

الوحدة التاسعة

الملفات و المجلدات

الملفات

المسار هو طريقة للتعبير عن عنوان ملف أو مجلد في نظام التشغيل , و المسارات نوعان :

مسارات نسبية ومسارات مطلقة , **المسارات النسبية** تبدأ من المسار الحالي حتى نصل الى القيد المطلوب (القيد = مجلد أو ملف) مثلا مسار الملف file1.txt الموجود في المجلد folder الموجود في مجلد البرنامج الذي نقوم بتنفيذه يكون كالتالي :

`\htdocs\unit9\file.txt`

`folder/file1.txt`

اما إذا كان الملف file1.txt موجوداً في المجلد الأب للمجلد التالي (أي المجلد الذي يسبقه) يكون المسار كالتالي :




`../file1.txt`

**الروابط المطلقة** فهي تُشير الى مسار الملف أياً كان المجلد الذي يوجد فيه البرنامج

`C:\xampp\htdocs\unit9\file.txt`

# دوال المجلدات والملفات:

```
echo realpath("file.txt");
```

 google  whatsapp  FreeLa

C:\xampp\htdocs\unit9\file.txt

```
if(file_exists('file.txt') === true)
```

```
{  
    echo 'file "file.txt" exists';  
}
```

```
echo '<br>';
```

```
if(file_exists('file2.txt') === false)
```

```
{  
    echo 'file "file2.txt" does not exists';  
}
```

**1** realpath التي تقبل وسيطاً وحيداً هو المسار النسبي للحصول على المسار كاملاً (المطلق)

**2** file\_exists التي تقبل وسيطاً وحيداً هو مسار الملف و تُعيد القيمة true في حال وجوده

**3** filesize في حال اردنا معرفة حجم ملف , نستخدم الدالة filesize التي تقبل وسيطاً واحداً هو مسار

الملف , وتُعيد هذه الدالة حجم الملف مقدراً بالبايت , وللحصول على الحجم مقدراً بالكيلوبايت أو الميغا بايت , نقسم على 1024 أو (1024\*1024) على التوالي وبالترتيب :

```
$size = filesize('file1.txt');
```

```
echo 'The size of file1.txt is : '. floor($size / 1024) . ' KB';
```

```
$file = 'file.example.txt';
```

```
$ext = explode('.', $file);
```

```
echo 'The file extension is : ' . $ext[count($ext) - 1];
```

explode استخراج الامتداد

**4**

الحصول على وقت تعديل أو تغيير أو الوصول لملف :

للحصول على بصمة الوقت التي تمثل آخر وقت لتغيير ملف ما , نستخدم الدالة filetime , حيث تقبل هذه الدالة وسيطاً واحداً هو مسار الملف .

أما للحصول على بصمة الوقت لآخر تعديل على الملف , نستخدم الدالة filemtime , وتقبل هذه الدالة - كما في الدالة السابقة - وسيطاً وحيداً هو مسار الملف .

```
echo date("m/d/Y H:i:s", filemtime('file.txt'));
echo "<br>" ;
echo date("m/d/Y H:i:s", fileatime('file.txt'));
```

11/29/2021 07:53:27

11/22/2022 17:32:08

الفرق التقني بين الدالة filetime و الدالة filemtime هو أن الدالة filetime تُعيد جميع

التغييرات على ملف سواءً على محتوياته أم على صلاحيات الوصول إليه أم تغيير المستخدم المالك له . أما الدالة filemtime فهي تشير الى آخر تعديل في محتويات الملف فقط .

