

بسم الله الرحمن الرحيم



اللهم لا علم لنا إلا ما علمتنا إنك انت علام الغيوب

هذا الكتيب مدعوم من قبل

شبكة رسوم العربية للتصميم والبرمجة



<http://www.roosom.com>

الموقع العربي للمصادر المفتوحة



<http://www.arabicosd.info>

قرية PHP العربية



<http://www.phpvillage.com>

بعض الأشياء قبل البدء

تحتوي مكتبة Directory Function Library على مجموعة من الدوال وكائن واحد سأشير إليه باللون الأحمر

Directory Functions Library

opendir
closedir
readdir
rewinddir
chdir
getcwd
chroot
dir (Object)
scandir

وسأقوم بإذن الله تعالى بشرح كل دالة على حدا راجين من الله العلي القدير ان يوفقنا لما يحبه ويرضاه
وأن نقدم الخير اليوم عسى ان نجده في آخرتنا بإذن الله تعالى .

وأرجوا ممن يجد أي أخطاء ضمن الكتيب ان يحاول إبلاغي عبر البريد الإلكتروني او الماسنجر وهذا بريدي
وهو نفسه الماسنجر ايضاً فكما يقولون (جل من لايسهو)

eid200@gawab.com

أما بالنسبة للذين يريدون الحصول على نفس هذا الكتيب ولكن لشرح مكتبة Winzip فستجدونه في مكتبة
الموقع العربي للمصادر المفتوحة وهذا عنوان المكتبة

<http://www.arabicosd.info/download>

والآن اتركك عزيزي القارئ تقلب سطور هذا الكتيب الذي أرجوا ان تستفيد منه بإذن الله تعالى ولو بحرف
واحد

opendir

تستخدم هذه الدالة لفتح مجلد (Directory) معين والتعامل معه عن طريق المقبض (Handle) الذي تعيده هذه الدالة في حالة نجاح عملية الفتح وذلك بأخذ مسار المجلد المراد فتحه .

الشكل العام للدالة

```
resource opendir (string path)
```

البارامترات

كما تلاحظ عزيزي القارئ فإن الدالة من الصورة العامة لها تأخذ بارامتر واحد كالتالي :

path

وهذا البارامتر (path) يستخدم لتحديد مسار المجلد المراد فتحه بالدالة opendir فكما تلاحظ ان (path) هو من نوع (string) أي سلسلة حرفية ويكل بساطة حدد المسار كسلسلة حرفية مثل : ("/home/tempdir/").

قيمة الإعادة

كما تلاحظ عزيزي القارئ فإن هذه الدالة من الصورة العامة تعيد قيمة من النوع (resource) وهي تمثل مقبض (Handle) الإتصال مع المجلد المفتوح ولكن هذه القيمة في حالة نجاح عملية الإتصال اما في حالة فشل الإتصال بالمجلد وفتحها فإن الدالة تعيد FALSE .

مثال الاستخدام

```
$dhand = @opendir("images/");  
if(!$dhand)  
    die("عذراً لقد تعذر فتح المجلد");  
  
// else some code here if success open
```

و كما ترى عزيزي القارئ في المثال الموجود في الأعلى فلقد قمنا باستخدام الدالة opendir لفتح مجلد خاص بالصور اسمه images وإعادة النتيجة في متغير بإسم dhand والذي سيمثل مقبض الإتصال بالمجلد .

وبعد استخدام الدالة opendir يجب علينا اختبار النتيجة المخزنة في المتغير dhand لمعرفة حالة فتح المجلد (هل هي ناجحة ام فشلت عملية فتح المجلد) حيث قمنا بوضع جملة if والتأكد من نجاح الدالة او فشلها وفي حالة الفشل استخدمنا الدالة die لطباعة رسالة للمستخدم وإنهاء السكريبت اما في حالة النجاح فإنه يمكنك التعامل مع المجلد المفتوح وذلك عن طريق المتغير الذي يحتوي على مقبض الإتصال وهو المتغير dhand .

closedir

وأظن ان هذه الدالة لا تحتاج إلى شرح مطول وأظن أنك عزيزي القارئ قد فهمتها فالمهم أن هذه الدالة تستخدم لإغلاق المجلد الذي قمنا بفتحه باستخدام الدالة opendir .

