تعلم لغة

الفهرس

مقدمه للغة PHP 1
- تشغيل Windows IIS 5.0 3
- إضافة PHP الى IIS 6
- إضافة MySQL الى IIS 15
- بنية ملفات PHP 21
- بروتوكولات الأنترنت 24
- التعليـــــقات 27
- المتغيـــــرات 28
- الأرقام 32
- العمليات الحسابية 32
- متغيرات النظام 34
- الثوابت 34
- معرفة وتحويل أنواع البيانات 35
- دوال الوقت والتاريخ 37
- النماذج 38
  - GET 39
  - POST 40
- الأوامر الشرطية 58
  - العبارة IF 58
  - المعاملات المنطقية 60
  - تعدد الشروط 69
  - تداخل العبارات الشرطية 70
  - العبارة Switch 72
  - التخلص من وسوم الـ html 75
- التكرارات والمصفوفات 77
- دوال المصفوفات 85
- فرز المصفوفات 90
- دوال المصفوفات الاضافية 95
- مصفوفات متعدده الابعاد 98
- ترتيب الكود البرمجي 101
  - Function 101
  - Print 104
  - مدى المتغيرات 106
  - المتغيرات المستقره 108
  - أشتمال الملفات 110
- تتبع وتصيد ومنع الأخطاء 111
  - أنواع الأخطاء 111
  - الأخطاء المنطقية 114
  - تفادي الأخطاء 115
  - Regular Expressions 116
  - صناعة فئة الحروف 119
- التعامل مع العميل 126
- Cookies 131
- Session 134
- قراءة وكتابة معلومات في ملف txt 139

مقدمة للغة PHP

تتميز لغة PHP بالكثير من الخصائص التي جعلتها الخيار الأمثل لمبرمجي الويب في العالم :

السهولة

تعتبر لغة PHP من أسهل لغات البرمجة تعلما، فهي تريحك من جميع تعقيدات إدارة الذاكرة وتعقيدات معالجة النصوص الموجودة في C من جهة، والكثير من الضعف الموجود في بينية وتصميم لغة البرمجة Perl من جهة أخرى.

تمتلك لغة PHP بنية وقواعدا ثابته وواضحة جدا، معظم قواعد اللغة مأخوذة من كل من C و Java و Perl لصنع لغة برمجة عالية السهولة والسلاسة دون فقدان أي من القوة في اللغة، يفيدك ذلك إذا كنت تعلم أي شيء عن لغات البرمجة الأخرى مثل Visual Basic أو C أو Java حيث ستجد دائما بأنك تفهم مواد الدورة بسرعة، وستكتشف كيف تقوم PHP بتسهيل أصعب الأمور وإذلال العقبات التي تواجه المبرمج حتى يتفرغ تماما للإبداع فقط، كل ما تفكر به تستطيع تنفيذه بلغة PHP.

السرعة

لغة PHP من اللغات المعروفة بسرعتها العالية في تنفيذ البرامج، وخاصة في الإصدارة الرابعة من المترجم، حيث تمت كتابة مترجم PHP من الصفر ليعطي أداءا في منتهى الروعة، كما أن لغة PHP مصممة أصلا كنواة لمترجم، بحيث يمكن أن تضع هذه النواة في عدة قوالب أو أغلفة لتعمل مع التقنيات المختلفة، فيمكنك تشغيل مترجم PHP كبرنامج CGI مثلا، ولكن الأفضل هو إمكانية تركيب مترجم PHP على مزود IIS في صورة وحدة إضافية تضاف إلى المزود عن طريق دوال ISAPI، وتوجد نسخة أخرى منه تركب على مزود Apache أيضا في صورة وحدة خارجية، وتوجد أيضا نسخة مخصصة للدمج مع شفرة مزود Apache بحيث تصبح جزءا من برنامج Apache نفسه، وهي الطريقة الأكثر استخداما الآن في مزودات الويب التي تعمل على أنظمة UNIX وهي الطريقة التي تعطي أفضل أداء لمترجم PHP، حيث يصبح المترجم جزءا من المزود، وبالتالي فإنه سيكون محملا في الذاكرة بانتظار صفحات PHP ليقوم بترجمتها وعرضها للزوار مباشرة دون التأخير الإضافي الذي تتطلبه برامج Perl/CGI مثلا حيث يجب أن يتم تشغيل مترجم Perl مع كل زيارة للصفحة لترجمة الصفحة، ثم يتم إغلاق المترجم، ثم استدعاءه مجددا عند الزيارة الثانية وهكذا، وهذا يشكل فارقا كبيرا في المواقع ذات الضغط العالي بالذات، ويكون استخدام PHP حلا أفضل بكثير.

المزايا

يأتي مترجم PHP لوحده محملا بعدد هائل من الدوال الجاهزة الاستخدام في جميع المجالات، من دوال المعالجة الرياضية والحسابية إلى دوال الوصول إلى قواعد البيانات ومزودات FTP، توفر لك دوال PHP مثلا وصولا إلى مزودات البيانات MySQL و PostgreSQL و MS SQL و Oracle وغيرها من مزودات قواعد البيانات، وهنالك أيضا مجموعة من الدوال لمعالجة ملفات XML، ودوال أخرى لإرسال واستقبال الملفات عن بعد باستخدام بروتوكول FTP، وهنالك مجموعة من الدوال لمعالجة وإنتاج الصور ديناميكيا وملفات Flash ديناميكيا، ناهيك عن جميع الدوال الخاصة بمعالجة النصوص والمصفوفات.

التوافقية

كما قلنا سابقا، فعلى الرغم من أن هنالك الكثير من نسخ PHP التي يعمل كل منها في بيئة مختلفة، إلا أنها جميعا تشترك في النواة الأصلية التي تقوم بالمعالجة الحقيقة لملفات PHP لذا فإن جميع مترجمات PHP تتصرف بنفس الطريقة فيما يتعلق بتنفيذ السكريبتات، فإذا كان السكريبت الذي عملته يعمل على نظام Windows مع مزود IIS فيجب أن يعمل دون الحاجة لأية تغييرات عند نقله إلى مزود Apache، بالطبع تظل بعض الأمور البسيطة جدا التي يوفرها بعض المزودات دون غيرها، ولكن جميع البرامج التي كتبتها منذ أن بدأت تعلمي للغة إلى الآن تعمل على جميع المزودات دون الحاجة لأي تغييرات، إضافة إلى ذلك فإن التغييرات التي حدثت باللغة الأساسية من الإصدارة الثالثة إلى الرابعة قليلة جدا، وأغلب التغييرات كانت في البنية التحتية للمترجم.

الحماية

يوفر PHP الكثير من المزايا المتقدمة، ولكنه يوفر لك الطرق المناسبة لوضع الحدود على هذه المزايا، فيمكنك التحكم بعدد الإتصالات المسموحة بقاعدة البيانات مثلا، أو الحجم الأقصى للملفات التي يمكن إرسالها عبر المتصفح، أو السماح باستخدام بعض الميزات أو إلغاء استخدامها، كل هذا يتم عن طريق ملف إعدادات PHP والذي يتحكم به مدير الموقع.

قابلية التوسع

يمكنك توسعة مترجم PHP بسهولة وإضافة الميزات التي تريدها إليه بلغة C، وحيث أن الشفرة البرمجية للمترجم مفتوحة فإنك تستطيع تغيير ما تريده مباشرة لتحصل على النسخة التي تناسبك من المترجم، ويمكنك أيضا عمل الوحدات الإضافية التي تركب على المترجم لزيادة ميزاته والوظائف المبيته فيه، وفي قد قام فريق تطوير مترجم PHP مسبقا بعمل هذه المهمة وتحويل كمية ضخمة من المكتبات المكتوبة بلغة C إلى مكتبات مخصصة لتضاف إلى المترجم، ومنها حصلنا على جميع الميزات التي تحدثنا عنها مثل الوصول إلى قواعد البيانات ومعالجة ملفات XML.

تاريخ PHP

بدأت PHP كمكتبة من الدوال تضاف على لغة Perl لتسهل عمل برامج CGI بلغة Perl، وبعد أن تلقى Rasmus Lerdof بعض الاقتراحات بتحويلها إلى مترجم بسيط، قام بعمل ذلك المترجم وطرحه على الإنترنت وسماه PHP أو Personal Home Pages أي الصفحات الشخصية، فقد كان عبارة عن نسخة مصغرة من Perl مع بعض الميزات الإضافية للويب، ثم أضاف إليه دعما لنماذج HTML وسماه PHP2/FI، فقام مجموعة من المبرمجين بالعمل على مترجم PHP وأضافوا إليه واجهة تطبيقات برمجية API لتسهيل عملية توسعته فأصبح لدينا PHP 3، بعد فترة من الزمن قامت شركة Zend للتقنيات بعمل مترجمها الخاص للغة والذي سمي zend أيضا، وقد اتصف هذا المترجم بالسرعة العالية وقدراته المحسنة، وجمع مع مكتبات PHP الأخرى لتكوين نواة المترجم PHP، مترجم PHP الآن مقسم على قسمان: المترجم zend ويتم تطويره على مزودات CVS الموجودة في موقع zend والقسم الثاني يسمى PHP وهو عبارة عن المكتبات والدوال الأساسية التي تأتي مع البرنامج، يقوم مترجم zend بقراءة الملفات ومعالجتها والتعامل مع المتغيرات وتنفيذ البرنامج وتوفير واجهة تطوير للتطبيقات API لتوسعة اللغة، أما PHP فتحتوي الآن على مكتبات مكتوبة بلغة C ومتوافقة مع واجهة التطبيقات التي يوفرها مترجم zend، وبالتالي يعمل القسمان معا لتكوين مترجم PHP، وعندما تزور موقع PHP الرسمي الآن وتحصل على مترجم PHP جاهزا أو تحصل على الشفرة البرمجية الخاصة بك، فإنك تحصل على كل من مترجم zend ومكتبات PHP معا.

تطور PHP تطورا مفاجئا في الفترة الأخيرة، وتشير إحصائيا Net Craft إلى أن مترجم PHP هو أكثر وحدات مزود Apache انتشارات على الإنترنت، كما أن مترجم PHP مركب على حوالي مليوني مزود ويب على الإنترنت.

تشغيل نظام Windows IIS 5.0 plus أوWindows PWS 4.0 plus

متطلبات تحميل البرنامج

- قرص نظام Win2000 أو WinXP لتحميل.IIS 5.0 plus
- قرص نظام Win98 SE أو WinME لتحميل.PWS 4.0 plus

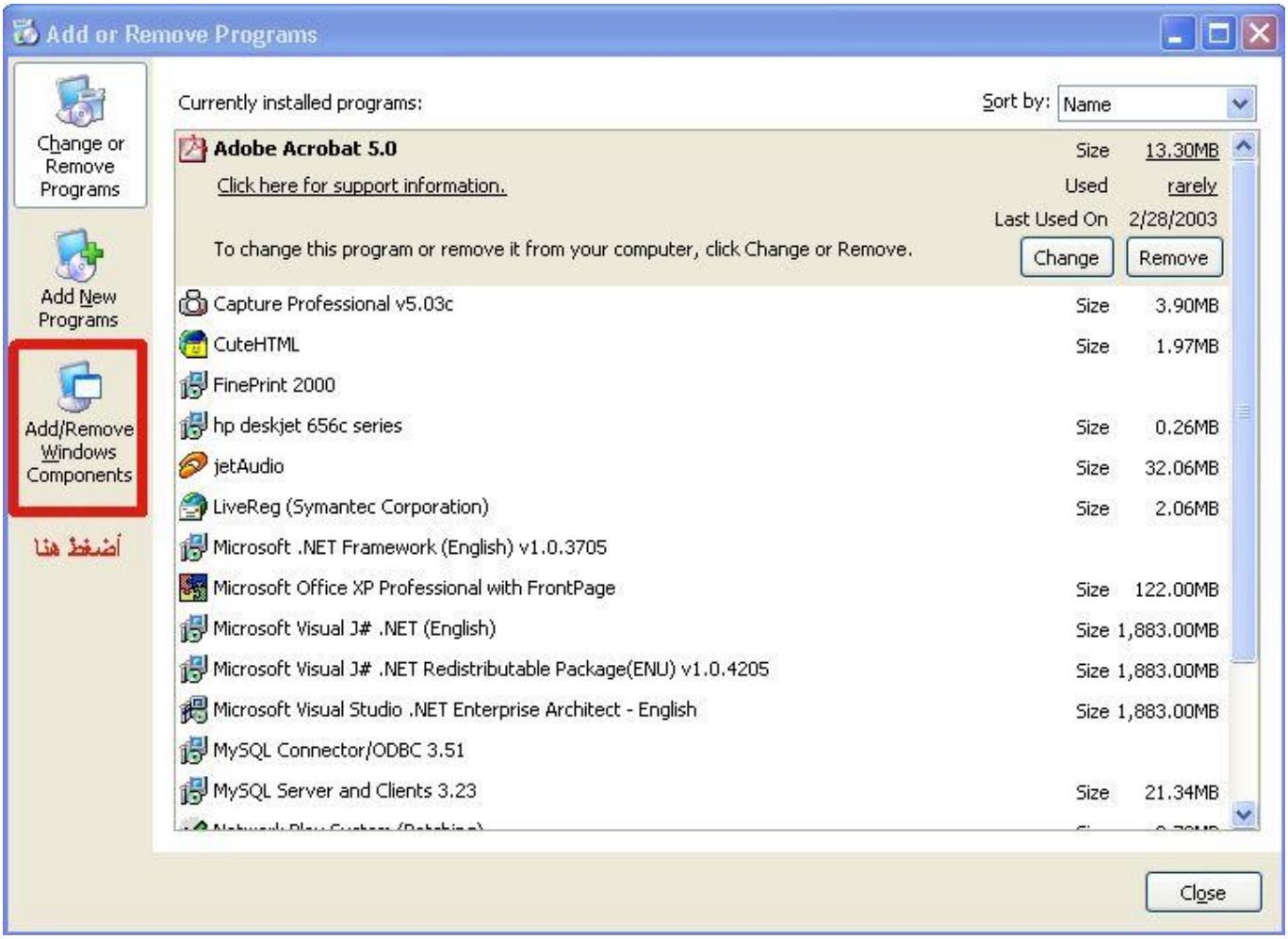
لابد أن يتوافق القرص مع النظام المحمل لديك. يعني، ما ينفع تحمل IIS 5 على ويندوز 98 أو ME

ملحوظة

الصور الرفقة هي لنظام WinXP فقط، لكن كافة العملية هي متشابهة بجميع أنظمة مايكروسوفت.

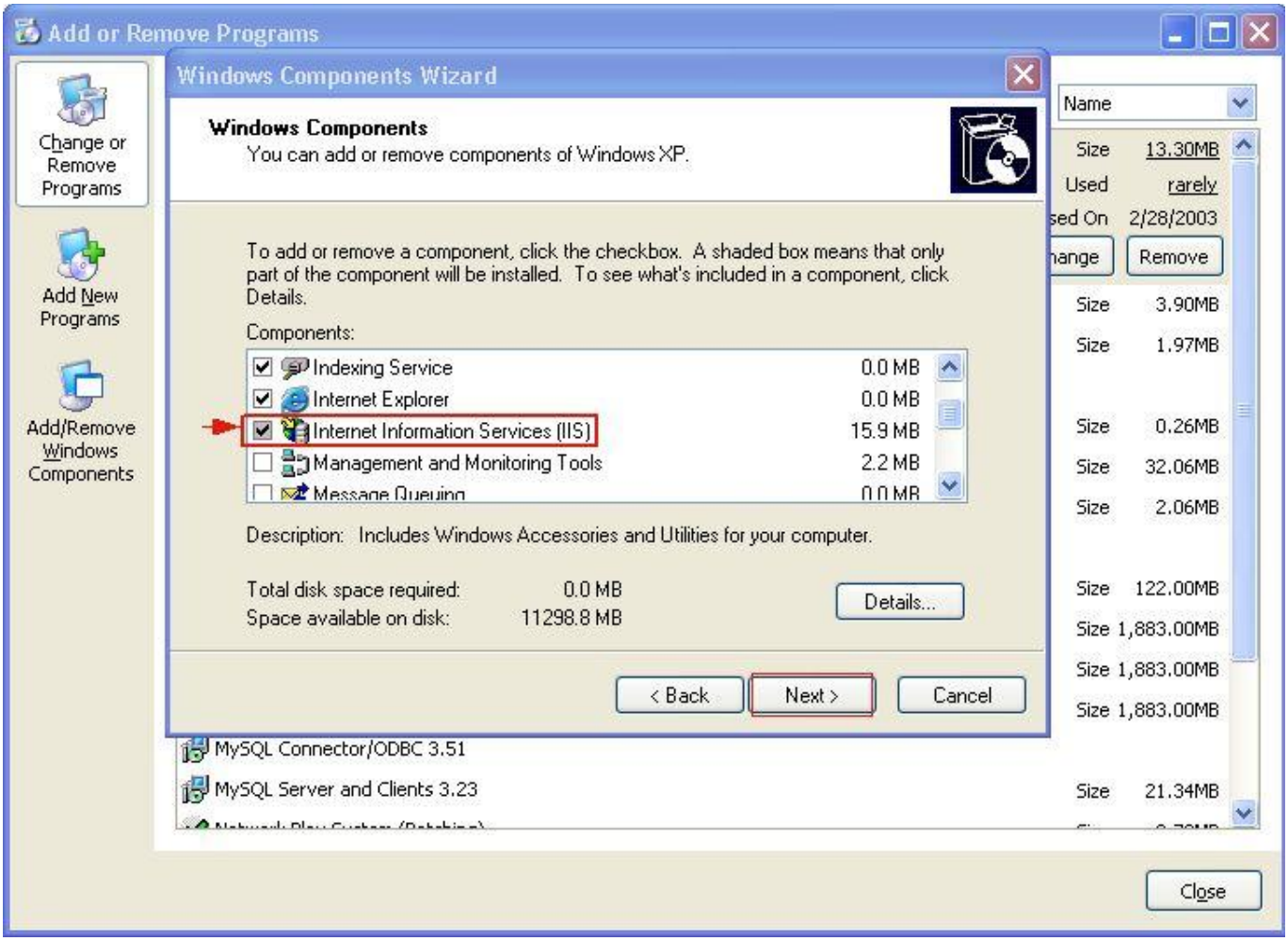
1 - ضع القرص المرن الخاص بنظام Windows في مكانه.

2 - إذهب إلى الـ Control Panel من My Computer وأختار Add or Remove Programs



(الشكل 1)

3 - ستظهر لك شاشة بقائمة برامج يمكنك إختيارها ومن ضمنها برنامج Internet Information Services أو بإسم Personal Web Server تحت نظام تشغيل Win98, ME



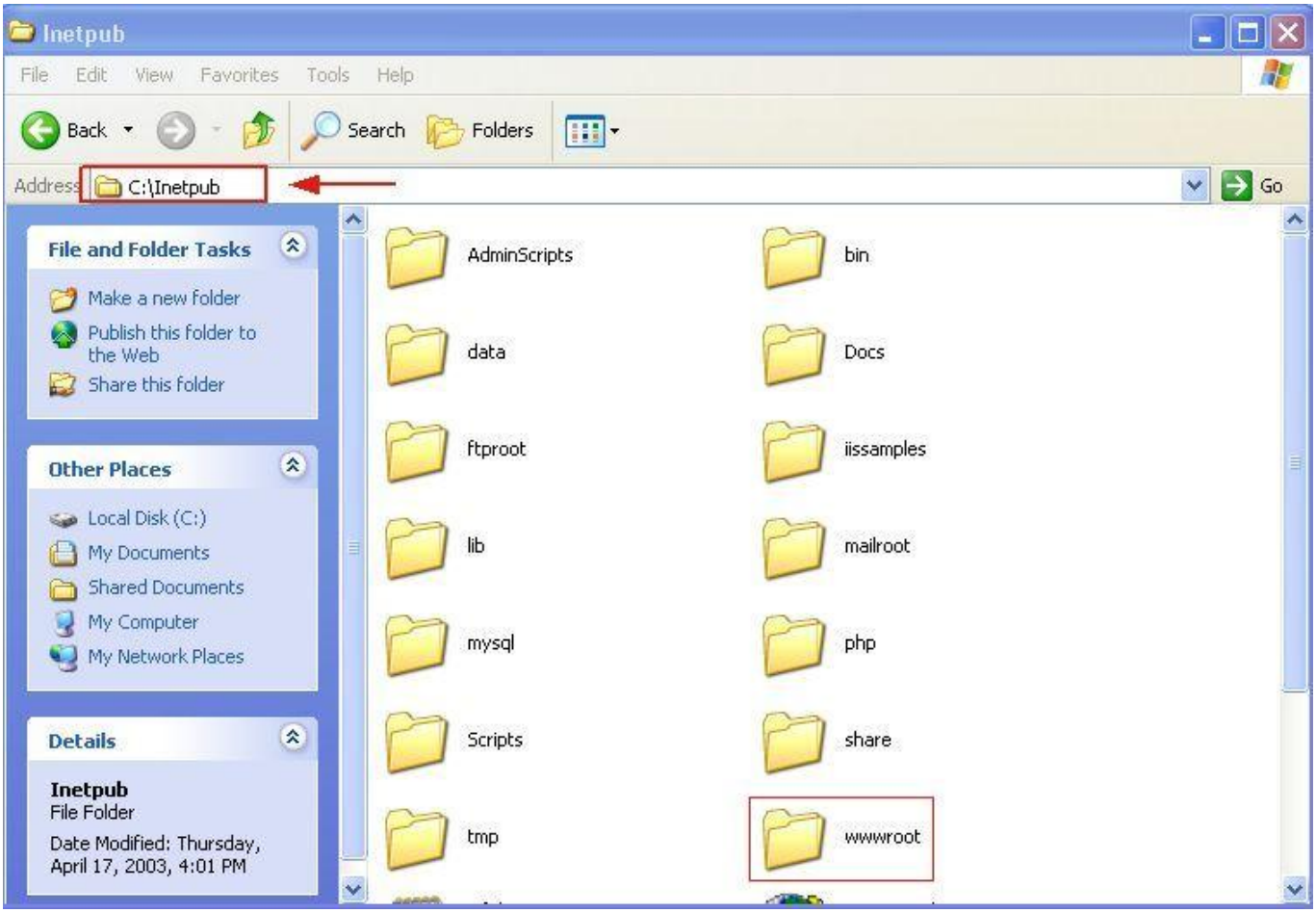
(الشكل 2)

4 – إن نقرت على Internet Information Services مرتين وكأنك تفتح ملف، ستلاحظ وجود العديد من الخدمات ومن ضمنها Frontpage 2000 Server Extensionsللذين يريدون إستخدام الفرونت بيج لتحكيل صفحاتهم المحلية.

5 – بعد إختيار ما تحب من المواصفات (أنا شخصياً إخترتها كلها ما عدى MSN Explorer) إضغط على Next ومن ثم سيقوم الـ Windowsبتحميل وتثبيت برنامج السرفر المحلي تلقائياً.

6 – بعد التحميل يستحسن إعادة تشغيل الكمبيوتر حتى تكون عملية تثبيت وتعريف السرفر صحيحة.

7 – بعد إعادة التشغيل، إذهب إلى القرص الصلب المحلي ...C ستجد مجلد جديد هناك تحت إسم Inetpub وبداخلة مجلد إسمة wwwroot



(الشكل 3)

8 - مجلد wwwroot هو ما يعادل المجلد public_html في نظام اللينكس... فهو مجلد نشر الصفحات.

9 - إفتح برنامج الـ Internet Explorerوضع http://localhost/ أو http://127.0.0.1/ تلاقي أن الملفات الموجودة في مجلد wwwrootفتحت وهي مهمه فلا تمحيها لأنها تحتوي على ملفات المساعدة إن إحتجتها فيما بعد.

