الفصل الحادى عشر

التعامل مع الملفات

التعامل مع الملفات

تعطى لغة PHP الكثير من الأدوات التى تمكنك من التعامل مع الملفات مثل القراءة و الكتابة

الدالة **Include()**:

تستخدم هذه الدالة لتضمين ملف خارجى الى الملف الحالى فيصبح الملف الخارجى كأنه جزء من الملف الحالى و هذا مفيد حيث أنك تستطيع تضمين ملف به العديد من الدوال الجاهزة الأستخدام (مكتبه) الى الملف الحالى مما يعطى أمكانيه سهلة و سريعة للوصول الى هذه الدوال من اى صفحة تستخدم هذه المكتبه .

و أذا أردت أن تقوم بتطوير المكتبه أو تصليح خطأ أكتشفته فيها فليس عليك الا التعديل فى هذه المكتبه فقط و ليس فى كل الصفحات التى تستخدمها

مثال:

- قم بأنشا ملف أسمه myfunc.php و به الكود الأتى:

```
1: <?php
2: $retval = ( 4 + 4 );
3: return $retval;
4: ?>
```

- قم الأن بأنشاء ملف أخر بأى أسم ثن أكتب الكود الأتى به

```
1: <html dir="rtl">
2: <head>
3: <title>مثال على الدالة تضمين الملفات</title>
4: </head>
5: <body>
6: <?php
7: $addResult = include("myfunc.php");
```

```
8: print "الملف الخارجى قام بأرجاع القيمة $addResult";
9: ?>
10: </body>
11: </html>
```

لاحظ أن الدالة include() قامت بإرجاع قيمة العملية الحسابية الى المتغير $addresult الى قمنا بعد ذلك بطباعة قيمته

يمكننا أيضا تضمين ملفات خارجيه بناء على تحقق شرط معين و يمكن أن يتم ذلك بالطريقه الأتيه

```
$test = false;
if ( $test )
{
include( "file.txt" );
}
```

لاحظ أن العبارة If لا يتم تحققها و بالتالى فأن الملف "file.txt" لن يتم تضمينه أبدا

وإذا قمت بتضمين الدالة include() فى حلقة تكراريه فأن محتويات الملف الخارجى يتم تنفيذها فى كل مرة يتم فيها النداء على الدالة و فيما يلى مثال يقوم بالنداء على ثلاثة ملفات مختلفة .

```
1: <html dir="rtl">
2: <head>
3: <title>تنفيذ الملفات الخارجية داخل حلقة</title>
4: </head>
5: <body>
6: <?php
7: for ( $x = 1; $x<=3; $x++ )
8: {
9: $incfile = "incfile$x".".txt";
10: print "بداية تنفيذ الملف $incfile<br>";
11: include( "$incfile" );
```

```
12: print "<p>";
13: }
14: ?>
15: </body>
16: </html>
```

يتم فى المثال السابق تنفيذ الحلقة ثلاثة مرات و يتم أيضا النداء على ثلاثة ملفات مختلفة هم "incfile1.txt" و "incfile2.txt" و "incfile3.txt"

يوجد منذ الأصدارة 4 من لغة PHP العبارة require() التى يمكنها عمل نفس وظيفة الدالة include() و لكن بدون أرجاع قيمة

الدالة **file_exists()**:

عادة قبل التعامل مع الملفات الخارجية سواء بالقراءة أو الكتابة فأن أختبار وجود الملف من الاشياء الهامة حتى لا ينتج خطأ وقت التشغيل و تستخدم الدالة file_exists() لهذا الغرض فهى تستقبل معامل واحد وهو مسار الملف المطلق او النسبى و ترجع أما القيمة true أذا وجدت الملف أو القيمة false أذا لم تجده

مثال:

```
if ( file_exists("test.txt") )
  print "الملف موجود";
```

يمكنك أيضا أختبار وجود ملف معين عن طريق الدالة is_file() التى ترجع قيمة من نوع boolean

```
if (is_file("test.txt"))
  print "test.txt ملف ";
```

نفس الطريقه يمكن أختبار وجود دليل فرعى عن طريق الدالى is_dir() التى ترجع أيضا قيمة boolean

```
if (is_dir("/php"))
  print "php دليل فرعى ";
```