الشكل العام للدالة

```
void closedir (resource dirhandle)
```

البارامترات

كما تلاحظ عزيزي القارئ فإن الدالة من الصورة العامة لها تأخذ بارامتر واحد كالتالي :

dirhandle

وهذا البارامتر (**dirhandle**) يستخدم لتحديد مقبض المجلد الذي تم فتحه بالدالة opendir فكما تلاحظ ان (**dirhandle**) هو من نوع (**resource**) وهي نفس القيمة التي تعيدها الدالة opendir والتي يتم الاحتفاظ بها في متغير و إذا انتهت العمليات على المجلد يتم تمرير المتغير الذي يحتوي على مقبض المجلد المفتوح إلى الدالة closedir لإغلاق المجلد .

قيمة الإعادة

كما تلاحظ عزيزي القارئ فإن هذه الدالة من الصورة العامة لاتعيد قيمة فالنوع المكتوب هو (**void**) وترجم في بعض الكتب على انها نوع عقيم أي لا يفعل شيئاً ويشيرون لها في الكتب الإنجليزية بكلمة (**nothing**) أي لا شيء .

مثال الإستخدام

```
$dhand = @opendir("images/");
if(!$dhand)
    die("عذراً لقد تعذر فتح المجلد");

// else some code here if success open

closedir($dhand);
```

و كما ترى عزيزي القارئ في المثال الموجود في الأعلى فلقد قمنا باستخدام الدالة opendir لفتح مجلد خاص بالصور اسمه **images** وإعادة النتيجة في متغير بإسم **dhand** والذي سيمثل مقبض الإتصال بالمجلد .

وبعد استخدام الدالة **opendir** يجب علينا اختبار النتيجة المخزنة في المتغير **dhand** لمعرفة حالة فتح المجلد (هل هي ناجحة ام فشلت عملية فتح المجلد) حيث قمنا بوضع جملة **if** والتأكد من نجاح الدالة او فشلها وفي حالة الفشل استخدمنا الدالة **die** لطباعة رسالة للمستخدم وإنهاء السكريبت اما في حالة النجاح فإنه يمكنك التعامل مع المجلد المفتوح وذلك عن طريق المتغير الذي يحتوي على مقبض الإتصال وهو المتغير **dhand** .

وبعد إجراء بعض العمليات على المجلد والإنتهاء منه تم إغلاق المجلد باستخدام الدالة **closedir** وذلك بتمرير مقبض المجلد المفتوح إليها لإغلاقه .

getcwd

تستخدم هذه الدالة بشكل فردي أي دون استخدام أي دالة معها مثل **opendir** أو **closedir** فمهمة هذه الدالة هي احضار مسار المجلد الحالي أو الذي أقوم بالعمل ضمنه وهذه الدالة مفيدة في حال كنت تعمل في مجلد معين وأنا لا أعرف مساره فهذه الدالة ستتأكد بجلب مسار المجلد الذي تعمل فيه حالياً (أو كما يقولون مجلد العمل النشط) .
أما **cwd** فهي اختصار لـ (**Current Working Directory**) أي مجلد العمل الحالي أو المجلد النشط والمقصود بالنشط أي الذي تعمل فيه حالياً

الشكل العام للدالة

```
string getcwd (void)
```

البارامترات

كما تلاحظ عزيزي القارئ فإن الدالة من الصورة العامة لها لا تأخذ أي بارامتر فالكلمة (**void**) كما أشرنا لها في السابق لاتعمل أي شئ وهذا لا يعتبر بارامتر للدالة

قيمة الإعادة

كما تلاحظ عزيزي القارئ فإن هذه الدالة من الصورة العامة تعيد قيمة من النوع (**string**) وهي سلسلة حرفية تمثل مسار المجلد الحالي الذي تعمل فيه

مثال الاستخدام

على افتراض ان المجلد الذي تعمل فيه هو ("**home/images/**") وأنا لا أعلم بمسار المجلد الحالي الذي تعمل فيه .

```
$cdir = getcwd();  
echo ($cdir); // output is : home/images/
```

و كما ترى عزيزي القارئ في المثال الموجود في الأعلى فكما افترضنا ان مسار العمل الحالي هو ("**home/images/**") وأنا لا أعرف المسار الذي تعمل فيه فإن الدالة تعيد هذا المسار وتحفظ به في المتغير **cdir** .

