

HTML

[php](#) [css](#) [js](#) [bootscrip](#)

-
- [SEO](#)
- [SVG](#)
- [php](#)



php.ini

php

php

```
php -m
```

```
<?php
?>
```



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



```
ceil(x)
```



```
floor(x)
```



```
;
upload_max_filesize = 50M

; POST
post_max_size = 200M

;
memory_limit = 512M
```

> POST >

```
; PHP
max_execution_time = 600
```

```
; PHP
max_input_time = 600
```

```
; MySQL
mysql.connect_timeout = -1
```

php config

```
;Opcache
opcache.enable = 1
```

```
;Opcache
opcache.memory_consumption = 256
```

```
;
;PHP
;
;1000
;999
;
;php-fpm
;php-fpm
;
;megabytes
;16
;16MB
;(default "4")
opcache.interned_strings_buffer = 8
```

```
;PHP
;PHP
;200 PHP 5.5.6
;PHP 5.5.6
;(default "2000")
opcache.max_accelerated_files = 4000
```

```
;Opcache
;
;"/"1"
```

```

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

;
;opcache.memory_consumption=8M
;opcache.interned_strings_buffer=8M
;opcache.max_wasted_percentage=5
;opcache.enable_cli=0
;opcache.enable_file_override=0
;opcache.preload_library=0
;opcache.preload=

;(default "2")
opcache.revalidate_freq = 60

;
;Zend OPcache
;

;opcache.preload_php_files=

;(default "0")
opcache.fast_shutdown = 1

;CLI OPcache
;opcache.enable_cli=1

;
;opcache.enable=0
;opcache.memory_consumption=8M
;(default "1")
opcache.use_cwd = 0

opcache.max_file_size = 0

```

PHP (memory_limit)

PHP memory_limit

php.ini

```
memory_limit = 256M
```

.htaccess

```
php_value memory_limit 512M
```

memory_limit

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

memory_limit

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

memory limit

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

phpinfo

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

php

```
<?php  
// PHP ...  
// PHP ...  
?>
```

```
<?php  
/*  
... PHP  
... PHP  
... PHP  
*/  
?>
```

```
<font size="1" color="blue"> 1 </font><br>
```

```
<a href="/html/image-img-tag.html"></a> Fooish  
<a href="#" style="color:red;" target="" rel=""></a>
```

HTML CSS

```
<style>
a {
  color:green;
}
</style>
<a href="#"></a>
```

```
<table 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;">
</DIV>
```

<a> (attributes)

href
URL target

target

target

- _self:
- _blank:
- _parent:
- _top:
- framename:

rel

- nofollow
- external target="_blank" _blank xhtml
- noopener target="_blank" window.opener.location = newURL
- noreferrer noopener ex
- noreferrer noopener SEO

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


□□□□□□□□ 1/2 □□□□

 

□□□□□□□□ 1 □□□□

□□□□

□ **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.

```

XXXXXXXXXX

```

```

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

```

PHP 5.3 DateTime() TimeInterval()

```

XXXX2XXX

```

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

```

```

XXXXXXXXXXXXXXXXXX

```

```

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;
?>
```

With 2 decimals: 123.00

With no decimals: 123

□□□□□□

php.ini display_errors = on

span□□

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

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

align="□□" right left center

```
<table style="border:3px #cccccc solid;" cellpadding="10" border='1'>
<tr><td>□□□□</td><td>□□□□</td></tr>
<tr><td>□□□□</td><td>□□□□</td></tr>
</table>
```

```
<table style="border:3px #FFD382 dashed;" cellpadding="10" border='1'>
<tr><td>□□□□</td><td>□□□□</td></tr>
<tr><td>□□□□</td><td>□□□□</td></tr>
</table>
```

```
<table style="border:8px #FFD382 groove;" cellpadding="10" border='0'>
<tr><td>□□□□</td><td>□□□□</td></tr>
<tr><td>□□□□</td><td>□□□□</td></tr>
</table>
```

border-top ← □□□□□□

border-right ← □□□□□□

border-bottom ← □□□□□□

border-left ← □□□□□□

border-color:□□□□

border-radius: 10px 20px 30px 40px;

border-style□□□□□□style□□□□□□

none(□□)

dotted

dashed