والدالة fileatime تُعيد بصمة وقت آخر وصول للملف أو false في حال فشلها

في php نستخدم الدوال is\_readable , is\_writable , is\_executable لمعرفة امكانية القراءة أو الكتابة أو التنفيذ على التوالي وبالترتيب .  
تُعيد هذه الدوال true في حال نجاحها أو false ماعدا ذلك , وتقبل وسيطاً وحيداً هو مسار القيد .

```
$file="file.txt";  
echo 'Is readable? : '  
echo is_readable($file) == true ? 'true' : 'false';  
echo '<br>';  
echo 'Is writable? : '  
echo is_writable($file) == true ? 'true' : 'false';  
echo '<br>';  
echo 'Is executable? : '  
echo is_executable($file) == true ? 'true' : 'false';
```

Is readable? : true

Is writable? : true

Is executable? : false



## حذف ملف :

كما تتيح لك php إنشاء الملفات تُتيح لك حذفها , للقيام بهذه العملية استخدم الدالة [unlink](#) ,  
تقبل هذه الدالة وسيطاً واحداً هو مسار الملف المُراد حذفه , وبالتأكيد يجب أن تكون لديك  
صلاحيات كتابة على الملف المُحدد حتى تستطيع حذفه عدا ذلك سيتم اظهار خطأ  
E\_WARNING .

```
<?php  
unlink( 'file1.txt' );  
?>
```

## chmod تغيير صلاحيات قيد

لكن php لا تقبل الاعلان عن الصلاحيات كسلسلة نصية مثلاً "a-wx", وإنما تحصرها فقط باستخدام الصلاحيات بالارقام في النظام الثماني , أي تكون الصلاحية مكونة من اربعة ارقام الرقم الاول هو صفر , أما الارقام الثلاث الباقية هي عبارة عن الصلاحيات للمستخدم و لمجموعة المستخدمين و لبقية المستخدمين على التوالي و بالترتيب , الجدول التالي يوضح الارقام والمصلاحيات المقابلة لها

نقبل هذه دالة chmod وسيطين , الاول هو مسار القيد الفراد تغيير صلاحياته, والثاني هو الصلاحية مثال :

الرقم 0	يشير الى عدم اعطاء أي صلاحية
الرقم 1	يشير الى اعطاء صلاحية التنفيذ فقط .
الرقم 2	يشير الى اعطاء صلاحية الكتابة فقط .
الرقم 3	يشير الى اعطاء صلاحية الكتابة والتنفيذ .
الرقم 4	يشير الى اعطاء صلاحية القراءة فقط .
الرقم 5	يشير الى اعطاء صلاحية القراءة والتنفيذ .
الرقم 6	يشير الى اعطاء صلاحية القراءة و الكتابة .
الرقم 7	يشير الى اعطاء صلاحية القراءة والكتابة والتنفيذ .

```
<?php
chmod('folder/file1.txt', 0600); # القراءة والكتابة للمستخدم , لاشيئ لبقية المستخدمين
chmod('folder/file1.txt', 0755); # القراءة والكتابة والتنفيذ للمالك , القراءة و التنفيذ لبقية المستخدمين
```



نسخ أو نقل ملف :

نستعمل الدالة copy لنسخ الملفات , تقبل هذا الدالة وسيطين الاول هو مسار الملف المُراد نسخة  
و الثاني هو المسار الجديد . في حال وجود ملف في المسار الجديد فسيتم استبداله تلقا

```
copy('filee.txt','file121212.txt');  
echo 'done';
```

اما لنقل ملف فنستخدم الدالة rename التي تقوم اساساً بتغيير اسم الملف لكن يمكن  
استخدامها لنقله , تقبل هذه الدالة وسيطين الاول هو مسار الملف و الثاني هو مسار الملف  
الجديد :

```
rename('file2.txt', '../file.txt');
```

قبل اجراء أي عمليات على الملف , علينا تهيئته وذلك بإنشاء مقبض للملف عن طريق الدالة fopen التي تقبل وسيطين الوسيط الأول هو مسار الملف , أما الوسيط الثاني هو الوضع المراد فتح الملف به , الجدول التالي يبين الاوضاع المختلفة لفتح ملف :

الفرق بين a و w يكمن في أن w يقوم بحذف محتويات الملف بينما الوضع a يحافظ على محتويات الملف و يضع المؤشر عند نهايته .  
يمكن اضافة الحرف b الى الاحرف السابقة لفتح الملف بالنظام الثنائي, ويفيد هذا الوضع عند القراءة أو الكتابة على ملفات غير نصية .  
و يمكنك باستخدام الدالة fopen للإشارة الى ملفات خارج الخادم المُنفذ عليه البرنامج , عن طريق بروتوكول HTTP أو ftp أو غيرهم ... لكن للقراءة فقط :

```
<?php
$handle = fopen("file.txt", "r");
$handle = fopen("./folder/file.zip", "wb");
$handle = fopen("http://www.example.com/", "r");
$handle = fopen("ftp://user:password@example.com/somefile.txt", "w");
?>
```

الوضع r يقوم بفتح الملف للقراءة فقط مع وضع مؤشر الملف في بدايته (سنتحدث لاحقا عن مؤشر الملف وكيفية تحريكه) .

