الفصل الثالث عشر
التعامل مع الملفات والمجلدات

كل مبرمج يجب أن يتعامل مع الملفات والمجلدات في بعض النقاط، برنامجك سوف يستخدم الملفات لكي يقوم بتخزين معلومات الإعداد للسكريبت، أو يقوم بتخزين البيانات لقراءتها وكتابتها، أو لكي يقوم بحفظ البيانات المؤقتة، وكمثال فإن أتفه برنامج عداد يحتاج إلى ملف يقوم بتخزين آخر رقم تم الوصول إليه..

الملف: ليس عبارة عن أكثر من بايتات متسلسلة يتم تخزينها على القرص الصلب أو أي مادة تخزينية أخرى.
والمجلد: هو عبارة عن نوع محدد من الملفات يحتفظ بأسماء ملفات أخرى ومجلدات أخرى (تسمى بالمجلدات الفرعية)، كل ما تحتاجه لتتعامل مع الملفات والمجلدات هو كيف يمكنك ربط سكريبتك بهم..

هذا الفصل سيأخذك في جولة لتعلم التعامل مع الملفات والمجلدات وفي نفس الوقت يوفر لك مرجعية لبعض الدوال التي تساعدك في ذلك مما يجعل مهمتك أسهل...

سيقوم هذا الفصل بتغطيه المواضيع التالية:

١ فتح وإغلاق الملف.
٢ القراءة من الملف والكتابة فيه.
٣ مسح وإعادة تسمية الملفات
٤ استعراض وتجوال في الملف.
٥ فتح وإغلاق المجلدات.
٦ نسح وإعادة تسمية المجلدات.

obeikandi.com

ملاحظة:

قبل أن نبدأ دعنا ننبهك أن التعامل مع الملفات يختلف من نظام تشغيل إلى آخر ففي أنظمة اليونكس تستخدم المسارات العلامة المائلة للأمام.

مثال

```
/home/usr/bin/data.txt
```

بينما في الويندوز فإن المسار يكون كالتالي:

```
C:\usr\bin\perl
```

وإذا استخدمنا العلامة الأمامية في PHP للويندوز فإنه يقوم بتحويلها بشكل تلقائي إلى علامة خلفية بينما إذا أردنا استخدام العلامة الأمامية فإننا يجب أن نقوم بتكرار العلامة لكي يتم التعرف عليها.

```
PHP\C:\\windows\
```

التعامل مع الملفات

يوفر الـPHP نوعين من الدوال المتعلقة بالملفات فهناك نوع من الدوال يستخدم مقبض للملف (file handle) أو ما يسمونه بالمؤشر (pointer) في العادة، بينما بعض الدوال تستخدم قيمة حرفية تشير إلى موضع الملف مباشرة...

مقبض الملف ليس أكثر من عدد صحيح (integer) يقوم بتعريف الملف المراد فتحه حتى يتم إغلاقه، إذا كان هناك أكثر من ملف مفتوح فإن لكل ملف مقبضه التعريفي الخاص به، وبالطبع فإنه لا يتوجب عليك معرفة هذا الرقم....

على سبيل المثال فإن الدالة ()fwrite تقوم بفتح الملف لكتابة بيانات إليه وهي تستخدم مقبض لكي تقوم بالتعرف إلى الملف وفتحه..

```
'Hello World'); ،Fwrite ($fp
```

بينما الدالة ()file التي تستخدم للقراءة من الملف تقوم باستخدام قيمة نصية تقوم بالإشارة إلى مكان الملف بشكل مباشر لكي يتم التعامل معه..

openkanal.com

لا تصب بالرعب والخوف من هذا الكلام فأنا أعلم أنه قد يكون غامضاً عليك.. تنفس الصعداء وجهز لنفسك كأساً من الشاي لأننا سنبدأ في الجد الآن....

ملاحظة: ستجد أن أغلب الدوال أو معظمها أو كلها تقريباً تقوم بإرجاع القيمة True إذا تمت بنجاح والقيمة False إذا فشلت في الحصول على هدفها..
لنبدأ الآن مع سكريبتات مبسطة للعمل مع الملفات..

فتح وإغلاق الملفات

Fopen:

تستخدم هذه الدالة ثلاثة عوامل هي مسار الملف (path) والوضع له (للقراءة، للكتابة.....) بالإضافة إلى مسار الـInclude فيه وتقوم هذه الدالة بإرجاع مقبض للملف...

قد تواجهنا مشاكل أحياناً فقد يكون الملف غير منشأ أو أننا لا نملك صلاحيات عليه ولذلك فإنه يمكننا اختبار القيمة التي ترجعها هذه الدالة فإذا كانت القيمة صفراً فهذا معناه أن الدالة فشلت في إرجاع مقبض الملف أو نوعه، أما إذا كانت القيمة هي واحد فهذا معناه أن الدالة قد نجحت في فتح الملف.

مثال

```
$fp=fopen ("./data.txt", "r");
if (!$fp) die ("فشل في قراءة الملف تأكد من التراخيص ومن مسار الملف");
```

يمكننا كتابة المثال أيضاً بالشكل التالي:

```
If (!($fp=fopen ("./data.txt", "r"))) die ("لا يمكن القراءة من الملف");
```

obeikandi.com

لاحظ أننا قلنا سابقاً إن هناك دوال تستخدم للتعامل مع الملفات تستخدم مقبض وهذا المقبض هو عبارة عن رقم، في مثالنا هذا يتحدد رقم المقبض الذي هو المتغير $fp الذي يخزن فيه مكان الملف وما إذا كان قابلاً للفتح أو لا أو يعمل أو لا يعمل، والنتيجة التي تتخزن في المتغير $fp هي رقم مثلما قلنا سابقاً وهو صفر إذا كان الملف لا يعمل أو واحد إذا تم فتح الملف بنجاح..

الآن دعنا نناقش معاملات الدالة fopen التي تقوم بإعطائنا رقم المقبض..

أول معامل هو مسار الملف على القرص الصلب

لنفرض أن لديك مجلداً قمت بإنشائه في مجلد السكريبتات الرئيسي لديك الذي يسمى htdocs وأسميته data

ولنفرض أن سكريبتك يستخدم ملفين:
١- ملف للقراءة والكتابة يسمي data.txt.
٢- وملف يقوم بعرض المدخلات والإضافة إليها اسمه script.txt.

حسناً لدينا الآن ثلاث حالات للسكريبت

الحالة الأولى:
أن يكون الملفان في نفس المجلد (data) وعند ذلك يمكنك فتح الملف الذي تريد فتحه بذكر اسمه فقط من غير إضافات.

```
“r”);،$fp=fopen (“data.txt”
```

obeikandi.com

الحالة الثانية:

أن يكون هناك مجلد في نفس مجلد الـdata باسم آخر ولنقل أن هذا الاسم هو gb وفيه ملف data.txt على ذلك فإننا نكتب المسار المطلق لهذا المجلد كالتالي:

```
“r”);،$fp=fopen (“./gb/data.txt”
```

الحالة الثالثة:

أن يكون الملف الذي تريد قراءته موجود في المجلد htdocs بينما السكريبت موجود في المجلد data الموجود داخل htdocs على ذلك نكتب المسار النسبي كالتالي:

```
“r”);،$fp=fopen (“../data.txt”
```

لاحظ النقطة التي تسبق العلامة الأمامية جيداً..

أتمنى أن تكون فهمت من هذا الكلام ما هو المقصود بالمسار المطلق والمسار النسبي..

يمكننا أيضاً وضع رابط صفحة في موقع آخر ولكننا لن نستطيع الكتابة عليه بل قراءته فقط.

مثال:

```
“r”))) ،If  (!($fp=fopen (“http://www.swalif.net/softs/index.php”
”);die (“لا يمكن القراءة من الملف
```

ينقصنا نقطة يجب أن نتكلم عنها وهي عند تحديد العامل use_include_path

obeikandl.com

العامل الثاني الذي نستخدمه للملفات هو حالة الملف:

(للقراءة، للكتابة، للإضافة إليه) يحدد وضعية الملف حال فتحه إذا كان للقراءة فقط أو للكتابة فقط أو للاثنين معاً أو للإضافة، وأرتبها هنا في جدول بسيط..

القيمة	الوصف
r	تفتح الملف للقراءة فقط ويكون المؤشر في بداية الملف
r+	يفتح الملف للقراءة والكتابة ويضع المؤشر في بداية الملف
w	يفتح الملف للقراءة فقط، أي بيانات موجودة سيتم مسحها، إذا لم يكن الملف موجوداً سيحاول PHP إنشاؤه.
w+	يفتح الملف للقراءة والكتابة، أي بيانات موجودة سيتم مسحها، إذا لم يكن الملف موجوداً سيحاول PHP إنشاؤه.
a	يفتح الملف للإضافة فقط، إذا لم يكن الملف موجوداً سيحاول PHP إنشاؤه، سيكون المؤشر في نهاية الملف.
a+	يفتح الملف للقراءة و للإضافة، إذا لم يكن الملف موجوداً سيحاول PHP إنشاؤه، سيكون المؤشر في نهاية الملف.
b	يستخدم لفتح وقراءة ملفات الصور على نظام أو سيرفرات الويندوز فقط.. أما اللينوكس فالعوامل السابقة تتعامل مع ملفات الصور بشكل عادي..

هناك مؤشر للملفات يحدد إذا ما كنت ستكتب من نهاية أو بداية الملف أو حتى من وسطه أو من أي مكان بالملف، ستعرف كيفية التحكم بهذا المؤشر بعد قليل.

obeikandi.com

العامل الثالث هو تحديد **use_include_path**

فإذا قمت بتحديد قيمته إلى (١) وقمت بكتابة اسم الملف مباشرة فسيبحث الـPHP عن الملف في نفس المجلد الموجود به السكريبت ثم سيقوم بالبحث عن الملف في المجلدات التي تم تحديدها في المتغير use_include_path في ملف php.ini

```
;;;;;;;;;;;;;;;;;;;;;;
; Paths and Directories ;
;;;;;;;;;;;;;;;;;;;;;;

; UNIX: "/path1:/path2"
;include_path=".:/php/includes"
;
; Windows: "\path1;\path2"
;include_path=".;c:\php\includes"
```

مثال:

```
1);، "r"،$fp=fopen ("./data.txt"
```

fclose

عندما تنتهي من التعامل مع الملف، تحتاج إلى إغلاقه لكي يتم حفظ التعديلات عليه، إذا تم إحباط سكريبتك لأي سبب أو أن السكريبت انتهي عمله فإن الـPHP يقوم بإغلاق جميع الملفات تلقائياً.