ملاحظة : في هذه اللحظة سرفرك الآن يدعم لغة الـ asp و قاعدة البيانات Access فقط

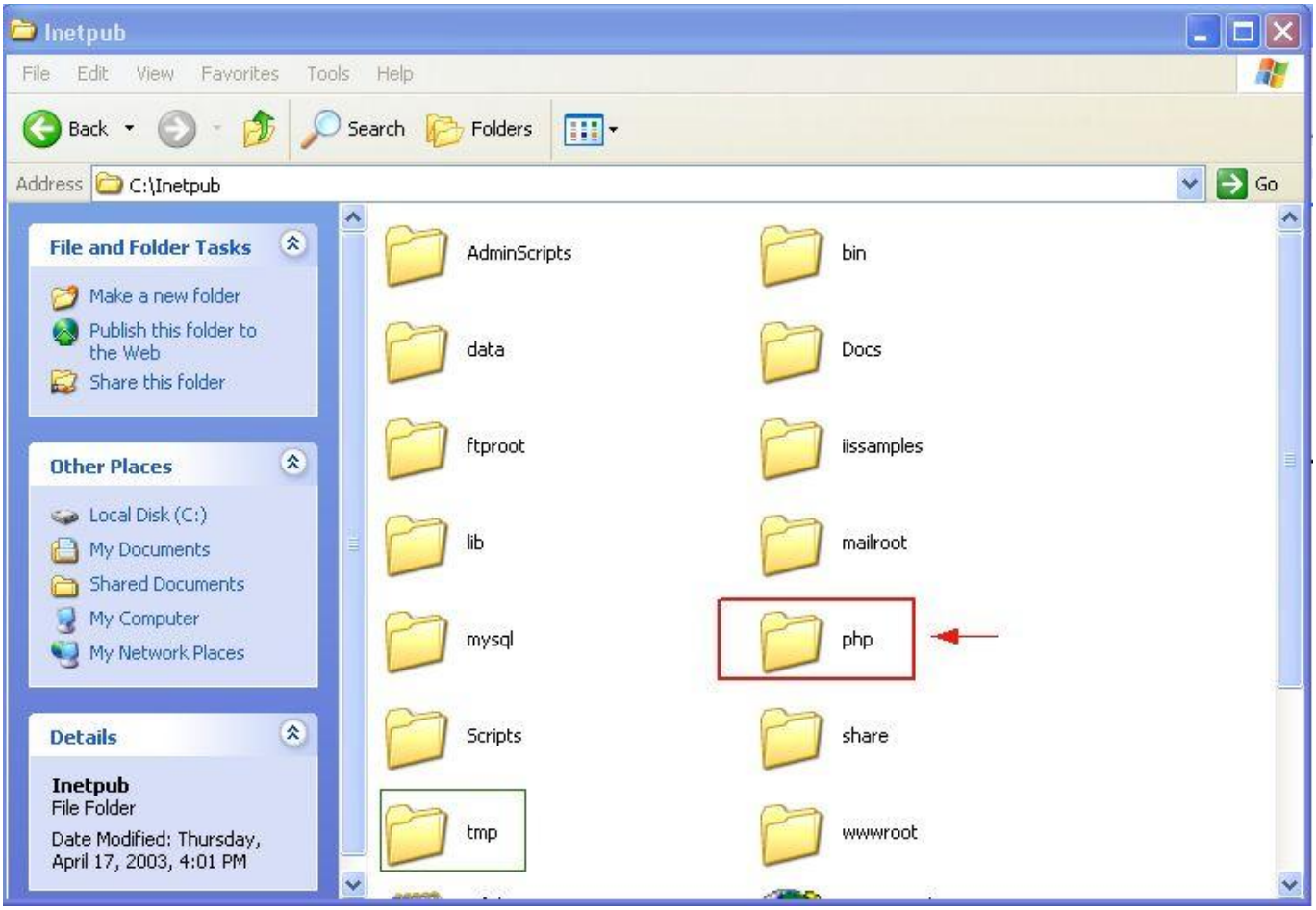
كيف تضيف لغة الـ PHP 4.3.1 لسرفرIIS

متطلبات هذا الجزء PHP4.3.1 : وممكن تحميلها من الموقع عن طريق هذا الرابط - إختار ما يناسبك حجمه MB5.71

http://www.php.net/get/php-4.3.1-Win32.zip/from/a/mirror

1 - فك الضغط من على ملف الـ ZIP وغير إسم المجلد إلى php حروف صغيره وليستPHP

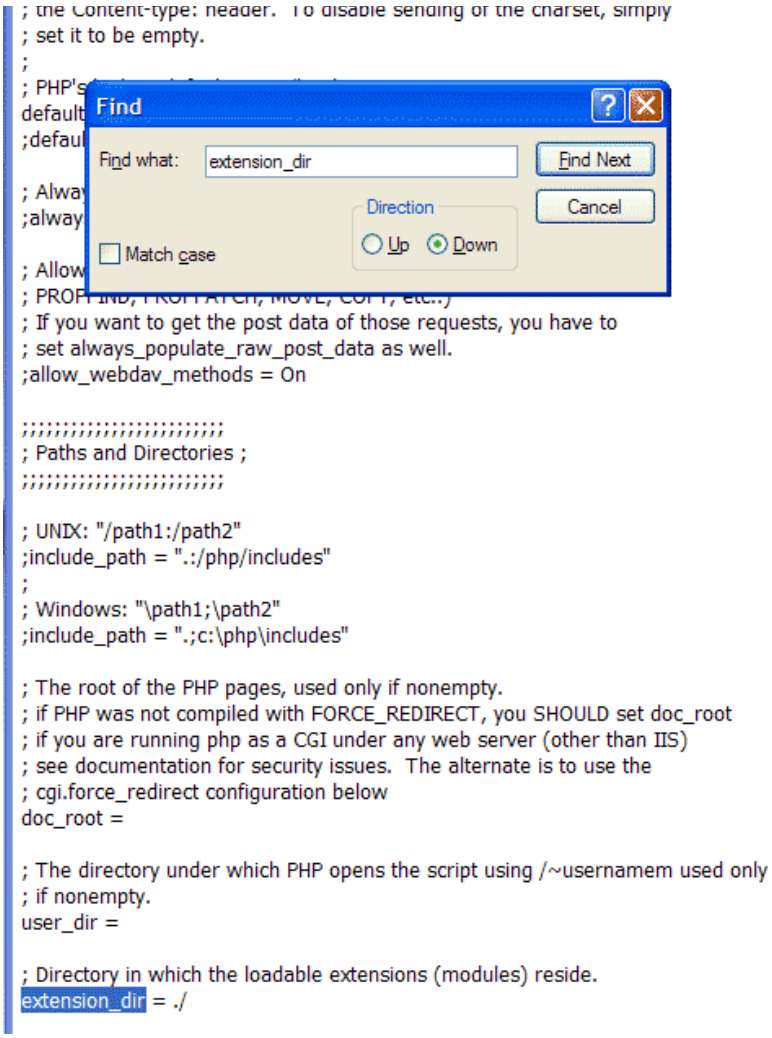
2 - إنقل هذا المجلد تحت مجلد C:\Inetpub



(الشكل 4)

3 - إفتح المجلد php ودور على ملف إسمه "php.ini-recommended". غير إسم الملف إلى php.ini وإنقله إلى مجلد الـ Windowsأو WINNT

4 - إفتح المجلد وستقوم الآن بتغيير بعض تعريفات الـ PHP حتى تمشى معاك أستخدم أحد المحررات

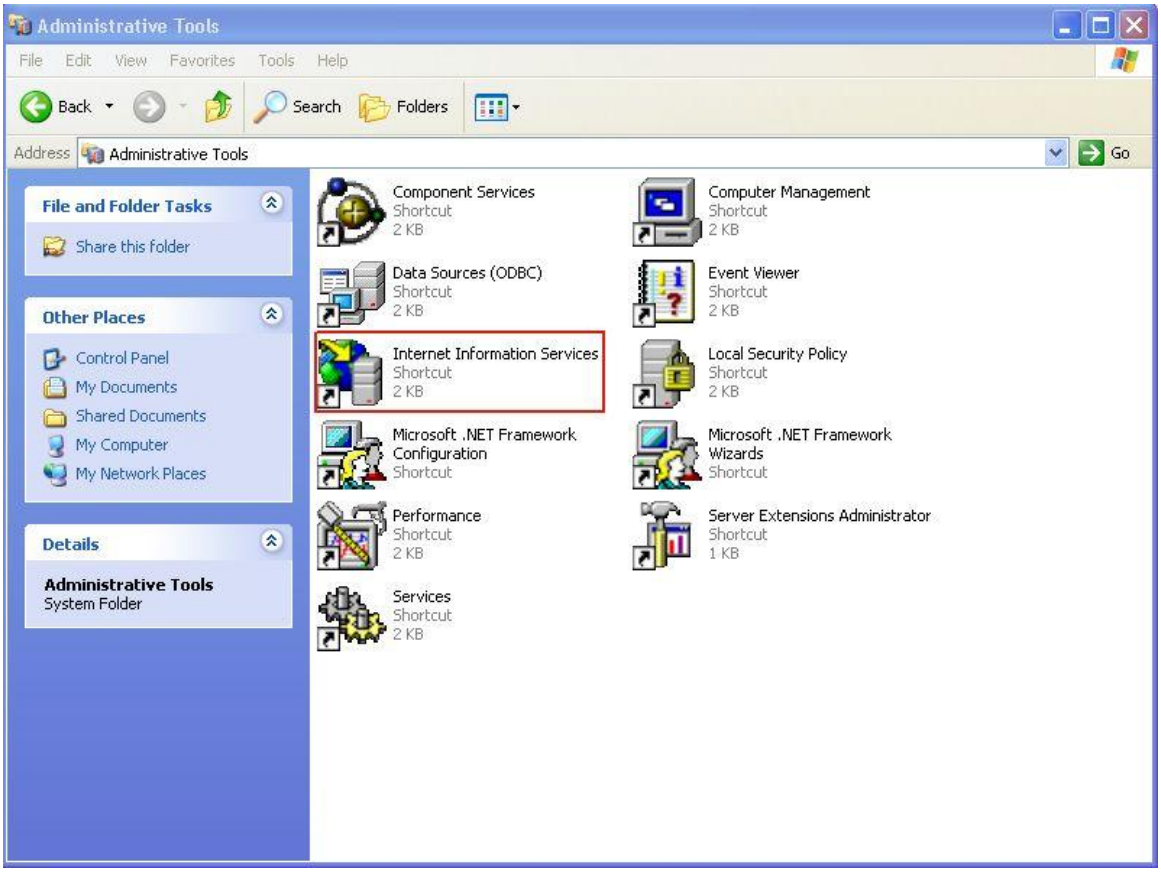


(الشكل 5)

- قم بتغيير extension_dir من ./ إلى C:\Inetpub\php
- قم بتغيير browscap إلى C:\WINDOWS\SYSTEM32\inetsrv\browscap.ini أوWINNT
- قم بتغيير cgi.force_redirect = 1 ; إلى cgi.force_redirect = 0 ملاحظه: إحذف هذه ";"
- قم بتغيير session.save_path من /tmp إلى C:\Inetpub\tmp وإذهب وإنشئ مجلد جديد تحت C:\Inetpub إسمه tmp كما في (الشكل 4) في الصفحه السابقه . المجلد المحدد بي الأخضر

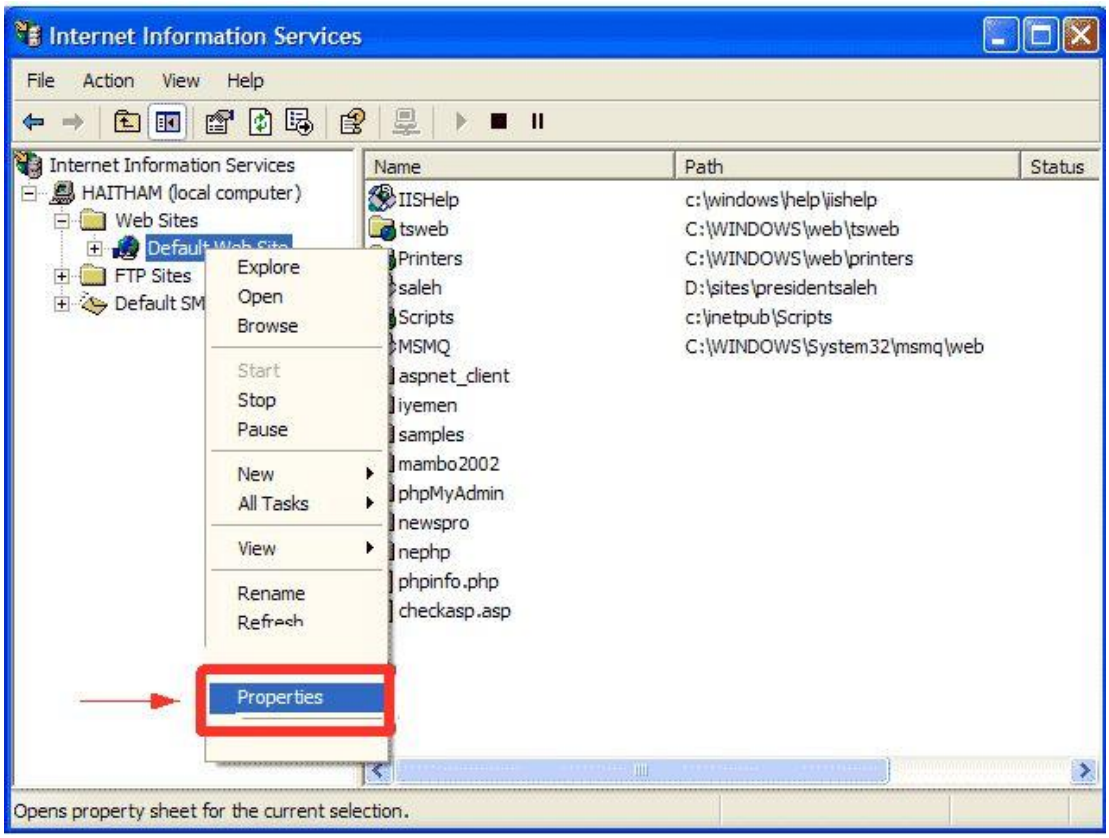
هنا تكون قد قمت بإكمال عملية تثبيت الـ PHPعلى جهازك. والآن عليك أن تجعل سرفر IIS تتعرف على ملفات الـPHP

5 – إذهب إلى Control Panel > Administrative Tools > Internet Information Services



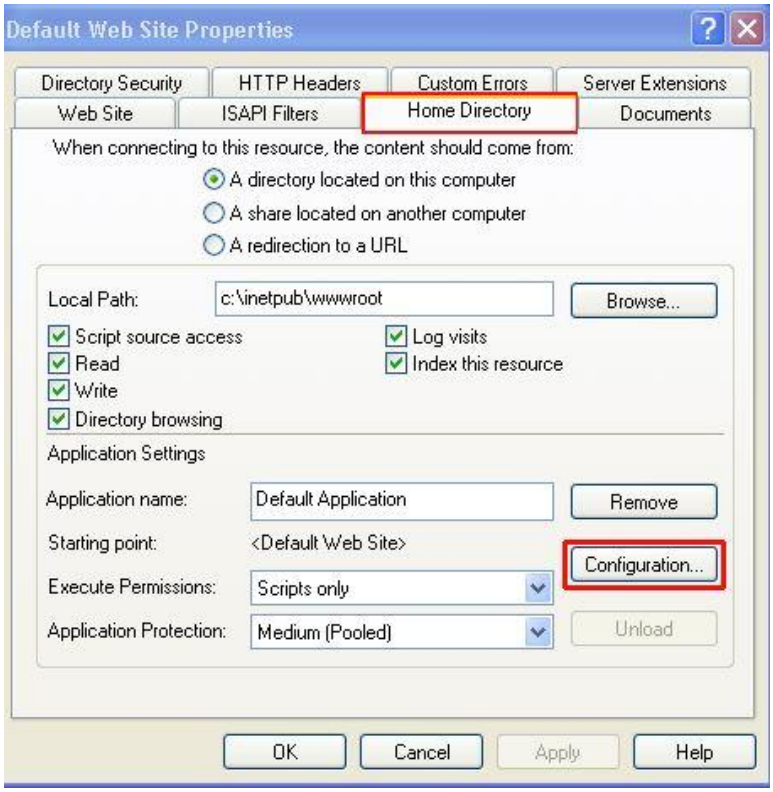
(الشكل 6)

6 - إدرج خواص سرفرك كما هو ملحوظ في الصوره التالية (الشكل 7)



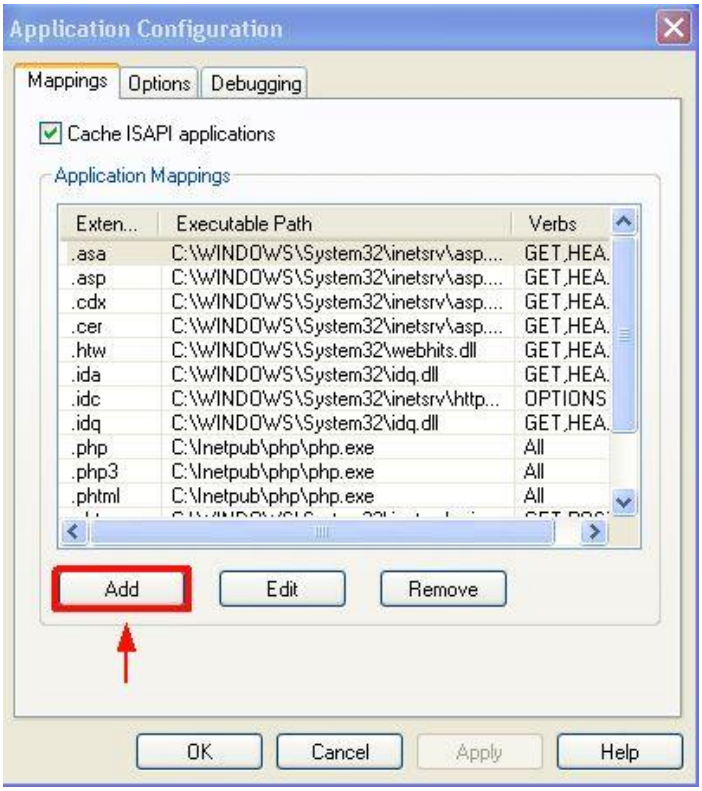
(الشكل 7)

7 - إختار Home Directory ومن ثم Configuration كما في (الشكل 8)



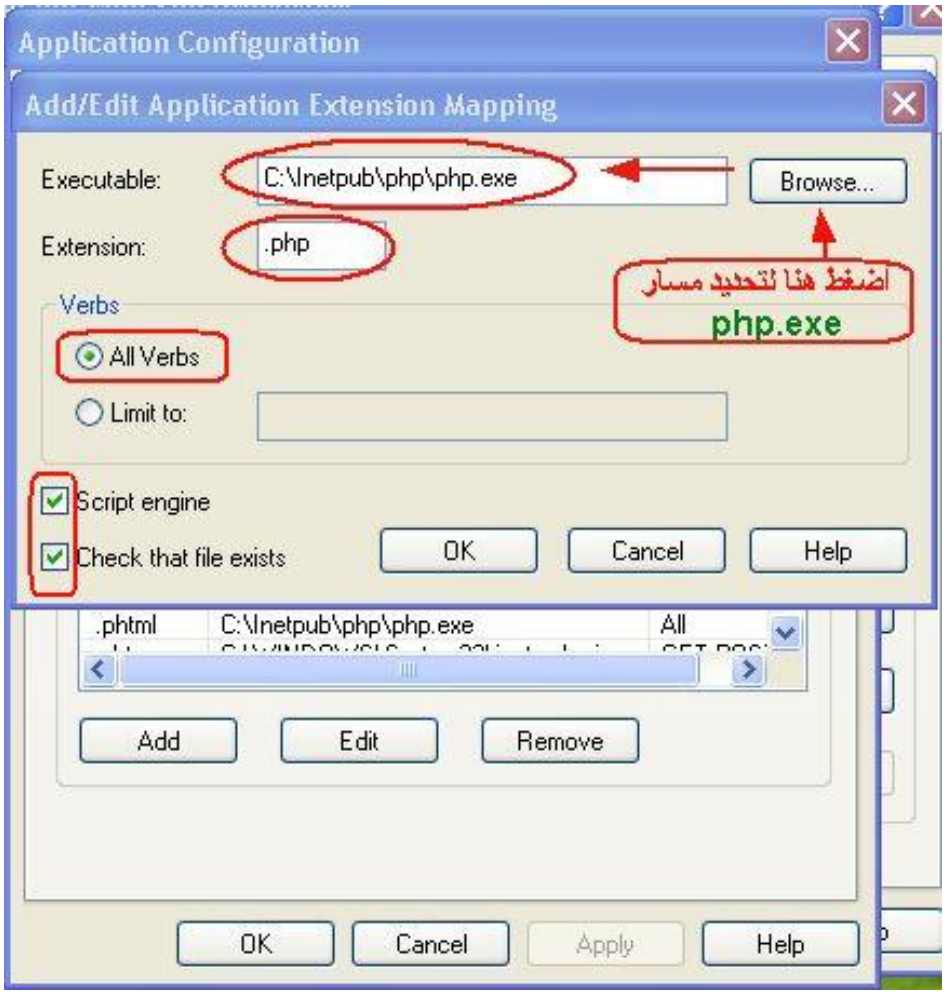
(الشكل 8)

8 - سنقوم الآن بإضافة إمتدادات PHP و PHP3 و PHTML للسرفر حتى يعرف كيف يتعامل معهن .إضغط على Add كما هو موضح في الصورة التالية (الشكل 9)



(الشكل 9)

9 - في الشاشة التالية لابد من تبيين مكان ملف PHP.EXE الموجود في مجلد البرنامج. قم بإختيار المجلد كما هو مبين بالصورة العاشرة ووضح إمتداد الملف .php وإنهي العملية بالضغط على .OK قم بنفس هذه العملية لإضافة الإمتدادات .php3 و.phtml

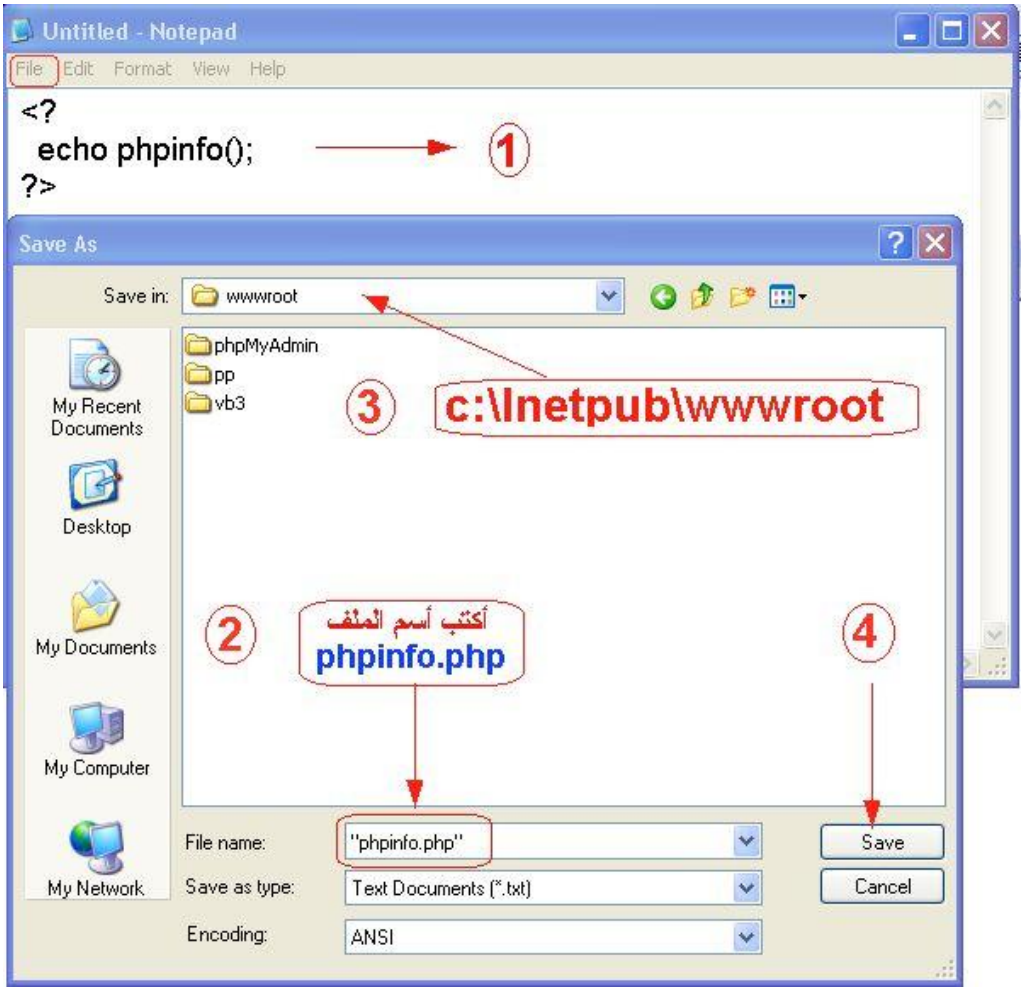


(الشكل 10)

10 - لقد إنتهيت!! إحفظ عملك بالكامل وأعد تشغيل الكمبيوتر .

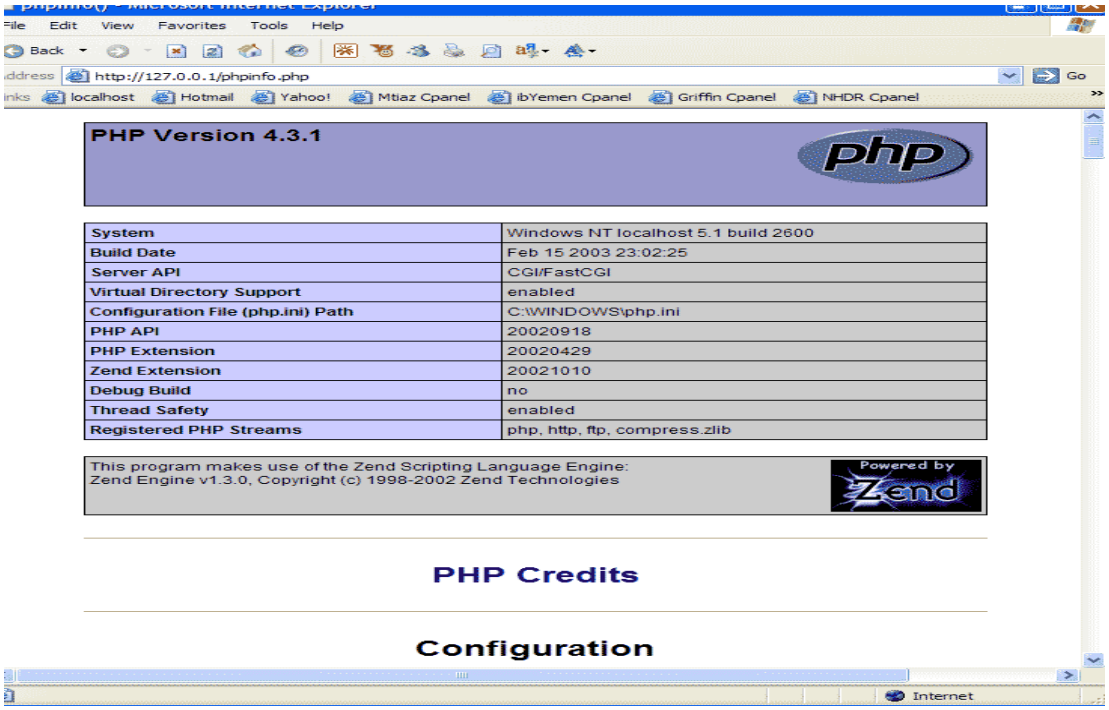
11 - الان قم بأنشاء ملف بأسم phpinfo.php وقم بوضعه داخل C:\Inetpub\wwwroot وإفتح إنترنت إكسبلورر على العنوان http://localhost/phpinfo.php .