```
</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 [ ] -->
<input type="file" name="myFile" accept="image/jpeg,image/jpg,image/gif,image/png">

<input type="submit" value="[ ]">
</form>

</body>
</html>

```

[]doAction.php

```

<?php
/**
 * [ ]
 */
// [ ]
header('content-type:text/html;charset=utf-8');
// [ ] PHP [ ] function
include_once 'upload.func.php';
// [ ] HTTP [ ]
$fileInfo = $_FILES['myFile'];
// [ ] function
$newName = uploadFile($fileInfo);

print_r($newName);

```

[]2(upload.func.php)

```

<?php
/**
 * 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') {
    // []
    $mes = "";

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

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

    // []
    if ($fileInfo['error'] > 0) {
        // []
        switch ($fileInfo['error']) {
            case 1:
                $mes = '[] php.ini [] upload_max_filesize []';
                break;
            case 2:
                $mes = '[] HTML [] MAX_FILE_SIZE []';
                break;
            case 3:
                $mes = '[]';
                break;
            case 4:
                $mes = '[]';
                break;
            case 6:
                $mes = '[]';
                break;
            case 7:
                $mes = '[]';
                break;
            case 8:
                $mes = '[] PHP []';
                break;
        }
    }
}

```

```

    exit($mes);
}

// 检查 HTTP POST 数据
if (!is_uploaded_file($fileInfo['tmp_name']))
    exit('非法 HTTP POST 数据');

// 检查文件扩展名
if (!is_array($allowExt)) // 检查是否数组
    exit('非法扩展名 array');
else {
    if (!in_array($ext, $allowExt)) // 检查是否在允许列表中
        exit('非法扩展名');
}

// 检查文件大小
if ($fileInfo['size'] > $maxSize)
    exit('文件大小超过限制');

// 检查文件类型
if ($flag && !@getimagesize($fileInfo['tmp_name']))
    exit('非法文件类型');

// 检查上传目录是否存在
if (!file_exists($uploadPath))
    mkdir($uploadPath, 0777, true); // 创建目录

// 检查文件是否移动成功
if (!@move_uploaded_file($fileInfo['tmp_name'], $destination)) // 移动失败
    exit('文件移动失败');

return $destination;
}

```

检查文件类型

```

<!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 [ ] name [ ] 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;">

  <!-- [ ] html 5 [ ] multiple [ ] -->
  <!-- <input type="file" name="myFile[]" id="" accept="image/jpeg,image/jpg,image/gif,image/png"
multiple> -->

  <input type="submit" value="[ ]">
</form>

</body>
</html>

```

[]doAction.php

```

<?php
/**
 * [ ]
 */
// [ ]
header('content-type:text/html;charset=utf-8');
// [ ] PHP [ ] 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'] . '  HTML  MAX_FILE_SIZE  ';
    break;
case 3:
    $res['mes'] = $fileInfo['name'] . '  ';
    break;
case 4:
    $res['mes'] = $fileInfo['name'] . '  ';
    break;
case 6:
    $res['mes'] = $fileInfo['name'] . '  ';
    break;
case 7:
    $res['mes'] = $fileInfo['name'] . '  ';
    break;
case 8:
    $res['mes'] = $fileInfo['name'] . '  PHP  ';
    break;
}

```

```

//  return
return $res;
}

```

```

//  HTTP POST
if (!is_uploaded_file($fileInfo['tmp_name']))
    $res['mes'] = $fileInfo['name'] . '  HTTP POST  ';

```

```

//
if (!is_array($allowExt)) //
    $res['mes'] = $fileInfo['name'] . '  array';
else {
    if (!in_array($ext, $allowExt)) //
        $res['mes'] = $fileInfo['name'] . '  ';
}

```