تقوم الدالة ()fclose بإغلاق الملف عندما تريد إغلاقه وهي تحتاج إلى معامل واحد فقط وهو مقبض الملف الذي تريد إغلاقه.

مثال:

```
Fclose ($fp) ;
```

openkandil.com

قراءة وكتابة الملفات

لقد تعرفنا الآن كيفية فتح وإغلاق الملف، لنقم الآن بالتعرف علي كيفية قراءة و كتابة البيانات من الملف،

Fread

تقوم هذه الدالة بقراءة واستخراج البيانات الموجودة في الملفات ووضعها بمتغير وهي تأخذ معاملين المعامل الأول هو مقبض الملف والعدد الثاني هو عدد الحروف المراد قراءتها..

مثال:

```
"r");،$fp=fopen("data.txt"
10);،$data=fread($fp
```

وخذ باعتبارك نقطتين وهما:

١- إذا مثلاً قرأت عشر حروف من الملف وكان في الملف عشرين حرف وقمت بطلب الدالة fread مرة أخرى فسيتم قراءة العشر أحرف الثانية..

٢- إذا كان في الملف أقل من عشر أحرف فسيتم قراءة الموجود.

Fwrite

تقوم هذه الدالة بالكتابة إلى الملف وتحتاج إلى عاملين وهي مقبض الملف والقيمة المراد كتابتها إلى الملف، فعلى افتراض أنك قد فتحت الملف والمقبض هو $fp فإننا نكتب الكلمة PHP إلى الملف بالطريقة التالية:

```
"PHP");،Fwrite ($fp
```

وهناك معامل ثالث لهذه الدالة يحدد كم حرفاً سنقوم بكتابته من القيمة الحرفية الموجودة في المعامل الثاني فلو كتبنا مثلاً

```
1);، "PHP"،Fwrite ($fp
```

فسوف يتم كتابه أول حرف فقط...

obeikandl.com

قراءة وكتابة الحروف في الملفات

Fgetc

تستخدم هذه الدالة لقراءة حرف واحد من الملف في كل مرة، وهي تستخدم معاملاً واحداً وهو مقبض الملف

وتقوم بإرجاع حرف واحد من الملف أو (False) عند الوصول إلى نهاية الملف..

Feof

تقوم هذه الدالة بخدمتنا في هدف بسيط وشيء ممتاز وهي معرفة إذا ما كنا قد وصلنا إلى نهاية الملف عند قراءته وتقوم بإرجاع (true) عند الوصول إلى نهاية الملف أو حصول خطأ ما، وهي تأخذ معاملاً واحداً وهو مقبض الملف.

فقد تكون مثلاً تريد أن تتأكد أن المؤشر لم يصل إلى نهاية الملف بعد استخدامك لأحد الدوال التي تقوم بنقل المؤشر من مكان إلى آخر، عند ذلك ستكون هذه الدالة مفيدة لتخبرك إذا ما وصلت إلى نهاية الملف أو لا...

تطبيق عملي:

قم بإنشاء ملف سمه file.txt واكتب فيه أكثر من سطر ثم قم بإنشاء ملف PHP وسمه بأي اسم وضع فيه الشيفرة التالية ثم اختبره، لكي ترى عمل الدالتين:

```
<?
"r");،$fp= fopen("file.txt"
While (!feof($fp))
{
 $char=fgetc($fp);
    echo $char;
} ?>
```

obeikandl.com

Fgets

إذا استخدمنا الدالة fgetc لقراءة الملفات الطويلة فإنها ستأخذ وقتاً وعمراً حتى تتم قراءتها، يقوم الـPHP بتوفير دالة fgets لتساعدنا في قراءة عدد محدد من البايتات وهي تأخذ معاملين، المعامل الأول هو مقبض الملف والمعامل الثاني هو عدد الحروف المراد قراءتها +١، فإذا مثلاً أردت قراءة ملف يتكون من خمس حروف فسيكون المعامل الثاني للدالة هو الرقم ٦ وتتوقف الدالة عند حدوث أحد من ثلاث حالات

الأول: هو إذا تم قراءة عدد البايتات المحددة.

الثاني: إذا تم الوصول إلى نهاية سطر في الملف.

الثالث: إذا وصلت إلى نهاية الملف.

مثال:

```
 "r");،$fd = fopen ("/tmp/inputfile.txt"
while (!feof ($fd)) {
 4096);،    $buffer = fgets($fd
   echo $buffer;
}
fclose ($fd);
```

Fputs

تقوم بنفس وظيفة الداله fwrite وتأخذ نفس معاملاتها ونفس طريقتها..

القراءة داخل الملفات

File

تحتاج هذه الدالة إلى معامل واحد هو مسار الملف ولا تحتاج إلى مقبض، وعملها هو قراءة ما بداخل الملف وتخزينه سطراً سطراً في مصفوفة حيث إن هذه المصفوفة تقوم بأخذ كل سطر في الملف كأنه عنصر لوحده وتظل السطور سطوراً (أي أن المصفوفة تحتفظ بالمعامل للسطر الجديد (\n) بداخلها)، هذه الدالة لا تحتاج إلى

obeikandl.com

مقبض للملف بل تحتاج إلى مسار الملف فقط، وهي تقوم بفتح وقراءة وإغلاق الملف تلقائياً...

وكغيرها من الدوال فإنها تستطيع قراءة صفحات الإنترنت الخارجية..

مع ذلك يستحسن أن لا تقوم باستعمال هذه الدالة لقراءة الملفات الطويلة لأنها تقوم باستخدام قدر كبير من الذاكرة المحجوزة للـ PHP وقد تستخدمها كلها...

مثال:

```
<?
$fcontents = file ('file.txt');
 $line) = each ($fcontents)) {،while (list ($line_num
echo "<b>Line $line_num:</b>  $line  <br>\n";
}
?>
```

Fpassthru

تقوم هذه الدالة بقراءة محتويات الملف بداية من النقطة التي توقف منها المؤشر الوهمي عند أي عملية قراءة أخرى، وتقوم بالتوقف عند نهاية الملف وتقوم بإغلاق الملف من تلقاء نفسها لذلك لا داعي لإغلاق الملف بواسطة الدالة fclose بعد استخدامك لهذه الدالة، وتقوم الدالة بقراءة المحتويات وطباعتها بشكل قياسي، وهي تحتاج إلى معامل واحد فقط وهو مقبض الملف...

مثال:

```
<?
"r");،$fp=fopen("file.txt"
fpassthru($fp)
?>
```

obeikandi.com

Readfile

تقوم هذه الدالة بقراءة جميع محتويات الملف ولا تحتاج إلى مقبض بل إلى مسار الملف فقط وتقوم بقراءة كامل محتويات الملف ثم طباعتها بشكل قياسي وتقوم بإرجاع عدد البايتات التي تم قراءتها أو (false) عند حدوث خطأ ما

```
<?
Readfile ("file.txt");
?>
```

الوصول العشوائي إلى الملفات

أخبرناك سابقاً بأن هناك طريقة تجعلك تتحكم في التحكم بالمؤشر الوهمي للملف والوصول إلى أي مكان في الملف أو عند أي حرف تريده، بالدوال السابقة كنا عندما نصل إلى حرف معين مثلاً بدالة من الدوال فإننا نقوم بإغلاق الملف ثم نعاود فتحه كي نكمل القراءة من عند الحرف الذي تم الوصول إليه ولكن هذه الطريقة غير عملية نهائياً.....

يوصل لنا الـPHP بعض الدوال التي تمكننا من الوصول إلى الملف بالمكان الذي نريده ومن هذه الدوال:

Fseek

تحتاج هذه الدالة إلى عاملين، العامل الأول هو مقبض الملف $fp والعامل الثاني هو عبارة عن رقم صحيح يسمونه كمصطلح بالـ(offset) أي المكان الذي سيتوقف فيه المؤشر، سيقوم الـPHP بالتحرك في الملف إلى أن يصل إلى المكان الذي تم تحديده.. أي إنه إذا كان في الملف سطر واحد مكون من عشرة حروف وقمنا بجعل الـoffset خمسة، سيقوم الـPHP بالتحرك حتى يصل إلى نهاية الحرف الخامس...

وهناك معامل ثالث اختياري لهذه الدالة ويسمونه كمصطلح بالـ (whence) وله إحدى ثلاث خيارات:

obeikandi.com

Seek_set ويقوم بقراءة الملف من بدايته حتى يصل إلى المكان المطلوب بالـoffset

Seek_cur يقوم بالقراءة من المكان الحالي حتى يصل إلى المكان المطلوب بالـoffset

Seek_End يقوم بالقراءة من نهاية الملف حتى يصل إلى المكان المحدد بالoffset

تعتبر هذه الدالة نادرة في عملها (أو كما يسميها المبرمجون شاذة) بسبب أنها تقوم بإرجاع القيمة (٠) عند نجاحها والقيمة (-١) عند حصول خطأ ما..

مثال:

قم بفتح ملف واكتب فيه ثمان حروف متتالية ثم قم بحفظه باسم file.txt ثم قم بوضعه مع ملف PHP فيه الشيفرة التالية، ثم بعد ذلك شغل ملف الـPHP وانتظر النتيجة:

```
 <?
$fp = fopen("file.txt");
SEEK_SET);،4،fseek($fp
fpassthru($fp);
?>
```

Ftell

هذه الدالة من الدوال المفيدة فهي تقوم بإرجاع مكان الـoffset (أو المؤشر الوهمي) في الملف وتحتاج إلى معامل واحد وهو مقبض الملف...

```
<?
$fp = fopen ("file.txt");
$p = ftell($fp);
echo $p;
?>
```

obeikandl.com

Rewind

تقوم بإرجاع المؤشر إلى بداية الملف...

```
<?
$fp = fopen ("file.txt");
rewind($fp)
?>
```

جلب معلومات الملف

يوفر الـPHP دوال تساعدنا في معرفه حجم الملف وما إذا كان الملف موجوداً أم لا من هذه الدوال:

File_exists

تقوم هذه الدالة بالقيام بالتأكد ما إذا كان الملف موجوداً أم لا وهي تحتاج على معامل واحد وهو مسار الملف، وتقوم بإرجاع true (١) إذا كان الملف موجوداً و false إذا كان الملف غير موجود.

```
<?
$Th=File_exists("file.txt");
echo $Th ;
?>
```

Filesize

تقوم هذه الدالة بإرجاع حجم الملف بالبايتات أو false عند حصول خطأ...

obeikandl.com

دوال الملفات المتعلقة بالوقت:

هذه الدوال تقوم بإرجاع معلومات مفيدة عن وقت التغيير الذي طرأ على الملف أو آخر مره تمت قراءته وهي على حسب نظام التشغيل فإذا كان نظام السيرفير هو يونكس أو لينوكس ستقوم الدوال بإرجاع الوقت بنظام (timestamp) وهو الوقت مترجم إلى عدد الثواني منذ صدور يونكس ومولده على العالم، بينما تقوم بإرجاع وقت التعديل على نظام الويندوز مباشرة...