هناك دوال أخرى تمكنك من التعامل مع الملفات بأحترافية فهى تعبر عن حالة الملفات الموجودة على السيرفرات المختلفة التى يوجد بها نظام آمان معين قد يمنعك من التعامل مع الملف وفيما يلى الدوال التى تمكنك من معرفة صلاحياتك بالنسبة لملف معين

الدالة **is_readable()**:

تمكنك الدالة من معرفة أذا كنت تستطيع قراءة الملف أم لا (أى تستطيع فتحه أم لا) و فى سيرفرات يونيكس قد تستطيع رؤية الملفات الملف و لا تستطيع فتحه و يمكنك تنفيذ هذه الدالة كما يلى :

```
if ( is_readable( "test.txt" ) )
  print "يمكنك فتح وقراءة الملف ";
```

بالمثل يمكنك تحديد أمكانية التعديل فى ملف عن طريق الدالة is_writable()

```
if (is_writable( "test.txt" ) )
  print "يمكنك فتح وقراءة الملف ";
```

وأيضا فى أذا كنت تريد تنفيذ ملف معين و بناء على أمتداد الملف (نوعه) وصلاحيات الدخول الخاصة بك فى السيرفر يتم التحقق من ذلك عن طريق الدالة is_executable() كما يلى :

```
if ( is_executable( "test.txt" )
print "الملف قابل للتنفيذ";
```

حجم الملف:

يمكنك معرفة حجم ملف معين عن طريق الدالة filesize() التى تقبل معامل واحد هو مسار الملف و ترجع حجمه بالبايت .

```
print "حجم ملف قاعدة البيانات: "
print filesize("mydb.mdb");
```

الحصول على تاريخ الملف:

أحيانا قد تريد معرفة متى أخر مرة تم فيها فتح الملف أو تعديله و تقدم لغة PHP العديد من الدوال التى تمكنك من معرفة ذلك

فأذا أردت الحصول على تاريخ أخر وصول الى الملف يمكنك النداء على الدالة filetime() التى تقوم بأرجاع التاريخ فى صورة رقمية و هى عدد الثوانى منذ 1 يناير 1970 (عصر اليونيكس) و يمكن أستخدام الدالة date() حتى يمكن تحويل الرقم الناتج الى تاريخ قابل للفهم

```
$atime = fileatime( "test.txt" );
print "أخر مرة تم فيها الوصول الى الملف ";
print date("D d M Y g:i A", $atime);
// سيتم طباعة تاريخ مثل هذا: Thu 17 Jan 2001 2:26 PM
```

راجع فصل التعامل مع البيانات الزمنية

ويمكن أيضا معرفة متى أخر مرة تم فيها تعديل الملف أو تغيير صلاحياته عن طريق الدالة filectime()

```
$ctime = filectime( "test.txt" );
print "تم تعديل الملف أخر مره فى ";
print date("D d M Y g:i A", $ctime);
// سيتم طباعة تاريخ مثل هذا: Thu 17 Jan 2001 2:26 PM
```

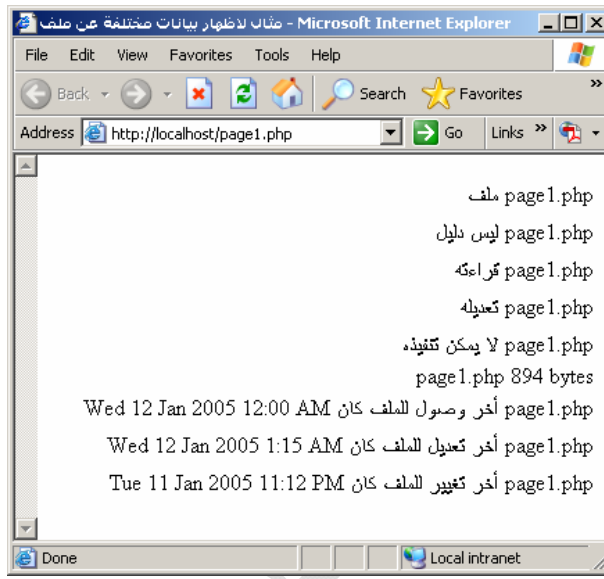
مثال عام:

سنقوم الأن ببناء دالة تقوم بأرجاع بيانات مختلفة عن ملف معين بأستخدام المهارات السابقة

```
1: <html dir="rtl">
2: <head>
```

```
3: <title>مثال لاظهار بيانات مختلفة عن ملف</
title>
4: </head>
5: <body>
6: <?php
7: $file = "page1.php";
8: outputFileTestInfo( $file );
9: function outputFileTestInfo( $f )
10: {
11: if ( ! file_exists( $f ) )
12: {
13: print "$f الملف غير موجود<BR>";
14: return;
15: }
16: print "$f ".(is_file( $f )?"":"ملف").") ليس<br>";
17: print "$f ".(is_dir( $f )?"":"دليل").") ليس<br>";
18: print "$f ".(is_readable( $f )?"":" لا يمكن
قراءته").("<br>";
19: print "$f ".(is_writable( $f )?"":" لا يمكن
تعديله").("<br>";
20: print "$f ".(is_executable( $f )?"":" لا يمكن
تنفيذه").("<br>";
21: print "$f ".(filesize($f))." bytes<br>";
22: print "$f أخر وصول للملف كان ".date( "D d M Y g:i
A", fileatime( $f ) )."<br>";
23: print "$f أخر تعديل للملف كان ".date( "D d M Y g:i A",
filemtime( $f ) )."<br>";
24: print "$f أخر تغيير للملف كان ".date( "D d M Y g:i A",
filectime( $f ) )."<br>";
25: }
26:
27: ?>
28: </body>
29: </html>
```

و يكون تنفيذ المثال يشابه الشكل القادم :



لاحظ انه تم أستخدام المعامل الثلاثى الذى يمكن عن طريقه أختبار بعض الحالات فى سطر واحد بدون كتابة الكثير من عبارات IF و يمكن تنفيذ هذا المعامل كما يلى:

```
"$f ".(is_file( $f )?"":"ملف".(" ليس<br>";
```

لاحظ أن العلامة ? تغنى عن العبارة IF فأذا تحقق الشرط تم تنفيذ الجزء الذى يأتى بعدها و هو طبع نص فارغ "" أما أذا لم يتحقق الشرط فيتم طبع العبارة "ليس" و يتم وصل كل النصوص الحرفية عن طريق معامل الربط ".".

ويمكن تنفيذ نفس الكود السابق بالطريقه الأتيه :

```
if ( is_file( $f ) )
print "$fi ملف<br>";
else
```

```
print "$fi ليس ملف<br>";
```

حذف و أنشاء الملفات:
أذا لم تجد الملف بأستخدام الدالة file_exists() فيمكنك أنشاؤه بأستخدام الدالة touch() التى تقبل معامل واحد هو مسار الملف و أسمه و تقوم بأنشاؤه فارغا أما أذا كان الملف موجودا من قب فأن تنفيذ هذه الدالة لا يقوم بإلغاء الملف القديم و لكن يتم تعديل تاريخ أخر وصول فقط

```
touch("file.txt");
```

و بنفس الطريقة يمكنك حذف ملف عن طريق الدالة unlink() التى تقبل مسار الملف الذى تريد حذفه

لاحظ أنه يجب وضع صلاحيات معينه للملف الذى تريد العمل عليه

فتح الملف للكتابة أو القراءة:

قبل أن تقوم بالتعامل مع الملف فيجب أن تقوم بفتحه أولا و يتم ذلك عن طريق الدالة fopen() التى تقبل معاملين الأول هو مسار الملف المراد فتحه و الثانى هو الحالة أو نوع الفتح (mode) .