ولمعرفة كيف يتم أنشاء الملف قم فتح النوت باد كما هو مبين في الصوره وكتب الكود الموجود به وحفظه



(الشكل 11)

12 - هنا يتبين إن قمت بتثبيت php بشكل صحيح لأن هذا الملف سيعطيك معلومات كاملة عن الإصدار اللي أنت مركبه

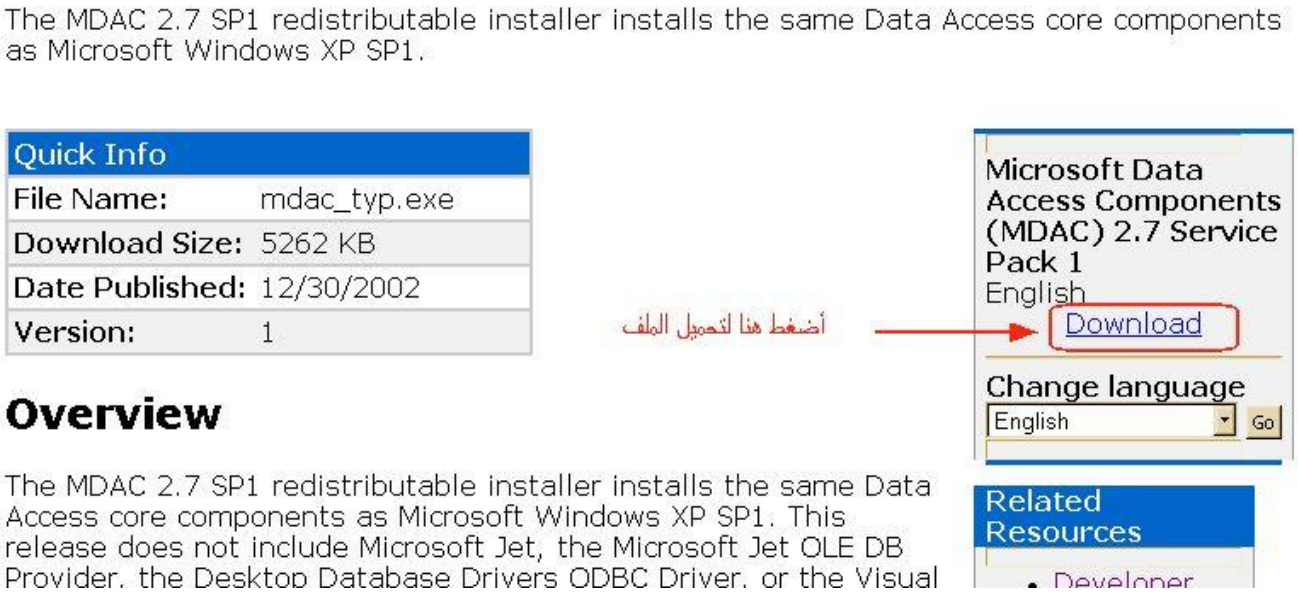


(الشكل 12)

متطلبات الدرس :

- Microsoft MDAC 2.7 هو أحدث إصدار لبرنامج مساعد للسرفر وهو ضروري جداً. وسوف تجده على العنوان التالي /http://www.microsoft.com/data

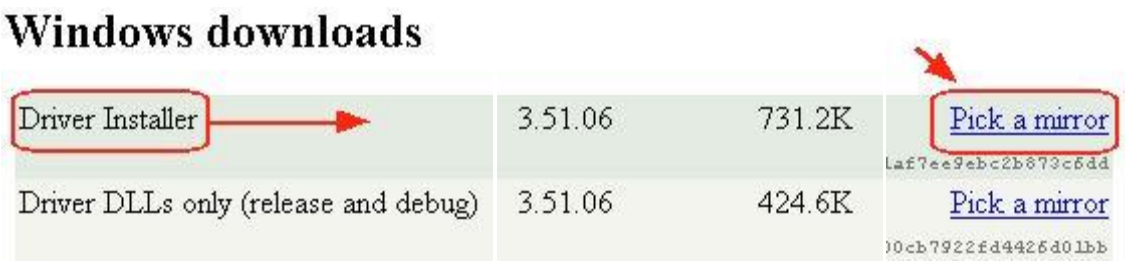
(الشكل 13)



(الشكل 14)

- **MyODBC 3.51.06 هو أيضاً أحدث إصدار لبرنامج الوصل بين اللغة والقاعدة**

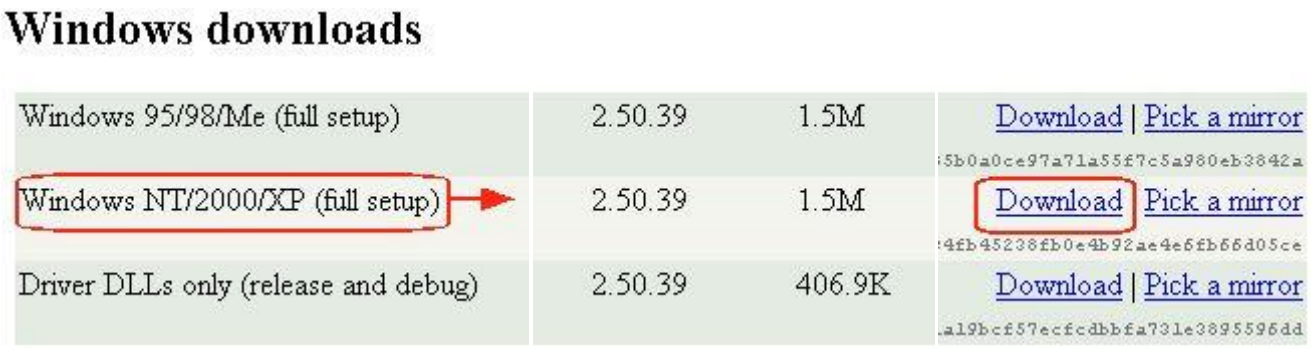
http://www.mysql.com/downloads/api-myodbc-3.51.html



(الشكل 15)

- **MyODBC 2.50.39 إصدار قديم من MySQL ولكنه ضروري لأن قاعدة البيانات تدور على الإثنين عند بدإ التشغيل**

http://www.mysql.com/downloads/api-myodbc-2.50.html



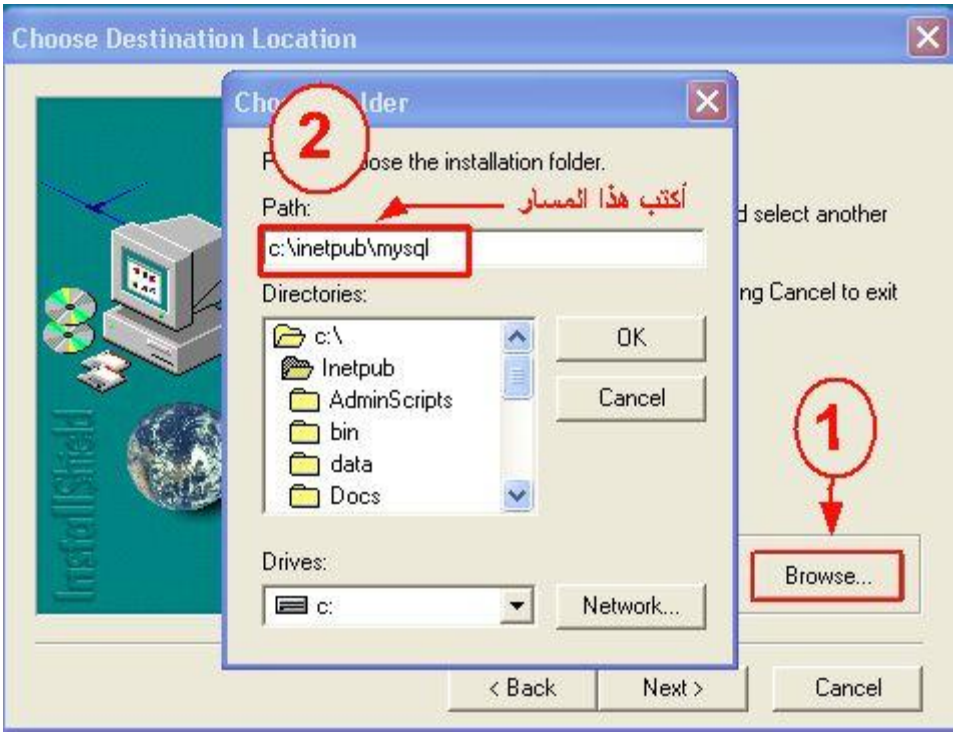
(الشكل 16)

- **MySQL 4.0.12 طبعاً قاعدة البيانات الشهيرة**

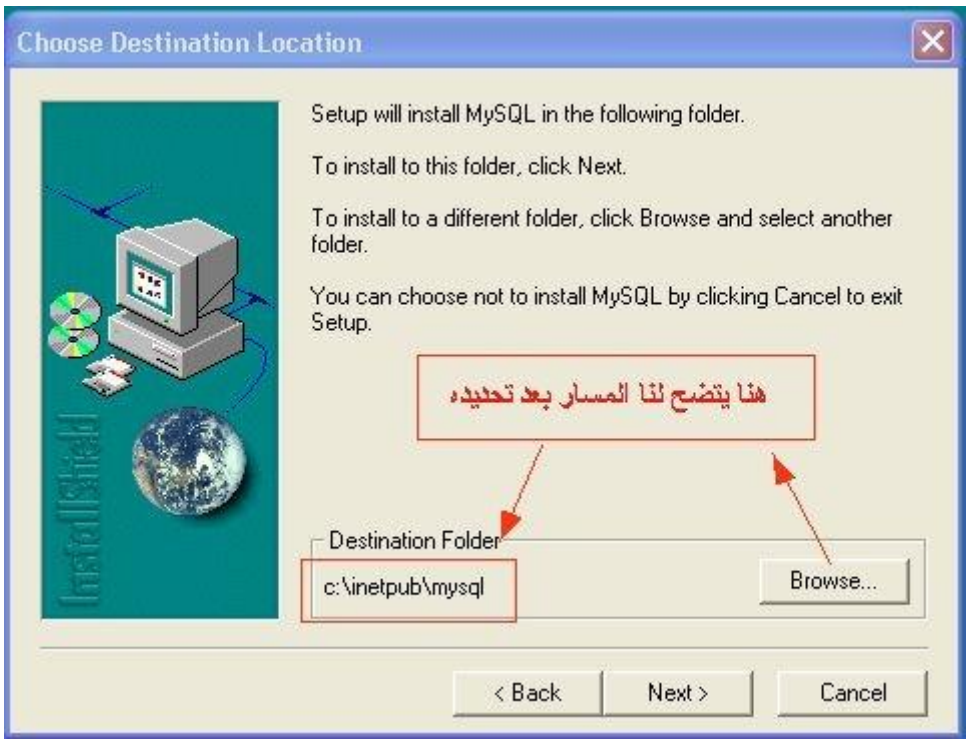
ملاحظة : يمكنك إستخدام إصدار أقدم للـ mySQL إن أردت لكنني لا أنصح بذلك

بعد تحميل وتثبيت أول ثلاث برامج بشكل عادي نبدأ عملية تثبيت الـmySQL

1 - حين تثبيت الـ mySQL ستسأل أين تريد تثبيتها، أنا محددها في C:\Inetpub\mysql بجانب مجلد الـ php



(الشكل 17)



(الشكل 18)

2 - وبعدها البرنامج سيثبتها لك أوتوماتيكياً

3 - أفتح مجلد بإسم bin بداخل mysql وحينها أنقر مرتين على ملف إسمهwinmysqladmin.exe

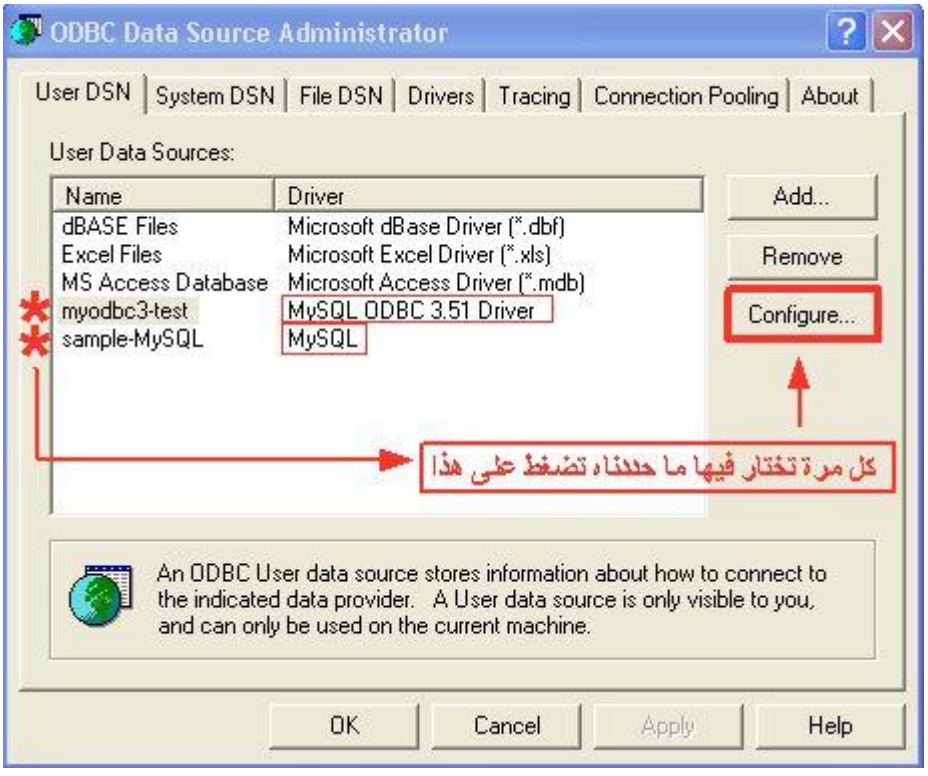
(الشكل 19)

4 - ستطلب منك mySQL أن تحدد إسم الدخول وكلمة السر التي تريها لتجعل سكربتاتك تتعامل معها

5 - أعد تشغيل الجهاز.

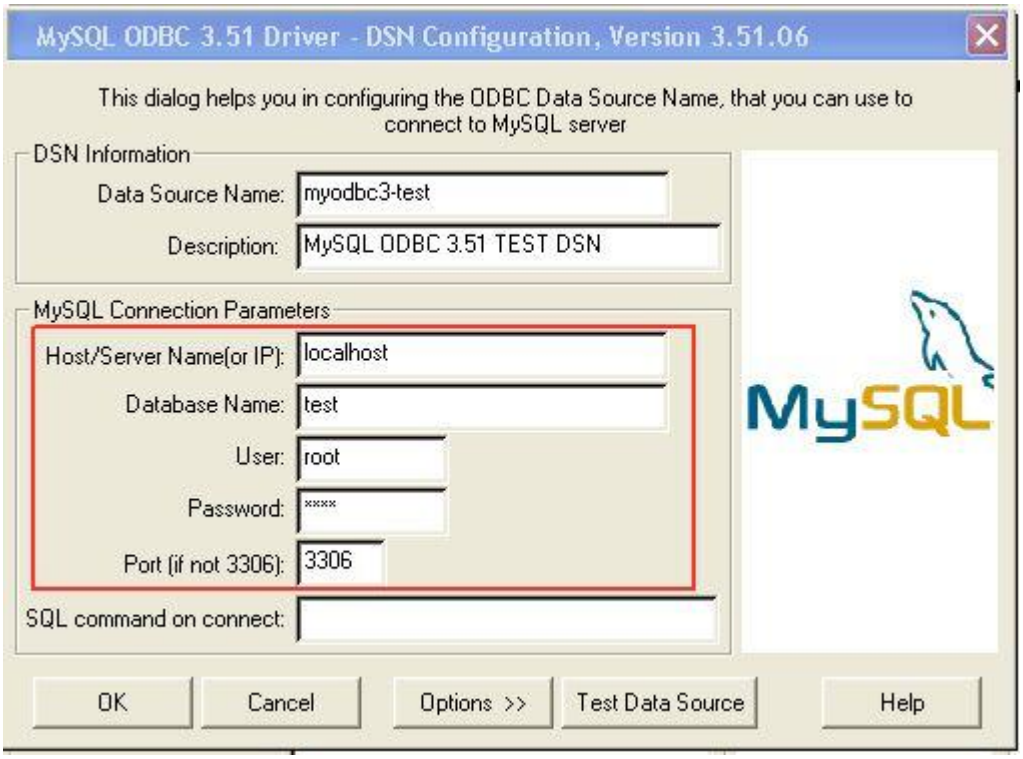
6 - أدخل الى المجلد التالي Control Panel > Administrative Tools > Data Sources ODBC

7 - حين تنقر على Data Sources ODBC تأتيك قائمة كالصورة الشكل 20 وإختار الـ MySQL و MySQL ODBC 3.51 Driver كما مبين لك في الصورة رقم إثنين. حينها ستبان أمامك



(الشكل 20)

سوف تشاهد بعد ذالك شاشة تستطيع أن تكتب فيها معلومات قاعدة بياناتك



(الشكل 21)

ملاحظة: لابد أن تكون متطابقة لنفس البيانات التي كتبتها في الخطوة الرابعة لهذا الدرس .

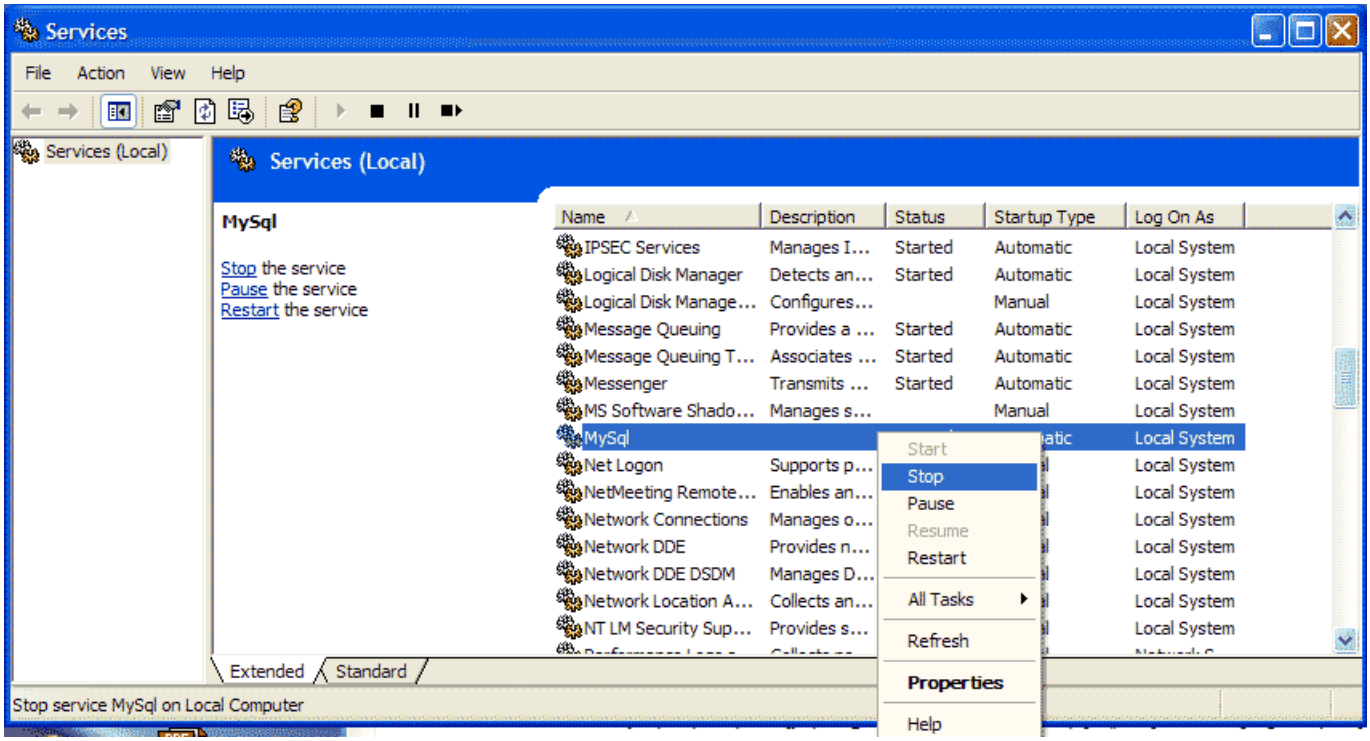
8 - قم بنفس العملية لكلا الـ MySQL Drivers في تلك الشاشة ثم أخرج بعد حفظ التغييرات

9 - أنقر بالزر الأيمن على مؤشر الـ MySQL في الـ Task Bar كماهو في الصورة الشكل 22 ثم إختار Stop This Service

(الشكل 22)

10 - إذهب الى المجلد Control Panel > Administrative Tools > Services

11 - إبحث عن MySQL كما هو موجود بالصورة الشكل 23 وقم بإختيار Stop



12 - أعد تشغيل جهازك الآن

13 - حين تتم عملية إعادة التشغيل، ستلاحظ أن برنامج MySQL بدأ تلقائياً وقد تحول النور من الأحمر إلى الأخضر، وهذا يعني أن قاعدة البيانات MySQL شغالة لديك الآن .

وسيكون الآن سرفرك يشغل :
PHP - ASB - Access - MySQL

بنية ملفات PHP

ملفات PHP هي ملفات نصية بسيطة، تشبه في تركيبها ملفات ASP وملفات HTML بشكل عام، يتكون ملف PHP من قسمان، قسم HTML وقسم PHP، الملف بالصورة الطبيعية عبارة عن ملف HTML عادي، ولكنك تستطيع تحديد أجزاء معينة من الملف ليخرج فيها الملف من وضعية HTML إلى وضعية PHP، لإخراج الملف إلى وضعية PHP توجد عدة طرق :

1 - استخدام زوج الوسوم <?php و ?> كالتالي :

```
<?php

echo 'This is PHP output!';

?>
```

2 - استخدام زوج المختصر <? و ?> وهو يستخدم بنفس الطريقة السابقة ولكنه يكون بدون الكلمة php في وسم البداية، هذا النوع من الوسوم يحتاج إلى كمية أقل من الكتابة بالطبع، ولكنه يتعارض مع وسوم xml، لذا يقوم البعض بإغلاق ميزة الوسوم القصيرة حتى لا يحصل هذا التعارض (يمكنك اغلاق هذه الميزة بسهولة عن طريق ملف إعدادات PHP).

3 - استخدام زوج الوسوم ASP، وهو من اسمه زوج الوسوم المستخدم في ملفات ASP وهما <% و %>، ميزة وسوم ASP لا تكون فعالة بشكل قياسي ولكنك تستطيع تفعيلها عن طريق ملف إعدادات مترجم PHP.

4 - الطريقة الأخيرة هي استخدام زوج الوسوم التالي :

```
<script language="php" >

    echo 'This is PHP output!';

</script>
```

ولكن هذه الطريقة غير مستخدمة الآن، حيث أنها تصعب عملية التمييز بين شفرات PHP وباقي ملف HTML، وكذلك بالنسبة لبرامج كتابة ملفات HTML التي تعطي تلوينا للشفرة فأغلبها لا يتعرف على هذا النوع من الشفرة ويعتبره جزءا من ملف HTML الاعتيادي.

أفضل الطرق السابقة للتحويل إلى وضعية PHP هو استخدام زوج الوسوم الأول بالطبع، حيث أنه الأكثر استخدامها، ولا يحتوي على أية تعارضات كما أنه يعمل على جميع مترجمات PHP مهما كانت إعداداتها، ولهذا السبب سنستخدمها في جميع الأمثلة التي ستجدها في هذه الدورة.

كتابة ملفات PHP

ملفات PHP هي ملفات نصية بسيطة تماما كما هي ملفات HTML، يمكنك كتابة سكريبت PHP بأي برنامج كتابة نصوص يتيح لك كتابة الملفات النصية البسيطة Plain Text مثل Notepad على النظام ويندوز، ولكن أغلبية مبرمجي PHP يستخدمون أدوات أخرى تسهل عليهم عملية البرمجة عن طريق تلوين الشفرات البرمجية، وتسهل عملية البحث عن الملفات واستبدال المقاطع من عدة ملفات في نفس الوقت، مثل HomeSite، على الرغم من أنك لن تحتاج إلى الكثير من هذه الميزات إلا أن استخدام Notepad في عمل ملفات PHP يعتبر أمرا صعبا جدا وخاصة في الملفات الضخمة حيث أن Notepad لا تتيح فتح الملفات الكبيرة، والمشكلة الأكبر هي أنها لا توفر ترقيما للأسطر، فإذا ظهرت لك رسالة الخطأ تشير إلى وجود خطأ في السطر 53 فلن تستطيع معرفة السطر المطلوب في Notepad إلا إذا قمت بالعد يدويا من السطر الأول وحتى 53 .. حسنا ماذا لو كان الخطأ في السطر 652، يمكنك البدأ بكتابة سكريبتاتك بالبرنامج المتوفر الآن إلى أن تحصل على برنامج آخر، يمكنك بالطبع فتح ملفاتك بأي محرر نصوص، فإذا كتبتها باستخدام Notepad فهذا لا يعني بأنك ملزم باستخدام Notepad في جميع ملفاتك أو حتى في هذا الملف.

لعمل ملف PHP الآن قم بفتح محرر النصوص الذي اخترته وابدأ بكتابة الصفحة التي تريدها، ولا تنسى إحاطة شفرات PHP بالوسوم الخاصة بها، ثم احفظ الملف في أي مكان في دليل مزود الويب الخاص بك وأعطه الإمتداد المناسب .php أو .php3 حسب إعدادات مزودك، ثم قم بزيارة الصفحة باستخدام المتصفح وستجد الصفحة وقد تمت ترجمتها وعرضها عليك.

تذكر بأنك يجب أن تزور الصفحة مرور بمزود الويب، ولا يمكنك عرض الصفحة عن طريق فتحها كملف خارجي، على سبيل المثال، إذا كان الدليل الجذري لصفحات مزودك هو : C:\Inetpub\wwwroot\

وقمت بعمل صفحة أسميتها test.php في ذلك الدليل، يجب أن تقوم الآن بتشغل مزود الويب وزيارة الصفحة على العنوان http://localhost/test.php، إذا قمت باستخدام الأمر Open من القائمة File في المتصفح لفتح الملف C:\Inetpub\wwwroot\test.php فلن ترى صفحة PHP مترجمة، وسترى شفرة PHP فقط.

تدريب

قم بتنفيذ ملف PHP التالي :

```
This is the normal html page.<br>

<?php

        echo "This is inside PHP<br>";

        echo "Hello World!<br>";

?>
```

ما الذي تشاهده عند تنفيذ البرنامج السابق؟ من المفترض أن تشاهد الخرج التالي :

```
This is the normal html page.

This is inside PHP

Hello World!
```

ها قد انتهيت من كتابة برنامجك الأول بلغة PHP، لا تقلق إذا لم تفهم أي شيء فيه، سنتعلم الآن كيفية استخدام المتغيرات والعبارات بلغة PHP.