يقوم الـPHP بتزويدنا بدالتين لمعرفة الوقت:

Filectime وتقوم بإرجاع آخر وقت تم فيه التغيير على الملف على شكل timestamp ويشمل هذا آخر وقت تم فيه إنشاء الملف أو الكتابة إليه أو تغيير تراخيصه...

Filemtime تقوم بإرجاع آخر وقت تم فيه التعديل على الملف على شكل timestamp ويشمل هذا إنشاء الملف أو تغيير محتوياته...

تقوم الدالة getdate بعمل مفيد وهو تحويل الوقت من timestamp إلى الوقت العادي

الملكية والتراخيص

على أنظمة تشغيل اليونكس مثل يونكس ترتبط الملفات مع مستخدم خاص أو مجموعة من المستخدمين (group) وتحتوي على علامات وتراخيص تقوم بتوضيح من له صلاحية على استخدامها..

يمكننا أن نلخص التراخيص كالتالي:

١ / ممتلك الملف (owner)، بشكل افتراضي، وهو المستخدم الذي تم استخدام حسابه في استخدام الملف.

obeikandi.com

٢ / مجموعة من المستخدمين (group)، بشكل افتراضي، المجموعة التي يكون ضمنها مالك الملف.

٣ / جميع المستخدمين (all) كل شخص له حساب على النظام.

المستخدمين والمجموعات في أنظمة اليونكس يتم تعرفهم عن طريق رقم (ID) مثلما يتم تعريفهم عبر أسمائهم، إذا كنت تريد معرفة معلومات شخص عن طريق رقمه، يمكنك استخدام هذه الدالة:

Posix_getpwind

التي ستقوم بإعطائنا مصفوفة تحتوي على المعلومات التالية:

Name	اسم المستخدم الذي يدخل به في حسابه.
passwd	كلمة السر المشفرة للمستخدم.
uid	رقم الحساب للمستخدم.
gid	رقم حساب المجموعة التي فيها المستخدم.
gecos	اسم المستخدم الكامل، رقم هاتف ومعلومات إضافية.
dir	المجلد الرئيسي للمستخدم.
shell	المسار الرئيسي لحساب المستخدم.

Posix_getgrgid

تقوم هذه الدالة بإرجاع مصفوفة عن معلومات المجموعة، وهي تحتاج إلى معامل واحد فقط وهو رقم الـID للمجموعة...

وسوف تحتوي على العناصر التالية:

Name	اسم المجموعة
Gid	رقم المجموعة
members	عدد أعضاء المجموعة

obeikandi.com

وهناك أيضاً خمس دوال تساعدنا في معرفة معلومات أكثر عن الملفات وتحتاج فقط إلى مسار الملفات.

Fileowner

تقوم بإرجاع رقم المعرف (ID) لمالك الملف...

Filegroup

تقوم بإرجاع رقم المعرف (ID) لرقم المجموعة التي يعتبر مالك الملف ضمنهم..

Filetype

تقوم بإرجاع رقم نوع الملف وقد تعود بإحدى هذه القيم (file، dir، char، fifo، block، link) والذي يهمنا منهم هو file وdir...

Is_dir

وتقوم بإرجاع True إذا كانت قيمة المسار هو مجلد..

Is_file

وتقوم بإرجاع True إذا كانت قيمة المسار هي ملف..

الحصول على اسم الملف من وسط مسار الملف..

Basename()

مثال:

```
<?
$path = "/home/httpd/html/index.php3";
$file = basename ($path);
echo '$file <br>';
".php3"); ،$file = basename ($path
echo '$file <br>';
?>
```

هذه الدالة مفيدة جداً للحصول على الملف من وسط مسار مجلد..

obeikandi.com

نسخ، اعادة تسمية وحذف الملفات

تسمح لك الـPHP بنسخ، وإعادة تسمية، وحذف والدوال التي تستخدم لتنفيذ هذه العمليات هي

Copy ()

تقوم بأخذ قيمتين حرفتين وتشير إلى مصدر الملف الرئيسي الذي يوجد فيه الملف والمصدر الهدف الذي سيتم نسخ الـPHP إليه...

```
<?
 $file.'.bak')) {،if (!copy($file
print ("failed to copy $file...<br>\n");
}
?>
```

Rename

نستطيع الآن استخدام هذه الدالة لإعادة تسمية الملف وهي تحتاج إلى قيمتين حرفتين وهي مصدر الملف أو مكانه واسمه الرئيسي ثم الاسم الجديد الذي تريد إعادة التسمية به..

مثال:

```
<?
'newfile.txt');،Rename ('file.txt'
?>
```

Unlink()

تحتاج إلى قيمه حرفية واحدة وهي مسار الملف الذي تريد حذفه.

```
<?
unlink ('file.txt');
?>
```

obeikandl.com

العمل مع المجلدات

مثلما تعاملنا مع الملفات في الـPHP فإننا نتعامل مع المجلدات، فهناك دوال للمجلدات تتطلب مقبض المجلد، وهناك دوال تحتاج فقط إلى القيمة الحرفية فقط وبدلاً من الإطالة دعنا نقوم بالدخول في الموضوع مباشرة.

Opendir تقوم بفتح المجلد وإعطائنا مقبض المجلد.

Closedir() تقوم بإغلاق المجلد المفتوح وتحتاج فقط إلى مقبض المجلد...

Readdir تقوم بقراءة المدخل الحالي للمجلد...

Rewindir تقوم بإرجاع المدخل إلى الصفر..

Chdir للانتقال إلى مجلد آخر، وتتطلب المسار للمجلد الذي تريد الانتقال إليه..

Rmdir تقوم بمسح مجلد، ولكن يجب أن يكون المجلد خالياً من أي ملفات أو مجلدات، وتتطلب مسار المجلد الذي تريد مسحه..

Mkdir تقوم بإنشاء مجلد جديد وتتطلب أن يكون هذا المجلد غير موجود مسبقاً وتحتاج إلى قيمتين وهما اسم المجلد الجديد مع مساره، والترخيص المطلوب له..

Dirname تقوم بإعطائنا اسم المجلد الحالي الذي فيه الملف، وتحتاج إلى مسار الملف..

تطبيق عملي:

أنشئ مجلد اسمه tmp في مجلد الـ htdocs وضع فيه ملفات، ثم أنشئ ملف اسمه test.php في مجلد الـhtdocs واكتب الشيفرة التالية ثم شغله:

obeikandi.com

```
<?php
if ($dir = @opendir("/tmp")) {
while($file = readdir($dir)) {
echo "$file\n";
}
closedir($dir);
}

?>
```

Dir()

عبارة عن كائن يحتوي على ثلاث وظائف.. ونقوم بإعطائه مسار المجلد الذي نريده أن يتعامل معه ثم بعد ذلك نقوم بوضع قيمته في متغير يقوم بوراثة صفاته.

خصائص الكائن:

handle تقوم بإعطائك مقبض المجلد..

Path تقوم بإعطائك المسار للمجلد..

Read تقوم بإعطائنا المجلدات اعتماداً على المؤشر الحالي للمجلد..

Rewind تقوم بإرجاع مؤشر المجلد من الصفر.. تقريباً نفس عملية rewinddir..

Closedir تقوم بإغلاق المجلد..

دعنا نأخذ فكرة عن طرف التخزين في البداية وكيف كانت على الإنترنت في السابق....

كان من أكثر طرق التخزين انتشاراً في السابق على الإنترنت وربما هو لا يزال يستخدم في بعض المواقع والمنتديات يعتمد على الملفات.... فكان صاحب الموقع الذي

obeikandl.com

لديه هذه الطريقة في تخزين البيانات خوف وتعب من فقدانها مثلاً وكان الشبح الذي يكدر عليه صفوة نجاح موقعه هو عمل نسخ احتياطية لهذه الملفات لكي يتمكن من استرجعها في حال فقدانها فكانت هذه العملية تأخذ وقتاً وجهداً ومالاً.... كما كان من عيوب تخزين البيانات في الجداول ضغط الخادم أو (server) في حال الاستعلام عن معلومة معينة والبحث عنها كما أنه يستهلك الكثير من ذاكرة هذا الخادم في عملية بحث معينة فهو يحجز مساحة ليست بالهينة في عملية بحث عن اسم مثلاً أو ما شابهها.

ربما يتردد عند البعض ذالك السؤال وهو....ما هي قواعد البيانات بالضبط ؟

قواعد البيانات ببساطة جمع المعطيات أو المدخلات.
كل قاعدة بيانات ربما تتكون من جدول (Table) واحد أو عدة جداول هذه الجداول تحتوي علي أعمدة وصفوف تهيكل البيانات وترتبها.... سوف أجعل لك مهمة اكتشاف فوائد قواعد البيانات في آخر الفصل.
لترى الجدول الذي بالأسفل كمثال:

"Table "Coustomers#

Id	Fname	Lname
025	عبد الواهب	صالح
044	محمد	خالد

كما تلاحظ الجدول قسَّم البيانات إلى صفوف...مع كل إضافة عميل جديد سوف يكون هناك صف (سجل) جديد... ربما لو تطلق لخيالك العنان سوف تلاحظ أن هذا الجدول مشابه للدولاب والصفوف رفوف فإذا أردت أن تضيف كتب أو ملابس

obeikandl.com

أو أي كان سوف تضيفها في رف جديد.. كما يحصل في إضافة عميل جديد سوف تضيفه في صف (سجل) جديد.