نوع الفتح قد يكون قراءة (r) أو كتابة (w) او للأضافه (a) وتقوم الدالة بعد الفتح بإرجاع عدد صحيح Integer هو يعبر عن مؤشر فى الذاكرة لهذا الملف و يمكن عن طريقه الوصول الى الملف و يسمى (file pointer) و يجب تخزين هذه القيمه فى متغير

مثال:

```
$fp = fopen( "test.txt", 'r' );
```

العبارة السابقة تقوم بفتح الملف "test.txt" للقراءة فقط و يتم تخزين مؤشر الملف فى المتغير fb

مثال:

أيضا الدالة fopen() تقوم بإرجاع القيمة false أذا حدثت أى مشكله تعوق عملية الفتح و يمكن أختبارها مثلا قبل الكتابه فى ملف

```
if ($fp = fopen( "test.txt", 'w' )) {

// كود الكتابة
}
```

و يمكن تنفيذ المثال السابق بطريقه مشابه فى سطر واحد هكذا

```
($fp = fopen( "test.txt", "w" ) ) or die("لا يمكن فتح الملف");
```

إذا تم فتح الملف بطريقه صحيحه فإن الجزء الثانى من العبارة بعد or لا يتم تنفيذه أما أذا حدثت مشكله فى فتح الملف فأن الدالة die() يتم تنفيذها و التى تقوم بطبع رساله فى المتصفح و أيضا أنهاء الكود الحالى .

كخطوة أخيرة فى التعامل مع الملف يجب عليك عند الأنتهاء من القراءة أو الكتابة أن تقوم بإغلاق الملف و تقوم الدالة fclose() بهذه الوظيفه و التى تقبل معامل واحد هو مؤشر الملف الذى تم ارجاعه من تنفيذ صحيح للداله fopen()

```
fclose($fp);
```

القراءة من الملفات:

تعطى لك لغة PHP أكثر من طريقه للقراءة من الملفات فتستطيع قراءة الملف بالحرف أو بالبايت أو بالسطر .

- القراءة بالسطر تتم عن طريق الدالة fgets() التى تقبل معاملين الأول هو مؤشر الملف الذى تم إرجاعه من الدالة fopen() و الثانى عدد صحيح يعبر عن عدد البايت التى يجب للدالة قراءته أذا لم تصل الى نهاية السطر أو الى نهاية الملف .

وهكذا نستطيع أن نستنتج أن الدالة fgets() تستمر فى القراءة حتى تصل الى نهاية السطر و تجد الحرف "\n" أو الى نهاية الملف أو الى عدد البايت المحدد كمعامل ثانى لها .

```
$line = fgets( $fp, 1024 );
```

و رغم أن الدالة fgets() تنتهى عند نهاية الملف فأنك يجب أن تعرف متى يتم أيقاف القراءة عند نهاية الملف و تقوم الدالة feof() بهذه الوظيفه و التى تقبل معامل واحد هو مؤشر الملف وتقوم بأرجاع القيمة true عند الوصول الى نهاية الملف .

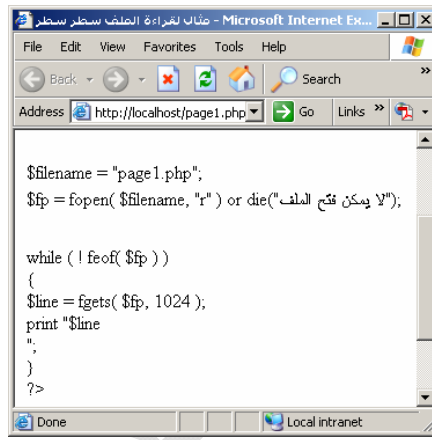
مثال:

```
1: <html>
2: <head>
3: <title>مثال لقراءة الملف سطر سطر</title>
4: </head>
5: <body>
6: <?php
7: $filename = "page1.php";
8: $fp = fopen( $filename, "r" ) or die("لا يمكن فتح الملف");
9: while ( ! feof( $fp ) )
10: {
11: $line = fgets( $fp, 1024 );
12: print "$line<br>";
13: }
14: ?>
15: </body>
16: </html>
```

لاحظ فى السطر 8 أن العبارة بعد or يتم تنفيذها عند وجود مشكله فى فتح الملف و بالتالى يتم أنهاء باقى الكود بالكامل وهذا عادة يحدث فى حالة عدم وجود الملف أو عد وجود صلاحيات للقراءة من الملف فى النظام unix مثلا

فى السطر 9 يوجد الشرط الخاص بالقراءة من الملف حتى نهايته فالمعامل ! لو تتذكر يعبر عن النفى و الشرط فى هذا السطر معناه أنه أذا لم تصل الى نهاية الملف فأستمر فى تنفيذ الحلقة و بالتالى القراءة من الملف .

