpg,image/gif,image/png" style="display:
block;margin-bottom: 5px;">
```

```
    <input type="file" name="myFile[]" accept="image/jpeg,image/jpg,image/gif,image/png" style="display:
block;margin-bottom: 5px;">
    <input type="file" name="myFile[]" accept="image/jpeg,image/jpg,image/gif,image/png" style="display:
block;margin-bottom: 5px;">
    <input type="file" name="myFile[]" accept="image/jpeg,image/jpg,image/gif,image/png" style="display:
block;margin-bottom: 5px;">

    <!-- 使用 html 5 實現單一上傳框可多選檔案方式, 須新增 multiple 元素 -->
    <!-- <input type="file" name="myFile[]" id="" accept="image/jpeg,image/jpg,image/gif,image/png"
multiple> -->

    <input type="submit" value="上傳檔案">
</form>

</body>
</html>
```

運作介面(doAction.php)

```
<?php
/**
 * 表單接收頁面
 */
// 網頁編碼宣告 (防止產生亂碼)
header('content-type:text/html;charset=utf-8');
// 封裝好的 PHP 多檔案上傳 class
include_once 'upload.class.php';

$upload = new Upload();
$upload->callUploadFile();

echo $upload->getDestination();  // 取得實際儲存檔名路徑
```

運作介面 (upload.class.php)

```
<?php
/**
 * 單一及多檔案上傳
 *
 * @author  smalljacky
```

```
 * @version 1.0
 */
class Upload
{
    /**
     * 檢查上傳檔案是否為允許的類型
     *
     * @var array
     */    private $allowMIME;

    /**
     * 允許上傳檔案的擴展名
     *
     * @var array
     */    private $allowExt;

    /**
     * 上傳檔案容量大小限制
     *
     * @var int
     */    private $maxSize;

    /**
     * 檢查是否為真實的圖片類型（只允許上傳圖片的話）
     *
     * @var boolean
     */    private $flag;

    /**
     * 存放檔案的目錄
     *
     * @var string
     */    private $uploadPath;

    /**
     * $_FILES 取得的 HTTP 檔案上傳項目
     *
     * @var array
     */    private $fileInfo;
```

```
    /**
     * □□□□□□
     *
     * @var array
     */    private $res;

    /**
     * □□□□□□□□
     *
     * @var array
     */    private $uploadFiles;

    /**
     * □□□□□
     *
     * @var string
     */    private $ext;

    /**
     * □□□□□□
     *
     * @param  array $allowMIME
     * @param  array $allowExt
     * @param  int $maxSize
     * @param  boolean $flag
     * @param  string $uploadPath
     */    public function __construct(array $allowMIME = array('image/jpeg', 'image/png', 'image/gif'), array $allowExt = array('jpeg', 'jpg', 'gif', 'png'), $maxSize = 2097152, $flag = true, $uploadPath = 'uploads')
    {
        $this->fileInfo = $this->getFiles();
        $this->allowMIME = $allowMIME;
        $this->allowExt = $allowExt;
        $this->maxSize = $maxSize;
        $this->flag = $flag;
        $this->uploadPath = $uploadPath;
    }

    /**
     * □□□□□□□□ array
     *
```

```
 * @return void
 */    public function callUploadFile()
{
    $res = '';

    foreach ($this->fileInfo as $file) {
        $res = $this->uploadFile($file);
        $this->showMessage();   // □□□□□□

        if (!empty($this->res['dest'])) {
            $this->uploadFiles[] = $res['dest'];
        }

        $this->res = array();  // □□□□□□
    }
}

/**
 * □□□□□□□□□□
 *
 * @return array
 */    public function getDestination()
{
    if (!empty($this->uploadFiles)) {
        print_r($this->uploadFiles);
    }
}

/**
 * □□□□□□□□□□□□□□□□□□□□□□ array
 *
 * @return array
 */    protected function getFiles()
{
    $i = 0;  // □□ array □□

    foreach ($_FILES as $file) {
        // string □□□□□□□□□□
        if (is_string($file['name'])) {
            $files[$i] = $file;
```

```
            $i++;
        }
        // array □□□□□□□□□□
        elseif (is_array($file['name'])) {
            foreach ($file['name'] as $key => $value) {
                $files[$i]['name'] = $file['name'][$key];
                $files[$i]['type'] = $file['type'][$key];
                $files[$i]['tmp_name'] = $file['tmp_name'][$key];
                $files[$i]['error'] = $file['error'][$key];
                $files[$i]['size'] = $file['size'][$key];
                $i++;
            }
        }
        return $files;
    }
}

