# [illegible]

## php.ini[illegible]

### php[illegible]

[illegible]php[illegible]

```
php -m
```

```
<?php
?>
```

### [illegible]

```
round(x,prec)
x:[illegible]prec[illegible]
```

### [illegible]([illegible])

```
ceil(x)
```

### [illegible]([illegible])

```
floor(x)
```

### [illegible]

```
; [illegible]
upload_max_filesize = 50M

; POST [illegible]
post_max_size = 200M

; [illegible]
memory_limit = 512M
```

[illegible] > POST [illegible] > [illegible]

```
; PHP [illegible]
max_execution_time = 600

; PHP [illegible]
max_input_time = 600
```

```
; MySQL [illegible]-1 [illegible]
mysql.connect_timeout = -1
```

## php config[illegible]

```
;OPcache[illegible]/[illegible]0[illegible]Opcache, [illegible]
opcache.enable = 1

;OPcache[illegible]opcode[illegible]MB[illegible]
opcache.memory_consumption = 256

;[illegible]
;PHP[illegible]string interning[illegible]
;[illegible]1000[illegible]"foobar"[illegible]PHP[illegible]
;[illegible]999[illegible]
;[illegible]——[illegible]
;[illegible]php-fpm[illegible]
;[illegible]php-fpm[illegible]
;[illegible]
;[illegible]megabytes[illegible]16[illegible]
;[illegible]16MB[illegible]4MB[illegible]
;(default "4")
opcache.interned_strings_buffer = 8

;[illegible]PHP[illegible]
;[illegible]PHP[illegible]
;[illegible] 200[illegible] PHP 5.5.6 [illegible] 100000[illegible]
;PHP 5.5.6 [illegible] 1000000[illegible]200[illegible]1000000[illegible]
;(default "2000")
opcache.max_accelerated_files = 4000

;[illegible]OPcache[illegible] opcache.revalidate_freq [illegible]timestamp
```

```
;[illegible]
;[illegible]/[illegible]"1"[illegible]"0"
;(default "1")
opcache.validate_timestamps = 1

;[illegible]
;[illegible]opcache[illegible]
;[illegible]PHP[illegible]opcode[illegible]
;[illegible]"0"[illegible]PHP[illegible]
;[illegible]stat[illegible]stat[illegible]
;[illegible]I/O[illegible]
;[illegible]CPU[illegible]CPU[illegible]
;[illegible]"0"[illegible]
;(default "2")
opcache.revalidate_freq = 60

;[illegible]
;Zend [illegible] [illegible]
;[illegible]
;[illegible]PHP 7.2.0[illegible]PHP[illegible]
;(default "0")
opcache.fast_shutdown = 1

;CLI[illegible]PHP[illegible]OPcache[illegible] PHP 7.1.2 [illegible]
opcache.enable_cli = 1

;[illegible]OPcache[illegible]
;[illegible]
;(default "1")
opcache.use_cwd = 0

opcache.max_file_size = 0
```

## [illegible] PHP [illegible] (memory_limit)

[illegible] PHP [illegible] memory_limit [illegible]

[illegible] php.ini [illegible]

```
memory_limit = 256M
```

[illegible] .htaccess [illegible]

```
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">[illegible]</a>[illegible] Fooish [illegible]
<a href="#" style="color:red;" target="[illegible]" rel="[illegible]">[illegible]</a>
```

[illegible] CSS [illegible]

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

## <a> [illegible] (attributes)

href
[illegible] URL [illegible]target[illegible]
target
target [illegible]
    _self: [illegible]
    _blank: [illegible]
    _parent: [illegible]
    _top: [illegible]
    framename: [illegible]
    [illegible]

rel[illegible]
    nofollow[illegible]
    external[illegible]target="_blank"[illegible]_blank[illegible]xhtml[illegible]
    noopener[illegible]target="_blank"[illegible]window.opener.location = newURL[illegible]
    noreferrer[illegible]noopener[illegible]ex[illegible]<a href=url" rel ="noreferrer noopener" target="_blank"></a
[illegible]noreferrer[illegible]noopener[illegible]SEO[illegible]

## [illegible]

[illegible]

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

□□□□□□□□□□□ id="some-section-id" □ HTML □□□□□
□□□□□□□

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

□□□□

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

□□□□□□□ mailto:

```
<a href="mailto:□□□□□□">□□□□</a>
```

□□□□□□ tel: □□□□ RFC 3966 □□□□□□

```
<a href="#" style="color:red;">□□□□□□□□□□</a>
<a href="tel:□□□□">□□□□</a>
```

```
<marquee>□□□</marquee>
```

direction="□□□"□□□□□□□□□□□□□□□ up□dun□left□right
align="□□□"□□□□□□□□□□□□ top□□□□□ midden□□□□□ botton
scrollamount="□□□"□□□□□□□□□□□□□□□□□□
height="□□□"□□□□□
width="□□□"□□□□□
behavior="□□□"□□□□□□□□□□□□□□ alternate□□□□□□□ slide
bgcolor="□□□"□□□□□□□□□□□□□□□□□
Loop="□□□□"
Behavior"□□□□" □ "Scroll"(□□□)□"Slide" □□□□□□□□□□□□"Alternate" □□□□□□□□□□
"
scrolldelay="0" □□□□