البيانات في كل صف قسمت إلى مدى أبعد في الخلايا (أو الحقول)، كل من هذه البيانات تحتوي على قيمة محددة وصفة محددة. على سبيل المثال محمد خالد سوف ترى أن هذا العميل انقسمت بياناته في الحقل إلى id والاسم الأول والاسم الأخير.

الصفوف في الجدول ليس لها ترتيب معين.. يمكن أن يكون الترتيب أبجدياً ويمكن أن يكون باسم العضو أو باسمه الأخير أو بأي معيار آخر يمكن أن تحدده مسبقاً لترتيب الصفوف ولهذا يكون من الضروري تحديد طريقة ليسهل عليك تحديد صف (سجل) معين....في المثال السابق نستطيع إخراج السجل من بين باقي السجلات بـ id وهو رقم العميل الذي هو عدد فريد لا يتكرر في أي صف (سجل) أخر وسبب استنادي في استخراج السجل علي id لأنه ربما يكون هناك عميلان لهما نفس الاسم.... وهذا ليس شرط أن يكون للجدول مفتاح فريد لكن هنا حددناه لكي يتم استخراج السجلات المطلوبة بسهولة وبسرعة أكبر.

العلاقات

الكثير من قواعد البيانات اليوم هي نظم إدارة قواعد بيانات علائقية (relational database management systems) تختصر في RDBMS، قواعد البيانات العلائقية هذه عبارة عن مجموعة من الجداول أو نموذج من الجداول النموذجية المتعددة التي تحتوي على معلومات مترابطة. ربما تسمع أيضاً الكثير عن SQL وهي اختصار ل(Structured Query Language) وهي تسمح لك أن توحد هذه المعلومات من الجداول المترابطة وبذالك تسمح لك بإنشاء وتحليل العلاقات الجديدة.

openkandil.com

المثال السابق للعملاء كان عبارة عن جدول واحد فقط، ولذلك لن تحتاج إلى ربط بينه وبين جدول آخر لأنه لا يجدي.

لكن إذا كان هناك أكثر من جدول وكانت هذه الجداول مترابطة مع بعضها البعض في البيانات سوف تلاحظ أنك بحاجة إلى نظم إدارة البيانات العلائقية (RDBMS).... فلنرى هذه المثال لكي تتضح الصورة أكثر:

#Table "Coustomers"

Id	Fname	Lname
025	عبد الواهب	صالح
044	محمد	خالد
022	حمد	طارق

#Table "Address"

Id	Tel	Street	City	Country
044	018522	شارع الأهرام	القاهرة	مصر
022	01225505	مساكن برزة	دمشق	سوريا
025	0122505	طريق الأربعين	الكويت	الكويت

#Table "Account"

Id	accountb
044	10.0000
025	20.0000
022	20.000

obeikandl.com

كل من هذه الجداول الثلاثة كيان مستقل.... لكن تلاحظ أنهم مرتبطين مع بعضهم البعض ب(id)، على سبيل المثال بإمكاننا أن نعرف رصيد العميل عبد الواهب صالح من id، كما يمكننا معرفة اين يسكن حمد طارق وكم رقم الهاتف وأيضاً يمكننا أن نعرف من هو صاحب الرصيد ٢٠,٠٠٠ أيضاً كم واحد من مدينة القاهرة والكثير الكثير..... ربما اتضحت لك أهمية العلاقات.

إذا عرفنا أن العلاقات هي الأساس الجوهري لنظم قاعدة البيانات العلائقية، بجعلها مرنة وسهلة بحث تتمكن من ربط السجلات المختلفة مع بعضها البعض في الجداول.

المفتاح الأجنبي

سوف تلاحظ أن حقل (id) الذي يظهر في الجداول في الثلاثة والذي جعل من الممكن ربط الجداول المختلفة معاً أنه مفتاح أجنبي لأنه بالأصل مفتاح فريد (primary key) في جدول (COUSTOMERS)...
ليس ضرورياً أن يكون هناك مفتاح أجنبي في كل جدول ولكن يتم إضافة على حسب حاجتك فإذا كنت تريد ربط بيانات الجداول مع بعضها فسوف تحتاج إليها.

في كل جدول يوجد به المفتاح الأجنبي سوف يكون له مرجعية للجدول الأصل فمثلاً هنا المرجعية ستكون جدول (customers).... بمعنى أن المفتاح الأجنبي سوف يقوم بربط البيانات ما بين الجدول الأصل وبين الجدول الذي يتواجد به كمفتاح أجنبي....من هنا يتضح لنا مفهوم الاستقامة المرجعية وهذا مفهوم أساسي ومهم عندما تصمم قاعدة بيانات بأكثر من جدول سوف يكون للمفتاح الأجنبي قيمة ثابتة في جميع الجداول بمعنى لو كان قيمة المفتاح الأجنبي في جدوله الأصل عدد صحيح فسوف يكون بنفس القيمة في جميع الجداول.... ونقطة أخرى إذا حدث تحديث أو

obeikandl.com

تغيير أو حذف لأحد القيم في المفتاح الأجنبي فسوف تتم في جميع الجداول...هذا هو مفهوم الاستقامة المرجعية.

كثير من قواعد البيانات اليوم يتم تعديل القيم بها تلقائياً كميكروسوفت أكسس وبعض قواعد البيانات الأخرى، لكن هناك بعض قواعد البيانات التي تحتاج إلى تعديل يدوي على كل قيمة يتم التعديل عليها... وهذا لاشك أنه متعب!!

الفهرسة

لو كان لديك جدول به الكثير من السجلات، يمكنك أن تستعلم بسرعة كبيرة عن أياً من هذه السجلات بواسطة "فهرسة" كل السجلات. هذا المفهوم تقريباً شبيه جداً بالفهرس الذي يوجد نهاية كل كتاب...كما يسهل عليك هذا الفهرس الموجود في الكتاب في سرعة البحث عن المواضيع التي يتضمنها الكتب، نفس الكلام ينطبق على فهرسة السجلات في الجدول....دعنا نرى مثالاً لتتضح الصورة:

```
= 220; SELECT * FROM names WHERE ID
```

سوف يقوم هذا الاستعلام في البحث في جميع المعلومات وإرجاع قيمتها بشرط أن يكون رقم السجل (الفهرس) ٠٢٢

هنا سهلت علينا المهمة كثيراً وذلك لأن السجلات مرتبة بأرقام فلكل سجل رقم فريد يميزه عن الآخر وبهذه الحالة سوف يقوم هذا الاستعلام السابق بجلب جميع بيانات العميل "حمد طارق "....

تدريب عملي

سوف نبدأ بأخذ فكرة عن البرنامج الذي سوف أشرح عنه وهو عبارة عن دليل مواقع بسيط جداً يعتمد علي قواعد البيانات.

بالطبع لا يوجد فيه تقسيمات والسبب لأننا سوف نضطر إلى وضع أو خلق أكثر من جدول (Table) في قواعد البيانات وسوف نضطر إلى وضع علاقات بينهم.

وهذا حالياً يعقد الموضوع فسأكتفي الآن بجدول واحد (one table) وسوف أشرح إذا اتسع لدي الوقت الكثير عن (sql) عامة وعن العلاقات (Relationstips) لأنها مثيرة أيضاً جداً.

نبدأ بالخطوة الأولى وهي قاعدة البيانات الخاصة بدليل الموقع الذي لدينا، فالبيانات والمعلومات التي غالباً ما يهتم بها دليل الموقع هي عنوان الموقع واسم الموقع والبريد الخاص بالموقع.

هذا هو الكود الخاص **بإنشاء قاعدة البيانات الخاصة بدليل الموقع**.

```
CREATE TABLE dalal (
، id int not null auto_increment
،sitename varchar(100)
،add_date date
،email varchar(100)
،site_url varchar(100)
 ،description text
primary key (id));
```

هنا سوف يخزن في قواعد البيانات المعلومات التالية:

id وهي المرشح الأساسي والفريد الذي يُتعرف على الجدول بواسطته.

sitename وهو اسم الموقع وهو من نوع char

add_date تاريخ إضافة الموقع وهو من نوع date

email البريد الإلكتروني وهو من نوع char وطوله نفس طول اسم الموقع (١٠٠)

site_url وهو عنوان الموقع أيضاً من نوع char

سوف أكتفي بشرح id لأنه ربما يكون جديد على البعض وسوف أشرح عمله وليس المهم أن تفهم عمله في هذه الخطوة إنما سوف تفهمها بالتفصيل في الخطوات القادمة بعد قليل إن شاء الله.

id هو عبارة عن المرشح الأساسي لهذا الجدول (primary key)

openkandil.com

وتلخص في آخر سطر من الجدول وجود :

```
primary key (id)
```

ربما ترجمة المصطلحات العلمية وخاصةً في الكمبيوتر تضر أكثر مما تنفع وهذا سبب وجيه لي في كتابتي السكريبت مهما كان بلغة إنجليزية ومن ثم تعريبها.

المهم نحن الآن نريد أن نعرف ماذا يعني id سوف أعطي مثالاً لكي يتضح فلو أردنا مرشحاً أساسياً أو كوداً فريداً نميز به هذا الجدول فلن نستطيع أن نضع هذا الكود الفريد لاسم الموقع لأنه ربما يكون لدي موقع بنفس الاسم والاختلاف بينهما في الوصلة وربما يكون هناك موقع بعنوانين مثل سوالف وياهوو إذا اسم الموقع لا يمكن أن نضعه كوداً أساسياً وأيضاً نفس الكلام ينطبق على عنوان الموقع وأيضاً على التاريخ وحتى البريد لا نستطيع أن نضع البريد هو المرشح الوحيد والأساسي للجدول ربما يأتي هذا لكن منطقياً لا يجوز.

ربما ترى id من نوع عدد صحيح (int) لكن ربما يصعب عليك الذي بجانبه وهو (not null) وهي تعني بعدم السماح لهذا الحقل أن يكون بدون قيمة.

not null تعني بكل وضوح ربما يسمح بقيمة فارغة ربما تفكر في الصفر فهذا غير صحيح.