وفي الأخير قمنا بطباعة المسار الحالي باستخدام الدالة **echo** وتمرير المسار الحالي الذي تم حفظه في المتغير **cdir** وأرجوا من الله العليّ القدير أن تكون الفكرة قد وصلت إلى ذهنك عزيز القارئ حول وظيفة هذه الدالة .

chdir

تستخدم هذه الدالة بشكل فردي أي دون استخدام أي دالة معها مثل **opendir** أو **closedir** أيضاً فمهمة هذه الدالة هي تغيير مسار المجلد الحالي أو النشط والذي يمكننا معرفته بالدالة **getcwd** السابق شرحها .

ملاحظة

الخلاصة الأساسية من وراء الدالة **chdir** هي لتغيير مسار المجلد الحالي أو النشط الذي يمكننا معرفته باستخدام الدالة **getcwd** السابق شرحها اما اسم الدالة فهو اختصار للعبارة (**Change Directory**) وهي تؤثر على قيمة الدالة **getcwd** حيث تغير القيمة التي تشير إليها دالة **getcwd** وتتحكم فيها.

دالة chroot

تتضمن المكتبة Directory Function Library دالة باسم **chroot** وهي تشبه لحد كبير الدالة **chdir** لذلك لن اتطرق لشرحها .

الشكل العام للدالة

```
bool chdir (string directory)
```

البارامترات

كما تلاحظ عزيزي القارئ فإن الدالة من الصورة العامة لها فإنها تأخذ بارمتر واحد كالتالي :

directory

وهذا البارامتر (**directory**) وهو من نوع (**string**) أي سلسلة حرفية ويتم فيه تحديد مسار المجلد المراد جعله المجلد النشط ولمعرفة المجلد النشط الرجاء **راجع شرح الدالة getcwd** لمعرفة المجلد النشط.

قيمة الإعادة

كما تلاحظ عزيزي القارئ فإن هذه الدالة من الصورة العامة تعيد قيمة من النوع (**bool**) وهي قيمة بوليانية ولذلك ستكون النتيجة في حالة نجاح تغيير المجلد النشط هي **TRUE** أما في حالة الفشل فهي **FALSE** .

مثال الاستخدام

على افتراض ان المجلد الذي اعمل فيه هو ("home/images/") لذلك ستكون قيمة الدالة **getcwd** هي نفس المسار وعلى افتراض ان مجلد **images** الذي اعمل فيه يحتوي بداخله على مجلد باسم **abdullaheid** وأنا اود تغيير المسار إلى داخل هذا المجلد .
فيما أنني داخل مجلد **images** حالياً فسأكتفي بتحديد اسم المجلد **abdullaheid** للدالة **chdir** وإليك المثال

```
$cdire = getcwd();  
echo ($cdire); // output is : home/images/  
chdir ('abdullaheid');  
  
$cdire = getcwd();  
echo ($cdire); // output is : home/images/abdullaheid/
```

و كما ترى عزيزي القارئ في المثال الموجود في الأعلى فكما افترضنا ان مسار العمل الحالي هو ("home/images/") فإن الدالة **getcwd** تعيد هذا المسار وتحتفظ به في المتغير **cdire** .

وبعد استدعاء الدالة **chdir** لتغيير المسار والدخول إلى مجلد **abdullaheid** فإن الدالة **getcwd** قد تأثرت وأصبح مخرجاتها هي المسار الذي قمنا بالدخول إليها بواسطة الدالة **chdir** وأرجوا من الله العلي القدير أن تكون الوظيفة الخاصة بهذه الدالة قد تم فهمها على الوجه الأكمل بإذن الله تعالى .