/**
 * □□□□□□□□□□□□□□□□ + md5 □□□□□□□ + □□□
 *
 * @return array
 */    private function uploadFile($file)
{
    $uniName = '';
    $destination = '';

    if ($this->checkError($file) && $this->checkHttpPost($file) && $this->checkMIME($file) && $this-
>checkExt($file) && $this->checkSize($file) && $this->checkTrueImg($file)) {
        $this->checkUploadPath();
        $uniName = $this->getUniName();
        $destination = $this->uploadPath . '/' . $uniName . '.' . $this->ext;

        if (!@move_uploaded_file($file['tmp_name'], $destination)) {
            $this->res['error'] = $file['name'] . '□□□□□□';
        } else {
            $this->res['succ'] = $file['name'] . '□□□□';
            $this->res['dest'] = $destination;
        }
    }
    return $this->res;
```

```
}

/**
 * □□□□□□□□□□□
 *
 * @param  array $files □□ $_FILES □□□ HTTP □□□□□□□ array
 * @return boolean
 */    protected function checkError($file)
{
    if ($file['error'] > 0) {
        switch ($file['error']) {
            case 1:
                $this->res['error'] = $file['name'] . ' □□□□□□□ php.ini □ upload_max_filesize □□□□□□□□□□□';
                break;
            case 2:
                $this->res['error'] = $file['name'] . ' □□□□□□□□□ HTML □□□ MAX_FILE_SIZE □□□□□□';
                break;
            case 3:
                $this->res['error'] = $file['name'] . ' □□□□□□□□';
                break;
            case 4:
                $this->res['error'] = $file['name'] . ' □□□□□□□□□□□□□□□□□□□□□□□';
                break;
            case 6:
                $this->res['error'] = $file['name'] . ' □□□□□□□';
                break;
            case 7:
                $this->res['error'] = $file['name'] . ' □□□□□□';
                break;
            case 8:
                $this->res['error'] = $file['name'] . ' □□□□□□ PHP □□□□□□';
                break;
        }
        return false;
    }
    return true;
}

/**
 * □□□□□□□□ HTTP POST □□□
```

```
     *
     * @param  array $files □□ $_FILES □□□ HTTP □□□□□□□ array
     * @return boolean
     */    private function checkHttpPost($file)
    {
        if (!is_uploaded_file($file['tmp_name'])) {
            $this->res['error'] = $file['name'] . '□□□□□□ HTTP POST □□□□□';
            return false;
        }
        return true;
    }

    /**
     * □□□□□□□□□□□□□□□
     *
     * @param  array $files □□ $_FILES □□□ HTTP □□□□□□□ array
     * @return boolean
     */    private function checkMIME($file)
    {
        if (!in_array($file['type'], $this->allowMIME)) {
            $this->res['error'] = $file['name'] . '□□□□□□□□□';
            return false;
        }
        return true;
    }

    /**
     * □□□□□□□□□□□□□□□
     *
     * @param  array $files □□ $_FILES □□□ HTTP □□□□□□□ array
     * @return boolean
     */    private function checkExt($file)
    {
        $this->ext = pathinfo($file['name'], PATHINFO_EXTENSION);  // □□□□□□□□□□

        // □□□□□□□□□□□□□□□□□□□□□□□□□□□□
        if (!is_array($this->allowExt)) {
            $this->res['error'] = $file['name'] . ' □□□□□□□□ array';
            return false;
        } else {
```

```
        // □□□□□□□□□□□□□□□
        if (!in_array($this->ext, $this->allowExt)) {
            $this->res['error'] = $file['name'] . ' □□□□□□';
            return false;
        }
    }
    return true;
}

/**
 * □□□□□□□□□□□□□□□□□□
 *
 * @param  array $files □□ $_FILES □□□ HTTP □□□□□□□ array
 * @return boolean
 */    private function checkSize($file)
{
    if ($file['size'] > $this->maxSize) {
        $this->res['error'] = $file['name'] . '□□□□□□□□□□';
        return false;
    }
    return true;
}