وترى أيضاً هذه العبارة بجانب not null وهي (auto_increment) وهي تعني بكل وضوح الإضافة التلقائية أو الأتوماتيكية ودائماً ما يبدأ بالقيمة (١)

وهذا يعني مع كل إضافة موقع سوف تزيد قيمة (id) وهي تبدأ من (١)

config.php

وهو عبارة عن ملف التوجيه الذي يحتوي معلومات الموقع وهي المعلومات التالية :

```
<?
//this moudel config
$dbname = "links";
$dbuname = "root";
$dbpws = "";
$dbhost = "localhost";
?>
```

obeikandi.com

لا يحتاج إلى شرح لأنه واضح فالذي جرب أن يركب أي برنامج php يعتمد على قواعد البيانات فالأكيد سيعدل في ملف مشابه له أو نفسه ومع ذلك سنشرحه:

الآن نبدأ في **add.php** وهذا هو:

```
<html>
<head>
<style type="text/css">
verdana; font-size:10pt}،body {font-family:verdana
verdana; font-size:10pt}،TD {font-family:verdana
#header {color:white; font-weight:bold; font-
verdana}،family:verdana
A {color:navy; text-decoration:underline}
A:hover {color:red}
A:visit {color:navy}
</style><title>Site Insertion</title></head>
<?
if ($action == 'addsite')
{
include("config.php");
 $dbpw)؛ $dbuname،    mysql_pconnect($dbhost
@mysql_select_db("$dbname") or die ("Unable to select database"
);
،'$email'،''،'$name'،    mysql_query("insert into dalal values ('
'$des')");،'$siteurl'
mysql_close();
echo "Thank you for add site ";
}
?>
<body bgcolor=white>
<form method="POST" action="add.php?action=addsite">
<p>SiteName <input type="text" name="name" size="40" style="
border-style: double; border-color: #000080"></p>
<p>Email       <input type=
"text" name="email" size="40" style="border-
style: double; border-color: #000080"></p>
<p>SiteUrl    <input type="text" name="siteurl
```

obeikandi.com

```
" size="40" style="border-style: double; border-
color: #000080" value="http://"></p>
<p>description<textarea rows="5" name="des" cols="33" style="
border-style: double; border-color: #000080"></textarea></p>
<p align="center"><input type="submit" value="Submit" name="
B1" style="border-style: double; border-
color: #000080"><input type="reset" value="Reset" name="B2" s
tyle="border-style: double; border-color: #000080"></p>
</form>
<br>
<a href="show.php"><b>View</b></a>
<br>
</body>
</html>
```

من رؤيتك له تلاحظ أنه عبارة عن html و php لكن كود php أكثر

بالنسبة html فهي بسيطة بإمكانك فهمها وهي عبارة عن حقول تطلب من المستخدم إدخال معلومات الموقع سوف أبدأ في شرح كود php فتلاحظ بدايته ونهايته بهذين الوسمين <؟ و ؟>

وبعدها يأتي شرط التحقق وهو:

```
if ($action == 'addsite')
```

فهذا يعني التحقق من إضافة جديدة على الحقول الخاصة بعملومات الموقع وتلاحظ أن التحقق من هذا الشرط بهذه القيمة ('addsite') بين وسمين اقتباس وحيد ربما يكون هناك بعض التساؤل قليلاً عليه لكن لو تلاحظ في كود html في form وضعت الانتقال بعد إضافة المعلومات إلى (add.php?addsite) هنا

```
<form method="POST" action="add.php?action=addsite">
<p>SiteName <input type="text" name="name" size="40" style="
border-style: double; border-color: #000080"></p>
<p>Email       <input type=
"text" name="email" size="40" style="border-
style: double; border-color: #000080"></p>
<p>SiteUrl    <input type="text" name="siteurl
```

obeikandl.com

```
" size="40" style="border-style: double; border-
color: #000080" value="http://"></p>
<p>description<textarea rows="5" name="des" cols="33" style="
border-style: double; border-color: #000080"></textarea></p>
<p align="center"><input type="submit" value="Submit" name="
B1" style="border-style: double; border-
color: #000080"><input type="reset" value="Reset" name="B2" s
tyle="border-style: double; border-color: #000080"></p>
</form>
```

هذا هو كود الفورم الخاص بإدخال معلومات الموقع فإذا لاحظته في بدايته هنا

```
<form method="POST" action="add.php?action=addsite">
```

وضعت الإرسال بالـ (post) وقمت بوضع قيمة (action) الصفحة (add.php?addsite)

هذا يعني أن قيمة (action) في حال ضغط submit ستكون ما هو مكتوب بعد علامة الاستفهام (؟) بعد اسم الصفحة وهي (add.php?addsite)

أتمنى أنك فهمت المقصود.

بعد ذلك إذا تحقق الشرط ووجد من أرسل سوف ينفذ الأوامر والأولى منها.

```
include("config.php");
```

وأمر include يعني استجلاب هذا الملف وقيم المتغيرات الموجودة به

بعد هذا الأمر طلبت الاتصال بقواعد البيانات لكي يتم تجهيز الجدول للإضافة

وهو هنا:

```
$dbpw); ، $dbuname، mysql_pconnect($dbhost
@mysql_select_db("$dbname") or die ("Unable to select database"
);
```

الدالة mysql_pconnect تطلب الاتصال بخادم قاعدة البيانات على أساس المعلومات الموجودة بها

وهنا نحن استعنا بعد الله بالمتغيرات الموجودة في ملف config.php

وربما هذا يوضح لنا أكثر فائدة الدالة include

Orekandi.com

ربما لاحظت في طلب الاتصال في الدالة mysql_pconccet في السطر الثاني بعدها هذا

```
@mysql_select_db("$dbname") or die ("Unable to select database");
```

هنا في الدالة mysql_select_db وضعنا قيمة المتغير dbname وهو اسم قاعدة البيانات

فهذه الدالة تقوم بتحديد اسم قاعدة البيانات المراد الاتصال بها وهذا ربما يوضح استخدام علامة @

لكي يعمل كل شيء بنفس الوقت

بعد هذا نلاحظ السطر التالي:

```
mysql_query("insert into dalal values ('','$name','$email','$siteurl','$des')");
```

قمنا في هذا السطر بتنفيذ الدالة mysql_query وهذه الدالة تقوم باختصار تنفيذ استعلامات (sql) وفي هذا السطر استخدمنا أمر الاستعلام المراد تنفيذه بواسطة الدالة mysql_query هو (Insert)

هذا الأمر أو الدالة في sql تقوم بإضافة البيانات في قاعدة البيانات والصيغة أو القاعدة الثابتة له هي:

```
INSERT INTO tablename VALUES (var and values)
```

INSERT INTO

هذا ثابت لدينا بعده يأتي اسم الجدول الذي نريد إضافة البيانات بداخله

وبعد ذلك تأتي كلمة VALUES (var) وتفتح قوس وتيحط المتغيرات والقيم المراد إضافتها بأقواس اقتباس (" ") وتنهيه بفاصلة منقوطة ;

تلاحظ حين استخدمت mysql_query و insert لم أضع القيم بل وضعت المتغيرات التي تحمل القيم المراد تخزينة في جدول dalal ربما هذا يسهل علينا الكثير

بعد ذلك تأتي الدالة mysql_colse وهي تعني طلب إغلاق الاتصال بخادم قاعدة البيانات

obeikandl.com

بهذا انتهينا من ملف add.php و config.php

هنا سوف نعرف كيف نستخرج البيانات الموجودة داخل قاعدة البيانات

وهي في غاية السهولة

بدايةً سوف أضع كود show.php كاملاً واشرحه مثل سابقه:

```
<html>
<head>
<style type="text/css">
verdana; font-size:10pt}،body {font-family:verdana
verdana; font-size:10pt}،TD {font-family:verdana
#header {color:black; font-weight:bold; font-
verdana}،family:verdana
A {color:navy; text-decoration:underline}
A:hover {color:red}
A:visit {color:navy}
</style>
</head><titel><center><b>view The sites</b></center><br><
center><a href='add.php'>Add Your Site</a></center> <hr color
="#000080"></titel>

<body>
<?php
include("config.php");
 $dbpw);، $dbuname،  mysql_pconnect($dbhost
@mysql_select_db("$dbname") or die ("Unable to select database"
);

$query="select * from dalal ";

$result=mysql_query($query);

mysql_close();

/*Display Results*/
```

obeikandl.com

```
$num=mysql_numrows($result);

$i=0;
while ($i < $num) {


"sitename");،$i،    $sitename=mysql_result($result
"add_date");،$i،    $add_date=mysql_result($result
"email");،$i،    $email=mysql_result($result
"site_url");،$i،    $siteurl=mysql_result($result
"description");،$i،    $description=mysql_result($result
"id");    ،$i،    $id=mysql_result($result
?>
<table border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="100%"><b>SiteName:</b><a href='<?echo $siteurl;
?>'><i><? echo $sitename;?></i></a>
<p><b>Date Add:</b>  <i> <?echo $add_date;?></i></p>
<p><b>description:</b><?echo$description;?></p>
<p><b>URL:</b><i><?echo$siteurl;?></i></p>
<p><b>Email:</b><i> <a href="mailto:<?echo$email;?>"><?ec
ho $email;?></a></i> </p>
</td>
</tr>
</table>
<a href='edit.php?id=<?echo $id;?>'><b><b>[Edit This Site]</b
></b></a>
<hr color="#000080">
<?
++$i;
}

?>
</body>
</html>
```

obeikandl.com

كما تلاحظ تقريباً شبيه بالملف add.php في عملية الاتصال لكن الجديد لدينا هنا

السطر الذي يلي عملية الاتصال وهو:

```
$query="select * from dalal ";
```

في هذا السطر قمنا بتعريف متغير يدعى query وقمنا بوضع قيمة هذا المتغير عملية الاستعلام عن المعلومات الموجودة في الجدول dalal

أظن هذا واضح لكن دعني أشرح عمل select فهي تقوم بتحديد الحقول بداخل الجداول أو الجدول وهي دائماً تستخدم في إخراج البيانات من الجداول فتلاحظ وضعنا قيمة الحقل أو الحقول في استعلامنا هذا علامة الضرب (*) وهي تعني تحديد جميع الحقول الموجودة وبعدها استعملنا from وهذا يعني من ودائماً يأتي بعده أسماء الجدول أو اسم الجدول الذي قمت بتحديد الحقول بداخله.