و يتم طباعة محتويات الملف سطر بسطر فى السطر 12 الذى يقوم بطباعة المتغير $line ووضع الوسم
 حتى تتم الطباعة أسفل بعضها البعض و يمكن قراءتها بسهولة .



لاحظ كيف تمكنت من قراءة ملف الكود الأصلى ("page1.php") و عرضه فى المتصفح بهذه الطريقه

الدالة fread():

بدلا من قراءة الملف سطر بسطر يمكنك قراءة مقدار معين من البيانات دفعة واحدة عن طريق هذه الدالة التى تقبل معاملين الأول هو مؤشر الملف و الثانى مقدار البيانات التى تريد قراءتها بالبايت و تقوم الدالة بإرجاع البيانات التى طلبتها أو حتى تصل الى نهاية الملف .

```
$chunk = fread( $fp, 16 );
```

مثال:
سنقوم بتنفيذ نفس المثال السابق لكننا سنقوم بقراءة 24 بايت من محتويات الملف فى كل مرة بدلا من سطر سطر

```
1: <html dir="rtl">
2: <head>
3: <title>مثال على الدالة fread()</title>
4: </head>
5: <body>
6: <?php
7: $filename = "test.txt";
8: $fp = fopen( $filename, "r" ) or die("لا يمكن فتح
الملف");
9: while ( ! feof( $fp ) )
10: {
11: $chunk = fread( $fp, 24 );
12: print "$chunk<br>";
13: }
14: ?>
15: </body>
16: </html>
```

و لا تعطيك الدالة fread() أمكانية تحديد بداية موضع القراءة و لكن يمكنك تحقيق ذلك عن طريق الدالة fseek() التى تقبل معاملين الأول هو مؤشر الملف و الثانى هو رقم الموضع الذى تريد الأنتقال اليه بالبايت .

```
fseek($fp, 43);
```

مثال:

```
1: <html>
2: <head>
3: <title>مثال على الدالة fseek()</title>
4: </head>
```

```
5: <body>
6: <?php
7: $filename = "myfile.txt";
8: $fp = fopen( $filename, "r" ) or die("لا يمكن فتح الملف");
9: $fsize = filesize($filename);
10: $halfway = (int)( $fsize / 2 );
11: print "يوجد منتصف الملف عند : $halfway <BR>\n";
12: fseek( $fp, $halfway );
13: $chunk = fread( $fp, ($fsize – $halfway) );
14: print $chunk;
15: ?>
16: </body>
17: </html>
```

هل تستطيع أن تستنتج عمل المثال السابق ؟
يقوم المثال السابق ببساطة بطباعة نصف الملف "myfile.txt" و أستطعنا أتمام ذلك عن طريق الدالة filesize() و قمنا بقسمة الناتج على 2 فى السطر 10

ثم قمنا بالأنتقال الى نقطة المتصف عن طريق الدالة fseek() فى السطر 12 و قمنا بتخصيص باقى المساحة المتبقية من الملف الى الدالة fread() حتى يتم قراءة باقى الملف دفعة واحدة و أخير ا تم طباعة الملف فى السطر 14

الدالة **fgetc()**:

وهذه الدالة تشبه الى حد كبير الدالة fgets() و لكنها تقوم بإرجاع حرف واحد فى كل مرة يتم فيها النداء على هذه الدالة و نظرا لان حرف واحد يمثل بايت واحد دائما فأن معامل طول القراءة غير موجود فى هذه الدالة

```
$char = fgetc( $fp );
```

مثال:

```
1: <html dir="rtl">
2: <head>
3: <title>مثال على الدالة fgetc()</title>
4: </head>
5: <body>
6: <?php
7: $filename = "myfile.txt";
8: $fp = fopen( $filename, "r" ) or die("لا يمكن فتح الملف");
9: while (! feof($fp))
10: {
11: $char = fgetc( $fp );
12: print "$char<BR>";
13: }
14: ?>
15: </body>
16: </html>
```