/**
 * □□□□□□□□□□□□
 *
 * @param  array $files □□ $_FILES □□□ HTTP □□□□□□□ array
 * @return boolean
 */    private function checkTrueImg(array $file)
{
    if ($this->flag) {
        if (!@getimagesize($file['tmp_name'])) {
            $this->res['error'] = $file['name'] . '□□□□□□□□□';
            return false;
        }
        return true;
    }
}

/**
```

```
    * □□□□□□□□□□□□□□□□□□□□
    *
    * @return void
    */    private function checkUploadPath()
    {
        if (!file_exists($this->uploadPath)) {
            mkdir($this->uploadPath, 0777, true);
        }
    }

    /**
    * □□□□□□□□□
    *
    * @return string
    */    private function getUniName()
    {
        return md5(uniqid(microtime(true), true));
    }

    /**
    * □□□□□□
    *
    * @return void
    */    private function showMessage()
    {
        if (!empty($this->res['error'])) {
            echo '<span style="color: #ff0000;">' . $this->res['error'] . '</span><br />';
        } else {
            echo '<span style="color: #0000ff;">' . $this->res['succ'] . '</span><br />';
        }
    }
}
<img
  src="images/dinosaur.jpg"
  alt="The head and torso of a dinosaur skeleton;
         it has a large head with long sharp teeth"
  width="400"
  height="341"
  title="A T-Rex on display in the Manchester University Museum" />
```

background的設定方式

background-repeat:repeat; ＃預設
值：no-repeat 不重複;repeat-x x軸重複;repeat-y y軸重複;space round space：盡可能在不裁切、不變形、不縮放圖片的情況下，最大量化的重複背景圖片，若剩餘空間無法拼接，則自動均分空白空間。 round：使用縮放或變形的方式，讓背景圖片在不裁切的情況下填滿空間。

background-position:top left;靠左上角 ＃預設值：top靠上、right靠右、bottom靠底、left靠左、center置中

background-position:X-axie y-axie;
設定xy軸的位置，可以使用%跟px等單位。

background-position: bottom 50px right 100px; 如 padding: background-position: 底部推50px 右推100px;

background-size:contain;
＃圖片高度自適應容器高度如有設定容器寬度,圖片寬度達不到容器寬度就重複顯示
值：cover 自適應容器;30% 原本圖片的30％;200px 100px 直接設定寬高

縮寫：
* {
background: url(“xxx")green no-repeat right top; background-size: 100px;
}

. image{
height:380px; 【設定高度】
background-image:url(“https://背景圖片網址.png");【背景圖連結】 background-position: center;【圖片置中,適合圖片中間為重點的圖】 background-size: cover;【自適應容器】
background-repeat: no-repeat;【不重複】
}

background-image: url("amos.png");
background-image: url("01.jpg"), url("02.jpg"), url("03.jpg"); ＃可以設多張圖
background-repeat: no-repeat;
background-position: left top, right bottom, center center;

background-image: linear-gradient(to 目標方向, 起始色彩, 結束色彩); #漸層 div{
width: 200px;
height: 200px;
background-image: linear-gradient(to left top, red,yellow); }