readdir

تقوم هذه الدالة بقراءة المجلد الذي تم فتحه باستخدام الدالة `opendir` وتعيد لنا اسم أول ملف أو مجلد موجود داخل المجلد الذي قمنا بفتحه بالدالة `opendir` و إذا استخدمت الدالة `readdir` مباشرة وبدون أي شيء فإنها سترجع فقط الملف أو المجلد الأول داخل المجلد الذي تم فتحه والسبب في ذلك أنك عندما تفتح أي مجلد بالدالة `opendir` فإن هناك شيء يسمى (Virtual Pointer) أو المؤشر الافتراضي وهو يشير إلى أول ملف أو مجلد في المجلد المفتوح وللوصول لجميع الملفات والمجلدات الموجودة ضمن المجلد المفتوح فإننا نستخدم حلقة مثل `while` لكي نقوم بتحريك المؤشر الافتراضي إلى الملف أو المجلد الثاني ثم الثالث وهكذا حتى نصل إلى النهاية .

الشكل العام للدالة

```
string readdir (resource dirhandle)
```

البارامترات

كما تلاحظ عزيزي القارئ فإن الدالة من الصورة العامة لها تأخذ بارامتر واحد كالتالي :

dirhandle

وهذا البارامتر (`dirhandle`) يستخدم لتحديد مقبض المجلد الذي تم فتحه بالدالة `opendir` فكما تلاحظ ان (`dirhandle`) هو من نوع (`resource`) وهي نفس القيمة التي تعيدها الدالة `opendir` والتي يتم الاحتفاظ بها في متغير وللقراءة يجب ان نحدد مقبض المجلد الذي تم فتحه بالدالة `opendir` ونمرره للدالة `readdir` لكي تعرف الدالة المجلد الذي يجب القراءة منه

قيمة الإعادة

كما تلاحظ عزيزي القارئ فإن هذه الدالة من الصورة العامة تعيد قيمة من النوع (`string`) وهي سلسلة حرفية تمثل اسم المجلد أو الملف الذي تمت قراءته باستخدام الدالة .
تعيد الدالة القيمة `FALSE` في حال فشلها أو انتهائها من قراءة كامل المجلد والوصول إلى نهايته

ملاحظة

هذه الدالة لا تقرأ الملفات التي بداخل المجلد أي هي لا تقوم بفتحها وقراءتها ومعرفة محتويات الملف أو المجلد وإنما فقط تقرأ أسماء الملفات والمجلدات الموجودة داخل المجلد الذي قمنا بفتحه باستخدام الدالة `opendir` .

مثال الاستخدام

على افتراض ان المجلد الذي سأفتحه هو ("`home/images/`") ويحتوي هذا المجلد على الملفات التالية :
-1 `Abdullah.gif`
-2 `eid.swf`

```
$dhand = @opendir("home/images/");  
if (!$dhand)  
    die("عذراً لقد تعذر فتح المجلد");  
  
while (false !== ($file = @readdir($dhand))) {  
    echo "$file\n";  
}  
closedir($dhand);
```

و كما ترى عزيزي القارئ في المثال الموجود في الأعلى فلقد قمنا باستخدام الدالة `opendir` لفتح مجلد خاص بالصور اسمه `home/images/` وإعادة النتيجة في متغير بإسم `dhand` والذي سيمثل مقبض الإتصال بالمجلد .

وبعد استخدام الدالة `opendir` يجب علينا اختبار النتيجة المخزنة في المتغير `dhand` لمعرفة حالة فتح المجلد (هل هي ناجحة ام فشلت عملية فتح المجلد) حيث قمنا بوضع جملة `if` والتأكد من نجاح الدالة او فشلها وفي حالة الفشل استخدمنا الدالة `die` لطباعة رسالة للمستخدم وإنهاء السكريبت اما في حالة النجاح فإنه يمكنك التعامل مع المجلد المفتوح وذلك عن طريق المتغير الذي يحتوي على مقبض الإتصال وهو المتغير `dhand` .

والآن إذا نجح فتح المجلد فإنه ستنتم الإشارة إلى أو ملف في المجلد بواسطة المؤشر الافتراضي وسيكون أول ملف هو `Abdullah.gif` فستقرأ الدالة `readdir` اسم الملف الأول وتخزنه في المتغير `file` وبعد ذلك يتم التحقق من قيمة المتغير `file` في نفس رأس الحلقة `while` ثم يطبع الملف الأول وهو `Abdullah.gif` ثم يكرر الحلقة فيجد الملف `eid.swf` ويطبعه ثم يعيد الحلقة ولا يجد شيئاً فيعيد `false` وينتهي الحلقة `while` .