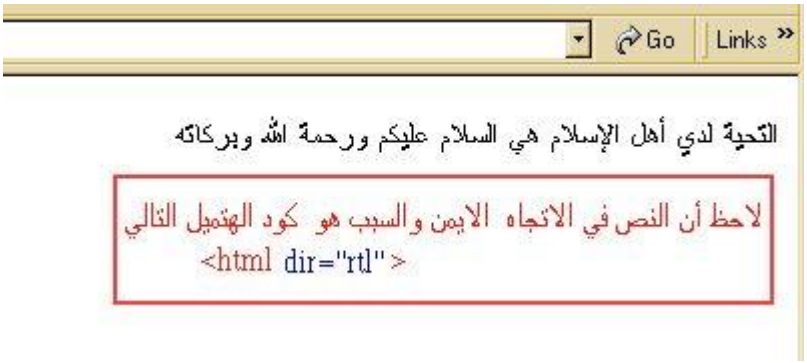
لنكتب سكربتاً بسيطاً (فاتح شهية) :

```
<“html dir = “rtl>
التحية لدي أهل الإسلام هي
؟>
 Echo (“السلام عليكم ورحمة الله وبركاته”);
<؟
<html/>
```

قم بحفظ الملف باسم echo.php

ستعرض علينا عباره مكتوب فيها

التحية لدي أهل الإسلام هي السلام عليكم ورحمة الله وبركاته



شي بسيط أليس كذلك ؟

يتكون كود الـphp من نصوص و كود و علامات ولغة html وقد لاتحتوي على نصوص html .

لكي يعمل الكود يجب أن يكون إمتداد الملف php أو بأي إمتداد من إمتدادات الـphp

مثلاً php3 و phtml

عندما تطلب صفحة في الإنترنت فإنك تجري اتصالاً مباشراً مع السيرفر هذه العملية تدعي request للسيرفر (يعني طلبية للسيرفر) يقوم السيرفر بتفسير طلبك والبحث عن الصفحة المطلوبة ويرسل اليك الصفحة المطلوبة كجزء مما يسمي response (استجابة) لمستعرض الانترنت لديك يقوم بعدها المتصفح لديك بأخذ الكود الذى ارجع إليه ويقوم بتجميعه (compile) لكي يصبح صفحة صالحة للعرض هذه العملية التي حصلت تشبه نظرية العميل للخادم(client to server) بحيث أن المتصفح هو العميل والخادم هو السيرفر .

الخادم يقوم بعملية تخزين وترجمة وتوزيع البيانات بينما يقوم العميل (مستعرض الانترنت لديك) بالعبور الى السيرفر واحضار البيانات

لانريد هنا أن نذهب إلى التكلم عن تاريخ انترنت العتيق ، النقطة المهمة هي الشبكة المربوطة بنقاط nodes الانترنت صممت لكي تقوم بالحفاظ على المعلومات لكي يتم نقلها من مكان إلى آخر وهي تستخدم مجموعة من البروتوكولات مثل Tcp/Ip لكي يتم نقل البيانات عبر الشبكة .

بروتوكول Tcp/Ip

من مميزات هذا البروتوكول أنه بإستطاعته إعاده تمهيد طريقه للبيانات إذا تم خلل في نقطة أو مكان أثناء نقلها ويتم ذلك بسرعة شديدة.عندما يطلب المستخدم من المستعرض أن يجلب له صفحة من الانترنت فإن المستعرض يجلب هذه الأوامر باستخدام بروتوكول يدعي بروتوكول التحكم في نقل البيانات TCP هذا البروتوكول هو بروتوكول نقل للبيانات وهو يضمن أن البيانات قد تم إرسالها ووصولها بشكل صحيح .

قبل أن يتم إرسال البيانات عبر الشبكة يجب عنونتها والبروتوكول الذي يقوم بعنونة البيانات يدعي HTTP يقوم هذا البروتوكول بوضع عنونة للبيانات لكي يعرف البروتوكول TCP أين سينقل البيانات (فهو لايستطيع نقل البيانات إذا لم يكن لها هدف أو مكان) يستخدم البروتوكول HTTP عن طريق الويب في عملية نقل البيانات من كمبيوتر إلى آخر عندما ترى الصفحة متبوعة بـhttp:// فانك تعلم مباشرة أن الانترنت يستخدم البرتوكول HTTP لإحضار هذه الصفحة يمكنك أن تاخذ صورة بأن الـTCP عبارة عن ساعي بريد الذي يقوم بإيصال رسالة ، هذه الرسالة فيها طابع بريد وعنوان وهو مانسميه بالـHTTP .

يتم تمرير الطلب من المستعرض إلى ملقم أو سيرفر الويب وهو مايعرف بـ HTTP request ويقوم السيرفر برؤية مستودع البيانات لديه لكي يحصل على البيانات المطلوبة فإذا وجد الصفحة في المستودع قام بإرسالها على شكل حزم الى الجهة التي قامت بالطلب باستخدام بروتوكول TCP ويعنون هذه الحزم لمستعرض الانترنت لديك باستخدام بروتوكول http (ننبه دائما الى أنه يرسلها على شكل حزم لكي تعرف السبب عند عدم ظهور صفحة ويب كاملة أن هناك حزمة لم ترسل بشكل جيد) ولكن إذا لم يجد السيرفر الصفحة المطلوبة فانه يقوم بإرسال صفحة تحتوي على رسالة خطا 404 وهذه الصفحة التي أرسلت من ملقم الويب الى المستعرض لديك تسمي HTTP response .

بروتوكول الـ HTTP

رغم ما أخذناه من معلومات كثيرة وقصص كثيرة تشبه قصص ألف ليلة أو حكايات الأطفال إلا أنه رغم ذلك يفوتنا الكثير من التفاصيل في هذا الموضوع لذلك دعنا نغوص قليلاً في التفاصيل عن بروتوكول HTTP بشكل خاص.

عندما تقوم بعملية طلب لصفحة من السيرفر هناك أمور إضافية ترسل مع عملية الطلب http request غير الـ URL وهي ترسل كجزء من http request .

نفس الموضوع مع الـhttp response هناك أمور أخرى تصل معه كجزء منه .

الكثير من هذه المعلومات تولد تلقائياً في رسالة الـHTTP ولايقوم المستخدم بالتعامل معها مباشرة ، إذن لايحتاج أن تقلق نفسك بشأن هذه المعلومات إذا أنت لم تنشأها في الأصل ويجب أن تأخذ أيضا في معلوماتك أن هذه المعلومات ترسل كجزء من الـ HTTP request والـHTTP response لأن سكربت الـ PHP الذي نصنعه يمنحنا تحكماً إضافيا بهذه المعلومات .

كل رسائل الـHTTP تأخذ تنسيقاً معيناً سواء كانت Request أو Response . نستطيع أن نقوم بتقسيم هذا التنسيق إلى ثلاثة أقسام :

1 – Request / response line
2 - HTTP header
3 - HTTP body

المحتوي من هذه الأشياء الثلاثة يعتمد على نوع الرسالة إذا كانت HTTp Request أو HTTP response لذلك سنتكلم عنهم بتعمق أكثر .

HTTP Request

يجب أن يحتوي الـrequest على الأقل الـ request line (سطر الطلب) والـHOST .
يرسل مستعرض الانترنت طلبية (HTTP request) إلى ملقم الويب تحتوي على التالي :

1- The Request Line

السطر الأول من كل طلبية (http request) هي Request Line الذي يحتوي على ثلاثة أنواع من المعلومات:

أ - أمر HTTP وهو مايعني بـ method .
ب - المسار من السيرفر إلى المصادر المطلوبة (صفحات الانترنت) المطلوبة من قبل العميل (المستعرض)
جـ - إصدارة الـHTTP .
الـ method يخبر السيرفر كيف يتعامل مع الطلب هناك ثلاثة أنواع شائعه من الـ method

2- HTTP Header

البت الثاني من المعلومات هو الهيدر HTTP Header .الذي يحتوي على تفاصيل أو وثائق عن العميل مثل نوع المتصفح (نتسكيب أو إكسبلور) الذي قام بطلب الصفحة والوقت والتاريخ والإعدادات العامة

الـHTTP Header يحتوي على معلومات نستطيع تقسيمها الى ثلاث فئات وهي :

أ - عامة GENERAL : تحتوي معلومات إما عن العميل أو السيرفر ولاتخصص إلى فرد أو مجموعة .
ب - شخصية Entity : تحتوي على معلومات عن البيانات التي أرسلت بين المتصفح والسيرفر .
جـ - مطلوبة Request : تحتوي على بيانات عن إعدادات العميل والأنواع المختلفة المقبولة من البيانات .

3- The HTTP Body

إذا تم استخدام الأمر POST في الـHTTP Request Line عندها يقوم الـ HTTP بطلب المعلومات التي ارسلت في الـbody الى السيرفر .

HTTP Response

يرسل من السيرفر إلى المستعرض ويحتوي على ثلاثة أشياء :

1- The Response Line

2 - HTTP header

3 - HTTP Body

1 - The Response Line

الـ response line يحتوي فقط على نوعين من المعلومات :

1 - رقم إصدارة الـHTTP .

2 - شفره أو كود الـhttp request التي تقوم بتحديد إذا كان الـrequest ناجحاً أم فاشل .

2 - HTTP Header

الـ response header يعتبر مشابه request hader الذي ناقشناه في الأعلى .وتنقسم المعلومات التي فيه أيضا إلى ثلاثة أنواع :

أ - عامة GENERAL : معلومات عن الـ client أو السيرفر ولاتخصص إلى واحد منهما .

ب - شخصية Entity : يحتوي على معلومات عن البيانات التي يتم ارسالها بين السيرفر والعميل .

جـ - الإجابة Response : يحتوي معلومات عن السيرفر الذي قام بإرسال الرد وكيفية تعامله ومعاجلته للرد (Response) .

3 - HTTP Body

إذا تم معالجة الطلب بنجاح ، فإن الـHTTP response Body يحتوي على كود الـ HTML ويقوم مستعرض الانترنت بتفسيرها وتحويلها إلى الصفحة النهائية التي تراها .

أين سكربت الـPHP من ذلك كله ؟

أصبح الآن لدينا مفهومية جيدة عن طريقة إرسال المستعرض طلب صفحة من السيرفر وكيفية استجابة السيرفر لهذا الطلب .

تكلمنا عن أن سكربت الـ php يتكون من ثلاثة أشياء : نص وكود php وكود HTML ، لانسطيع وصف الـ HTML بأنها لغة برمجة بشكل جيد ونستطيع أن نقول أن الـ php لغة سكربتات Scripting Language لأنها تضيف قدرات HTML عليها مثل الجداول والفريمات بكود HTML بداخل كود الـ php هناك لغات تسمي لغات سكربتات قد تكون متآلفاً معها مثل الجافا سكربت والفجول بيسك سكربت بإستثناء أن الفرق بينها وبين الـ php هو أن الـ php لغة تعتمد على جهة المزود أي السيرفر ويمكنك تخصيص المتصفح الذي يستعرضها .

تجعلنا الـ HTML نضمن سكربتات الـ php فيها ضمن قواعد لذلك لكي نستطيع تشغيلها ولكننا لاننسي أن إمتداد الملفات يظل كما هو php أو php3 بدون تغير فيه لكي يتم إرسال السكربت الى مكتبة الترجمة (scripting engine) التي تقوم بترجمة السكربت إلى HTML (كأنك تترجم من عربي لإنجليزي أو العكس)

ممكن أن نقسم عملية الترجمة الذي يقوم بها سيرفر php إلى قسمين أو عمليتين :

العملية الأولى : هي أن السيرفر يقوم أولا بفحص قواعد اللغة وهذا لايضمن أن السكربت صحيح مائة بالمائة ولكنه تدقيق في الأوامر وقواعد اللغة وهذا مايسمونه بالـ Parsing

العملية الثانية : هي تنفيذ السكربت بعدها وإخراجه على شكل كود HTML وهذا مايسمي بالـ Execution .

بقي أن نقول أمراً معروفاً وهو أن السكربتات نوعين :

1 - وهو ماينفذ من جهة المزود

Server – Side scripting

2 - ماينفذ من جهة المستعرض (صفحة انترنت) .

التعليقات

ما رأيك إذا كنت في شركة وكان معك أكثر من مبرمج وأردتم تصميم برنامج ، اإذن قد تحتاجون لتنظيم العمل وتعديله لذا من اللازم أن تقوم بعمل توضيح لفائدة الكود الذي كتبته كي يسهل فهمه عليهم وإضافة تعديلات مناسبة ، إذن التعليقات تستخدم في الإفاده عن شرح الأكواد أو إضافة معلومات لاتستعمل إلا كتوضيح أو أي شي آخر .

يمكنك عمل تعليق من سطر واحد كالتالى :

```
<?
//هذا تعليق لا فائدة له اي معني
?>
```

مثال آخر :

```
<?
//هذه الداله تقوم بطباعه الكلمه تعليق
Echo "تعليق";
?>
```

وأيضا يمكنك استخدام تعليق من أكثر من سطر كالتالي :

```
<?
/* تعليق يتكون من
 اكثر من سطر بعلامة السلاش والنجمه
*/
?>
```

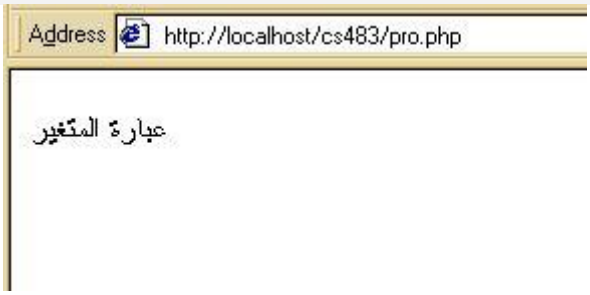
المتغيرات

ماهي المتغيرات ؟

أبسط تعريف يمكن أن نقوله عن المتغير هو أنه مساحة من الذاكرة تستخدم لتخزين المعلومات ويتم التحكم فيها عن طريق المبرمج في الـPHP ، المتغيرات تبدأ بعلامة الـ$ ولكي تقوم بإدخال قيمة في المتغير فإنك تستخدم المعامل (=) إذن لكي تقوم بإنشاء متغير يحتوي على قيمة يمكنك القيام بذلك كالتالى :

```
$char = "عبارة المتغير";
$اسم_المتغير = قيمه ;
```

```
<?
$char = "عبارة المتغير";
Echo $char;
?>
```



لاحظ أن السطر السابق يتكون من خمسة أشياء :

1 / المتغير وهو char
2 / وقبله علامة الـ$ لكي يعرف مترجم الـPHP أنه متغير
3 / المعامل (=)
4 / الفاصلة المنقوطة (;)
5 / القيمة وهي How Are You Every Body? وهي القيمة الموجودة في المتغير أو التي اقترحناها للمتغير أو التي وضعناها فيه (لأن الذي اقترح القيمه هو أنت (مبرمج الـphp))

ملاحظات :

١ ـ اسماء المتغيرات حساسة لحالة الأحرف إذا كانت كبيرة وصغيرة

```
<?
$Majed  = "العبارة الأولى";
$majed  = "العبارة الثانية";
echo  $majed;
echo  $Majed;
?>
```

المتغيرين الذين بالأعلي مختلفين بسبب حالة الأحرف.

2 ـ يمكنك استخدام المعامل (_)

$First_name

3 - يمكنك استخدام ألف حرف في تسميه المتغيرات (وفي الواقع هي غير محدده) .

علامات التنصيص

وهذه نقطة مهمة وهي لماذا وضعنا علامات التنصيص هذه ؟ فالإجابه تكون هي أن القيمة التي وضعناها حرفية أي تتكون من نصوص وهناك أنواع للمتغيرات وعلى ذلك سنفصل ونقول

هناك انواع للبيانات وهي :

1 - strings (حروف)

```
$Exa = “Just An Example”;
$Exa2 = “2.5”;
$Exa3 = “2”;
```

2 - Integer (ارقام)

```
$Exam = 5;
```

3 - Double (ارقام ذات فواصل)

```
$num= 5.4
```

4 - array (ياتي تفصيلها فيما بعد)

5 - objects (تفصيلها في دروس اخري)

6 - Unknown (ياتي تفصيلها في درس اخر)

المتغيرات لايتم تعريف نوعها من قبل المبرمج إنما مترجم الـPHP يقوم بالتعرف عليها لكي يتم إتمام العمليات المختلفه عليها

البيانات الحرفيه :

في الـ PHP أي قيمة تكون بين علامتي تنصيص عادية أو علامة تنصيص مفردة يعتبرها الـPHP قيمة حرفية أمثلة :

”هذا النص بين علامتي تنصيص عاديه اومزدوجه“

’هذا النص بين علامتي تنصيص مفرده او وحيده‘

يجب أن يبدأ النص وينتهي بنفس علامة التنصيص ، وإلا فلن يتعرف الـPHP على القيمة الحرفية أو على النص .

```
<?
$d=”غلط ‘
echo “خطا ‘
?>
```

لايمكنك أيضاً أن تقوم بوضع علامة تنصيص من نفس النوع التي تستخدمه القيمة الحرفية في وسط العبارة الحرفية أو النص

```
<?
$variable = “"هذا النص "خطا بسبب وجود علامة في النص من نفس النوع";
?>
```

وتصحيحه

```
<?
$variable  = “’هذا النص ’ صحيح”;
?>
```

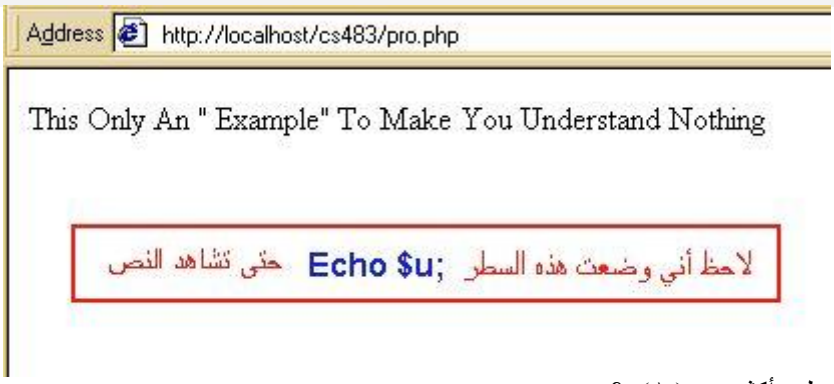
وأيضا مثال آخر

```
<?
$r = “This is”BAD”; // خطأ
$t = “This is ‘good”; // صحيح
?>
```

أما إذا كنت مصراً على ذلك أو تحتاج إليها في عمليات ضرورية (كما سوف نري فيما بعد حاجتنا إليها في صناعة النماذج) فيمكنك وضع معامل (\ ) قبل علامة التنصيص .
لكي تعمل معك بكل سهولة .

مثال :

```
<?
$u = “This Only An \” Example\” To Make You Understand Nothing”;
?>
```



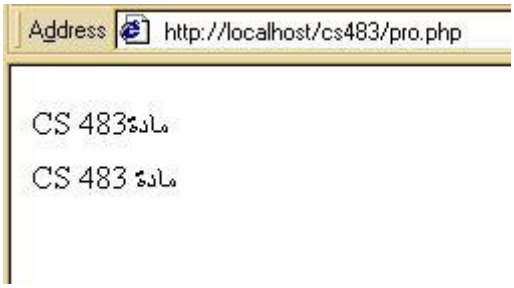
طيب ما رأيك لو أردنا أن نطبع أكثر من (\ ) ؟
الحل هو أن نتبعه بمثله ، وبالمثال يتضح المقال :

```
$file = "c:\windows\system.ini";
echo $file;   // النتيجه c:\windows\system.ini

$file = "c:\\\windows\\\system.ini";
echo $file;   // النتيجه c:\\windows\\system.ini
```

يمكنك الجمع بين أكثر قيم المتغيرات في متغير واحد عن طريقة الـ (.)

```
<?
```

```php
$first = “CS 483” ;
$last = “مادة”;
$fullname = $first.$last;
Echo $fullname ;
Echo "<br>";
//ولكننا نريد وضع فراغ بين الكلمتين
$fullname=  $first . ’ ’ . $last  ;
Echo $fullname ;
?>
```



وأيضا يمكننا أن ضيف إلى متغير قيمة متغير آخر :

```php
<?
$f="I Love M" ;
$k= "y Country" ;
//إضافه القيمة الى المتغير
$f = $f . $k;
echo $f;
?>
```

```php
<?
//تقريباً نفس العملية
$f="I Love M" ;
$k= "y Country" ;
$f.=$k;
echo $f;
?>
```

الارقام

العدد الفردي والمزدوج
الاختلاف المعروف لدي أنا حتى الآن هو أن الفرق بينهما هو الفاصلة العائمة (والله حتي اعطاءها هذا الاسم يجعل الواحد يشعر بالاحباط والخوف)
لاحظ أننا لا نستخدم علامات التنصيص وذلك ليعرف الـPHP أنها بيانات رقمية قد نستخدمها في عمليات حسابية معقدة ويمكننا تطبيق عمليات حسابية بسيطة عليها إذا كانت حرفية .

```
//هذا عدد فردي
$j=2;
//هذا عدد مزدوج
$h=4.5;
```

العمليات الحسابيه

هي مثل الجمع والطرح والضرب والقسمة وهي مرتبة كالتالي :
أولاً / الأقواس
ثانياً / الضرب ثم القسمه .
ثالثاً / الطرح ثم الجمع

```
<?
Echo 5*2/5;
Echo 5*(2/5) ;
?>
```

مثال آخر :

```
<?
Echo 5-6+9 ;
?>
```

مثال لعملية حسابية نستخدم فيها متغير حرفي

```
<?
$W="2L";
$E= 2;
$F = $W * $E;
echo $W .' ' .$E .' ' .$F;
?>
```

مثال لعملية أخري لكنها لم تعمل وعليك استنباط السبب بنفسك (هاه طل زين) :

```
<?
$W=”L10”;
$E= 2;
$F = $W * $E;
echo $W .’ ‘ .$E .’ ‘ .$F;
?>
```

يمكننا إضافة رقم واحد الى متغير بثلاث طرق متنوعة :
مثال

```
$j++
```

أو

```
$j = $j+1
```

أو

```
$j += 1
```

ويمكننا على ذلك إضافه المتغير إلى نفسه كالتالي :

```
$j += $j
```

أو كالتالي :

```
$j = $j + $j
```

متغيرات النظام

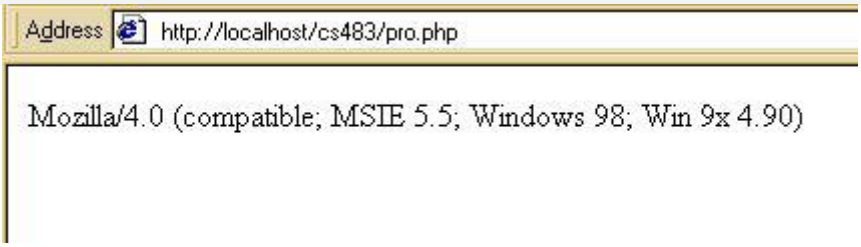
هناك متغيرات يستخدمها النظام يمكنك أن تستعملها ومنها

$HTTP_USER_AGENT

التي تظهر لديك نوع المستعرض الذي يستخدمه العميل

مثال :

```
<?
Echo $HTTP_USER_AGENT ;
?>
```



الثوابت

يمكننا تعريف الثوابت بقول أنها قيم ثابتة لاتتغير ونعرفها عن طريق الدالة define
الثوابت حساسة أيضا لحالة الأحرف

```
<?
Define (“author”, “Majed”);
Echo “author is “ . author ;
?>
```

هناك ثوابت يستخدمها النظام مثل

PHP_OS

التي تقوم بعرض نظام التشغيل الذي يستخدمه السيرفر

مثال :

```
<?
Echo PHP_OS;
?>
```

إذا أردت أن تعرف نوع متغير ما يمكنك استخدام الدالةgettype

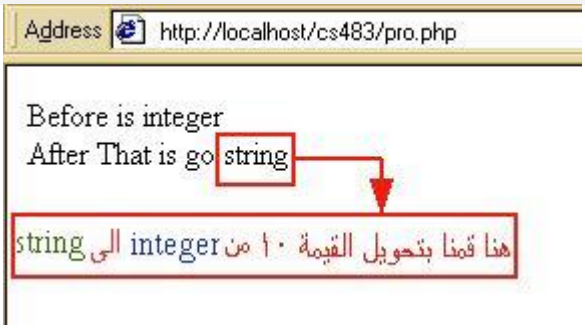
مثال :

```
<?
$n=5;
$l ="hi";
echo "The n Is " . gettype ($n) . "<br>";
echo "The l is " . gettype ($l);
?>
```

إذا أردت تحويل نوع متغير ما يمكنك ذلك باستخدام الدالةsettype :

مثال :

```
<?
$n = 10 ;
echo "Before is " . gettype ($n) ."<br>";
settype ($n,"string");
echo "After That is go " . gettype ($n);
?>
```



الدالة isset

لمعرفة إذا كان المتغير منشأ مسبقاً أم لم يتم انشاؤه وهي لا تتطلب غير اسم المتغير الذي تريد فحص وجوده وتقوم بارجاع القيمة (1) إذا كان المتغير تم انشاؤه ولا ترجع أي قيمة إذا كان المتغير غير منشأ أو موجود .

مثال :

```
<?
$n = “n”;
Echo isset ($n);
?>
```

الدالة unset

تقوم بحذف المتغير إذا كان موجوداً وتحرير الذاكرة منه (لذلك تأكد جيداً قبل استخدام هذه الدالة من اعطاء دمعة الوداع للمتغير المسكين)

```
<?
$n = “n”;
unset ($n);
Echo isset ($n);
?>
```

الداله empty

تقوم بإرجاع القيمة (1) إذا كان المتغير غير منشأ أو أن القيمة التي فيه صفر (0) أو نص فارغ ("") ولاتقوم بإرجاع أي شي اذا كان المتغير منشأ وفيه قيم غير المذكورة .

```
<?
$n = "n";
$a = " ";
$c = "0";
$x = "";
unset ($n);
Echo empty ($n);
Echo "<br>";
Echo empty ($a);
Echo "<br>";
Echo empty ($c);
Echo "<br>";
Echo empty ($x);
?>
```

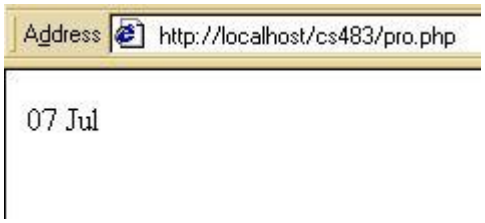
داوال الوقت التاريخ

نستطيع إيجاد الوقت و التاريخ عن طريق دوال في الـPHP من تلك الدوال الدالة

gmdate ()

مثال :

```
<?
Echo gmdate (m);
Echo "\t"; //ما هي الى مسافه بين الناتجين
Echo gmdate (M);
?>
```



لاحظ أن هناك فرق في النتائج مع أننا نستخدم نفس الحرف لكن طريقه العرض تختلف عندما يكون الحرف كبيراً أو صغيراً .