background-image: linear-gradient(角度, 起始色彩 起始色彩位置, 結束色彩 結束色彩位置); background-image: linear-gradient(115deg, #F00 50%, #FF0 50%); background-image: linear-gradient(115deg, #F00 50%, transparent 50%); #50%變空白

background-attachment: scroll;
值：fixed local

background: gray url("amos.png") scroll no-repeat 50% 50% / 50% 50%;

border: 線條尺寸 線條樣式 線條色彩;
outline: 線條尺寸 線條樣式 線條色彩;
EX:

border: 1px solid red;
outline: 1px solid red;

border
border-width
border-style
border-color

上邊框線(可以再分上下左右)
border-top-width: 1px;
border-top-style: solid;
border-top-color: red;

精簡寫法
border-top: 1px solid red;
border-right: 1px solid red;
border-bottom: 1px solid red; border-left: 1px solid red;

精簡寫法2
border-width: 1px 1px 1px 1px; border-style: solid solid solid solid; border-color: red red red red;

border-width: 上邊框粗細 右邊框粗細 下邊框粗細 左邊框粗細; border-style: 上邊框樣式 右邊框樣式 下邊框樣式 左邊框樣式; border-color: 上邊框色彩 右邊框色彩 下邊框色彩 左邊框色彩;

/*四個值，個別指定*/
border-width: 上 右 下 左;
/*三個值，左右採用同一個值*/
border-width: 上 [右左] 下;
/*二個值，上下採用同一值，左右採用同一值*/
border-width: [上下] [右左];

/*一個值，上下左右都使用相同的值*/
border-width: [上右下左];

outline
outline-width
outline-style
outline-color
可以分開特性寫

list-style: list-style-type list-style-position list-style-image; list-style: 清單符號 符號位置 清單符號圖片;
list-style: circle outside url("gold-fish.png");

list-style-type的參數
none (沒有符號)cjk-ideographic (一、二、三)
decimal (1、2、3)decimal-leading-zero (01、02、03)
lower-alpha (a、b、c)upper-alpha (A、B、C)
lower-roman (i、ii、iii)upper-roman (I、II、III)
initial

position: relative;
static (靜態定位),relative (相對定位),fixed (固定定位),absolute (絕對定位),sticky (黏貼定位)

---粗體字的使用---
<p>
<strong style="color:#f00;">床前明月光，疑是地上霜，舉頭望明月，低頭思故鄉。 </p>
<p>
<strong style="color:#f90;">春眠不覺曉，處處聞啼鳥，夜來風雨聲，花落知多少。 </p>
<p>
<strong style="color:#00f;">千山鳥飛絕，萬徑人蹤滅，孤舟蓑笠翁，獨釣寒江雪。 </p>
備註：也可以使用<b>

SEO設定

1. Title 標籤
<title>頁面標題</title>
<meta name="description" content="頁面的描述">　(60字以內)
標題標籤是 HTML 標籤, HTML <h1>–<h6> 標題 (Heading) 元素呈現了六個不同的級別的標題, <h1> 級別最高, 而 <h6> 級別最低。它可以用來識別內容的結構

圖片的 alt 屬性被添加到<img />標籤以描述其內容

<META NAME="ROBOTS" CONTENT="INDEX,FOLLOW">: 可以抓取本頁，而且可以順著本頁繼續索引別的連結
<META NAME="ROBOTS" CONTENT="NOINDEX,FOLLOW">: 不許抓取本頁，但是可以順著本頁抓取索引別的連結
<META NAME="ROBOTS" CONTENT="INDEX,NOFOLLOW">: 可以抓取本頁，但是不許順著本頁抓取索引別的連結
<META NAME="ROBOTS" CONTENT="NOINDEX,NOFOLLOW">: 不許抓取本頁，也不許順著本頁抓取索引別的連結。

SVG□□

SVG Circle

Sorry, your browser does not support inline SVG.

Example

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

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>

</body>
</html>
```