وبعد إجراء العمليات على المجلد و الإنتهاء منه تم إغلاق المجلد باستخدام الدالة `closedir` وذلك بتمرير مقبض المجلد المفتوح إليها لإغلاقه .

ملاحظة هامة

عند ما يتم القراءة بواسطة الدالة `readdir` إلى نهاية المجلد ومن ثم لم يتم الإغلاق باستخدام الدالة `closedir` أي تم إجراء عمليات أخرى ثم العودة للقراءة مرة أخرى من المجلد فإن المؤشر الافتراضي (`Virtual Pointer`) سيكون يشير إلى آخر ملف أو مجلد ضمن المجلد الذي قمنا بفتحه ولم نغلقه عندها يجب عليك استخدام الدالة `rewinddir` لإرجاع المؤشر الافتراضي إلى أول ملف أو مجلد ضمن المجلد الذي قمنا بفتحه وذلك إذا لم تقم بإغلاق المجلد باستخدام الدالة `closedir` .

وستكون الدالة `rewinddir` هي محور حديثنا بإذن الله تعالى في الصفحة القادمة .

rewinddir

وظيفة هذه الدالة هي إرجاع مؤشر القراءة الافتراضي ليشير إلى أول ملف أو مجلد موجود ضمن المجلد الذي فتحناه بواسطة الدالة **opendir** وذلك قبل إغلاق المجلد فلو أغلق المجلد ثم فتحته مرة أخرى فإن الدالة **opendir** أصلاً تضع المؤشر الافتراضي عند أول ملف داخل المجلد ولكن سبب استخدام هذه الدالة له حالات عدة منها نذكر على سبيل المثال لو كان لديك مجلد وأردت القراءة منها ثم قرأت جميع مداخله من ملفات ومجلدات باستخدام حلقة **while** مثلاً ثم وصلت إلى آخر ملف ضمن المجلد المفتوح وبعد ذلك أنت لا تريد إغلاق المجلد بدالة **closedir** بل تريد إجراء بعض العمليات على الملفات فيجب عليك إرجاع المؤشر الافتراضي لأول ملف في المجلد الذي فتحناه بالدالة **opendir** و **rewinddir** هي الدالة المخصصة لإرجاع المؤشر الافتراضي (**Virtual Pointer**) والذي شرحناه سابقاً .

وهذه الدالة تشبه إلى حد كبير (بل تستطيع أن تقول أنها مثل) الدالة **reset** الخاصة بالمصفوفات (**array**) فالدالة **reset** تقوم بإرجاع المؤشر الافتراضي الخاص بالمصفوفة إلى أول عنصر في المصفوفة .

فلو تخليت الآن أن المجلد المفتوح عبارة عن مصفوفة فإنك ستفهم الدالة **rewinddir** بسهولة

الشكل العام للدالة

```
void rewinddir (resource dirhandle)
```

البارامترات

كما تلاحظ عزيزي القارئ فإن الدالة من الصورة العامة لها تأخذ بارامتر واحد كالتالي :

dirhandle

وهذا البارامتر (**dirhandle**) يستخدم لتحديد مقبض المجلد الذي تم فتحه بالدالة **opendir** فكما تلاحظ أن (**dirhandle**) هو من نوع (**resource**) وهي نفس القيمة التي تعيدها الدالة **opendir** والتي يتم الاحتفاظ بها في متغير و إذا انتهت العمليات على المجلد يتم تمرير المتغير الذي يحتوي على مقبض المجلد المفتوح إلى الدالة **rewinddir** لإرجاع المؤشر الافتراضي إلى أول ملف في المجلد .

قيمة الإعادة

كما تلاحظ عزيزي القارئ فإن هذه الدالة من الصورة العامة لاتعيد قيمة فالنوع المكتوب هو (**void**) وترجم في بعض الكتب على أنها نوع عقيم أي لا يفعل شيئاً ويشيرون لها في الكتب الإنجليزية بكلمة (**nothing**) أي لا شيء .