تحتجز الـ php بكثير من الدوال والكلمات المحجوزة التي تقوم بعمليات مختلفة مثل العمليات الحسابية المعقدة والقيام بإيجاد الوقت والتاريخ وإرسال الرسائل البريدية وإيقاف السكربتات لعدة ثواني هذه الدوال ليس مطلوب منك أن تحفظها كما تحفظ اسمك إنما المطلوب منك أن تفهم ماهية عملها واستخدامها في الوقت الذي تراه مناسباً .

يمكنك أيضا عرض اليوم والشهر

مثال

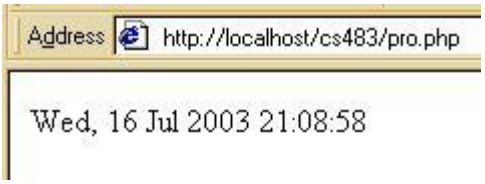
```
<?
Echo gmdate ("M D");
?>
```

لاحظ أننا استخدمنا علامات التنصيص لكي تنجح العملية عندما قمنا باستخدام أكثر من عامل في الدالة

جرب استخدام الكود التالي :

هذا سوف يعرض لك اليوم والتاريخ والساعة

```
<?

Echo gmdate ("D, d M Y H:i:s")

?>
```



النمـــاذج

النماذج في الويب أو صفحات الانترنت عبارة عن استمارات تقوم بتعبئتها ثم عند إرسالها لخادم الويب (السيرفر) يتلقاها برنامج يقوم بإجراء العمليات عليها مثل JavaScript أو ASP أو php (في حالتنا) .

فائدة النماذج

لنقل أنك مثلاً أردت شراء كتاب من الانترنت فإنك في الواقع تحتاج إلى تعبئة استمارة ببياناتك ورقم بطاقة الائتمان وغير ذلك من المعلومات ويتم ذلك عن طريق نموذج (فورم) .

في الواقع أنت تقوم بإختيار الكتاب الذي تريد وتكتب اسمك ورقم هاتفك وصندوق بريدك (ربما) في فراغات أو عن طريق الإشارة الى مجموعة من الخيارات .
يتم تخزين هذه القيم في المتغيرات التي يتم كتابتها في الخاصيه(name) نتكلم عنها في هذا الدرس ويتم ارسالها عند ضغط زر - ارسال البيانات - (submit) الى (البرنامج) الصفحة التي سوف تقوم بمعالجة هذه البيانات (والتي يتم تحديدها في الخاصية ACTION) وإجراء العمليات عليها مثل تخزينها مثلاً في قاعدة البيانات أو ارسالها إلى البريد الالكتروني وذلك عن طريق الـphp

ماذا يعمل العميل في النماذج ؟

إنه باختصار يقوم بتعبئة مربعات نصوص (textbox) ويقوم بوضع علامة صح في مربعات الاختيار (check boxes) أو يقوم بالتصويت أحيانا لشي معين فيختار زر اختيار (ازرار الراديو) .
هذه الأشياء كلها يتم انشاءها بواسطة الـ html ودرسنا لهذا اليوم يناقش كيفيه انشاءها وكيفيه التعامل والحصول على البيانات منها ، بقي علينا كبداية أن نعرف أن هذه الأدوات تنشأ في الواقع بين وسمين من وسوم لغه الـ html وهي الوسمين

```
<form>
</form>
```

خصائص النماذج

يجمع النموذج جميع خصائص المضيف لكننا هنا سنتطرق الى اثنين منهما وهما ACTION و METHOD التي تستخدم بكثرة و مهمة لنا في دروسنا القادمة
اما (ID;CIASS;NAME) فيلزمها تعمق في HTML خاصة عندما ندخل في ACCEPT-CHAR و ENCTYPE وستكون خارج نطاق موضوعنا حالياً وقد نفصلها في دروس قادمة إن شاء الله .

ACTION

وظيفة هذه الخاصية أن تخبر السيرفر مكان الصفحة التي يقوم بارسال معلومات النموذج إليها أوعنوانها أيا كان نوعها ، وطبعاً في حالتنا ستكون الصفحة الثانية هي الصفحة التي تحتوي على سكربت الـphp .
ليس مهما أن تكون الصفحة php فقد تكون html ولكنها تحتوي على كود يختص بالتعامل مع برنامج تفاعلي لصفحات الويب مثل الجافا .
ولانريد أن نخرج عن نطاق الموضوع فدعنا نعطي مثالاً على هذه الخاصية :

```
<FORM  ACTION =”TEST.PHP”>
…..
</FORM>
```

METHOD

هذه الخاصية تقوم بإخبار النموذج طريقة ارسال المعلومات الى الصفحة الهدف وفي الواقع هناك طريقيتين مشهورتين ومعروفتين لارسال المعلومات هما GET وPOST .

```
<FORM ACTION =”test.php” METHOD = “GET”>
```

أو

```
<FORM ACTIN = “test.php” METHOD =”POST”>
```

ملاحظه : في الواقع يوجد اكثر من هذه الطريقيتن لارسال المعلومات وهي
(CONNECT;HEAD;OPTIONS:DELETE:TRACE) وغيرها ولكن لاتستخدم الا بشكل نادر .

دعنا الآن نفصل هاتين الطريقتين بشكل أوسع :

GET

تقوم هذه الخاصية بإخبار مستعرض الانترنت لديك بأن يقوم بإضافة المعلومات التي تمت كتابتها في النموذج إلى متصفح الانترنت لديك وتكون طريقة كتابته كاالتالي :

١ ـ كتابه عنوان الصفحه المصدر .
٢ ـ اتباعها بعلامة استفهام .
٣ ـ كتابة العناوين والقيم .

http://localhost/test.html?name=value

قد تكون النقطيتن الأخيرتين غير مفهومتين بشكل جيد بسبب أنك لم تتعامل مع النماذج من قبل .
لكن الحقيقة أن النموذج يتكون من عناصر (مربع علامة ، مربع نص ، زر اختيار) ولكل من هذه العناصر عنوان خاص بها (name) ولكل منها قيمه خاصه بها (value) .
وهي مشابهة للمتغيرات ويمكن أن يحتوي عنوان الصفحة على أكثر من عنوان (name) واكثر من قيمة (value) ويقوم بالتعريف عنهما باستخدام المعامل (&) .

مثال :

http://localhost/test.html?animal=cat&age=30

تسمى الإضافة التي تظهر بعد علامة الاستفهام (query String) نتيجه الاستعلام الحرفية.
العنوان دائما يكون باللغه الانجليزيه (name) ونعامله كانه اسم متغير من المفترض تعريفه في الصفحه الهدف (التي سنكتبها بالPHP).

قد تحتوي القيم على فراغات او معاملات مثل (+،-،\،#،%)
يقوم المتصفح باستخدام لغه تشفير الصفحات URL ENCODING .
أيضا يستخدم الـ URL ENCODING مع الأحرف العربية أو اللغات الأخري غير الإنجليزية في كتابة الحرف .

URL Encoding

هناك بعض الأحرف لايستطيع المتصفح إضافتها لعنوان الصفحة بصيغتها الحقيقية بل يستخدم لغة التشفير في التعريف عنها وهذه جداول بالرموز الذي يستخدم المتصفح كود بدلا من عرضها بصيغتها الحقيقية

الحرف	شفرته	الحرف	شفرته	الحرف	شفرته
Tab	%09	(	%28	;	%3B
Space	%20	)	%29	<	%3C
!	%21	+	%2B	>	%3E
”	%22	،	%2C	=	%3D
#	%23	.	%2E	?	%3F
@	%40	/	%2F	%	%25
\	%5C	:	%3A	&	%26

لاتقلق فليس عليك أن تحفظ كل هذه العلامات وتشفيراتها بل سيقوم المتصفح بالعمليه كلها بدلا عنك .

POST

في الواقع وظيفتها هي نفس وظيفة الـ get ولكنها لاترسل المعلومات في عنوان صفحة الانترنت بل تقوم وضعها في الـ body التابع لـ http response .
بالإضافه إلى أنه يستطيع ارسال البيانات بكمية أكبر من الـGET .

أيهما تستخدم GET أم POST ؟

قد يكون العيب في الخاصية GET عدم سرية المعلومات التي تقوم بكاتبتها ومن الممكن أن تظهر للشخص الذي يجلس الى جوارك ... خاصة عندما تريد الحفاظ على سرية معلوماتك .
أضف إلى ذلك أنها غير مفيدة في النصوص الكبيرة الحجم .
ولكنها مفيدة في أشياء كثيرة فمثلاً محركات البحث يجب أن تستخدم هذه الخاصية لكي يستطيع المستخدم أن يستخدم عنوان البحث ويحتفظ به لوقت آخر ولا يقوم من جديد بكتابة الكلمة التي يبحث عنها .

أيضا الـPOST مفيدة في إخفاء المعلومات وإحتواء كميات كبيرة من البيانات ولكن لايمكن الاحتفاظ بعنوان الصفحة .... مع ذلك فإنها أيضا ليست جيدة في الحماية بحيث أن أي هاكر خبير يمكنه الحصول على المعلومات إذا لم يكن لها تشفير معين في نقلها ..لكن اذا اردت فعلاً ان تجعلها محمية فيجب عليك استخدام اتصال محمي الى سيرفر محمي .... او مايسمونه (SCURE CONNECTION TO SCURE SERVER) .

أدوات التحكم في النماذج :

في الواقع أن أدوات التحكم عبارة عن مربعات النصوص العاديه (التي يدخل فيها المستخدم اسمه وعنوانه) وازرار الراديو (والتي يقوم المستخدم فيها باختيا شي معين (مثل الوجبه المفضلة لديه او المشروب المفضل اليه) ومربعات الاختيار (التي تتيح للمستخدم أن يختار مايشتهي ويحب من الخيارات المعروضة)
وأيضا القوائم التي تساعدك على اختيار أكثر من شي أو شي واحد .

في أغلب هذه الاشياء يتم استعمال الوسم

<INPUT>

وتلخيص تفصيله كالتالي :

```
<INPUT TYPE= type NAME= name VALUE= value other attribute>
```

الشرح :

1 - TYPE= type
نحدد نوع الكائن إذا كان زر راديو أو مربع نص عادي أو مربعات الاختيار .

2 - NAME= name
تقوم فيها بإعطاء اسم لمتغير يتم حفظ القيمة فيه .

3 - VALUE= value
سيتضح وظيفته أكثر عندما ندرج عليه أمثله إذ أن عمله يختلف من أداة إلى أخرى .

تطبيقات عمليه

سنقوم في هذه التطبيقات بصنع برامج بسيطة تتكون من ملفين ، الملف الاول يحتوي على كودHTML يقوم بتكوين النموذج والملف الثاني يقوم بإستقبال النتائج وطباعتها .

مربعات النصوص (TEXT Box) :

نقوم بعمل ذلك كالتالى :

1 - قم بتشغيل محرر النصوص لديك .

2 - اكتب الكود التالي :

```
<html dir =”rtl”>
<FORM METHOD  = “GET” ACTION = “pro.php”>
ماهي وجبتك المفضلة في الصباح ؟

<br>
<INPUT TYPE =”text” NAME = “food” value=”جبنة ومربي”>
<br>
<INPUT TYPE= submit  VALUE=”إرسال”>
<INPUT TYPE= reset  VALUE=”مسح”>
</form>
</html>
```

3 - قم بحفظ الملف كصفحة HTML . وقم بتسميته (prohtml.html) .

4 - افتح محرر النصوص إذا كنت أغلقته .

5 - اكتب الكود التالي :

```
<?
Echo “وجبتك المفضله والى تموت في حبها هي” . “ “ . $food ;
?>
```

6 - قم بحفظ الملف كـphp . وقم بتسميته pro.php .

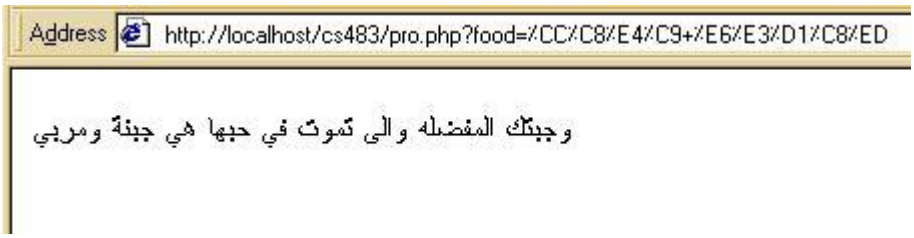
7 - الآن قم بأخذ الملفين وضعهما في مجلد السيرفر لديك .

8 - قم بتشغيل السيرفر واكتب في مستعرض الانترنت لديك

http://localhost/prohtml.html

9 - قم بكتابة وجبتك المفضلة واضغط زر إرسال .

10- ستظهر النتيجة .



لاحظ كيف ظهر العنوان :

http://localhost/pro.php?food=%CC%C8%E4%C9+%E6%E3%D1%C8%ED

الشرح

لقد قمنا في البداية بعمل صفحة تتكون من نص و مربع نص وزر يقوم بعملية إرسال البيانات
قمنا بصناعة بداية النموذج بواسطة الوسم<FORM> وقمنا بتحديد المكان الذي سيتم ارسال البيانات إليه بواسطة

ACTION=“pro.php“

وقمنا بصنع مربع النص بواسطه الوسم INPUT واخترنا الـ

TYPE=“text“

كما قمنا بوضع القيمة الإفتراضية فيه بواسطة القيمة

Value= “جبنه ومربي”

وقمنا بوضع الناتج الذي يضعه المستخدم في مربع النص في المتغير food .

(لاحظ ان تسميه المتغيرات حساسه لحاله الاحرف في PHP واننا لم نقم بوضع $ في صفحه المتغير في كود الـhtml).

وأيضا لقد قمنا بإضافه زر بواسطة

TYPE=SUBMIT

وقمنا بوضع كلمة على الزر وهي كلمة (إرسال)

VALUE = “إرسال”

أيضا قمنا بصنع زر آخر

Type =reset

وقمنا بجعل العبارة التي عليه (مسح)

Value=”مسح”

هناك نوعين من الازرار هي SUBMIT و RESET
1- الـ submit يقوم بإرسال المعلومات .
2- الـ reset يقوم بمسح البيانات في جميع الأدوات في النموذج لإعاده إدخالها من جديد .

بعد ماقمنا بادخال البيانات وضغط زر الارسال قام النموذج بارسال البيانات إلى الصفحة المحددة في الخاصية ACTION وقامت الصفحة المحددة بإستقبال النتائج الموجودة في النموذج وهي نتيجة واحدة في مربع نصوص تم حفظ قيمته في المتغير food .
وقامت بطباعتها بواسطة الدالة echo .

نظرا لاننا استخدمنا الاسلوب GET فقد تم اعطاءنا عنوان الصفحه بالاضافه الى (?) وايضا المعلومات المسجله في المتغيرات والتي تم استخدام الURL ENCODING فيها لانها تستخدم حروف عربيه .

مربعات النصوص الكبيره (text area) طلبات اكبر للطعام الشهي !
إذا كنت تريد أن تكتب رسالة متعددة الأسطر فإنك تحتاج إلى أداة تحكم تختلف تماماً عن مربع النص العادي وهي مربعات النصوص الكبيرة التي يمكنك فيها من إدخال نصوص كبيرة الحجم ومتعددة الاسطر .

تستخدم هذه الأداة وسم فتح ووسم إغلاق

<TEXTAREA>
</TEXTAREA>

ويمكنك تحديد حجمها بواسطه تحديد الصفوف بالخاصية rows والأعمدة بالخاصية cols .

تمرين عملي

١ - قم بفتح محرر النصوص لديك
٢ - قم بكتابة الكود التالي :

```
<html dir=”rtl” >
<FORM ACTION = “TAREA.PHP”  METHOD=”POST”>
ما هي وجبتك المفضلة ؟
<br>
<TEXTAREA NAME = “food” ROWS=”10” COLS =”50” >
جبنة
مربي
مكرونة
بيف برغر
سمبوسة
معصوب
مطبق
معجنات
ماخلص لو قعدت اكتب هاها
</TEXTAREA>
<br>
<INPUT TYPE = SUBMIT VALUE =”قم بإرسال الطلبات إلى الجرسون”>
</FORM>
</html>
```

٣ - قم بحفظ الملف باسم TAREA.html .

٤ - الآن قم بفتح ملف جديد في محرر النصوص .
٥ - قم بكتابة الكود التالي :

```
<html dir=”rtl”>
وجبتك المفضلة هي :
<br>
<?
Echo $food;
?>
</html>
```

٦ - قم بحفظ الملف باسم tarea.php

٧ - قم بوضعهما في مجلد السيرفر لديك .

٨ - قم بتشغيل البرنامج .

http://localhost/tarea.html

٩ - قم بضغط الزر لارسال البيانات .
١٠ شاهد النتيجه.

الشرح

لانضيف شيئاً على قولنا هنا سوى أننا نريدك أن تلاحظ كيف جهزنا القيمة الإفتراضية بكتابة نصوص بين وسومات الـ textarea وأيضا أننا استخدمنا الأسلوب POST في ارسال البيانات مما جعلها لاتظهر في شريط العنوان .
وأن الـNAME تحدد اسم المتغير التي ستذهب إليه القيمة واسم المتغير في الكود لايحتوى على $ لأنه كود HTML وليس PHP .

مربعات الاختيار (Check Box) اكثر من خيار في وقت واحد !
في الواقع قد نرى مربعات الاختيار في صفحات الويب عندما نريد الاشتراك في موقع معين لرؤيه محتوياته أو عندما نريد تسجيل بريد إلكتروني أو حجز مساحة عند موقع .
وفائدتها هي إتاحه فرصة للمستخدم لتحديد أنواع الأشياء التي يريد أن يشترك فيها مثلاً أو إتاحه فرصة له لقبول إتفاقية أو غير ذلك أو رفض الجميع أو قبول الجميع .

يمكننا صنع مربع العلامة بواسطه الوسم INPUT

```
<INPUT TYPE = ”CHECKBOX” NAME = “majed” value= ”Majed” checked>
```

نقوم بتحديد نوع الأداة بأنها مربع علامة في هذا الجزء

```
TYPE = ”CHECKBOX”
```

نقوم بتحديد اسم المتغير في هذا الجزء

```
NAME = “majed”
```

ونقوم بتحديد القيمه التي يتم وضعها في المتغير اذا قام المستخدم باختيار مربع العلامه في هذا الجزء :

```
value= ”Majed”
```

اذا لم تقم بوضع الخيار value فستكون القيمه الافتراضيه هي on عند اختيار المستخدم مربع العلامه وستكون فراغ اذا لم يقم المستخدم باختيار المربع.

ونقوم بوضع القيمه الإفتراضية بإضافه الكلمة checked فإذا تم وضع هذه الكلمة يكون مربع العلامة مختار تلقائياً أما إذا لم نتكبها فسيكون بدون علامة الاختيار .

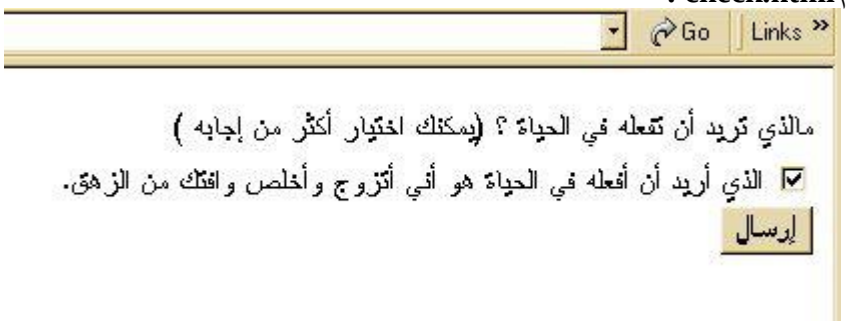
```
Checked
```

تطبيق عملي (1) :

١ ـ قم بفتح المفكرة وقم بكتابة الكود التالي :

```
<html dir=”rtl”>
<FORM ACTION=”CHECK.PHP” METHOD = “POST”>
مالذي تريد أن تفعله في الحياة ؟ (يمكنك اختيار أكثر من إجابه )
<br>
<INPUT TYPE=”CHECKBOX” NAME = “WIFE” CHECKED>
الذي أريد أن أفعله في الحياة هو أني أتزوج وأخلص وافتك من الزهق.
<br>
<input type= submit value = “إرسال”>
</FORM>
</html>
```

٢ ـ قم بحفظ الملف باسم **check.html** .



٣ ـ قم بفتح ملف جديد في المفكرة وقم بكتابة التالى :

```
<?
Echo $WIFE ;
?>
```

٤ ـ قم بحفظ الملف باسم **check.php** .
٥ ـ قم بنقل الملفين الى مجلد السيرفر .
٦ ـ اكتب في المتصفح

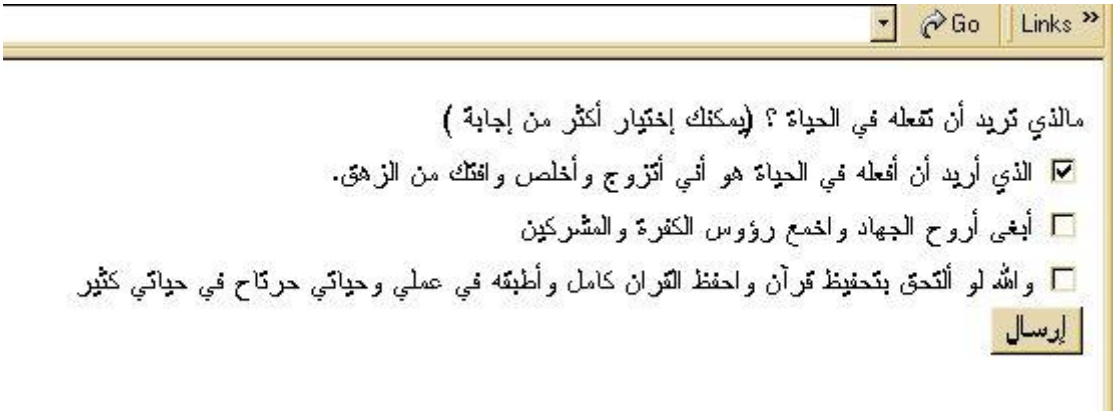
http://localhost/check.html

٧ ـ النتيجه

تطبيق عملي (2) :

١ - افتح المفكرة واكتب الكود التالي وقم بحفظه في ملف جديد باسم check2.html

```
<html dir=”rtl”>
<FORM ACTION=”CHECK2.PHP” METHOD = “POST”>
مالذي تريد أن تفعله في الحياة ؟ (يمكنك إختيار أكثر من إجابة )
<br>
<INPUT TYPE=”CHECKBOX” NAME = “WIFE” value= “زوجة” CHECKED>
الذي أريد أن أفعله في الحياة هو أني أتزوج وأخلص وافتك من الزهق.
<br>
<INPUT TYPE=”CHECKBOX” NAME = “jihad” value= “جهاد” >
أبغى أروح الجهاد واخمع رؤوس الكفرة والمشركين
<br>
<INPUT TYPE=”CHECKBOX” NAME = “qran” value= “قران” CHECKED>
والله لو ألتحق بتحفيظ قرآن واحفظ القران كامل وأطبقه في عملي وحياتي حرتاح في حياتي كثير
<br>
<input type= submit value = “إرسال”>
</FORM>
</html>
```



٢ - قم بفتح ملف جديد وقم بوضع الكود التالى فيه :

```
<html dir = “rtl”>
<?
Echo $WIFE . “ “ . $jihad . “ “ . $qran ;
?>
</html>
```

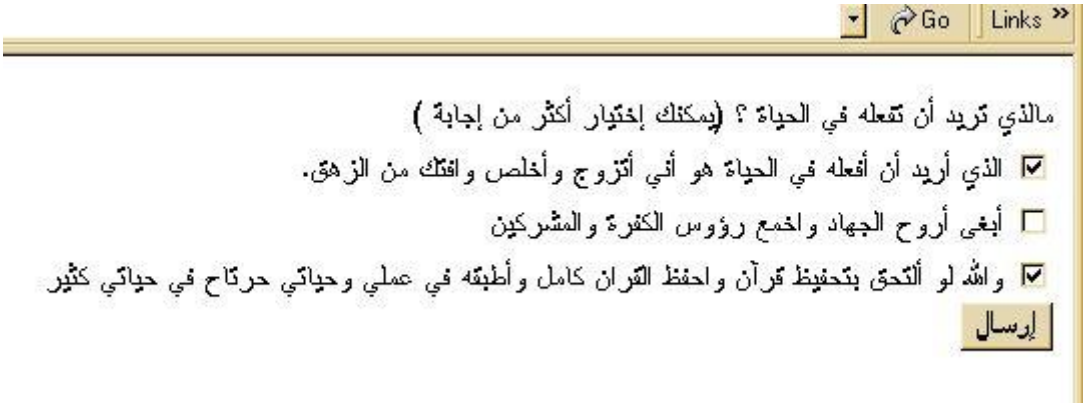
٣ - قم بحفظه باسم check2.php
٤ - قم بتشغيل الملف .
٥ - النتيجه

تطبيق عملي (3)

١ - افتح محرر النصوص واكتب الكود التالي :

```
<html dir=”rtl”>
<FORM ACTION=”CHECK3.PHP” METHOD = “POST”>
مالذي تريد أن تفعله في الحياة ؟ (يمكنك إختيار أكثر من إجابة )
<br>
<INPUT TYPE=”CHECKBOX” NAME = “alswalif[]” value= “زوجة” CHECKED>
الذي أريد أن أفعله في الحياة هو أني أتزوج وأخلص وافتك من الزهق.
<br>
<INPUT TYPE=”CHECKBOX” NAME = “alswalif[]” value= “جهاد” >
أبغى أروح الجهاد واخمع رؤوس الكفرة والمشركين
<br>
<INPUT TYPE=”CHECKBOX” NAME = “alswalif[]” value= “قران” CHECKED>
والله لو ألتحق بتحفيظ قرآن واحفظ القران كامل وأطبقه في عملي وحياتي حرتاح في حياتي كثير
<br>
<input type= submit value = “إرسال”>
</FORM>
</html>
```

٢ - قم بحفظه باسم check3.html



٣ - افتح محرر النصوص من جديد واكتب الكود التالي :

```
<html dir=”rtl”>
<?
Echo “$alswalif[0] <br>” ;
Echo “$alswalif[1] <br>” ;
Echo “$alswalif[2] <br>” ;
?>
</html>
```

٤ - قم بحفظه باسم check3.php وقم بنقلهما الى ملف السيرفر .