مثل ماسبق تماما مع الدالة fgets() يتم القراءة من الملف داخل حلقة تكرارية حتى نهاية الملف و لكن تتم القراءة حرف واحد فى كل مرة

تعديل الملفات:

تعديل الملف لا يختلف كثيرا عن فتحه للقراءة فيمكنك أن تقوم بتغيير المعامل الثانى للداله fopen() من "r" الى "w" للكتابة

```
$fp = fopen( "myfile.txt", "w" );
```

أذا كان الملف غير موجودا فيتم أنشاؤه و أذا كان موجودا فيتم تدمير البيانات القديمه و أستبدالها بالبيانات الجديده عن البدء فى الكتابه

و بالمثل يمكنك تغيير المعامل الثانى للداله من "r" الى "a" للأضافه

```
$fp = fopen( "myfile.txt", "a" );
```

و تتم الأضافه الى البيانات الموجودة و لا يتم حذفها

الكتابه الى الملف:

يمكنك الكتابة الى الملف عن طريق الدالة fwrite() او الدالة fputs() و هما يعملان بنفس الطريقة تماما فيتم تحديد معاملين الأول هو مؤشر الملف و الثانى هو النص الحرفى المراد كتابته .

```
fwrite( $fp, "مرحب بك" );
fputs( $fp, "مرحب بك" );
```

مثال:

```
1: <html dir="rtl">
2: <head>
3: <title>مثال على الكتابة داخل الملفات</title>
4: </head>
5: <body>
6: <?php
7: $filename = "myfile.txt";
8: print "سيتم الكتابة الى الملف $filename<br>";
9: $fp = fopen( $filename, "w" ) or die("لا يمكن فتح
الملف ");
10: fwrite( $fp, "مرحب بك\n" );
11: fclose( $fp );
12: print "الأضافة الى الملف  $filename<br>";
13: $fp = fopen( $filename, "a" ) or die("لا يمكن فتح
الملف");
14: fputs( $fp, "أضافة نص أخر الى الملف\n" );
15: fclose( $fp );
16: ?>
17: </body>
18: </html>
```

أقفال الملف:

فى الحياة العملية الكثير من المستخدمين أو الزائرين لموقعك سيقوموا بفتح الملف و الكتابه فيه و هذا يعنى أن أكثر من مستخدم يقوم بالكتابه فى نفس الوقت الى نفس الملف مما يتلف الملف و يجعله غير صالح للعمل و لحل هذه المشكله يجب أستخدام الدالة flock() التى تقوم بإغلاق الملف لأى محاولات أخرى من قبل مستخدمين أخرين للكتابه فى الملف أو تعديله .

و تقبل الدالة معاملين الأول هو مؤشر الملف المراد إغلاقه و الثانى هو عدد صحيح يعبر عن نوع الإغلاق و يتضح ذلك من الجدول الأتى :

الكود	النوع	الشرح
1	shared	يسمح الوصول المتعدد للملف لقراءته فقط و لا يسمح بالكتابه فيه و يستخدم فى حالة القراءة من ملف مشترك
2	exclusive	يمنع الوصول المتعدد للملف بغرض القراءة أو الكتابة و يستخدم فى حالة القراءة من الملف
3	release	يقوم بإلغاء القفل أو تحرير الملف من النوعين السابقين

ويجب عليك أن تقوم بتنفيذ الاغلاق مباشرة بعد فتحه و الغاؤه مباشرة قبل أغلاق الملف كما يتضح من الكود الاتى:

```
$fp = fopen( "myfile.txt", "a" );
flock( $fp, 2 ); // exclusive lock
// write to the file
flock( $fp, 1 ); // release the lock
fclose( $fp );
```

لاحظ أن عملية الاغلاق باستخدام الدالة flock() هو عمل احتراسى اى ان البرامج الاخرى التى لا تهتم باستخدام الدالة قد يسمح لها بالوصول الى نفس الملف فى نفس الوقت.

التعامل مع الأدلة الفرعية:

فيما يلى سنتعرف على الدوال التى تسمح لنا بالتعامل مع الادلة الفرعية بالقراءة او الكتابة او الحصول على معلومات عنها .