ببساطة واختصاراً لهذا السطر select * from dalal

حدد جميع الحقول من جدول الدليل.

هنا سوف يحدد جميع الحقول الموجودة في جدول الدليل ويقوم بتخزينها في المتغير query

بسيطة.

لا تنس أن تضعها بين أقواس اقتباس مزدوجة (" ")

بعد ذلك يأتي هذا السطر

```
$result=mysql_query($query);
```

قمنا هنا أيضاً بتعريف المتغير result وقمنا بتخزين قيمة الدالة mysql_qurey وقيمة المتغير الذي استعلمت به الدالة mysql_query في المتغير result

بوضوح هنا قمنا بالاستعانة بالدالة mysql_query ونسبنا إليها قيمة المتغير query الموجود به مسبقاً قيمة الاستعلام مسبقاً في السطر الذي يسبق هذا السطر وقمنا بهذا لكي يتم فرز جميع القيم الموجودة في الجدول dalal بواسطة الدالة mysql_query

openkandil.com

بعد ذلك قمنا بإغلاق قاعدة البيانات بالدالة mysql_colse

وبعد ذلك قمنا بتنويه بسيط وتعليق يوضح أننا سوف نعرض ونطبع القيم الموجودة في الجدول في هذا السطر.

/❖Display Results❖/

بعد هذا السطر قمنا بتعريف متغير يدعى num وأضفنا القيمة إليه باستخدام الدالة mysql_numrows مسندين إليه قيمة المتغير result

الدالة mysql_numrows تقوم باسترجاع ومعالجة نتيجتنا المخزنة في المتغير result وتقوم بتخزينها في المتغير num على شكل مصفوفة

بعد ذلك قمنا بتعريف عداد يدعى i ووضعنا قيمته صفر لكي نستخدمه مع أمر التكرار while وقمنا بوضع الشرط في أمر التكرار while

$i < $num

هذا يعني إذا كان قيمة العداد i أصغر من قيمة المتغير num المتوجد بداخله جميع المعلومات فاستمر بطباعة التالي.

وهي هذه السطور.

```
$sitename=mysql_result($result،$i،"sitename");
$add_date=mysql_result($result،$i،"add_date");
$email=mysql_result($result،$i،"email");
$siteurl=mysql_result($result،$i،"site_url");
$description=mysql_result($result،$i،"description");
$id=mysql_result($result،$i،"id");
?>
<table border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="100%"><b>SiteName:</b><a href='<?echo $siteurl;?>'>
<i><? echo $sitename;?></i></a>
<p><b>Date Add:</b>  <i> <?echo $add_date;?></i></p>
<p><b>description:</b><?echo$description;?></p>
<p><b>URL:</b><i><?echo$siteurl;?></i></p>
<p><b>Email:</b><i> <a href="mailto:<?echo$email;?>"><?echo $e
mail;?></a></i> </p>
</td>
</tr>
```

obeikanal.com

```
</table>
<a href='edit.php?id=<?echo $id;?>'><b><b>[Edit This Site]</b></b
></a>
<hr color="#000080">
<?
++$i;
}
```

هنا قمنا بالكثير في أمر التكرار WHILE فسوف يطبع جميع القيم الموجودة في الجدول كما رمزت إليها في هذا السطر

```
"sitename"); ،$i ،$sitename=mysql_result($result
```

قمنا في هذا السطر والسطور الخمسة التي تليه بتعريف العديد من المتغيرات وقمنا بتخزين قيم المعلومات بداخله باستخدام الدالة mysql_result مسندين إليه المتغيرات result و i و الحقل الذي نريد إخراج القيمة منه واستخدمنا في هذا السطر الحقل sitename

كما استخدمنا كود html لكي ننسق ونرتب شكل ظهور المعلومات وفي آخر سطر قبل أن ننهي عملية التكرار while بالقوس { جعلنا قيمة العداد i تزيد بمقدار واحد مع كل لفة أو بمعنى طباعة قيم من قاعدة البيانات لكي يتم توقفها أرجو أن تنتبه لهذه الأمور الصغيرة لأنها من مشاكل البرمجة المشهورة.

هنا انتهينا من ملف show.php يتبقى لنا ملف edit.php وهو الملف الخاص بتعديل الموقع فربما يخطئ أحدهم في إدخال اسم أو عنوان موقعه فهنا يمكنك تعديلها حتى بعد تخزينها في قاعدة البيانات.

هذا آخر ملف سوف أشرحه اليوم من ضمن مشروع دليل المواقع وربما أشرح عملية الحذف والبحث في وقت آخر.

obeikandi.com

سوف أضع كود **edit.php** كاملاً كسابقيه

```
<html>
<head>
<style type="text/css">
body {font-family:verdana،verdana; font-size:10pt}
TD {font-family:verdana،verdana; font-size:10pt}
TH {font-size:10pt; color:white; font-weight:bold; font-
family:verdana،verdana}
A {color:navy; text-decoration:underline}
A:hover {color:red}
A:visit {color:navy}
#mySelect {font-family:verdana،arial; font-size:9pt}
</style>
</head>
<titel><center><b>Edit Site</b></center><br><center><a href='add
.php'><b>Add Your Site </b></a> |
<a href='show.php'><b>View Edit database</b></center></a> <hr c
olor="#000080"></titel>
<body>

<table border=0 cellpadding=4 cellspacing=0 width=600>
<? echo "<form method='post' action='$PHP_SELF'>"; ?>
<table border="1" width="100%" bordercolor="#000080" cellspacing="
1" cellpadding>
<tr>
<td width="100%">
<p align="center">Edit The Info </td>
<?
include("config.php");
mysql_pconnect($dbhost، $dbuname، $dbpw);
@mysql_select_db("$dbname") or die ("Unable to select database");

$query1 = "select * from dalal where id = '$id'";

$result1 = mysql_db_query('links'،$query1);

while($row = mysql_fetch_object($result1)) {
echo "<td align=middle>SiteNmae<br><input type=text name='name'
value='$row->sitename'></td>";
echo "<td align=middle>Add Date<br><input type=text name='add_da
```

obeikandl.com

```
te' value='$row->add_date'></td>";
echo "<td align=middle>Email<br><input type=text name='email' value
='$row->email'></td>";
echo "<td align=middle>SiteUrl<br><input type=text name='siteurl' val
ue='$row->site_url'></td></tr>";
echo "<td align=middle>Description<br><input type=text name='desc'
value='$row->description'></td></tr>";
$id = $row->id;
}
echo "</table><p>";
echo "<br><input type=hidden name='id' value='$id'>";
?>

<input type=submit name="update" value="Edit The Info">
</form>
</tr>
</table>
<?
if ($update) {
if($name == "" || $add_date == "" || $email == "" || $siteurl == "" || $
desc == "") {
die("<b>You left one or more fields blank.</b>");
}

$query2 = "update dalal set sitename='$name'، add_date='$add_date'،
email='$email'، site_url='$siteurl'، description='$desc'
where id='$id'";
mysql_db_query('links'،$query2);
echo "<b>Your record has been updated</b></p>";
}
mysql_close();
?>
</body>
</html>
```

هنا سوف يسهل علي شرح هذا الملف أكثر من سابقيه لأنه يكرر ما يفعله إخوانه لكن مع بعض التجديدات التي لا تذكر فالجديد هنا وفي هذا السطر

```
$query1 = "select * from dalal where id = '$id'";
```

هنا تلاحظ عرفنا متغير يدعى query1 وقمنا بتخزين ناتج الاستعلام بداخله فهذا ليس بجديد الجديد ما يوجد في الاستعلام فقد استخدمنا where وهي تعني

openkanal.com

بالترجمة إلى العربية "بحيث" وتعني في البرمجة وخاصةً مع أوامر sql "بشرط" أي هنا اشترطنا في استعلامنا أن تكون قيمة المتغير id مساوية لنفس القيمة الموجودة في الجدول وهي id وربما هنا يسعني أن أوضح أكثر ما هو id ولماذا عرفته وجعلته primary key

هنا إذا ضغط أحد الذين أضافوا مواقعهم بعد أن أخطأ في إضافة موقعه على edit فسوف يعتمد على الرقم المخزن في id وهو كما سبق وأن عرفتها في قاعدة البيانات عدد صحيح ويزداد تلقائياً فلو كان لدينا مثلاً في قاعدة البيانات أربعة مواقع فلن يجري التعرف على الأربعة مواقع ومعلوماتهم إلا عن طريق id لأن لكل موقع منهم رقم، بمعنى الأول منهم ١ والأخير ٤ أرجو أنك فهمته وأرجو أن تحاول تشغيل السكريبت لكي تفهمه أكثر وأكثر.

بعد ذلك يأتي هذا السطر

```
$query1);،$result1 = mysql_db_query('links'
```

هنا أيضاً عرفنا المتغير result وخزنا بداخله قيمة استعلام الدالة mysql_db_query وأسندنا إليها اسم قاعدة البيانات والمتغير query

الدالة mysql_db_query وهي مشابة في عملها مع mysql_query ما عدا الفرق بينهم أنك تستطيع تحديد اسم قاعدة البيانات الذي تريد الاستعلام عنه واستعملتها هنا لكي يتم إعطاؤنا النتائج بوضوح أكثر.