٥ - قم بتشغيل البرنامج

http://localhost/check.html

٦ - قم بضغط زر ارسال وانظر للنتيجه

الشرح

في الواقع لقد قمنا بتطبيق ثلاث تمارين التمرين الاول أردنا لفت النظر إلى أننا قمنا بعدم استخدام value للمتغير وتم إعطاء القيمة on عند اختيار المستخدم مربع العلامة بالإضافة أن مربع العلامة كان مختاراً بسبب وضعنا الخاصية CHECKED ولكن التمرين غير عملي وغير جيد بدون وضع قيم VALUE عند وضعنا لأكثر من مربع اختيار لذلك فقد قمنا باضافه قيم يتم وضعها في المتغيرات عند اختيار المستخدم لها كما في التمرين الثاني وأردنا لفت النظر في التمرين الى شي يسمي بالمصفوفات فإذا أردنا مثلا أن نجعل اسم المتغير متشابها واجراء عمليات تكون أسرع عليه نستخدم المصفوفات ولن نتطرق إلى المصفوفات حالياً ولكن أردنا لفت نظرك فقط وسنقوم بالتكلم عن المصفوفات بالتفصيل في الدروس القادمه باذن الله هي والتكرارات بعد التكلم عن العبارات الشرطية في الـPHP .

ازرار الراديو(RADIO BUTTONS) (اختر المشروب المفضل !)
ماهو اختيارك المفضل ؟ علما بانه لايمكنك اختيار اكثر من خيار واحد !!

في الواقع إن زر الراديو يتيح لك أن تختار شي واحد من بين عدة اختيارات ونراه كثيراً عند اتفاقيات البرامج حيث يعطيك فرصه إما بقبول الإتفاقية أو رفضها ويكون واحد من الاختيارين محددا (وهو خيار الرفض!) .

يتم استخدام ازرار الراديو باستخدام العبارة<INPUT> كالتالى :

```
<INPUT TYPE = ”radio” NAME = “name” value= ”value” checked>
```

نقوم بتحديد نوع الكائن بانه زر راديو في هذا الجزء :

```
TYPE = ”radio”
```

نقوم بتحديد اسم المتغير في هذا الجزء :

```
NAME = “name”
```

نقوم بتحديد القيمة التي ستكون في المتغير هنا :

```
value= ”value”
```

في الواقع مع ازار الراديو نقوم بجعل اسم المتغير name هو نفسه والقيم مختلفة value لكل سؤال . وإذا لم نقم بوضع قيمة فسيقوم PHP بوضع القيمة on للمتغير .

تطبيق عملي :

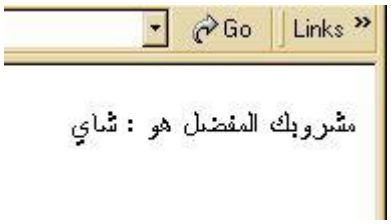
١ ـ قم بتشغيل محرر النصوص لديك واكتب الكود التالي وقم بحفظه في ملف اسمه radio.html .

```
<html dir=”rtl”>
<form action = radio.php method = “post”>
ماهو مشروبك المفضل ؟
<br>
<br>
<INPUT TYPE = ”radio” NAME = “mshroob” value= ”شاي” checked>
شاي
<br>
<INPUT TYPE = ”radio” NAME = “mshroob” value= ”قهوة” >
قهوة
<br>
<INPUT TYPE = submit  value= ”إرسال” >

</form>
</html>
```

٢ - قم بفتح محرر النصوص واكتب الكود التالي وقم بخفظه باسمradio.php


```
<html dir = "rtl">
<?
echo "مشروبك المفضل هو :" . " " . $mshroob;
?>
</html>
```

3 - قم باختيار المشروب المفضل واختر إرسال .



الشرح :

في الواقع لقد قمنا بصنع أزرار راديو ولقد قمنا بوضع قيمة لكل زر تكون تابعة للعباره التي بجوار الزر . ولقد قمنا بوضع عبارة checked لكي ترى كيف أن الأداة التي تحتوي على العبارة تكون محددة تلقائيا ولاحظ أن العبارة التي تكون بجانب الزر تكون موجودة أسفل كود الزر مثل :

```
<INPUT TYPE = ”radio” NAME = “mshroob” value= ”شاي” checked>
شاي
```

العبارة هي الملونة باللون الأحمر .
وأيضا لاحظ أننا استخدمنا متغيراً واحداً فقط لجميع الإختيارات بحيث أن جميع الأزرار قيمتها تعود إلى هذا المتغير .

القوائم (Lists Or drop down menus) اختر مواصفات زوجتك للمستقبل واسمها :
تستخدم القوائم في الـ html بشكل مختلف قليلاً عن الأداوت السابقة إذ أننا نستخدم وسمين من وسوم لغة html وهما :
<select> لنقوم بإنشاء القائمة و <OPTION> ونستخدم الخاصية MULTIPLE إذا كنا نريد إتاحه الفرصه للمستخدم أن يختار أكثر من قيمة ونقوم بوضع القيمة التي يختارها المستخدم في متغير بواسطه الخاصية NAME أو في مصفوفة متغيرات (وسيتضح مفهوم المصفوفات لديك جيدا في درس المصفوفات بإذن الله .

تطبيق عملي :

١ - قم بفتح محرر النصوص لديك واكتب الكود التالي واحفظه في ملف باسم lists.html :

```
<html dir=”rtl”>
<form action = “lists.php” method = “post”>
ماذا تريد ان يكون اسم زوجة المستقبل(لغير المتزوجين ) ؟
<br>
<select name = “wife” >
<option> هناء </option>
<option>جمانة </option>
<option>رزان </option>
<option> سحـــر </option>
<option>سارة </option>
<option>سمية </option>
<option>روان </option>
<option>دلال </option>
<option> اسم اخر </option>
</select>
<BR>
ماذا تريد أن تكون مواصفاتها ؟
<Br>
<select name=”dis[]” multiple>
<option>جميلة</option>
<option>متدينة</option>
<option>شقراء </option>
<option>جعداء الشعر</option>
<option>سوداء </option>
<option>سمراء </option>
<option>بيضاء </option>
</select>
<br>
<INPUT TYPE=SUBMIT VALUE=”إرســال”>
</html>
```

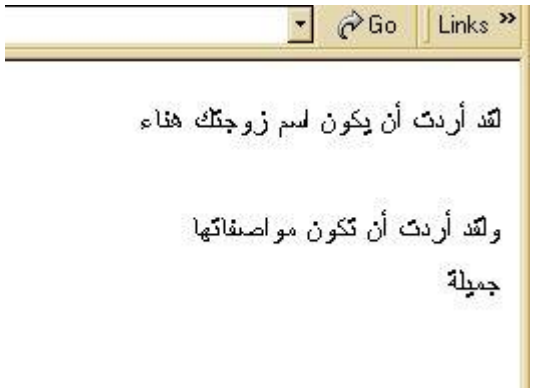
٢ - قم بفتح ملف جديد واكتب فيه الكود التالي وقم بحفظه باسم**lists.php** :

```
<html dir=”rtl”>
<?
Echo “لقد أردت أن يكون اسم زوجتك ” . “ “ . $wife ;
Echo “<br><br>”;
Echo “ولقد أردت أن تكون مواصفاتها“;

Echo “<br><br>”;
Echo “$dis[0] <br>”;
Echo “$dis[1] <br>”;
Echo “$dis[2] <br>”;
Echo “$dis[3] <br>”;
Echo “$dis[4] <br>” ;
Echo “$dis[5] <br>”;
Echo “$dis[6] <br>”;
?>
</html>
```

قم بتشغيل البرنامج

http://localhost/lists.html

واختر ماتريد ثم اضغط زر ارسال



الشرح :

لقد قمنا بصناعة قائمة تسمح بإختيار قيمة واحدة منها ثم تذهب هذه القيمة الى المتغير wife وصنعنا قائمة ثانيه تسمح بإختيار أكثر من عنصر واحد وقمنا بوضع هذه القيم في مصفوفه متغيرات (سيتضح معني المصفوفات في دروس قادمه ان شاء الله) .

الاداة الخفيه (والمعلومات السريه!) (hidden control)

هناك بعض الأوقات تحتاج فيها إلى إرسال بعض المعلومات من صفحة ويب الى صفحة ويب اخرى عن طريق النماذج وفي نفس الوقت أنت لاتريد المستخدم أن يقوم برؤية هذه المعلومات .

في الواقع هناك أداة تساعدك على اخفاء هذه المعلومات على المستخدم يسمونها بحقل النموذج المخفي أو الأداة الخفية (hidden form field or hidden control) .

هذه الأداة تلعب دوراً مختلفاً ومتميزاً عن بقية الأدوات وهي اخفاء المعلومات التي تم ادخالها كما شرحنا في السابق وهي مفيدة جدا مع النماذج المصنوع بواسطة الـ PHP إذ أنها تسمح لنا أيضا بان تكون المعلومات المخفيه هي متغيرات PHP .

يتم صنع هذه الحقول المخفية كالتالى :

```
”>الرسالة السرية<INPUT TYPE=HIDDEN NAME =hidden1 VALUE=”
```

نقوم بوضع HIDDEN لكي يعرف المتصفح أن هذه المعلومات خفية (لاتظهر للمستخدم) ونضع اسما للمتغير الذي يقوم بالاحتفاظ بالمعلومات والذي يتخزن اسمه في الـ NAME ونقوم بوضع المعلومات التي نريد اخفاءها في الـVALUE .

نستطيع الاستفاده أيضا منها عن طريق الـ php وذلك عن طريق كتابة كود الـHTML بواسطة الأمر ()echo في الـPHP كما في المثال التالى :

```
<?
”; هذه العبارة لن تظهر$msg1= “
echo “<form>” ;
echo “<input type=hidden name =secret value= ‘$msg1’>”;
echo “<input type=submit>”;
echo “</form>”;
?>
```

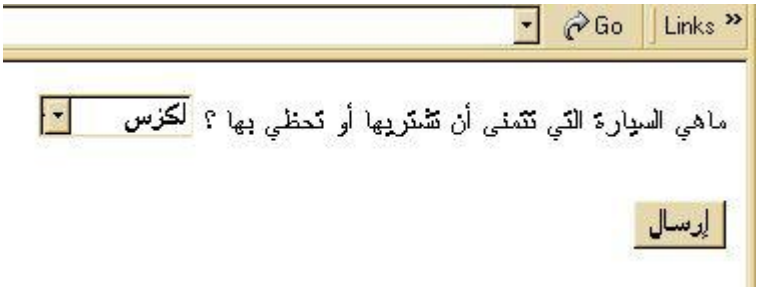
هذا الكود الذي تراه عبارة عن كود HTML تم كتابته بالـPHP عن طريق الامر ()echo ولقد استطعنا تخزين قيمة متغير php ($msg) في متغير html (secret) .

تطبيق عملي :

1 - افتح محرر النصوص واكتب الكود التالي واحفظه باسم hid.php :

```
<html dir=”rtl”>
<head></head>
<body>
<?
”;لكزس$car1= “
”;ماكسيما$car2= “
”;لاندكرزور$car3=”
Echo “<form method =get action=’hid2.php’>”;
”;ماهي السيارة التي تتمنى أن تشتريها أو تحظي بها ؟Echo “
Echo “
<select name= ‘favcar’>
<option>$car1</option>
<option>$car2</option>
<option>$car3</option>
</select><br><br>
<input type =hidden name = hid1 value=’$car1’>
<input type =hidden name = hid2 value=’$car2’>
<input type =hidden name = hid3 value=’$car3’>
’>إرسال<input type = submit  value=’
</form>”;
?>
</body>
</html>
```

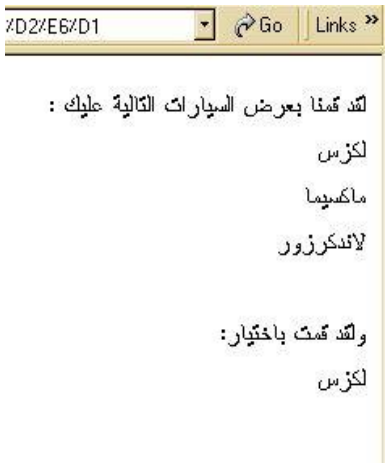
٣ - افتح محرر النصوص واكتب الكود التالي واحفظه باسم hid2.php




```
<html dir=”rtl”>
<head></head>
<body>
<?
<br>”;: لقد قمنا بعرض السيارات التالية عليكEcho “
Echo “$hid1<br>”;
Echo “$hid2<br>”;
Echo “$hid3<br>”;
<br>”;:ولقد قمت باختيارEcho “<br>
Echo $favcar;
?>
</body>
</html>
```

3- قمت بنقل الملفين الى مجلد السيرفر ثم قم بتشغيل السكربت :

http://localhost/hid.php



الشرح :

لقد قمنا بعمل نموذج بسكربت الـ php لاحظ أننا استخدمنا الـ(') بدلاً من (") كما كنا نعمل في الـ html وذلك لاننا قلنا سابق أن القيم الحرفية ..... (راجع درس المتغيرات) ولقد قمنا بإدراج قيم متغيرات الـphp في كود الـhtml مما يوفر علينا الكثير من إعاده الكتابة (في حال كان النص المستخدم طويلاً) .

اقرا المثال أكثر من مر وسيتضح لك المقال أكثر باذن الله .

استخدام حقل كلمات السر (Password fields)

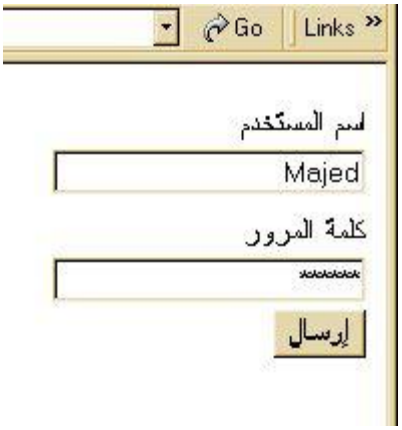
لكي تجعل المعلومات أكثر حماية من التعرض إلى السرقة أو غير ذلك يمكنك استخدام حقول كلمات السر الذي هو عبارة عن مربع نص بسيط يقوم بإظهار النص على شكل نجوم ** في حال كان الجهاز يستخدم على يد أكثر من شخص فان هذه الطريقة جيدة قليلاً في أن لايري شخص معلومات الآخر السرية .**

في الواقع مع ذلك فإنك لاتكون قد اديت حماية إذا كان الاسلوب المستخدم في ارسال بيانات المستخدم هو الاسلوب get إلا إذا كنت تستخدم تشفير البيانات ويكون أكثر جودة اذا استخدمت الاسلوب post وايضا لن يكون محمياً من الهاكر إذا لم تكن تستخدم SSL (Secure Socket Layer) لكي تقوم بتنشيط تشفير البيانات .

تطبيق عملي

قم بفتح محرر النصوص لديك واكتب الكود التالي واحفظه باسم pass.php

```
<html dir=”rtl”>
<body>
<form method=post action=”pass1.php”>
اسم المستخدم
<br>
>"user"<input type="text" name =
<br>
كلمة المرور
<br>
>"pass"<input type="password" name =
<br>
’>إرسال<input type = submit  value=’
</form>
<body>
</html>
```



قم بفتح محرر النصوص لديك واكتب الكود التالي واحفظه باسم pass1.php

```
<?
”; : اسم المستخدم هوEcho “
Echo “<br>$user<br>”;
”;: وكلمه المرور هيEcho “
Echo “<br><br>$pass”
<?
```

قم بنقل الملفين الى مجلد السيرفر لديك

قم بتشغيل البرنامج ولاحظ النتيجه .

ارسال البريد الالكتروني بواسطه الـphp :

البريد الإلكتروني هو الحياة التي تنبض بها السكربتات فمثلاً هناك سكربتات ارسال بريد الى صاحب الموقع تخبره بشي معين أو ملحوظة أو غير ذلك ويمكن استخدامها في أكثر من مجال .

والدالة التي تستخدم في ذلك هي الدالة mail()

```
mail("$to", "$sub", "$msg”,”From:$you”);
```

وتقوم بوضع بريد الذي ستصله الرسالة في الخانة $to وموضوع الرسالة في الخانة $sub والرسالة في الخانة $msg وبريدك أنت أو بريد المرسل في الخانة $you .

تطبيق عملي

قم بكتابة الكود التالي واحفظه في ملف باسم mail.html

```
<html dir=rtl>
<head>
</title>برنامج إرسال بريد  <title>
</head>
<body>
<form action="mail.php" method="post">
عنوان المرسل
<br>
<input type="text" name = "you">
<br>
عنوان المستقبل
<br>
<input type="text" name = "to">
<br>
موضوع الرسالة
<br>
<input type="text" name = "sub">
<br>
الرساله
<br>
<textarea rows=10 cols=20 name = "msg" >
</textarea>
<br>
">إرسال البريد الالكتروني<input type="submit" value = "
</form>
</body>
</html>
```

قم بانشاء ملف اخر وقم بكتابة الكود التالي وقم بإعطاءه الاسم mail.php .

```
<?
mail("$to", "$sub", "$msg","From:$you");
?>
```

قم بوضع الملفين في مجلد السيرفر وقم بتشغيل البرنامج واملا البيانات واضغط زر الارسال وستري ان الرساله تم ارسالها بنجاح .

الأوامر الشرطية

لقد أخذنا في الدروس السابقة فكرة عن المتغيرات وكيفية تعامل البيانات مع النماذج ...في هذا الدرس سنتعلم كيفية التحكم بالكود بمعني تنفيذ سطر معين من الكود عند حصول شرط معين وعند عدم حصوله نتجاهل السطر ونتجه الى السطر الذي يليه ..هذا يمنحنا تحكماً أكبر بالكود ويجعلنا نستخدم قرارات وتنفيذ أشياء ممتازة وبرامج رائعة بالـPHP .

دعنا نعطيك فكرة من حياتنا اليومية ....

تقوم في الصباح وتريد أن تحضر فطورك الذي يتكون من التالي :

عسل
جبنة
خبز
شاي

ستقوم بالذهاب إلى الثلاجة ثم تقوم بالبحث عن الأشياء التي يتكون منها فطورك ، فإذا لم تجد ما تريد تستعد للذهاب إلى المركز التجاري لشرائه حاجتك ، تذهب إلى المطبخ وتتأكد مره أخرى وتبحث عن المؤونة التي يحتاجها البيت بشكل عام .

١ - تبحث عن جبنة وإذا لم تجدها تنتقل إلى الخطوة 3 .
٢ - إذا وجدت جبنة فإنك تبحث عن العسل فإذا وجدته تنتقل الى الخطوة 4 ، واذا لم تجده تنتقل الى الخطوة 5 .
٣ - تقوم بكتابتها في ورقة جانبية وتقوم بالبحث عن العسل .
٤ - تتجهز للذهاب إلى المركز التجاري .
٥ - تكتبه في ورقه جانبيه ثم تتجهز للذهاب إلى المركز التجاري .

هل لاحظت انك كنت تقوم بالبحث عن أشياء معينة فاذا وجدتها (true) قمت بالبحث عن التي تليها وإذا لم تجدها (false) تقوم بتسجيلها في قائمة المشتريات لديك .

القيم المنطقية والدوال الشرطية

في الواقع لقد تكلمنا عن المتغيرات سابقاً وذكرنا بأن هناك متغيرات منطقية (قيمتها إما صحيح إم خطأ) ولم نقم بشرحها ، وهذا الدرس سيتولي شرحها وإعطاء أمثلة على كيفية التعامل معها .

العبارة IF

IF condition is true(إذا كان الشرط صحيحاً)

{

excute this code(قم بتنفيذ هذا الكود)

}

إن الدالة IF معروفة تقريباً في جميع لغات البرمجة ...حيث أنها تقوم بعملية التحقق من شي معين وتنفيذ بعض الأشياء إذا كان الشرط صحيحا (true) والقيام بتنفيذ أشياء أخرى أذا لم يكن صحيحا ....

سيقوم الـPHP بتنفيذ الكود التي بين { و } فقط اذا كان الشرط صحيحاً .

أما إذا لم يكن صحيحاً فسيقوم بتجاوزه وتنفيذ الكود الذي يليه .

ويمكنك أيضا أن تقوم بجعلها بسطر واحد ولا تستخدم الأقواس بل تكتب الأمر مباشرة :

```
IF condition is true excute function;
```

لاحظ أنه لابد من استخدام { و } إذا كان الكود يتكون من عدة أسطر أما إذا كان يتكون من سطر واحد فلا داعي لاستخدامها .

فالمثالين التاليين كلهما صحيحين

مثال(1)

```
<?
$S=10
IF ($S=10) echo 11;
?>
```

مثال (2)

```
<?
$S=10
IF ($S=10){
 echo 11;
}
?>
```

لنتخيل مثلاً أن الجو ممطر وسنقوم بإعطاء المطر متغيراً ونسميه rain ونقوم بإعطاء المظلة اسم متغير آخر ونسميه umbrella وسنقوم بإفتراض أن هناك أمر في الـ php يسمي go out حسناً الآن الكود الذي نريد أن نقوم بكتابته هو :

```
If $rain = true
{
$umberrlla = true
}
go out();
```

فائدة هذا الكود هو أن تأمر الـ PHP بحمل المظلة ($umberrlla=true) معه إذا كان الجو ممطراً ($rain=true)وإذا لم يكن ممطراً ولم يتحقق الشرط فإنه سيخرج إلى النزهه بدون أي مظلة .

طبعا ليس هناك دالة تقوم بذلك إنما قمنا بذلك من أجل التوضيح للمستخدم هيكلية عمل الدالة بشكل عام .

مقدمه الى القيم المنطقيه (Boolean Values)

القيم المنطقية ترمز إلى الأشياء التي لاتحتمل أكثر من احتمالين وهما إما صح وإما خطأ ، وهي نوع جديد من القيم غير التي كنت نعرفها سابقا (مثل الرقميه والنصيه) .

مثال

```
<?
$variable=true;
echo "$variable";
?>
```

لو قمت برؤيه النتيجة ستجد أنه يطبع الرقم واحد وهو قيمة المتغير إذا كان صحيحاً ، أما أذا كان خطأ أو غير صحيح فقيمته ستكون(0) .

المعاملات المنطقية

لقد أخذنا المعاملات الرياضية فيما سبق بشيء من التفصيل (+،-،/،*) والان سنأخذ شيئاً جديداً من المعاملات وهي المعاملات المنطقية التي تساعدنا في صناعة الشروط والتقييدات على شي معين وتعطينا تحكما أكبر في الكود .

المعاملات : > و <

من المفترض أن تكون متآلفاً مع علامتي الأكبر من والأصغر من في الرياضيات التي تتعلمها في المدرسة مما يجعل فهم هذا الأمثله بسيطاً .

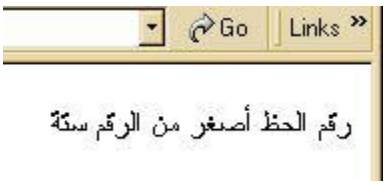
```
<?
If (6>5)
{
"; الرقم ستة أكبر من الرقم خمسةecho "
}
Echo "<br>end";
?>
```

سيقوم الـPHP في مثالنا هذا بفحص الشرط (6>5) فإذا كان صحيحاً (true) سيقوم بطباعة السطر (الرقم سته اكبر من الرقم خمسه) ثم يقوم بطباعة end ، وإذا لم يكن صحيحاً فسيقوم بتجاهل الكود وطباعة (end) فقط .

يمكننا أيضا استعمالها في المقارنة بين متغير ورقم أو بين متغير وثابت (constant) أو العكس أو المقارنة بين متغيرين .