الوضع r+ يقوم بفتح الملف للقراءة والكتابة مع وضع مؤشر الملف في بدايته .

الوضع w يقوم بفتح الملف للكتابة فقط ويقوم بمسح جميع محتوياته , وإذا لم يكن الملف موجودا سوف يقوم بإنشاءه .

الوضع w+ كما في الوضع w , لكنه يقوم بفتح الملف للقراءة والكتابة .

الوضع a يقوم بفتح الملف للكتابة فقط ويضع مؤشر الملف عند نهايته , إذا لم يكن الملف موجودا يقوم بإنشاءه .

الوضع a+ يقوم بفتح الملف للقراءة و الكتابة ويضع مؤشر الملف عند نهايته , إذا لم يكن الملف موجودا يقوم بإنشاءه .

يتم تحرير الذاكرة بإغلاق مقبض الملف عن طريق الدالة fclose التي تقبل وسيطا وحيدا هو مقبض الملف المُنشئ بواسطة الدالة السابقة .

# قراءة البيانات من ملف :

بعد انشاء مقبض الملف باستخدام الدالة `fopen` , نقوم باستخدام الدالة `fread` للقراءة من الملف وتقبل وسيطين : الاول هو مقبض الملف , والثاني هو عدد البايتات التي سيتم قراءتها من الملف بدءاً من مكان وجود مؤشر القراءة :

ولقراءة الملف بأكمله نقوم بتحديد قيمة الوسيط `length` بحجم الملف عن طريق الدالة `filesize` كما في المثال التالي الذي يقوم بطباعة محتويات الملف `file1.txt` :

```
<?php
$filename = 'file1.txt';
$handle = fopen($filename, 'r');
$contents = fread($handle, filesize($filename));
echo $contents;
fclose($handle);
?>
```

الدالة `fgets` شبيهة جداً بالدالة `fread` وتقوم بنفس العمل تقريباً حيث في معظم الاحيان يمكن استخدام `fgets` عوضاً عن `fread` , حيث الفرق الاساسي بينهما هو ان الدالة `fgets` لا تتطلب تحديد عدد البايتات التي يجب قراءتها من الملف حيث تكون القيمة الافتراضية للوسيط `length` هي 1024 بايت . وكلا الدالتين تقومان بالتوقف عن القراءة عندما تصلان الى نهاية الملف EOF

(End Of File)



# الكتابة على ملف :

php عن طريقة الدالة [fwrite](#) التي تقوم بكتابة البيانات المُمررة اليها الى ملف , طبعا يحب فتح الملف بوضع يسمح بالكتابة عليه كما في الوضعين (w , a) , الدالة fwrite تقبل ثلاثة وسطاء , الوسيط الاول هو مقبض الملف والثاني هو البيانات المُراد كتابتها و الثالث اختياري يمثل عدد البايتات التي سيتم كتابتها , فاذا تم تحديد الوسيط الثالث فان الكتابة على الملف سوف تتوقف عندما يصبح عدد البايتات المكتوبة مساويا لقيمة هذا الوسيط

```
<?php
$filename = 'file1.txt';
$handle = fopen($filename, 'w+');
fwrite($handle, 'Hello World !!');
fclose($handle);
?>
```

الدالة [fputs](#) هي دالة مكافئة Alias للدالة fwrite , اي انها تقوم بنفس العمل تماما وتأخذ الوسائط ذاتها