بعد هذا السطر يأتي أمر التكرار while مرة أخرى في هذا الملف

```
while($row = mysql_fetch_object($result1))
```

هنا قمنا بخلق المتغير row ولا تنس في php ليس من الضروري تعريف المتغيرات قبل استخدامها وقمنا بمساواته (وهو الشرط الخاص بأمر التكرار while) بالدالة mysql_fetch_object مسندين إليه قيمة المتغير result1

الدالة mysql_fetch_object وظيفته أو من ضمن الوظائف التي تقوم بها أنها تقوم بانتزاع وعرض نتائج الاستعلام.

obeikandl.com

فإذا تحقق شرط أمر التكرار while فستنفذ هذا الكود الطويل قليلاً.

```
while($row = mysql_fetch_object($result1)) {
echo "<td align=middle>SiteNmae<br><input type=text name='name'
value='$row->sitename'></td>";
echo "<td align=middle>Add Date<br><input type=text name='add_da
te' value='$row->add_date'></td>";
echo "<td align=middle>Email<br><input type=text name='email' value
='$row->email'></td>";
echo "<td align=middle>SiteUrl<br><input type=text name='siteurl' val
ue='$row->site_url'></td></tr>";
echo "<td align=middle>Description<br><input type=text name='desc'
value='$row->description'></td></tr>";
$id = $row->id;
}
echo "</table><p>";
echo "<br><input type=hidden name='id' value='$id'>";
?>

<input type=submit name="update" value="Edit The Info">
</form>
</tr>
</table>
<?
if ($update) {
if($name == "" || $add_date == "" || $email == "" || $siteurl == "" || $
desc == "") {
die("<b>You left one or more fields blank.</b>");
}

add_date='$add_date'، $query2 = "update dalal set sitename='$name'
description='$desc' ، site_url='$siteurl'، email='$email'،
where id='$id'";
$query2)؛ mysql_db_query('links'
echo "<b>Your record has been updated</b></p>";
}
mysql_close();
?>
```

هنا قمنا بطباعة المعلومات والتحقق بواسطة شرط IF دعنا نأخذ عملية طباعة المعلومات:

Obeikandl.com

```
echo "<td align=middle>SiteNmae<br><input type=text name='name'
value='$row->sitename'></td>";
echo "<td align=middle>Add Date<br><input type=text name='add_da
te' value='$row->add_date'></td>";
echo "<td align=middle>Email<br><input type=text name='email' value
='$row->email'></td>";
echo "<td align=middle>SiteUrl<br><input type=text name='siteurl' val
ue='$row->site_url'></td></tr>";
echo "<td align=middle>Description<br><input type=text name='desc'
value='$row->description'></td></tr>";
$id = $row->id;
```

هنا في أول سطر قام بطباعة كود HTML وبداخله أيضاً بعض الأشياء من PHP لا تخف فالمقصود هنا أنه قام بتخزين جميع ما يطبعه بداخل المتغير الخاص بلغةHTML فهنا مثلاً

```
echo "<td align=middle>SiteNmae<br><input type=text
name='name' value='$row->sitename'></td>";
```

قام بطباعة كود HTML وهو عبارة عن حقل الإدخال (INPUT) من نوع (TEXT) وقام بترميزه إلى الحقل الموجود بقاعدة البيانات من المتغير row وتخزينه في قيمة المتغير في name) html)

والأكثر وضوحاً ما قمنا به في السطر السابق قمنا بوضع المتغير row ثم يليه اسم الحقل في الجدول وهنا اسم الحقل sitename.

أظن مع تشغيلك للسكريبت سوف تتضح أكثر وأكثر وهذا الذي استخدمته مع الجميع

بعد ذلك قمت بطباعة أكواد html وهي عبارة عن فورم للإدخال ويأتي بعده شرط if وهو هنا

```
<input type=submit name="update" value="Edit The Info">
</form>
</tr>
</table>
<?
if ($update) {
if($name == "" || $add_date == "" || $email == "" || $siteurl == "" || $
desc == "") {
```

```
die("<b>You left one or more fields blank.</b>");
}

add_date='$add_date'، $query2 = "update dalal set sitename='$name'
description='$desc' ، site_url='$siteurl'، email='$email'،
where id='$id'";
$query2)؛، mysql_db_query('links'
echo "<b>Your record has been updated</b></p>";
}
mysql_close();
?>
```

في الشرط نريد التأكد هل تم تعديل أحد البيانات أم لا وهو شبيه بالشرط الذي استخدمته مع ملف add.php بعده إذا كان هناك إدخال فسيقوم بعدة أوامر وأولها شرط آخر if يتأكد هل هناك من الحقول التي جعلت فارغة أم لا وهي هنا:

```
if($name == "" || $add_date == "" || $email == "" || $siteurl == "" || $
desc == "") {
die("<b>You left one or more fields blank.</b>");
}
```

إذا كان هناك حقول فارغة سوف يطبع له You left one or more fields blank أي أنك تركت حقل أو أكثر فارغ.

بعد ذلك يأتي هذا:

```
،$query2 = "update dalal set sitename='$name'
، site_url='$siteurl'، email='$email'، add_date='$add_date'
description='$desc'
where id='$id'";
$query2)؛، mysql_db_query('links'
echo "<b>Your record has been updated</b></p>";
}
mysql_close();
```

قمنا هنا في أول سطر بتعريف المتغير query2 بعد ذلك خزنا بداخله قيمة أمر SQL التحديث update .

أمر update يقوم بتحديث المعلومات في قواعد البيانات من تجديد وحذف.

openkanal.com

ويكتب هكذا:

Update tablename Set (values

الكلمات التي تبدأ بحروف كبيرة هي الثوابت بمعنى لا يتم تغييرها والتي تبدأ بحروف صغيرة مثل sitename و values فهي التي يتم تغييرها كما وضحت في الكود السابق فقد قمت بمساوات المتغيرات name، add_date، email، site_url، description بما يقابلها من الحقول في الجدول dalal ووضعت شرط where أن يكون المتغير id=id$

وفي السطر الذي يليه هذا قمت باستخدام الدالة mysql_db_query مسنداً إليه اسم قاعدة البيانات والمتغير الذي خزن قيمة التحديث وهو query2 لكي يتم تحديث قاعدة البيانات
وفي السطر الذي يليهما استعملت الدالة mysql_colse لكي يتم إغلاق الاتصال بقاعدة البيانات
وهو آخر سطر اليوم

السكريبت يتكون من ثلاثة ملفات، الأول view.php لعرض المدخلات في سجل الزوار، والثاني add.php لإضافة مدخل جديد إلى سجل الزوار، والثالث هو config.php ويحتوي على بيانات قاعدة البيانات.
في الملف view.php، يتم جلب البيانات المدخلة من قاعدة البيانات وعرضها واحدة تلو الأخرى في صورة جدول HTML، وأسفل هذا الجدول يوجد نموذج لإضافة تعليق جديد في سجل الزوار.
الملف add.php يقوم بأخذ البيانات المرسلة من النموذج الموجود في الملف view.php، ثم يقوم بمراجعة هذه البيانات والتحقق منها، ثم إضافتها إلى قاعدة البيانات وإعادة المستخدم إلى الملف view.php.

obeikandl.com

يجب أن نحدد الآن البيانات التي نريد تخزينها في قاعدة البيانات:

- الاسم.
- البريد الإلكتروني.
- الصفحة الشخصية.
- التعليق على الموقع.

❖ **قاعدة البيانات**

من خلال المعلومات السابقة نلاحظ بأننا سنحتاج إلى جدول واحد فقط في قاعدة البيانات، وليكن اسمه guestbook، هذه هي الشيفرة التي يجب تنفيذها للحصول على البنية الأساسية للجدول:

```
CREATE TABLE guestbook (
،   id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY
،   name CHAR(100)
،   email CHAR(128)
،   homepage CHAR(200)
،   date DATETIME
،   ip CHAR(15)
message TEXT
);
```

شيفرة SQL السابقة تعرف جدولاً اسمه guestbook يحتوي على الحقول التالية:

id - هذا الحقل يخزن رقماً تسلسلياً يزداد بمقدار واحد مع كل حقل جديد يضاف إلى الجدول، ومواصفاته:

رقم INTEGER

بلا إشارة (موجب دائماً) UNSIGNED

إجباري (لا يمكن أن يترك خالياً) NOT NULL

يتم تحديثه تلقائياً إلى رقم أكبر من السابق بواحد AUTO_INCREMENT

وهو المفتاح الأساسي للوصول إلى بيانات الجدول PRIMARY KEY

obeikandl.com

name - هـذا الحقـل يخـزن اسـم الـشخص الموقـع في سـجل الـزوار، ومواصـفاته كالتالي:
يخزن نصاً بطول ١٠٠ حرف كحد أقصى CHAR (100)

email - لتخزين البريد الإلكتروني، ١٢٨ حرفاً كحد أقصى (هذا الرقم قياسي تقريباً في أغلب البرامج القائمة على الويب لتخزين عناوين البريد الإلكترونية).

homepage - لتخزين عنوان الصفحة الشخصية للموقع، ٢٠٠ حرفاً بحد أقصى -- قيمة معقولة.
date - لتخزين تاريخ المدخل في سجل الزوار، وهو يخزن قيماً من النوع:
تاريخ ووقت DATETIME

message - لتخزين نص الرسالة (تعليقك على الموقع)، وهو حقل يتسع لتخزين عدد كبير جداً من الحروف TEXT

❖ الملف config.php

يحتوي على الملف على شيفرة برمجية تقوم بإعداد المتغيرات اللازمة للوصول إلى قاعدة البيانات، وسنقوم بإضافة شيفرة الاتصال بقاعدة البيانات إلى هذا الملف بدلاً من تكرارها في كل من الملفين view.php و add.php.
أولاً سنقوم بإعداد مجموعة من المتغيرات:

```
/* store the hostname of the MySQL server */
$dbhost = "localhost";
/* store the username to login to MySQL */
$dbuser = "root";
/* store the password to login to MySQL */
$dbpass = "";
/* store the name of the MySQL database */
$dbname = "";
```

obeikandl.com

كما تلاحظ، قمنا بتعريف أربعة متغيرات مختلفة هي:

$dbhost -- وتقوم بتخزين عنوان مزيد قاعدة البيانات.

$dbuser -- وتقوم بتخزين اسم المستخدم الذي سيستخدم للدخول إلى قاعدة البيانات.

$dbpass -- وتقوم بتخزين كلمة المرور التي ستستخدم مع اسم المستخدم للدخول إلى قاعدة البيانات.

$dbname -- تخزن اسم قاعدة البيانات التي ستحوي الجدول guestbook.

يمكن للمستخدم الآن تعديل معلومات هذه المتغيرة عن طريق فتح الملف في المفكرة مثلاً وتعديل القيم الموجودة بين علامات الاقتباس مقابل أسماء المتغيرات.