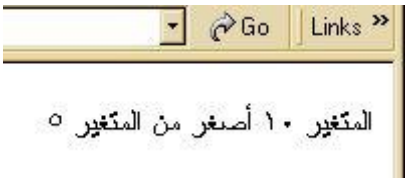
مثال (1)

```
<html dir ="rtl">
<?
$LuckeyNumber = 5;
If ($LuckeyNumber<6)
{
"); رقم الحظ أصغر من الرقم ستةecho ("
}
?>
```



مثال (2)

```
<html dir=”rtl”>
<?
$f=5;
$r=10;
If ($f >$r)
{
 $r";المتغير $f أكبر من المتغيرecho "
}
else{
 $f";المتغير $r أصغر من المتغيرecho "
  }
?>
</html>
```



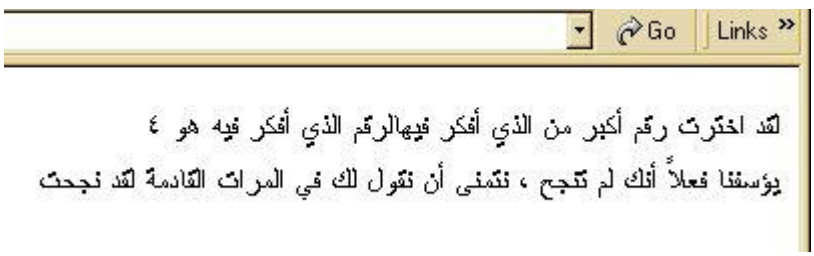
تطبيق عملي :

قم بتشغيل محرر النصوص واكتب الكود التالي واحفظه باسم thegame.php

```
<html dir = "rtl">
<body>
<form method =get action="game.php">
ماهو الرقم الذي أفكر به الآن والذي هو بين 1 و10 ؟
<br>
<input type="text" name="number">
<br>
<br>
<input type =submit>
</form>
</body>
</html>
```

قم بفتح محرر النصوص لديك من جديد واكتب الكود التالي واحفظه باسم game.php

```
<html dir=”rtl”>
<body>
<?
$num = rand (1,10);
if ($number>$num)
        {
” ; لقد اخترت رقم أكبر من الذي أفكر فيهecho “
“; الرقم الذي أفكر فيه هوEcho “
Echo $num;
“ ; يؤسفنا فعلاً أنك لم تنجح ، نتمنى أن نقول لك في المرات القادمةEcho “<br>”.“
}
 if ($number<$num)
        {
” ; لقد اخترت رقم أصغر من الذي أفكر فيهecho “
$num”; الرقم الذي كان في مخيلتي هوEcho “
“; يؤسفنا فعلاً أنك لم تنجح ، نتمنى أن نقول لك في المرات القادمةEcho “<br>”. “
}
?>
لقد نجحت
</body>
</html>
```



شرح التطبيق :

الداله rand

تقوم هذه الدالة باختيار رقم عشوائي من بين رقمين يتم اعطاءها إياها الرقم الاول(x) هو الأصغر والرقم الثاني هو الأكبر(y)

```
Rand (x,y);
```

يمكنك حفظ القيمة التي تقوم بإخراجها هذه الدالة في متغير مباشرة

مثال

```
$Num = rand (5.57);
```

وهذا يوضح ماقمنا به في الكود

```
$num=rand(1,10);
```

لقد قمنا باختيار قيمة عشوائية ثم قمنا بمقارنتها مع القيمة التي تم إدخالها من قبل المستخدم فإذا كانت القيمة التي أدخلها المستخدم أكبر من قيمة العدد العشوائي أخبرناه بأن الرقم الذي أدخله أكبر من الرقم الصحيح ... وهذا ماتجده جليا في الأسطر التالية :

```
if ($number>$num)
        {
" ; لقد اخترت رقم أكبر من الذي أفكر فيهecho "
"; الرقم الذي أفكر فيه هوEcho "
Echo $num;
" ; يؤسفنا فعلاً أنك لم تنجح ، نتمنى أن نقول لك في المرات القادمةEcho "<br>"."
}
```

فإذا لم ينطبق الشرط وكان الرقم الذي اختاره المستخدم أصغر من الرقم العشوائي فإنه يترك الشرط الأول ويتجه الى الشرط الثاني ويطبق الأوامر التي فيه والتي تقوم باخباره بأن الرقم الذي قام باختياره أصغر من الرقم المطلوب ، وهذا ماتجده جليا في الأسطر التالية :

```
if ($number<$num)
        {
" ; ”لقد اخترت رقم أصغر من الذي أفكر فيهecho “
$num”; الرقم الذي كان في مخيلتي هو Echo “
“; يؤسفنا فعلاً أنك لم تنجح ، نتمنى أن نقول لك في المرات القادمة Echo “<br>”. “
}
```

فإذا لم يتطبق الشرطين فإنه يتركهما ويكتب الكلمة (لقد نجحت) بدون أي كلمات آخرى مثلما كنا نكتب الكلمه (يؤسفنا فعلاً أنك لم تنجح ، نتمنى أن نقول لك في المرات القادمة) قبل كلمة (لقد نجحت) ، أتمني أنك قد فهمت جيداً ما أقول .... وتظهر هذه العبارة جلية في الأسطر التالية :

```
?>
لقد نجحت
</body>
</html>
```

على هذا نكون قد صنعنا لعبة كاملة تقوم بإخبار المستخدم عند نجاحه او خسارته .

معاملات المساواة : == و ===

لقد قمنا باستخدام علامة المساواة الفردية سابقاً في تخزين قيمة في متغير وهانحن نأخذ نوعاً من علامات المساواة وهو علامة الساواة المزدوجة (==) وعلامة المساواة المضاعفة (===) .

لقد كنا نستخدم علامة المساواة الفردية او العادية في تخزين القيم في المتغيرات .

مثال :

```
<?
$m=12;
?>
```

ولكن العلامات التي نتكلم عنها الآن تستخدم في تحديد إذا ماكانت قيمة معينة تساوي قيمة اخري .

مثال :

```
<?
$m=”11”;
$u=11;
If ($m==$u)
{
”; القيم متساويةEcho “
}
?>
```

القيم متساوية

لاحظ أن $m متغير حرفي وان $u متغير رقمي .

إذا كنا نريد ارجاع قيمة إلى متغير نستخدم علامة المساواة العادية (=) وإذا أردنا اختبار متغيرين أو قيمة معينة من أنها متساوية نقوم باختبار القيم بواسطة علامة المساواة المزدوجه (==) .

في الـ php4.01 تم إصدار علامة مساواة جديدة تقوم باختبار القيم ولاتعطي القيمه (true) إلا إذا كانت أنواع القيم متساوية وأنواع البيانات في المتغيرات ايضا متساوية .

مثال (1) :

```
<?
$m=”11”;
$u=11;
If ($m == $u)
{
”; القيم متساويةEcho “
}
?>
```

مثال (2) :

```
<?
$m=”11”;
$u=11;
If ($m === $u)
{
”; // القيم متساويةEcho “ لن يتم طباعة هذه الجمله على الشاشه لانهما غير متكافئتان
}
?>
```

التوضيح

لاحظ أننا في المثال الأول استخدمنا علامة المساواة المزدوجة لاختبار القيم وكانت القيم متساوية في المتغيرين فتم طباعة أن القيم متساوية (مع أن نوع البيانات مختلف) ولكن في المثال الثاني عندما استخدمنا علامة المساواة المضاعفة لم يتم طباعة أي شي وذلك لان القيم متساوية ولكن نوع البيانات مختلف فالمتغير $m حرفي بينما المتغير $u رقمي .

المعاملات : =! و <>

إن عكس علامة المساواة هي علامة عدم المساواة (=!)

مثال :

```
<?
If (5!=99)
"; القيم غير متساويةEcho "
?>
```

لاحظ أن 5 لاتساوي 99 لذلك فإن الشرط صحيح (true) لذلك قام بطباعة أن القيم غير متساوية .

إن الضد من علامة أكبر من وأصغر من هو علامة الـ(<>) وهو يقوم بارجاع قيمة (true) إذا كانت القيمتين مختلفتين عن بعضهما أي أنه مثل علامة =! تقريباً .

مثال :

```
<?
If (5<>99)
"; القيم غير متساويةEcho "
?>
```

تطبيق عملي على علامات المساواة وعدم المساواة

قم بفتح محرر النصوص لديك واكتب الكود التالي :

```
<html>
<head></head>
<body>
<Form method =get ACTION= "quiz.php">
ماهو اسم الرجل الذي يسمي بالفاروق ؟
<br><br>
">عمر<input type ="radio" name = "man" value="
عمر بن الخطاب رضي الله عنه
<br>
">أبوبكر<input type ="radio" name = "man" value="
أبو بكر الصديق رضي الله عنه
<br>
">عثمان<input type ="radio" name = "man" value="
عثمان بن عفان رضي الله عنه
<br>
<input type = submit>
</form>
</body>
</html>
```

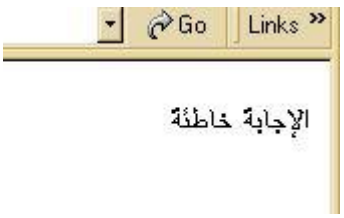
احفظها باسم quiz.html ...

قم بفتح محرر النصوص لديك واكتب الكود التالي :

```
<html dir="rtl">
<head></head>
<body>
<?
") عمر If ($q=="
";الإجابة صحيحة Echo "
") عمر If ($q!="
";الإجابة خاطئة Echo "
?>
```

قم بحفظه باسم quiz.php وضعهما في مجلد السيرفر

قم بتشغيل الملف quiz.html



المعاملات المنطقيه (AND,OR,NOT)

إن هذه المعاملات المنطقية تتيح لك بتنفيذ الكود بعد التحقق من مجموعة شروط وأيضا تنفيذ الكود إذا تحقق أكثر من شرط : (AND)

أو تحقق شي معين من بين عده أشياء : (OR)

ويمكنك مثلاً التحقق من عدم صحة شي لكي تقوم بتنفيذ شي آخر : (NOT)

فيمكنك مثلاً أن تقول : إذا كان الجو ممطراً والعاصفة شديده فلن أخرج من البيت .

ويمكنك أن تقول : إذا كان الجو هادئاً أو لايوجد أمطار فسأقوم بالخروج الى المنتزه .

ويمكنك أيضا أن تقول : إذا لم يكن الجو ممطراً سأقوم بالخروج إلى نزهة .

ولكن عند استخدامك لهذه الدوال عليك مراعاة أن تقوم بجعل هذه الشروط بين قوسين .

المعامل (AND) ونظيره (&&)

يمكننا استعمال المعامل (AND) والمعامل (&&) للتحقق من صحة عدة شروط لتنفيذ شي معين

مثال(1)

```
<?
$w=10;
$g=12;
IF ($w=10 and $g=12)
Echo ("لقد تحققت جميع الشروط");
      ?>
```

مثال (2)

```
<?
$w=10;
$g=12;
IF ($w=10 && $g=15)
Echo ("لقد تحققت جميع الشروط");
?>
```

في المثالين السابقين قمنا بعملية التحقق من أكثر من شرط باستخدام المعاملين (&& و and) فعندما تحققت جميع الشروط تم تنفيذ الأمر .... وعندما لم تكن جميع الشروط صحيحة تم تجاهل الأمر .

لاحظ أننا قمنا بجعل الشروط بين قوسين () لكي يعمل الكود بشكل صحيح :

```
($w=10 && $g=15)
($w=10 and $g=12)
```

المعامل (OR) ونظيره (||)

المعامل OR يقوم بالتحقق من عدة شروط وإذا تحقق أي واحد منها فإنه يقوم بتنفيذ الكود ونظيره (||) الذي يقوم بنفس العمليه .

مثال (1)

```
<?
$E=100;
$T=8;
IF ($E=14 OR $E=55 OR $E = 10 OR $T=8 )
Echo ("لقد تحقق أحد هذه الشروط");
?>
```

مثال (2)

```
<?
$E=100;
$T=458;
IF ($E=14 || $E=55 || $E = 10 || $T=8 )
Echo ("لقد تحقق أحد هذه الشروط");
?>
```

إذن عندما تحقق واحد من هذه الشروط تم طباعة السطر (لقد تحقق أحد هذه الشروط) .

ملحوظة قد لاتكون بتلك الأهمية لكن يجب أن تعرف أن الرموز && و || لها الأسبقية والأفضلية على استخدام AND و OR .

المعامل NOT ونظيره (!)

في الواقع لايمكنك استخدام NOT أبدا لأنها ليست أصلاً موجودة في لغة PHP لكن يمكنك استخدام المعامل (!) كبديل لها فهو يؤدي نفس وظيفتها وهي القيام بالتأكد من أن هناك قيمة غير صحيحة (FALSE) لكي يتم تنفيذ شي معين .

```
<?
$F="ماجد";
IF !($F=="نعمان")
Echo ("أهلاً بك");
?>
```

في المثال السابق يقوم الـPHP بالتأكد من أن المتغير $F لايحتوي على القيمة الحرفية (نعمان) ويتم ذلك باستخدام المعامل (!) وعندما يتم التأكد من ذلك يقوم بطباعة السطر (أهلاً بك)

ونشير إلى أننا عندما نقوم باختبار متغير بواسطة المعامل (!) فإن الـPHP إذا وجد المتغير فارغاً أو لم يتم انشاؤه يعطيه القيمة صفر وهي FALSE .

مثال

```
IF (!($R))
Echo (10);
```

استخدام المعاملات => و =<

من المعاملات المعروفة والمشهورة في الرياضيات هي علامتي أصغر من أو يساوي =< أو أكبر من أو يساوي => وهي تستخدم بنفس وظيفتها بالـphp وهي معرفة إذا ماكانت قيمة أصغر أو أكبر من أو تساوي قيمة آخري ، وهذه الأمثلة تعطيك مدخلاً أشمل لفهم هذه الدوال :

```
<?
$t = 15;
If ($t >= 10 )
" . "<br>"); ممتازEcho ("
$t = 5;
If ($t <= 9 )
");جيد جداEcho ("
?>
```

يمكننا في الشرط أن نتحقق من مجموعة من القيم باستخدام مجموعة من المعاملات ، ونقوم بتجميع هذه المجموعات داخل أقواس () مثلما كنا نستخدم سابقاً أكثر من معامل (+ ، - ، / ، *) باستخدام الأقواس .

وسيبدو ذلك واضحاً وجلياً في مثالنا التالى :

```
<?
$a=10;
$y=5;
$t =29;
If (($a == 10) or ($a==54) and ($y !=25) and ($t >= 11))
; "تحققت جميع الشروط"Echo
?>
```

سيتم طباعة لأنه قيمة تجميع التعبير السابق تكون صحيحة ولو قمنا بشرح المثال فسنقوم برؤية القسم الأول وهو :

```
($a == 10) or ($a==54)
```

وطبعاً المتغير يحمل القيمة 10 فسيكون هذا الجزء صحيحاً .

ثم نقوم برؤية الجزء :

```
($y !=25) and ($t >= 11)
```

وطبعاً تم التحقق من جميع الشروط وتم طباعة الكلمة (تحققت جميع الشروط) .

تعدد الشروط (else if و else)

يمكننا استخدام أكثر من هيكلية للعباره if فهناك مثلاً الهيكلية التالية :

```
If condtion is true
{
Excute code
}
Else
{
Excute other code
}
```

وهي تقوم بالتحقق من الشرط فإذا وجدته صحيحاً قامت بتنفيذ الكود الأول وإذا لم تجده صحيحاً ستقوم بتنفيذ الكود الآخر .

مثال

```
<?
$age=10;
If ($age>18)
 {
;"مرحبا بك في أكبر موقع تجاري إلكتروني"echo
}
else
{
";ممنوع دخول الأطفال الموقع لأنهم لايملكون المالecho "
}
?>
```

ويمكننا أيضا استخدام الهيكلية التالية :

```
If condtion is true
{
Excute code
}
Elseif
{
Excute other code
}
Else
{
Excute other code
}
```

وهي تقوم بتطبيق أكثر من شرط فإذا لم يكن أي شرط من الشروط صحيحاً سيتم تنفيذ الكود الذي يقع بعد كلمه **else**.

مثال :

```
<?
$age=10;
 If ($age<=18)
 {
;"مرحبا بك في أكبر موقع تجاري إلكتروني"echo
}
;)$y >= 44(elseif
{
”; مافي مشكلة برضه إذا كنت كبيرecho “
}
else
{
”;ممنوع البقيةecho “
}
?>
```

تداخل العبارات الشرطية

يمكنك تداخل العبارات الشرطية ، ونعني بتداخل العبارات الشرطية هي أن تقوم بأكثر من عملية شرطيه متداخله فمثلاً إذا كان شرط ما صحيحاً فإنه يجب أن يكون شرط آخر صحيحاً لكي يتم حصول شي معين وغير ذلك .

مثال :

```
<?
$h=”majed”;
$f=45;
If  ($h = = “majed” )
If ( $f= = 45)                {
{
”;الاسم والرقم صحيحانecho “
}
 else
{
 “ );الرقم غير صحيحecho (“
}        }
{  else
 “ ; اسم تسجيل الدخول غير صحيحecho “
}
      ?>
```

هذا مجرد مثال بسيط جداً لتداخل الدوال الشرطية حيث يقوم بإجراء اختبار على قيمة معينة ثم يقوم عند تجاوزه ذلك الاختبار بنجاح بإجراء اختبار ثاني فإذا تم تجاوز الاختبار الثاني يتم طباعة الاسم والرقم صحيحان وإذا لم يتم الاجتياز يتم طباعة عبارة الفشل في الاجتياز .

تطبيق عملي

سنقوم في هذا التطبيق بصناعة مسابقة بسيطة نستخدم فيها ماتكلمنا عنه سابقاً

١ -قم بانشاء ملف Msabqa.html .
٢ -قم بكتابه الكود التالي فيه :

```
     <html>
<body>

<form method="POST" action="msabqa.php" dir="rtl">
من هو أول الخلفاء الراشدين <br>
<br><input أبوبكر الصديق<p><br><input type="radio" value="abubaker" name="s">
عمرtype="radio" value="3mar" name="s">
<br><br><br>عثمان<br><input type="radio" value="3thman" checked name="s">

</p>

">    <input type="reset" ارسال  <p><input type="submit" value="
"></p>حذف_____value="
</form>

</body><html>
```

قم بفتح ملف وقم بتسميته msabqa.php

```
<?
<html dir = “rtl”>
If $s == “3mar”{
الإجابه صحيحة
}
else
{
”; الإجابه خاطئةecho “
}
?>
```

العبارة Switch

```
Switch (VARIABLE) {
CASE THING1 :
Excute code ;
        break;
CASE THING2 :
Excute code ;
break;
 Default;
Excute code ;

 }
```

تقوم العبارة بنفس عملية العبارة if ولكن بهيكلية أسهل ومحببة أكثر وتتيح لك اختبار قيمة متغير وإجراء أكثر من اختبار عليه .

```
break;
```

تقوم بالخروج من عباره معينه مثل switch و if والذهاب الى الأوامر والعبارات التي بعدها .

```
EXIT;
```

break; **تقوم بعملية الخروج من الكود نهائياً ولاتطبق أي أوامر بعدها ، وفي الأمثله التوضيحية التالية ستجد أن تخرج من العبارة فقط (Statement) بينما الـexit; تقوم بالخروج من كامل الكود (code).**

مثال :

```
<?
$s=10;
if ($s=10) {
echo “number=10”;
exit;
}
elseif ($s<11) {
  echo “number is less than 11”
{
echo “hello”;
?>
```

مثال :

```
<?
$s=10;
if ($s=10) {
echo “number=10”;
break;
}
elseif ($s<11) {
 echo “number is less than 11”
{
echo “Hello”;
?>
```

```
Defualt;
```

إذا لم تصلح جميع الحالات (Cases) في العبارة (Switch) فسيتم تنفيذ الأوامر التي تقع بعد هذه الكلمة وهي تؤدي نفس عمل else تقريباً في العباره if .

مثال (1)

```
<?
$g= “ahmed”;
Switch ($g) {
Case “ahmed”:
 “; مسموحEcho “
Break ;
 Case “khaled “ :
 “ ;ممنوعEcho “
Break ;
Case “salem“ :
 “ ;ممنوعEcho “
Break ;
Case “Mohmed “ :
 “ ;مسموحEcho “
Break ;
Default ;
”;لقد ادخلت اسم غير صالحEcho “
}
?>
```

مثال (2)

```
Switch ($g) {
Case $g>50:
 "; كبيرEcho "
Break ;
 Case  40 :
 " ;لاباسEcho "
Break ;
Case ($g<15) :
 " ;أطفال ممنوعEcho "
Break ;
Case 30 :
 " ;مسموحEcho "
Break ;
}
```

لاحظ أننا عند اختبارنا لنصوص نحتاج الى علامتي تنصيص مزدوجة وعند الارقام فاننا لانحتاج الي ذلك .

تطبيق عملي

قم بفتح محرر النصوص لديك واكتب الكود التالي واحفظه باسم age.html

```
<html>

<form method=post action="age.php">

كم عمرك ؟

<br>

<input type="text" name = "g">

">ارسال<input type=submit value="

</form>

</html>
```

قم بفتح محرر النصوص لديك واكتب الكود التالي واحفظه باسم age.php

```
<?
Switch ($g) {
Case $g>50:
 "; كبيرEcho "
Break ;
        Case  40 :
 " ;لاباسEcho "
Break ;
Case ($g<15) :
 " ;أطفال ممنوعEcho "
Break ;
Case 30 :
 " ;مسموحEcho "
Break ;
}
?>
```

الشرح

تقوم العبارة **Switch** باختبار قيمة متغير ما ويمكنك إجراء أكثر من افتراض عليه ويجب عليك كتابة الكلمة **break;** لكي تقوم بإيقاف تنفيذ العبارة **switch** فمثلاً لو قمت بكتابة الكود التالي :

```
<?
$g=40
   Switch ($g) {
Case $g<50:
Echo "1 ";

         Case  40 :
Echo "2" ;
}
?>
```

فاذا ادخل المستخدم الرقم **40** فسيتم طباعة الرقمين واحد واثنين كلاهما وذلك لأنك لم تقم بإيقاف العبارة فاكملت التحقق وطبقت جميع العمليات المطلوبة .

إذا قمت بوضع مربع نص وأردت من المستخدم كتابة شي فيه فإنه يستطيع ادخال أي شي ولنفترض أنه كتب في مربع النص كالتالى

```
I am <b>ahmed</b> ...
```

فسيقوم المتصفح بعرضها بعد معالجتها كالتالى :

```
I am ahmed ....
```

ولنقم بتطبيق عملي على ذلك

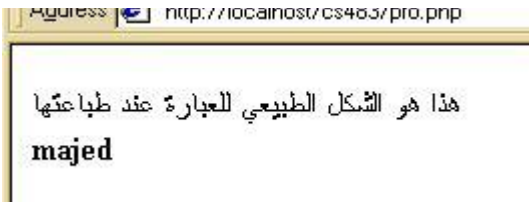
قم بفتح محرر النصوص واكتب الكود التالي واحفظه باسم **htmlch.html**

```
<html dir="rtl">
<form method=post action="html.php">
أدخل اسمك الكريم
<br>
<input type="text" name = "fname">
">ارسال<input type=submit value="
</form>
</html>
```

قم بفتح محرر النصوص واكتب الكود التالي واحفظه باسم **html.php**

```
<?

"; هذا هو الشكل الطبيعي للعبارة عند طباعتها Echo "

Echo "<br>" . $fname;

 ?>
```

قم بوضع الملفات في مجلد السيرفر ثم قم بتشغيل الملف **htmlch.html** واكتب في مربع النص أي شي وضعه بين وسوم **html**



مثال :

I am alfareees

ستجد أنه قدتم التعامل مع الوسوم كـ html وليس كنص عادي ولكي تعرضها كنص عادي فإنك تقوم باستخدام الدالة

HtmlSpecialChars();

حيث أنها ستقوم بمعاملة كود الـhtml كنص عادي وطبيعي تماماً .
إذاً نقوم بتعديل ملف الـhtml.php ليصبح كالتالى :

```
<?

$fname = HtmlSpecialChars($fname);

"; هذا هو الشكل بعد استخدام الدالةEcho "

Echo "<br>" . $fname;

 ?>
```

التكرارات والمصفوفات

لقد اخذنا في الدرس السابق شيئاً من أساسيات البرمجة وهو الدوال الشرطية وصناعة القرارات والآن نحن نتجه إلى شي يحب جهاز الكمبيوتر عمله وهو التكرارت والمصفوفات .

في الواقع قد يكون لديك يومياً شي تفعله بشكل مستمر مثل الإفطار في الصباح الباكر والنوم مساء ، انك تستمر على هذا الروتين دائماً .... نحن نسمي هذا الشي في لغة البرمجه التكرار .

هناك شي آخر يسمي المصفوفات ... في الواقع قد يحتوي درج مكتبك الخاصة بالكتب على عدة أدراج الدرج الاول منها يحتوي على الكتب الإسلامية والدرج الثاني منها يحتوي على الكتب الرياضية والدرج الثالث يحتوي على كتب الرياضيات ... أو لنفرض أنك مدرس في إحدي المدارس ولديك جدول للحصص ففي الحصة الأولى لديك مثلاً تدريس مادة الرياضيات .... والحصة الثانية لديك تدريس مادة العلوم والثالثة لديك تدريس مادة الكيمياء .... إن حصصك مرتبة بشكل معين مع أنها كلها تسمي حصص إلا أن كل حصة تختلف عن الأخري في المادة ! وهي مرتبه بشكل تصاعدي (الحصة الاولى ، الثانية ،الثالثة .....) .

نسمي هذه التقنية بالمصفوفات .... المصفوفات عبارة عن متغير اسمه ثابت ولها اكثر من قيمة وكل قيمة لها رقم معين ولكي تحصل على القيمة فانك تكتب المتغير ثم رقم القيمة التي فيه، لايشترط أن تكون هذه القيم متسلسلة فقد يكون هناك قيمتين ولكل قيمة رقم يختلف تماماً ويبعد كل البعد عن القيمه الثانية مثال رقم 1 و 258 كلاهما مختلف تماماً ويبتعد كل البعد عن الاخر .

إن دمج ميزة التكرارات مع المصفوفات يساعدك على توفير عدد الأسطر للكود ويساعدك على صنع أشياء عجيبة في أقل عدد ممكن من الأسطر .

التكرارات

التكرارات عبارة عن تكرار أمر معين بعدد معين من المرات ولقد اخذنا سابقاً الدوال الشرطية أو العبارات الشرطية بالأصح فوجدنا أن الكود الذي نكتبه في العبارات الشرطية لا تتنفذ إلا عندما يكون الشرط صحيحاً

أيضا التكرارت فهي تختبر الشرط فإذا كانت قيمته صحيحة فإنها تقوم بعمل الكود المطلوب ثم تقوم بإعادة اختبار القيمة فإذا كان صحيحاً فإنها تقوم بإعادة تنفيذ الكود وهكذا ، أما عندما لا يكون الشرط صحيحاً فإنها تتوقف عن تنفيذ الكود ويتم اكمال البرنامج بشكل عادي ... هناك ثلاثة أنواع من التكرارات .

إن أول دالة نقوم بأخذها في البداية هي الدالة while

التكرار while

لقد قمنا بأخذ التكرار while لأنه بسيط جداً وصيغه هذا التكرار هي :

```
While (condition شرط )
{
code
}
```

مثال :

```
<?
$d =10 ;
while ($d<15)
{
echo "$d <br>";
$d++;
}
?>
```

10
11
12
13
14

سيقوم الـPHP أولا بإعطاء المتغير $d القيمة 10 ثم يقوم بعد ببدء التكرار while فإذا كان الشرط صحيحاً (وهو أن المتغير أصغر من الرقم 15) فإنه يقوم بتنفيذ الكود الذي بين الأقواس وعمل هذا الكود أن يقوم بطباعة المتغير ثم يقوم بإضافة واحد على القيمة الموجودة في المتغير $d ثم بعد ذلك سيتم اختبار الشرط مرة ثانية فإذا كان صحيحاً فسيتم نفس العملية حتي يكون الشرط غير صحيح فيتوقف عندها التكرار ويتم إكمال الكود التي تقع بعد الاقواس .