مثال الاستخدام

على افتراض أن المجلد الذي سأفتحه هو ("**home/images/**") ويحتوي هذا المجلد على الملفات التالية :

Abdullah.gif - ١
eid.swf - ٢

```
$dhand = @opendir("home/images/");  
if (!$dhand)  
    die("عذراً لقد تعذر فتح المجلد");  
  
while (false != ($file = @readdir($dhand))) {  
    echo "$file\n";  
}  
  
rewinddir($dhand);  
$xx = readdir($dhand);  
echo ($xx);  
closedir($dhand);
```

و كما ترى عزيزي القارئ في المثال الموجود في الأعلى فلقد قمنا باستخدام الدالة **opendir** لفتح مجلد خاص بالصور اسمه **home/images/** وإعادة النتيجة في متغير بإسم **dhand** والذي سيمثل مقبض الإتصال بالمجلد .

وبعد استخدام الدالة **opendir** يجب علينا اختبار النتيجة المخزنة في المتغير **dhand** لمعرفة حالة فتح المجلد (هل هي ناجحة أم فشلت عملية فتح المجلد) حيث قمنا بوضع جملة **if** والتأكد من نجاح الدالة أو فشلها وفي حالة الفشل استخدمنا الدالة **die** لطباعة رسالة للمستخدم وإنهاء السكريبت أما في حالة النجاح فإنه يمكنك التعامل مع المجلد المفتوح وذلك عن طريق المتغير الذي يحتوي على مقبض الإتصال وهو المتغير **dhand** .

والآن إذا نجح فتح المجلد فإنه سستتم الإشارة إلى أو ملف في المجلد بواسطة المؤشر الافتراضي وسيكون أول ملف هو **Abdullah.gif** فستقرأ الدالة **readdir** اسم الملف الأول وتخزنه في المتغير **file** وبعد ذلك يتم التحقق من قيمة المتغير **file** في نفس رأس الحلقة **while** ثم يطبع الملف الأول وهو **Abdullah.gif** ثم يكرر الحلقة فيجد الملف **eid.swf** ويطبعه ثم يعيد الحلقة ولا يجد شيئ فيعيد **false** وينتهي الحلقة **while** .

والآن بعد انتهاء الحلقة فإن المؤشر الافتراضي يشير إلى آخر المجلد المفتوح لذلك قمنا باستخدام الدالة **rewinddir** لإعادة المؤشر إلى بداية المجلد المفتوح ولاحظ أنه لم تغلق المجلد ثم بعد ذلك قمت باستخدام الدالة **readdir** بدون حلقة لقراءة الملف الأول في المجلد وحفظ اسم الملف في المتغير **xx** ثم طباعته .

وبعد إجراء العمليات على المجلد و الإنتهاء منه تم إغلاق المجلد باستخدام الدالة **closedir** وذلك بتمرير مقبض المجلد المفتوح إليها لإغلاقه .

وهكذا أرجوا أن تكون فكرة الدالة **rewinddir** قد وصلت إلى ذهنك عزيزي القارئ بإذن الله تعالى .

scandir

تستخدم هذه الدالة بشكل فردي أي دون استخدام أي دالة معها مثل **opendir** أو **closedir** أيضاً فمهمة هذه الدالة هي قراءة مجلد **بالكامل** ووضعه ضمن مصفوفة كل عنصر فيها يحتوي على اسم ملف أو مجلد موجود ضمن المجلد الذي تعاملنا معه باستخدام **scandir**

ملاحظة

لقد تم دعم الدالة **scandir** وإضافتها إلى مكتبة **Directory Function Library** في **PHP 5** .

الشكل العام للدالة

```
array scandir (string directory,[ int sorting],[resource context])
```

البارامترات

كما تلاحظ عزيزي القارئ فإن الدالة من الصورة العامة لها فإنها تأخذ ثلاث كالتالي :

directory

وهذا البارامتر (**directory**) وهو من نوع (**string**) أي سلسلة حرفية ويتم فيه تحديد مسار المجلد المراد قراءته .

sorting

وهذا البارامتر (**sorting**) وهو من نوع (**int**) أي متغير عددي (رقمي) وهذا البارامتر افتراضي أي بإمكانك وضعه أو تجاهله و أما الهدف من هذا البارامتر فهو لتحديد طريقة ترتيب الملفات ضمن المصفوفة فهل هو تصاعدي أو تنازلي فإذا تم ترك هذا البارامتر ولم يزود بقيمة فإنه سيقوم بترتيب المصفوفة بشكل تصاعدي أما إذا أعطيته القيمة ١ فإنه سيرتبه بشكل تنازلي حسب أسماء الملفات هجائياً .

context

وهذا البارامتر (**context**) وهو من نوع (**resource**) وهو افتراضي أيضاً ولكن لم أجد تعريف ثابت لهذا البارامتر ضمن **php Documentation** فقد يكون هذا البارامتر تم وضعه لأغراض مستقبلية .