## الدالة feof :

تقوم هذه الدالة بإعادة true في حال وصل مؤشر القراءة الى نهاية الملف و false عدا ذلك و تُفيد عندما نقوم بالدوران على محتويات ملف لقراءته , وتقبل هذه الدالة وسيطا وحيدا هو مقبض الملف.

```
<?php
$file = fopen('file1.txt', 'r');
while(!feof($file))
{
echo fgets($file). '<br>';
}
fclose($file);
?>
```



## تغيير مكان المؤشر :

لتغيير مكان المؤشر سواء عند القراءة أو الكتابة نستخدم الدالة [fseek](#) التي تقبل وسيطين اجباريين , الاول هو مقبض الملف والثاني هو offset الذي سوف يتم وضع المؤشر عنده .

```
fseek($handle, $offset);
```

ملاحظة : عند فتح الملف بوضع a أو a+ فإن الكتابة سوف تكون في اخر الملف حتى لو قمت بتغيير مكان المؤشر .

## الدالتين file\_get\_contents و file\_put\_contents :

تقوم الدالة [file\\_get\\_contents](#) بقراءة ملف بأكمله على شكل سلسلة نصية و يُمرر لها مسار الملف كوسيط , شكلها العام :

```
file_get_contents($filename);
```

اما الدالة [file\\_put\\_contents](#) فتقوم بكتابة البيانات المُمررة اليها بالوسيط الثاني على الملف الذي يتم تحديده بمساره والذي يشكل الوسيط الاول :

```
file_put_contents($filename, $data);
```

وتقوم هذه الدالة بانشاء الملف اذا لم يكن موجوداً , وفي حال وجوده تقوم بمسح جميع محتوياته !

المجلات

## القراءة من مجلد :

كما في دالة fopen عند التعامل مع الملفات , تُستخدم الدالة opendir للحصول على مقبض للمجلد , حيث تقبل الدالة opendir وسيطاً واحداً هو مسار المجلد

وايضاً لتحرير الذاكرة وإغلاق مقبض الملف , نستخدم الدالة closedir التي تقوم بعمل مشابه للدالة fclose , تقبل هذه الدالة وسيطاً واحداً هو مقبض المجلد الذي قُمتُ بإنشاءه باستخدام الدالة opendir :

1

## قراءة محتويات مجلد :

تستخدم الدالة readdir لقراءة القيد التالي من مجلد تم انشاء مقبضه بواسطة الدالة opendir , حيث تقوم هذه الدالة بقراءة قيود الملفات على التتالي وحسب ترتيب نظام الملفات المُستخدم . تقبل هذه الدالة وسيطاً واحداً هو مقبض المجلد و تُعيد القيد (اسم الملف أو المجلد) , وللمرور على جميع قيود المجلد نستخدم حلقة التكرار while . ولتطبيق الدوال الثلاث السابقة نجرب المثال التالي:

```
<?php
```

```
$dir = opendir('C:\xampp\htdocs\unit9');  
while (($file = readdir($dir) )!= false)  
{  
    echo $file.'<br>';  
}
```

localhost:8080/unit9/folder.php

google whatsapp FreeLancer platform منصة مستقل Arabi online

..  
file.txt  
file121212.txt  
file30.txt  
file33.txt  
filehh.txt  
folder.php  
index.php  
nbproject  
read.php  
write.php

## حذف المجلدات :

لحذف مجلد نستخدم الدالة [rmdir](#) التي تقبل وسيطاً واحداً هو مسار المجلد المراد حذفه , لكن يجب أن يكون هذا المجلد فارغاً أما اذا كان المجلد يحوي أي ملف او مجلد فرعي , فلن يتم تنفيذ هذه التعليمة و سيتم توليد رسالة خطأ .

إنشاء مجلد : بالطبع يمكنك باستخدام php انشاء المجلدات وتعيين صلاحيات الوصول إليها , ويتم ذلك بواسطة الدالة [mkdir](#) التي تقبل وسيطين , الأول اجباري هو اسم المجلد والثاني اختياري هو صلاحيات الوصول للمجلد , ويكون شكلها العام كالتالي :

```
mkdir($pathname, $mode);
```

وبشكل افتراضي يكون mode مساوياً للقيمة 0777 , أي صلاحيات القراءة والكتابة والتنفيذ لجميع المستخدمين .