بعد تعريف المتغيرات السابقة، سنقوم كما اتفقنا بوضع الشيفرة اللازمة للاتصال بقاعدة البيانات، وهي عبارة عن استدعاء لدالتين:

```
/* connect to the MySQL server */
$dbpass);،$dbuser،mysql_connect($dbhost
/* set $dbname as the database to be used */
mysql_select_db($dbname);
```

❖ **شرح الدوال**

```
password)، username،mysql_connect(hostname
```

تقوم الدالة السابقة بالاتصال بمزود قاعدة البيانات الموجود على العنوان hostname، وتحاول الدخول إلى المزود قاعدة البيانات عن طريق إرسال اسم المستخدم username و كلمة المرور password، في حال نجاح العملية تعيد مورداً (resource) نستطيع استخدامه مع دوال أخرى لتنفيذ أوامر قاعدة البيانات على هذا الاتصال المفتوح (يمكن فتح أكثر من اتصال بأكثر من قاعدة بيانات في نفس الوقت).

openkandil.com

جميع المتغيرات الممررة إلى الدالة اختبارية، إذا لم تدخل password فإن الدالة ستحاول إجراء الاتصال دون تمرير كلمة مرور، وإذا لم تدخل username فستحاول الدالة إجراء الاتصال دون اسم مستخدم، وإذا لم تحدد hostname فإن الدالة ستحاول الاتصال بقاعدة البيانات على العنوان المحلي (locahost).

إذا كنت قد ركبت MySQL مباشرة دون تغيير الإعدادات فإن مزود قواعد البيانات سيسمح مباشرة بالاتصال من الجهاز المحلي باسم المستخدم root والذي لن تكون له كلمة مرور افتراضية، وبالتالي يمكنك الاتصال مباشرة إلى مزود MySQL المحلي عن طريق القيم الموضحة في المتغيرات التي في الأعلى.

mysql_select_db(database_name، link_identifier)

تقوم هذه الدالة بتحديد قاعدة البيانات التي سيتم العمل عليها، حيث أن مزود قاعدة البيانات يمكن أن يحتوي على أكثر من قاعدة بيانات واحدة.

تحدد اسم قاعدة البيانات عن طريق المتغيرة database_name، أما المتغيرة الثانية فهي الاتصال الذي تريد أن تحدد قاعدة البيانات له، فإذا كنت قد خزنت القيمة التي أعادتها الدالة mysql_connect في متغيرة، وأردت الآن تعيين قاعدة البيانات الذي يجب استخدامها في ذلك الاتصال، فيمكنك تمرير المتغيرة إلى الدالة mysql_select_db في الوسيطة الثانية، ولكن هذه الوسيطة الثانية اختيارية، فإذا لم تقم بتمريرها فإن الدالة ستعيد قاعدة البيانات التي يجب العمل عليها لآخر اتصال تم فتحه بقاعدة البيانات، وإذا لم يكن هناك اتصال مفتوح من قبل ستحاول الدالة فتح اتصال بقاعدة البيانات كما لو كنت قد شغلت mysql_connect() بدون تحديد الـ hostname و الـ username والـ password وثم تعيين قاعدة البيانات التي يجب العمل عليها لهذا الاتصال الجديد.

في ملفنا المثال قمنا بتنفيذ هذه الدالة مباشرة بعد فتح اتصال بقاعدة البيانات باستخدام الدالة mysql_connect، وبالتالي عينا قاعدة البيانات $dbname كقاعدة البيانات التي يجب استخدامها مع آخر اتصال فتح بقاعدة البيانات.

هذه الدالة تغير true (صح) في حالة نجاحها و false (خطأ) في حالة فشلها.

يمكننا الآن النظر إلى نسخة نهائية كاملة من الملف config.php:

```
<?php
/* store the hostname of the MySQL server */
$dbhost = "localhost";
/* store the username to login to MySQL */
$dbuser = "root";
/* store the password to login to MySQL */
$dbpass = "";
/* store the name of the MySQL database */
$dbname = "";
/* connect to the MySQL server */
$dbpass); ،$dbuser ،mysql_connect($dbhost
/* set $dbname as the database to be used */
mysql_select_db($dbname);
?>
```

❖ الملف view.php:

في البداية المقدمات العادية ورأس المستند بلغة HTML:

```
<html dir="rtl">
<head>
</title>سجل الزوار<title>
<style>
textarea {،select ،input ،td ،body
font-family: Tahoma;
font-size: x-small;
}
h6 {،h5 ،h4 ،h3 ،h2 ،h1
font-family: Arial;
}
</style>
</head>
<body>
</h1>سجل الزوار<h1>
```

بعد أن تتم طباعة رأس المستند بلغة HTML نأتي إلى الجزء التالي وهو جلب البيانات من قاعدة البيانات وعرضها للمستخدم.

openkandil.com

في البداية نحتاج إلى اتصال بقاعدة البيانات، وكما قلنا من قبل، الملف config.php يحتوي على الشيفرة الكاملة لتجهيز اتصال بقاعدة البيانات، ما علينا القيام به الآن هو إعطاء أمر بتنفيذ البيانات الموجودة في الملف config.php، وهذا الأمر هو include، كالتالي:

```
/* get a database connection */
include 'config.php';
```

النتيجة التي يعطيها الأمر include السابق هي نفسها النتيجة التي كنا سنحصل عليها لو كنا وضعنا الشيفرة الموجودة في الملف view.php مباشرة، بمعنى أننا بدلاً من أن ننسخ شيفرة قاعدة البيانات الموجودة في الملف config.php يدوياً ونضعها في أعلى الملف view.php للحصول على وصول إلى قواعد البيانات، فإننا نطلب من مترجم PHP أن يقوم بالعملية آلياً، حيث نطلب منه تنفيذ الشيفرة الموجودة في الملف config.php كما لو كانت جزءاً من الملف الذي استدعى الأمر (view.php).

❖ الأمر include:

يستخدم الأمر include كالتالي:

```
include filename;
```

أي الكلمة include وبعدها اسم الملف الذي نريد إدراجه، وفي مثالنا كان الملف هو config.php.

❖ طلب البيانات من قاعدة البيانات

المفترض الآن بأن جميع الإضافات التي تمت إلى سجل الزوار مخزنة في الجدول guestbook في قاعدة البيانات، وأن لدينا اتصالاً الآن بقاعدة البيانات من خلال الملف config.php، نستطيع البدأ بعمل استعلام لقاعدة البيانات نطلب فيه البيانات التي نريدها، ويكون ذلك بالشيفرة التالية:

```
$res = mysql_query("SELECT
،homepage،email،name
،UNIX_TIMESTAMP(date) AS date
```


obeikandi.com

```
message
FROM guestbook
ORDER BY date DESC");
```

الدالة mysql_query:

```
mysql_query(query، link_identifier);
```

تقوم الدالة بإرسال الاستعلام (query) إلى قاعدة البيانات عبر الاتصال الذي يمرر إليها (link_identifier)، وإذا لم يكن هنالك اتصال ممرر، فإن الدالة تستخدم آخر اتصال فتح بقاعدة البيانات، وإذا لم يكن هنالك اتصال قد فتح من قبل ستحاول الدالة إجراء اتصال بقاعدة البيانات عن طريق تنفيذ الدالة mysql_connect وتمرر إليها قيم hostname و username و password فارغة.

في مثالنا السابق، قمنا بتمرير قيمة واحدة فقط إلى الدالة هي الاستعلام الطويل المحاط بعلامتي الاقتباس " و "، وستقوم الدالة باستخدام آخر اتصال فتح بقاعدة البيانات وهو الاتصال الذي فتح في الملف config.php الذي أدرجناه من قبل في ملفنا (view.php).

نأتي الآن إلى الاستعلام، وهو عبارة عن استعلام بسيط من النوع SELECT كتب بلغة SQL. وسأقدم شرحاً سريعاً هنا.

SELECT: يعني اختيار أو جلب.

```
message، UNIX_TIMESTAMP(date) AS date، homepage، email، name
```

هذه هي أسماء الحقول التي طلبناها (جلبناها) من الجدول، وهي عبارة عن قائمة بسيطة من أسماء الحقول، ما عدا التالي:

obeikandl.com

```
UNIX_TIMESTAMP(date) AS date
```

تقوم الدالة UNIX_TIMESTAMP بإعادة التاريخ المخزن في قاعدة البيانات في صورة TIMESTAMP الخاص بنظام unix، وهو عبارة عن نظام للتاريخ يحسب عدد الثواني منذ منتصف الليل في ١٩٧٠/١/١، وبالتالي فإن هذه القيمة تتغير في كل ثانية، تعتبر الـ Timestamp الوحدة القياسية لتخزين المعلومات عن التاريخ والوقت في PHP وأغلب لغات برمجة ونظم Unix، وبالتالي فإننا طلبنا من قاعدة البيانات أن تحول لنا قيمة التاريخ (date) إلى الـ Timestamp المقابل حتى يسهل التعامل معها في PHP وعرضها بالطريقة التي نريدها..

بعد ذلك أضفنا المقطع AS date، هذا المقطع يعني بأن الحقل هذا يجب أن يسمى بالاسم date، السبب في إضافتنا لهذا المقطع هي أن الحقول تسمى عادة بأسماء التعبيرات التي تتكون منها، فهذا الحقل مثلاً:

```
UNIX_TIMESTAMP(date)
```

سيكون اسمه هو:

```
UNIX_TIMESTAMP(date)
```

وهذا سيجعل التعامل معه في PHP صعباً نوعاً ما، لذا قلنا لقاعدة البيانات أن تسمي البيانات هذه بالاسم date بدلاً من الاسم السابق.

بعد الانتهاء من قائمة الحقول نجد المقطع التالي:

```
FROM guestbook
```

FROM

الكلمة FROM تعني (من) ونحدد بعدها اسم الجدول الذي نريد طلب (جلب) البيانات منه، وهو guestbook في مثالنا السابق.

```
ORDER BY date DESC
```

المقطع السابق يحدد الطريقة التي ترتب بها البيانات..

obeikandl.com

ORDER BY

تعني (ترتيب حسب) وهي تحدد الحقل الذي نريد الترتيب تبعاً له، وفي مثال الحقل date، أي أننا نريد ترتيب المدخلات حسب تاريخ إضافتها، ولكن استخدام ORDER BY وبعده اسم الحقل يعني فرز البيانات حسب ذلك الحقل تصاعدياً، أي من الأقل إلى الأكبر، ولكننا في الواقع نريد فرزها في برنامجنا تنازلياً (من الأحدث أو الأعلى تاريخاً إلى الأقدم) ولهذا السبب أضفنا المقطع DESC وهو اختصار لكلمة descending أي تنازلياً.

انتهى

obeikandl.com