الدالة mkdir():

تقوم الدالة mkdir() بانشاء دليل فرعى و هى تقبل معاملين الأول هو مسار الدليل المراد أنشاؤه و الثانى هو حالة الدليل التى نريد نجعله بها و هو رقم ثمانى يتم كتابته دائما عن طريق صفر فى أول الرقم و هذا الرقم له معنى فقط فى النظام يونيكس و يتم عن طريقه تحديد صلاحيات الملف و يتكون من ثلاث مجموعات من الأرقام من 0 الى 7 تمثل الصلاحيات owner, group, everyone على الترتيب .

تقوم هذه الدالة بإرجاع القيمة true اذا تم انشاء الدليل الفرعى و القيمة false أذا حدثت مشكله فى الانشاء و عادة يتم ذلك أذا كان المطور ليس له صلاحيه لانشاء أدله على السيرفر .

```
mkdir( "mydir", 0777 ); // global
read/write/execute permissions
```

الدالة rmdir():

يمكنك أن تحذف دليل فرعى بالكامل أذا كنت تملك صلاحية لذلك و أذا كان الدليل الفرعى فارغ من الملفات و تقبل الدالة معامل واحد هو مسار الدليل المراد حذفه .

```
rmdir("mydir");
```

الدالة opendir():

لا يمكنك ان تقوم بقراءة محتويات دليل فرعى قبل الحصول على مؤشر لهذا الدليل و تقوم الدالة opendir() بذلك مهيئة الدليل للفتح و ترجع القيمة true أذا تم فتح الدليل بنجاح أو القيمة false أذا كان الدليل غير موجود مثلا :

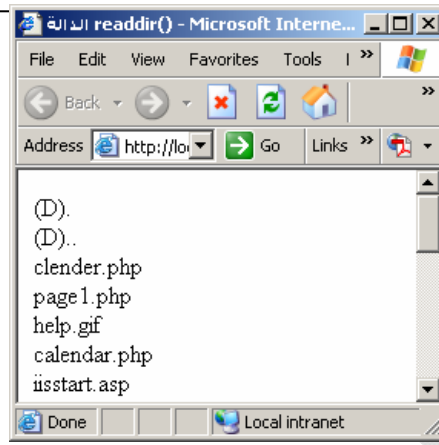
```
$dh = opendir( "mydir" );
```

الدالة readdir():

تقوم الدالة readdir() بقراءة محتويات الدليل الفرعى من أدله و ملفات مثلها مثل الدالة fgets() و تتطلب معامل واحد هو مؤشر الدليل التى تقوم بقراءته وتقوم بأرجاع الأسماء بدون المسار حتى نهاية الدليل التى تعطى عندها القيمة false .

مثال:

```
1: <html>
2: <head>
3: <title>الدالة readdir()</title>
4: </head>
5: <body>
7: <?php
8: $dirname = "mydir";
9: $dh = opendir( $dirname );
10: while ( gettype( $file = readdir( $dh )) !=
boolean )
11: {
12: if ( is_dir( "$dirname/$file" ) )
13: print "(D)";
14: print "$file<br>";
15: }
16: closedir( $dh );
17: ?>
18: </body>
19: </html>
```



لاحظ فى السطر رقم 10 أننا قمنا بأختبار نوع قيمة الدالة أذا كانت شرطية boolean و هذا لن يحدث الا أذا وصلت الدالة الى نهاية الدليل .
ثم نقوم فى السطر 12 بأختبار القيمة التى تم ارجاعها اذا كانت دليل يتم طباعة النص "(D)" بجانب أسم الدليل و بهذا نميزه عن الملفات الفرعيه .

لاحظ انه هناك طريقه أخرى للأختبار الموجود فى السطر 10 وهى :

```
while ( $file = readdir( $dh ) )
{
print "$file<BR>\n";
}
```

هذه الطريقه فعاله لانه عند عدم وجود قيم تستطيع الدالة readdir() قراءتها فأنها ترجع القيمة 0 لكن تحدث مشكله أذا كان هناك ملف أو دليل أسمه 0 فى هذه الحالة فأن القيمة false يتم تحقيقها و يتم الخروج من الحلقة بدون أستكمال القراءة.