&nbsp;(□□□□□)&ensp;□□□□□□□□□□□□&emsp;□□□□□□□□□□□□
onMouseOver="this.stop()" onMouseOut="this.start()" □□□□□□□□□
hspace="" □□□□
vspace="" □□□□

&nbsp;

□□□□□□□□□□□□□□□□□□□□(space key)□□□□□

&ensp;

□□□□□□□□□□□□ 1/2 □□□□□□

&emsp;

□□□□□□□□□□□□ 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.
```

## □□□□□□□□□□

```
$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;
 ?>
```

□□□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

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'>
```

## □□□□□□□□□

1□□□□$_FILES□□5□□□

$_FILES['userfile']['name']——□□□□□□□□

$_FILES['userfile']['type']——□□ MIME □□

$_FILES['userfile']['size']——□□□□□□□□□□□

$_FILES['userfile']['tmp_name']——□□□□□□□□□□□□□

$_FILES['userfile']['error']——□□□□□□□□□□

2□□$_FILES['userfile']['error']□□□□□□□□□

0——□□□□□□□□□□ □□□□□□□□□□□□□□□size□0□

1——□□□□□□ php.ini □ upload_max_filesize □□□□□

2——□□□□□□□ HTML □□ MAX_FILE_SIZE □□□□□

3——□□□□□□□

4——□□□□□□ □□□□file□□□□□□□□

## □□□□□□□

<input accept=".pdf,.xls" type="file" />  #□□□□

<input accept="image/*" type="file" />    #□□□□□□jpg□gif□png

<input accept="text/html" type="file" />  #□□□□□htm html

<input accept="video/*" type="file" />    #□□□□□□□ .avi□.mp4□.mpg □□□□

<input accept="audio/*" type="file" />    #□□□□□□□ .mp3□.wav □□□□

<input accept="image/*,.pdf,.xls" type="file" />  #□□□

# □□□□□□□□□

## □□□□□□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 □□□□□□□□ -->
    <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);
        }

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

        // □□□□□□□□□□□□□□□
        if (!is_array($allowExt))  // □□□□□□□□□
            exit('□□□□□□□□□ array');
        else {
            if (!in_array($ext, $allowExt))  // □□□□□□□□□□□□□□
                exit('□□□□□□');
        }

        // □□□□□□□□□□□□□□□□□
        if ($fileInfo['size'] > $maxSize)
            exit('□□□□□□□□□□');

        // □□□□□□□□□□□□
        if ($flag && !@getimagesize($fileInfo['tmp_name']))
            exit('□□□□□□□□□');

        // □□□□□□□□□□□□□□□□□□□
        if (!file_exists($uploadPath))
            mkdir($uploadPath, 0777, true);  // □□□□

        // □□□□□□□□□□□□□□
        if (!@move_uploaded_file($fileInfo['tmp_name'], $destination))  // □□□□□□□□
            exit('□□□□□□');

        return $destination;
    }
```

## □□□□(□□□□)

```
<!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 [illegible]
 *
 * @param files [illegible] $_FILES [illegible] HTTP [illegible]
 * @param allowExt [illegible] 'jpeg', 'jpg', 'gif', 'png'
 * @param maxsize [illegible] 2097152[illegible]2M * 1024 * 1024 = 2097152byte[illegible]
 * @param flag [illegible]true[illegible]false [illegible]
 * @param uploadPath [illegible] uploads
 *
 * @return [illegible] + md5 [illegible] + [illegible]
 */function uploadFile($fileInfo, $allowExt = array('jpeg', 'jpg', 'gif', 'png'), $maxSize = 2097152, $flag = true,
$uploadPath = 'uploads') {
    // [illegible]
    $res = array();

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

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

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

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

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

</body>
</html>
```

[illegible](doAction.php)

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

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

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

[illegible]upload.class.php[illegible]

```
<?php
/**
 * [illegible]
 *
 * @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;
```

```
                $i++;
            }
            // array □□□□□□□□□□□
            elseif (is_array($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□□□□□

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(□□, □□□□ □□□□□□, □□□□ □□□□□□);
background-image: linear-gradient(115deg, #F00 50%, #FF0 50%);
background-image: linear-gradient(115deg, #F00 50%, transparent 50%);
#50%□□□

background-attachment: scroll;
□□fixed local

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

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

border
    border-width
    border-style
    border-color

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

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

□□□□2
border-width: 1px 1px 1px 1px;
border-style: solid solid solid solid;
border-color: red red red red;

border-width: □□□□□   □□□□□   □□□□□   □□□□□;
border-style: □□□□□   □□□□□   □□□□□   □□□□□;
border-color: □□□□□   □□□□□   □□□□□   □□□□□;

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

```
/*□□□□□□□□□□□□□□□*/
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>
```

Revision #4

Created 13 September 2024 15:46:35 by Ron

Updated 27 October 2024 06:07:29 by Ron