قيمة الإعادة

كما تلاحظ عزيزي القارئ فإن هذه الدالة من الصورة العامة تعيد قيمة من النوع (**bool**) وهي قيمة بوليانية ولذلك ستكون النتيجة في حالة نجاح تغيير المجلد النشط هي **TRUE** أما في حالة الفشل فهي **FALSE** .

مثال الاستخدام

سأقوم بفتح المجلد ("**/tmp/**") وقراءة كامل الملفات والمجلدات الموجودة فيه و حفظها ضمن مصفوفة الأولى مرتبة تصاعدياً والثانية سأقرأ المجلد وارتبها تنازلياً وذلك بإعطاء البارامتر **sorting** القيمة ١ . ثم سأطبع المصفوفتين باستخدام الدالة **print_r** الخاصة بطباعة المصفوفات

```
$dir = '/tmp/';
$files1 = scandir($dir);
$files2 = scandir($dir, 1);

print_r($files1);
print_r($files2);
```

و كما ترى عزيزي الفارئ في المثال الموجود في الأعلى فلقد وضعنا مسار المجلد في متغير بإسم **dir** ثم استخدمنا الدالة **scandir** بفرز تصاعدي ووضعنا المصفوفة في المتغير **files1** ثم بنفس الطريقة استخدمنا الدالة **scandir** لقراء المجلد ولكن بفرز تنازلي وتخزين المصفوفة في المتغير **fiels2** ثم بعد ذلك طبعنا كلا المصفوفتين واليك النتيجة

```
Array //files1
(
    [0] => .
    [1] => ..
    [2] => bar.php
    [3] => foo.txt
    [4] => somedir
)
Array //files2
(
    [0] => somedir
    [1] => foo.txt
    [2] => bar.php
    [3] => ..
    [4] => .
)
```

Directory Function Library ومكتبة OOP

عزيزي القارئ بقي شيء آخر اقلوه وهو ان هذه المكتبة تحتوي على **class** باسم **dir** وهذا الكلاس يستخدم بعض الدوال التي شرحناها ويقوم بتغليفها فقط وهذا الصورة العامة للكائن .

```
class dir {  
  
    dir ( string directory)  
  
    string path  
  
    resource handle  
  
    string read ( void )  
  
    void rewind ( void )  
  
    void close ( void )  
  
}
```

أي لن تحتاج لتعلم شيء جديد فقط ستستخدم الدوال التي تعلمناها ولكن باختلاف بسيط حيث ان هذا الكائن يحتوي بداخله على متغير يحتفظ بمقبض المجلد المفتوح يعني ان الدوال التي تحتاج الى المقبض كـ **rewind** و **close** فإن دوال الكائن لا تحتاج لتمرير هذا البارامتر لها لأنها تحتفظ به في داخل الكائن وأنا أعرف ان هذا الكلام قد يكون صعب أو غير مفهوم نوعاً ما ولكن سأريك الطريقة بإذن الله .

المتغير **path** هو من نوع **string** أي سلسلة حرفية وهو يحتفظ بمسار المجلد الذي مررناه له .