```

//
if ($fileInfo['size'] > $maxSize)
    $res['mes'] = $fileInfo['name'] . '  ';

```

```

// []
if ($flag && !@getimagesize($fileInfo['tmp_name']))
    $res['mes'] = $fileInfo['name'] . ' []';

// array [] return []
if (!empty($res))
    return $res;
else {
    // []
    if (!file_exists($uploadPath))
        mkdir($uploadPath, 0777, true);

    // []
    if (!@move_uploaded_file($fileInfo['tmp_name'], $destination)) // []
        $res['mes'] = $fileInfo['name'] . ' []';

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

    return $res;
}
}

```

[][][]

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>PHP [] - [] class</title>
</head>
<body>

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

    <!-- accept [] name [] 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;">

<!-- [ ] html 5 [ ] multiple [ ] -->
<!-- <input type="file" name="myFile[]" id="" accept="image/jpeg,image/jpg,image/gif,image/png"
multiple> -->

<input type="submit" value="[ ]">
</form>

</body>
</html>

```

[](doAction.php)

```

<?php
/**
 * [ ]
 */
// [ ]
header('content-type:text/html;charset=utf-8');
// [ ] PHP [ ] class
include_once 'upload.class.php';

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

echo $upload->getDestination(); // [ ]

```

[]upload.class.php

```

<?php
/**
 * [ ]
 *
 * @author smalljacky

```

```
* @version 1.0
*/
class Upload
{
  /**
   * ██████████
   *
   * @var array
   */ private $allowMIME;

  /**
   * ████████
   *
   * @var array
   */ private $allowExt;

  /**
   * ████████
   *
   * @var int
   */ private $maxSize;

  /**
   * ████████████████████████████████
   *
   * @var boolean
   */ private $flag;

  /**
   * ████████
   *
   * @var string
   */ private $uploadPath;

  /**
   * $_FILES HTTP ████████
   *
   * @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;

```

```

        $i++;
    }
    // array [0][0][0][0][0][0][0][0][0][0]
    elseif (is_array($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;
}
}

/**
 * [0][0][0][0][0][0][0][0][0][0] + md5 [0][0][0][0][0] + [0][0]
 *
 * @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'] . '[0][0][0][0]';
        } else {
            $this->res['succ'] = $file['name'] . '[0][0][0]';
            $this->res['dest'] = $destination;
        }
    }
}
return $this->res;

```



```

*
* @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'] . ' MIME';
        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 />';
    }
}
}


```

background[]

```
background-repeat:repeat; [ ]  
[ ]no-repeat [ ];repeat-x x[ ];repeat-y y[ ];space round  
space[ ]  
round[ ]
```

```
background-position:top left;[ ] [ ]  
[ ]top[ ]right[ ]bottom[ ]left[ ]center[ ]
```

```
background-position:X-axie y-axie;  
[ ]xy[ ]%[ ]px[ ]
```

```
background-position: bottom 50px right 100px;  
[ ]padding: background-position: [ ]50px [ ]100px;
```

```
background-size:contain;  
[ ],[ ]  
[ ]cover [ ];30% [ ]30%;200px 100px [ ]
```

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

```
.image{  
height:380px; [ ]  
background-image:url("https://[ ].png");[ ]  
background-position: center;[ ],[ ]  
background-size: cover;[ ]  
background-repeat: no-repeat;[ ]  
}
```

```
background-image: url("amos.png");  
background-image: url("01.jpg"), url("02.jpg"), url("03.jpg");  
[ ]  
background-repeat: no-repeat;  
background-position: left top, right bottom, center center;
```

```
background-image: linear-gradient(to [ ], [ ], [ ]); # [ ]  
div{  
width: 200px;  
height: 200px;  
background-image: linear-gradient(to left top, red,yellow);  
}
```

```
background-image: linear-gradient(to top, #000 0%, #000 100%, #000 100%);
background-image: linear-gradient(115deg, #F00 50%, #FF0 50%);
background-image: linear-gradient(115deg, #F00 50%, transparent 50%);
#50% 0px
```

```
background-attachment: scroll;
background-attachment: fixed local
```