SVG Rectangle

Sorry, your browser does not support inline SVG.

Example

```
<svg width="400" height="120">
  <rect x="10" y="10" width="200" height="100" stroke="red" stroke-width="6" fill="blue" />
</svg>
```

SVG Rectangle with Opacity and Rounded Corners

Sorry, your browser does not support inline SVG.

Example

```
<svg width="400" height="180">
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"
  style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
</svg>
```

SVG Star

Sorry, your browser does not support inline SVG.

Example

```
<svg width="300" height="200">
  <polygon points="100,10 40,198 190,78 10,78 160,198"
  style="fill:lime;stroke:purple;stroke-width:5;fill-rule:evenodd;" />
</svg>
```

SVG Gradient Ellipse and Text

Example

```
<svg height="130" width="500">
  <defs>
    <linearGradient id="grad1">
```

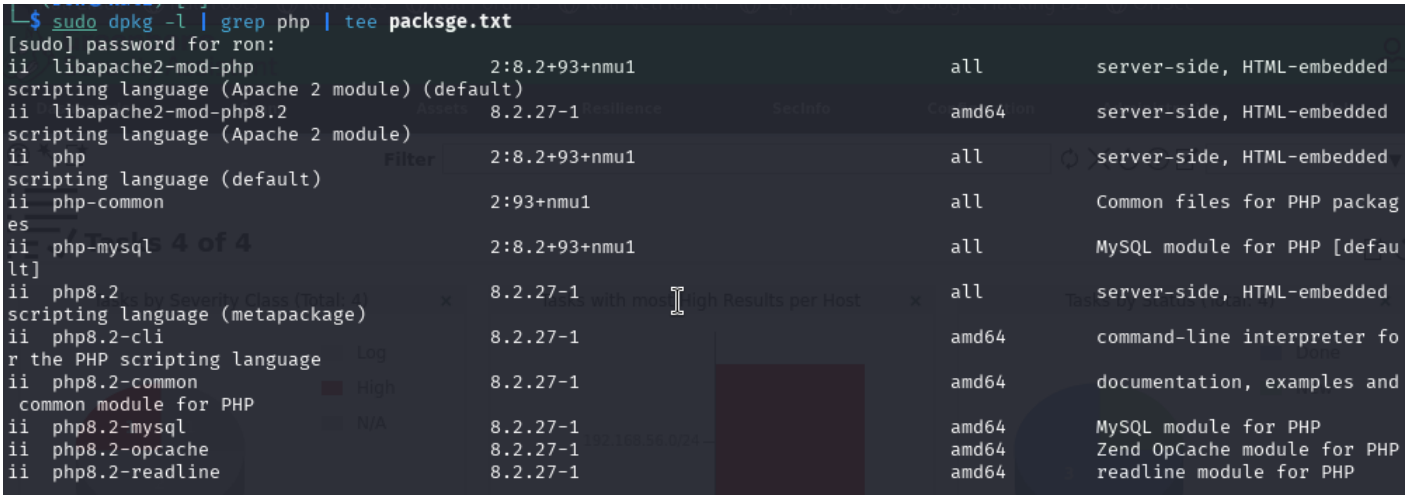
```
    <stop offset="0%" stop-color="yellow" />
    <stop offset="100%" stop-color="red" />
   </linearGradient>
  </defs>
  <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />
  <text fill="#ffffff" font-size="45" font-family="Verdana" x="50" y="86">SVG</text>
  Sorry, your browser does not support inline SVG.
</svg>
```

php□□□□

PHP□□□□□□

□□□□php□□□□□□□□□□

```
sudo dpkg -l | grep php | tee packages.txt
```



□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□

debian

```
sudo apt install apt-transport-https
sudo curl -sSLo /usr/share/keyrings/deb.sury.org-php.gpg https://packages.sury.org/php/apt.gpg
sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/deb.sury.org-php.gpg] https://packages.sury.org/php/
$(lsb_release -sc) main" > /etc/apt/sources.list.d/php.list'
sudo apt update
```

Ubuntu

```
sudo add-apt-repository ppa:ondrej/php
sudo apt update
```

□□□□php

```
sudo apt install php8.3 php8.3-cli php8.3-{bz2,curl,mbstring,intl}
```

備註：視需要做所需要安裝的**php**套件增減。

安裝跟nginx或apache運作的php模組

nginx

```
sudo apt install php8.3-fpm
```

備註：完成後需要在**nginx**設定好才會正常運作

apache

```
sudo apt install libapache2-mod-php8.3
```

備註：安裝後需要啟用新版，關閉舊版

檢視新的版本是否正常運作

nginx

```
sudo systemctl status php8.3-fpm.service
```

移除舊版

```
sudo apt purge php8.2*
```

效能相關設定

JIT設定（設定php.ini）

```
opcache.jit = 0    # 關閉
opcache.jit = 1235 # 適合一般 web（官方預設）
opcache.jit = 1256 # 更積極的編譯
```

```
[opcache]
opcache.enable=1
opcache.enable_cli=1     ; 若你希望 CLI 也加速
opcache.jit=1235         ; 官方推薦的 JIT 等級
opcache.jit_buffer_size=128M
```

確認是否啟用

```
php -i | grep JIT
```

也可以看

```
<?php phpinfo(); ?>
```

主要差異

- 科學計算
- 圖像處理
- 迴圈密集計算
- 伺服器長時間執行的 daemon
- 自動化、ETL 等非 Web 工作

原因: JIT 只加速 CPU-bound, 而 Web 程式大多是 I/O bound (資料庫 / 檔案 / 網路)。

一般差異

APCu設定

```
[apcu]
apc.enabled=1
apc.shm_size=128M
apc.enable_cli=0
```

參數說明:

- **apc.enabled=1** → 啟用 APCu
- **apc.shm_size=128M** → 設定分享記憶體大小 (快取容量)
  - 小網站: 32M
  - 中型: 64M
  - 大型: 128M~256M
- **apc.enable_cli** → 是否讓 CLI 也使用 APCu
  - [0]建議免干擾 CLI)

參考資料

https://php.watch/articles/php-8.3-install-upgrade-on-debian-ubuntu#php83-ubuntu-quick