انظر إلى المتغير **handle** الموجود بالأعلى فهو يمثل مقبض المجلد المفتوح وانظر إلى دالة **read** دالة **rewind** و **close** فإنها لا تحتاج إلى تمرير مقبض الإتصال كما كنا نفعل مع دوال المكتبة والسبب هو ان الكائن يحتفظ بمقبض المجلد المفتوح في المتغير **handle**

ولفتح المجلد استخدم **dir** مع مسار المجلد بين قوسين واحتفظ بالقيمة ضمن متغير والذي سيمثل الكائن وإليك مثال .

```
$d = dir("/etc");  
echo "Handle: " . $d->handle . "<br />\n";  
echo "Path: " . $d->path . "<br />\n";  
while (false != ($entry = $d->read())) {  
    echo $entry."<br />\n";  
}  
$d->close();
```

لاحظ عزيزي القارئ من المثال في الأعلى فلقد قمنا باستخدام الكلاس **dir** ومررنا المجلد المراد فتحه ثم حفظنا القيمة ضمن متغير باسم **d**

وبعد ذلك سهلت العملية فكل ما عليك هو استدعاء دوال الكائن فقط فـ **read** للقراءة وهي نفسها الدالة **readdir** ولكن بعد اجراء عملية فبركة بسيطة عليها .

وأرجوا أن تكون الفكرة الخاصة بالكائن قد وصلت إلى ذهنك عزيزي القارئ بإذن الله تعالى .

is_dir

هذه الدالة غير مدرجة ضمن مكتبة **Directory Function Library** ولكن تستخدم بكثرة مع هذه المكتبة وفائدتها هي انك تقوم بتمرير مسار خاص بمجلد معين فتتأكد هذه الدالة من ان هذا المسار هو مجلد او لا فإذا كان مجلد فإنها ترجع القيمة **TRUE** وإذا لم يكن مجلد ترجع القيمة **FALSE** وهذا مفيد جداً إذا يجب استخدام هذه الدالة قبل إجراء عملية فتح للمجلد باستخدام الدالة **opendir** وذلك للتأكد قبل إجراء عملية الفتح .

الشكل العام للدالة

```
bool is_dir (string directory)
```

البارامترات

كما تلاحظ عزيزي القارئ فإن الدالة من الصورة العامة لها فإنها تأخذ بارمتر واحد كالتالي :

directory

وهذا البارامتر (**directory**) وهو من نوع (**string**) أي سلسلة حرفية ويتم فيه تحديد المسار المراد التأكد من أنه مجلد او لا .

قيمة الإعادة

كما تلاحظ عزيزي القارئ فإن هذه الدالة من الصورة العامة تعيد قيمة من النوع (**bool**) وهي قيمة **بوليانية** ولذلك ستكون النتيجة في حالة نجاح تغيير المجلد النشط هي **TRUE** أما في حالة الفشل فهي **FALSE** .

مثال الاستخدام

```
if(!is_dir("home/images/"))  
    die("عذراً المسار المحدد لا يمثل مجلد");  
//else output is : home/images/
```

كما تلاحظ عزيزي القارئ فلقد قمنا باستخدام الدالة **is_dir** في البداية للتأكد من أن المسار الممرر يمثل مجلد او لا فإذا كان لا يمثل مجلد يتم إنهاء السكريبت و إعطاء المستخدم رسالة تنفيد بأن المسار المحدد لا يمثل مجلد صحيح او تكمل باقي السكريبت في حال كان المسار المحدد للدالة **is_dir** هو مجلد صحيح .

مثال شامل

هنا في هذه الفقرة الأخيرة سأعطي مثال شامل لما تمت دراسته وسيكون مفيد بإذن الله تعالى .

```
<?php
$dir = "/tmp/";

// Open a known directory, and proceed to read its contents
if (is_dir($dir)) {
    if ($dh = opendir($dir)) {
        while (($file = readdir($dh)) !== false) {
            echo "filename: $file : filetype: " . filetype($dir . $file) .
"\n";
        }
        closedir($dh);
    }
}
?>
```


وأخيراً

وليس أخيراً أرجو أن تكون قد استفدت من هذا الكتيب وأرجو منك الدعاء لي في ظهر الغيب وكما ذكرنا سابقاً فإن السلسلة ستستمر بهذا النمط حسب الوقت المتوفر لي و بعد انتهائي من مجموعة من الكتيبات سأقوم بضمها تحت كتاب واحد بإذن الله وسأصدر الكتيب الثالث بإذن الله تعالى في أقرب وقت ممكن
راجين من الله العلي العظيم أن يوفقنا لما يحبه ويرضاه

أخوكم عبد الله عيد الحمادي

eid200@gawab.com