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

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

EX:

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

```
border
border-width
border-style
border-color
```

```
border-top(1px solid red)
border-top-width: 1px;
border-top-style: solid;
border-top-color: red;
```

```
border
border-top: 1px solid red;
border-right: 1px solid red;
border-bottom: 1px solid red;
border-left: 1px solid red;
```

```
border
border-width: 1px 1px 1px 1px;
border-style: solid solid solid solid;
border-color: red red red red;
```

```
border-width: 1px 1px 1px 1px;
border-style: solid solid solid solid;
border-color: red red red red;
```

```
/*border*/
border-width: 1px 1px 1px 1px;
/*border*/
border-width: 1px [1px] 1px;
/*border*/
border-width: [1px] [1px];
```

```
/*[ ]*[ ]*/  
border-width: [ ];
```

outline

outline-width
outline-style
outline-color

[]

```
list-style: list-style-type list-style-position list-style-image;  
list-style: [ ] [ ] [ ];  
list-style: circle outside url("gold-fish.png");
```

list-style-type[]

none ([]) cjk-ideographic ([])
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 ([]), relative ([]), fixed ([]), absolute ([]), sticky ([])

---[]---

```
<p>  
  <strong style="color:#f00;">[ ]</strong>[ ]  
</p>  
<p>  
  <strong style="color:#f90;">[ ]</strong>[ ]  
</p>  
<p>  
  <strong style="color:#00f;">[ ]</strong>[ ]  
</p>  
[ ]<b>
```

SEO

1. Title

<title></title>

<meta name="description" content=""> 60

HTML HTML <h1>-<h6> (Heading) <h1> <h6>

alt

<META NAME="ROBOTS" CONTENT="INDEX,FOLLOW">

<META NAME="ROBOTS" CONTENT="NOINDEX,FOLLOW">

<META NAME="ROBOTS" CONTENT="INDEX,NOFOLLOW">

<META NAME="ROBOTS" CONTENT="NOINDEX,NOFOLLOW">

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
```

```
ron@ron:~$ sudo dpkg -l | grep php | tee packages.txt
[sudo] password for ron:
ii libapache2-mod-php 2:8.2+93+nmu1 all server-side, HTML-embedded
scripting language (Apache 2 module) (default)
ii libapache2-mod-php8.2 8.2.27-1 amd64 server-side, HTML-embedded
scripting language (Apache 2 module)
ii php 2:8.2+93+nmu1 all server-side, HTML-embedded
scripting language (default)
ii php-common 2:93+nmu1 all Common files for PHP packag
es
ii php-mysql 4 of 4 2:8.2+93+nmu1 all MySQL module for PHP [defau
lt]
ii php8.2 8.2.27-1 all server-side, HTML-embedded
scripting language (metapackage)
ii php8.2-cli 8.2.27-1 amd64 command-line interpreter fo
r the PHP scripting language
ii php8.2-common 8.2.27-1 amd64 documentation, examples and
common module for PHP
ii php8.2-mysql 8.2.27-1 amd64 MySQL module for PHP
ii php8.2-opcache 8.2.27-1 amd64 Zend OpCache module for PHP
ii php8.2-readline 8.2.27-1 amd64 readline module for PHP
```

████████████████████

██████████

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

□□□□

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

□□□

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

□□□

- □□□
- □□□
- □□□□
- □□□□□□□□ daemon
- □□□□ ETL □□ Web □□

□□□ JIT □□□ CPU-bound □□ Web □□□□ I/O bound □□□□ / □□ / □□□□

□□□

APCu□□

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

□□□□

- **apc.enabled=1** → □□ APCu
- **apc.shm_size=128M** → □□□□□□□□□□□□□□
 - □□□ 32M
 - □□□ 64M
 - □□□ 128M~256M
- **apc.enable_cli** → □□□ CLI □□□ APCu
 - **0** □□□□□□ CLI □

□□□□

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