إذا لم تقم بوضع حد للتكرار فلن يتوقف التكرار وقد يكون لانهائي ....

مثال :

```
<?
$d =10 ;
while ($d<15)
{
echo "$d <br>";
}
?>
```

سيتم طباعة الرقم 10 ولن يتوقف التكرار لأن الشرط صحيح دائماً وليس هناك مايوقفه بينما في الكود السابق استطعنا إيقاف الكود بسبب أننا كنا نضيف واحد على القيمة الموجودة في المتغير وكلما يتم اعادة اختبار الكود كل ما تتغير القيمة حتي يصبح الشرط غير صحيح بسبب أن $d أكبر من 15 .

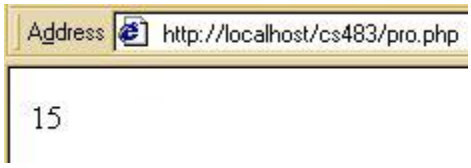
التكرار do - while

هذا التكرار يعمل بنفس طريقه التكرار الأول إلا أنه يوجد بعض الاختلافات البسيطة وصيغته كالتالى :

```
do

code

); شرطwhile (condition
```

مثال :

```
<?
$f=15 ;
do
{
echo $f;
$f++;
}
while ($f < 10) ;
?>
```



سيقوم التكرار بتنفيذ السطر الموجود بين القوسين أولا ثم يقوم بتنفيذ بإختبار الشرط فإذا كان الشرط صحيحاً قام بإعادة العملية الموجودة بين القوسين وهي إضافة واحد على المتغير $f وهكذا حتي يكون الشرط غير صحيح فيتم التوقف .. لاحظ أننا في التكرار الأول قمنا باختبار الشرط قبل صناعة أي عمل بينما في التكرار الثاني قمنا بتنفيذ الكود أولا ثم قمنا بإجراء الاختبار .

التكرار FOR

يختلف هذا التكرار عن سابقيه لكن وظيفته هي نفس وظيفتهما وهي تكرار الأوامر عند حصول شي معين

الصيغة :

```
For (counter عداد ; test value اختبار القيمة ; set counter اداء عمليه على العداد )
{
code شفرة
}
```

مثال :

```
<?
For ($u = 18 ; $u>10 ; $u--)
{
echo $u."\t\t";
}
 ?>
```

يتكون هذا التكرار من ثلاثة أقسام .... القسم الأول نضع فيه متغير يحتوي على قيمة حيث سيبدأ التكرار العمل من عند هذه القيمة والقسم الثاني نكتب فيه الشرط الذي سيقوم التكرار بفحصه (والذي هو كالمعتاد اختبار لقيمة المتغير

في القسم الاول) والقسم الثالث نضع فيه العمل الذي سيجري على المتغير عند كل تكرار ثم نقوم بكتابة كود التي سيقوم بتنفيذها التكرار بين القوسين .

كأننا نقول للـphp بشكل عامي أن يقوم في البداية بإعطاء المتغير $u القيمة 18 وقبل ان يقوم بتنفيذ الكود عليه أن يقوم بتحليل الشرط فإذا كان الشرط صحيحاً فإنه يقوم بإنقاص واحد من المتغير $u ويتم تنفيذ الكود حتي يصبح المتغير$u قيمته 9 فيقوم الـPHP آنذاك بالخروج من التكرار والذهاب الي الكود الذي يلى القوسين .

المصفوفات

لقد قمنا بتعريف المصفوفات سابقاً بشكل بسيط وحان الوقت الآن لنعرفها ونعرف كيفية عملها . المصفوفات عبارة عن متغير وهذا المتغير يحتوي على أكثر من قيمة أو عنصر (Element) وكل عنصر له فهرسة (Index) تبدأ هذه الفهرسة من الصفر إذا لم تقم بتحديدها

مثال :

```
<?
$A[ ] = "Majed";
$A[ ] = 13;
?>
```

في هذا المثال سيقوم الـ PHP بإعطاء الفهرسة تلقائياً فسيقوم بوضع الرقم فتصبح المتغير فهرسته كالتالى :

```
$A[0] = "Majed";
$A[1] = 13;
```

إننا لم نقم بإدخال هذه الأرقام من تلقاء أنفسنا ولكن الـPHP قام بوضعها مع أنه يمكننا أن ندخلها بشكل عادي فمثلاً لو كتبنا :

```
<?
$A[0]= "Majed";
$A[1] = 13;
?>
```

سيقوم الـPHP بأخذ الفهرسة المعتمدة ولن يضع أي فهرسة أخري يمكننا أيضا أن نكتب أي فهرسة ولانعتمد على الترتيب في الارقام

مثال :

```
<?
$A[10 ] = "Majed";
$A[ 25] = 13;
?>
```

هل لاحظت أيضا أننا لم نقم بتعريف نوع متغيرات المصفوفة وقام الـ PHP بتعريفها تلقائياً بدلاً منا فمرة استخدمنا قيمة حرفية ومرة استخدمنا رقماً ورغم ذلك فلم يقم الـ PHP بعمل أي اعتراض إضافة إلى ذلك فإن الـ PHP يقوم بتحديد عدد عناصر المصفوفة تلقائياً فهو يعرف مثلاً من المثال السابق أن عدد عناصر المصفوفة الكلي هو عنصرين .

يمنحنا الـPHP ميزة أخري وهي عدم التقيد بالأرقام في الفهرسة فمثلاً يمكننا استخدام حروف عادية .

مثال :

```
<?
$A["a" ] = "Majed";
$A["b" ] = 13;
?>
```

لاحظ أننا استخدمنا القيم الحرفية ولم يعترض الـ PHP بتاتاً ويمكننا طباعة أي عنصر من عناصر المصفوفة بكل بساطة .

مثال :

```
<?
$r [“aa”] = “Majed sa”;
$r [1] = 483;
$r [20] = 1424;
echo $r[aa] ."\t";  // "\t" ما هي الى مسافة فقط
echo $r[20] ."\t"; // "\t" ما هي الى مسافة فقط
];"aa"echo $r[
?>
```

لا فرق بين أن نكتب النص الحرفي (aa) بين علامتي تنصيص عند الطباعه وعند كتابته بدون علامات تنصيص ... سيقوم الـPHP بمعرفة ذلك تلقائياً .

يمكننا تعريف المصفوفات أيضا بطريقة أخري

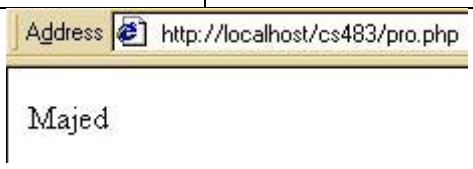
```
$variable = array (elements) ;
```

مثال :

```
<?
$t =array (“Majed”, “SA”, “Mohammad”, “Ali”);
echo $t [0];
?>
```

يقوم الـPHP بإعطاء كل عنصر من عناصر المصفوفة رقم فهرسة فتصبح كالتالى :

العنصرElement	الفهرسهIndex
Majed	0
SA	1
Mohammad	2
Ali	3



إذن القيمة التي سيطبعها الـPHP في النهايه هي Majed ، لاحظ أن الـPHP قام بإعطاء رقم الفهرسة وقام بالبدء من الصفر ولكن يمكننا جعل الـPHP يبدأ الفهرسة من الرقم واحد كالتالي :

```
<?
$r = array (1=>"Majed", "SA","Mohammad", "Ali");
?>
```

عند تعريفك لرقم الفهرسة للقيمة الأولى سيقوم الـPHP بإعطاء أرقام فهرسة بشكل تسلسلي ،

عندئذ ستصبح الفهرسة كالتالى :

Elementالعنصر	Indexالفهرسه
Majed	1
SA	2
Mohammad	3
Ali	4

هناك طريقة لتكون أيضا الفهرسة هي عبارة عن حروف :

```
<?
$r = array ("M1"=>"Majed", "M2"=> "SA", "M3"=>"Mohammad", "M4"=> "Ali");
?>
```

عندئذ ستصبح الفهرسة كالتالى :

Elementالعنصر	Indexالفهرسه
Majed	M1
SA	M2
Mohammad	M3
Ali	M4

عندما نريد تغيير أي عنصر في المصفوفة فيمكننا عمل ذلك ببساطه .

مثال :

```
”; لمياء$r [M2]= “
```

لاحظ أننا قمنا بتغيير القيمة من (SA) الى (لمياء) .... طريقة بسيطة أليس كذلك :)

قراءه المصفوفات واستخراج القيم

تكلمنا سابقا عن التكرار For

يمكننا استخراج عناصر مصفوفة وطباعتها في بساطة وتوفير وقت عن طريق التكرارات

لنفرض أن لديك هذه المصفوفة :

```
<?
$people =array (“Majed”, “SA”, “Mohammad”, “Ali”);
?>
```

واردت أن تطبع أسماء جميع الاشخاص المتواجدين فيها

أولاً نحن نعرف أن المصفوفة إذا لم نقم بتعريف رقم فهرسة لها فإن الـPHP يقوم ببداية فهرستها من الصفر وعلى ذلك فان رقم العنصر الأول 0 ورقم العنصر الرابع 3 ... على ذلك يمكننا بكل بساطه كتابة الكود التالي الذي يقوم بطباعة المصفوفة كالتالى :

```
<?
$people =array (“Majed”, “SA”, “Mohammad”, “Ali”);
echo “$people[0]. <br>”;
echo “$people[1]. <br>”;
echo “$people[2]. <br>”;
echo “$people[3]. <br>”;
?>
```

لنفرض أن لديك ثلاثين أو ثلاثة آلاف اسم في مصفوفة ألن تبدو هذه الطريقة متعبة قليلا !!!

هناك طريقة أخري وهي عن طريق التكرارات .

لنفرض أننا أردنا كتابة تكرار يقوم بطباعة الارقام من واحد الى عشرة فإننا نستطيع كتابة التكرار بالشكل التالى :

```
<?
;$I++) 11For ($I=1;$I<
{
Echo "$I <br>";
}
?>
```

والآن لنقل أننا نريد طباعة الأربعة عناصر في المصفوفة كل ماعلينا هو إجراء عملية بسيطة على الكود لكي يتم ذلك:

```
<?
$people =array ("Majed", "SA", "Mohammad", "Ali");

For ($I=0;$I<4;$I++)
{
Echo "$people[$I] <br>";
}
?>
```

لاحظ أننا بدأنا العداد بالقيمة صفر ثم اشترطنا أن يكون أقل من 4 لأن آخر عنصر في المصفوفة رقم فهرسته 3 ثم قمنا بجعله يزداد بقيمة 1 لأننا نريد طباعة جميع عناصر المصفوفة وقمنا بوضع رقم العداد في خانة الفهرسة وعلى ذلك سيتم في كل تكرار طباع عنصر المصفوفة الذي فهرسته تساوي رقم العداد .

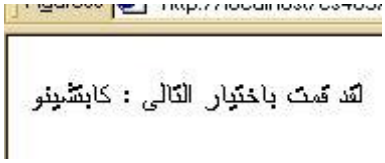
لقد تكلمنا سابقاً في درس النماذج عن إخراج القيم من قائمة على شكل مصفوفة .

مثال :

```
<form action = "array.php" method = post>
ما هو مشروبك المفضل ؟
<br>
<select name = "a[]" multiple>
</option>شاي<option>
</option>قهوة<option>
</option>كابتشينو<option>
</option>توت<option>
</option>برتقال<option>
</select>
<br>
" >لذيذ<input type=submit value = "
</form>
```

في ملف الـ array.php اكتب :

```
<html>
لقد قمت باختيار التالى :
<?
For ($I=0;$I<4;$I++)
{
Echo "$a[$I] <br>";
}
?>
</html>
```



لقد عرضنا في القائمة خمسة عناصر ... لاحظ أننا وضعنا في اسم المتغير للقائمة قوسين [ ] لكي يتعرف الـ html على أنه سيتم تخزين البيانات تلقائياً بعد ذلك قام الـ PHP بفهرسة العناصر التي تم إرسالها من قبل العميل سواء كانت ثلاثة أو أربعة ولكنها بالطبع لن تزيد على خمسة .... على ذلك سيكون آخر رقم تنتهي به المصفوفة هو 4 .

أتوقع أنك الآن بدأت تحب المصفوفات .... يمكننا صناعة القائمة عن طريق المصفوفة أيضاً ....

مثال :

```
<form action = "list.php" method = post>
ما هو مشروبك المفضل ؟
<br>
<select name = "s" >
<?
");برتقال","توت","كابتشينو","قهوة","شاي$shrab =array("
For ($k=0;$k<4;$k++)
{
echo "<option>".$shrab[$k]."</option>";
}
?>
</select>
</form>
```

عند اختيار المستخدم للقيمة سيتم وضعها في المتغير $s يمكنك مراجعة درس النماذج لكي تفعل ذلك ، هذا المثال يقوم بصناعة مصفوفة للمشروبات ثم يقوم بإخراجها في قائمة مما يوفر علينا الوقت في كتابة الكود فلو كان لديك مثلا حوالى مئة دولة فيمكنك مثلا وضعها في مصفوفة وبعد ذلك بناء القائمة التي سوف تقوم ببناء القائمة التي ستحتوي على هذه الدول عن طريق المصفوفات والتكرارات .

قم بحفظ التغييرات في ملف إمتداده php وقم بكتابة الملف list.php اعتمادا على معلوماتك السابقة في درس النماذج .

الدالة key

لنفرض أن لدينا مصفوفة مكونة من عنصرين :

مثال :

```
$s= array ("على","ماجد");
```

الآن لنضف إليها هذه السطور

```
<?
$s= array ("على","ماجد");
$t=key ($s);
echo $t;
?>
```

يقوم الأمر key بإيجاد رقم الفهرسه (index) العنصر النشط حالياً .... وهو الرقم صفر حيث أننا لم نضع فهرسة وهذه هي الفهرسة التي وضعها الـPHP تلقائياً عندما لم نضع فهرسة ... قد تحيرك كلمة النشط لكن ستعرف أننا نستطيع التجول بين عناصر المصفوفة لاحقاً .

قد يكون رقم الفهرسة حروف أو كلمات

مثال :

```
<?
$s= array ("ع"=>"م","على"=>"ماجد");
$t=key ($s);
echo $t;
?>
```

الدالة current()

تقوم الدالة current بإيجاد القيمة لعنصر المصفوفة الحالى (index value) .

مثال :

```
<?
");على"=>"ع" , "ماجد"=>"م$s= array ("
$p=current ($s);
echo $p;
?>
```

في المثال السابق قمنا بإيجاد القيمة الحالية للعنصر النشط .... لاحظ أننا أوجدنا بالأمر key رقم الفهرسة بينما أوجدنا بالأمر current القيمة للعنصر المفهرس .

كيف يمكننا تنشيط العناصر الاخري للمصفوفه ؟!

يمكننا ذلك عن طريق الدالتين next() و prev اللتان تقومان بالتجول بين عناصر المصفوفة ..... لنفرض أن لدينا مصفوفة تتكون من ثلاثة عناصر

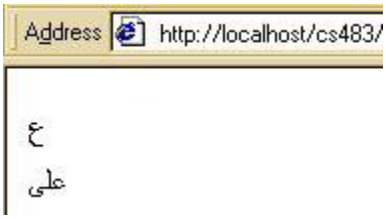
مثال :

```
<?
");أحمد"<="أ","على"<="ع","ماجد"<="م$s= array ("
echo key($s)."<br>";
echo current($s) ."<br>";
?>
```

لقد قمنا في هذا المثال بطباعة قيمة رقم الفهرسة للعنصر الحالى وقيمته (اقصد برقم الفهرسة الحرف(م) واقصد بالقيمة (ماجد) .... لنقم الآن بالتجول بين عناصر المصفوفة ولنر نتيجة الطباعة .

مثال :

```
<?
"); أحمد"<="أ","على"<="ع","ماجد"<="م$s= array ("
next($s);
echo key($s).”<br>”;
echo current($s) .”<br>”;
?>
```




```
<?
"); أحمد"<="أ","على"<="ع","ماجد"<="م$s= array ("
next($s);
next($s);
echo key($s).”<br>”;
echo current($s) .”<br>”;
?>
```

لاحظ أننا كتبنا الدالة **next()** قبل أن نقوم بالإنتقال لكي يتم تنشيط العنصر الثاني في أول مثال ولتنشيط العنصر الثالث في ثالث مثال (ولاحظ أننا كتبنا **next()** مرتين) .

يمكننا الرجوع لتنشيط العنصر السابق بوضع الدالة prev() فمثلاً يمكننا تعديل المثال التالى :

```
<?
");أحمد"<="أ"،"على"<="ع"،"ماجد"<="م$s= array ("
next($s);
next($s);
prev($s);
echo key($s).”<br>”;
echo current($s) .”<br>”;
?>
```

فسيقوم الـPHP في هذه الحاله طباعة العنصر الثاني وليس الثالث لأنه تم التراجع خطوه عن طريق prev()

ماذا سيحصل إذا قمنا بإضافة عنصر على مصفوفة غير محدودة الفهرسة ؟!

لنفرض أن لدينا مصفوفة وأضفنا إليها عنصر غير محدد الفهرسة .

مثل :

```
<?
”);احمد”,44=>”محمد”,5=>”ماجد$s= array (12=>“
”;هشام$s[ ]= “
Next($s);
Next($s);
Next($s);
Echo key ($s).”<br>”;
).”<br>”; $s Echo current(
?>
```

سيقوم الـPHP ببساطة بالبحث عن أكبر رقم فهرسة وبعد ذلك يبدأ بإعطاء الفهرسة تسلسلاً بعده فإذا كانت أرقام الفهرسة حروفاً بدأ من الصفر في اعطاء الرقم .. ولاحظ في هذا المثال بأنه قام بإعطاء العنصر الرقم 45 لأن أكبر عنصر في المصفوفة هو 44 وعلى ذلك قام بإعطاء الأرقام تسلسلاً بعد هذا الرقم .

الدالة List و Each

لنفرض أنك قد قمت بصنع مصفوفة غير مفهرسة بالترتيب

مثال :

```
<?
");احمد",44=>"محمد",5=>"على$s= array (12=>"
?>
```

على ذلك دعنا نخبرك بخبر سار وهو أنك تستطيع أن تجعل حياتك مع PHP أسهل مع حياتك مع نفسك !

```
While (list( ارقام الفهرسة Index,Element value قيمة العنصر ) = each (array)
```

تستطيع بواسطة هذه الدالتين وعن طريق التكرار while استخراج جميع العناصر الموجودة في المصفوفة

```
While (list($e,$r) = each ($s))
{
echo "$e \t\t <b>$r</b>\t\t";
}
```

أولاً أنت تقوم بتسمية متغيرين واحد منهما لرقم الفهرسة ($e) والثاني للعنصر ($r) ويمكننا تسميتهما بأي اسم وفي حالة ما إذا أردنا عرض العنصر فقط أو معرفة العنصر فقط فيمكننا حذف ($e) ولكننا لانحذف الفاصلة

```
While (list(,$r) = each ($s))
{
echo " $e $r<br>";
}
```

لنعد الى المثال الذي فيه رقم الفهرسة والعنصر ... سيقوم التكرار بوضع رقم الفهرسة (الذي قد يكون نصياً) في المتغير $e وسيضع قيمة العنصر الذي رقم الفهرسة له هو $e في المتغير $r ثم سيقوم بطباعة العناصر حتي ينتهي منها جميعها ...

ملاحظة مهمة : إذا لم تقم بتعريف فهرسة للمصفوفة (حروف أو أرقام أيا كان) فسيتم استخدام العناصر عندما يطلب التكرار الفهارس

مثال :

```
<?
$e=array("M100","M101","M102");
While (list ($I,$V)=each($e))
{
echo "<br>$e[$I]";
}
?>
```

لاحظ أننا طلبنا طباعة الفهرسة (index) إلا أنه تم أخذ العناصر(elements) بدلاً من الفهرسة

يمكننا بواسطة هذه الدالة صناعة أشياء مفيده وكمثال لذلك لنفرض أن لدينا مصفوفة أرقام هواتف ونريد أن نخرج هذه المصفوفة على جدول html فسنستطيع صناعة هذا الجدول عن طريق التكرار السابق بكل سهولة .

مثال :

```
<table align='center' dir = "rtl" border="1" width="50%" cellspacing="0"
bordercolorlight="#000000" bordercolordark="#000000" bordercolor="#000000">
<tr>
</td>الاسم    <td align='center'>
</td>رقم التلفون    <td align='center'>
 </tr>
<?
);465873 ، "سالم"<=456546، "ماجد$s = array (658=>"
While (list($e,$r) = each ($s))
{
echo "<tr><td align='center'>". $r . "</td><td align='center'>" .$e .
"</td></tr>";
}
?>
</table>
```

الاسم	رقم التلفون
ماجد	٦٥٨
سالم	٤٥٦٥٤٦
٤٦٥٨٧٣	٤٥٦٥٤٧

أرايت كيف استخرجنا جميع أرقام التلفونات في جدول بواسطه تكرار بسيط ، يمكنك صناعة الأكثر واختصار الكثير من الوقت على ذلك إذا كانت المصفوفة تحتوي على المئات من الأرقام بواسطة هذا الكود بدلاً من أن تكتب الكود على شكل html وتكتب البيانات وتتعب نفسك .

يمكنك أيضا معرفة عدد العناصر في مصفوفة معينة إذا كنت تريد معرفة عددها وذلك بالطريقة التالية :

```
<?
");احمد",44=>"محمد",5=>"ماجد$s= array (12=>"
$S=0;
While (list($E,$r) = each ($s))
{
$S++;
}
" . $S++;  عدد عناصر المصفوفهECHO "
?>
```

فرز المصفوفات

هناك العديد من الدوال التي يوفرها لنا الـ PHP لفرز المصفوفات . نحن سنأخذ نظرة عن الخمسة دوال الأكثر استخداماً :

الدالة Sort()

هذه الدالة من أساسيات فرز المصفوفات وهي جداً أساسية وهي تقوم بأخذ محتويات المصفوفة ومن ثم تقوم بفرزها هجائياً اعتماداً على الأحرف الكبيرة أولا ثم الصغيرة .. تتطلب هذه الدالة اسم المصفوفة التي سيتم عليها الفرز

```
Sort (ArrayName);
```

إذا قمنا بإنشاء مصفوفة بالشكل التالى :

```
;)$NaNo=array (“ali”,“Majed”,“hythem”,“Khaled”,“Ammar”,“Hesham”
```

فإذا أردنا فرزها عن طريق الدالة sort() فإننا نقوم باستخدامها كالتالى :

```
<?
$NaNo=array ("ali","Majed","hythem","Khaled","Ammar","Hesham");
sort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "$e\t\t$r<br>";
}
?>
```

لاحظ أنه عند تنفيذك للمثال ستجد أن الـPHP قام بالفرز اعتماداً على الأحرف الكبيرة أولاً ثم قام بالفرز بعدها اعتماداً على الأحرف الصغيرة .

الدالة Arsort()

هذه الدالة تعمل نفس عملية الدالة sort() ولكن هناك اختلاف بسيط فمثلاً لو كتبنا المصفوفه كالتالى :

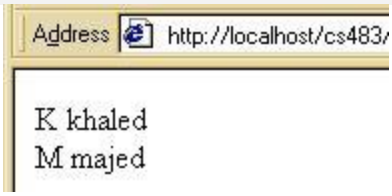
```
$NaNo=array ( "ad"=>"majed", "kh"=> "khaled");
```

وأردنا فرزها وطباعة الفهارس والقيم كما في المثال التالى :

```
<?
$NaNo=array ( "M"=>"majed", "K"=> "khaled");
sort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "$e \t\t $r<br>";
}
?>
```

قارن ناتج المثال السابق مع هذا المثال :

```
<?
$NaNo=array ( "M"=>"majed", "K"=> "khaled");
asort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "$e \t\t $r<br>";
}
?>
```



اعتقد انك قد عرفت الفرق ففي المثال الاول قامت الدالة sort باستبدال الحروف بأرقام في الفهرسة أما في المثال الثاني فقد تم وضع الحروف كما هي وتم فرزها كما تفعل الدالة sort في الفرز .

باختصار لايوجد فرق بين sort و asort إلا في أن الدالة sort تستبدل فهرسة الحروف بأرقام .

الدالة rsort() و arsort

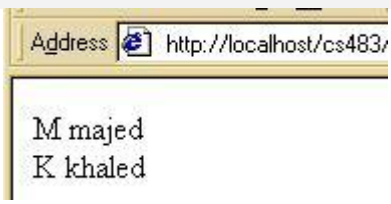
تقوم بنفس عمل sort و asort ولكن بشكل عكسي جرب الأمثلة التالية :

مثال :

```
<?
$NaNo=array ( "M"=>"majed", "K"=> "khaled");
rsort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "$e \t\t $r<br>";
}
?>
```

مثال :

```
<?
$NaNo=array ( "M"=>"majed", "K"=> "khaled");
arsort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "$e \t\t $r<br>";
}
?>
```



ستجد أن الدالة rsort تقوم بنفس عملية الدالة sort ولكن بشكل عكسي

أيضاً الدالة arsort تقوم بنفس عملية asort ولكن بشكل عكسي .

يمكنك استعمال كل هذه الدوال في الفرز مع الحروف العربية (إذا كان السيرفر يدعم اللغة العربية)

قم بتطبيق المثال التالى :

```
<table border =1><tr><td>
RSORT()<br>
<?
"); أحمد", "A"=> "ماجد$NaNo=array ( "M"=>"
rsort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "$e\t$r<br>";
}
?>
</td><td>
ARSORT()<br>
<?
"); أحمد", "A"=> "ماجد$NaNo=array ( "M"=>"
arsort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "$e\t$r<br>";
}
?>
</td><td>
ASORT()<br>
```

```
<?
"); أحمد", " A"=> "ماجد$NaNo=array ( "ad"=>"
asort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "$e\t$r<br>";
}
?>
</td><td>
SORT()<br>
<?
"); أحمد", " A"=> "ماجد$NaNo=array ( "ad"=>"
sort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "$e\t$r<br>";
}
?>
<td></tr></table>
```

الدالة ksort

تكلمنا سابقاً عن طريقة فرز المصفوفات ولكن نريد أن نلفت نظرك أننا كنا نعتمد على العنصر في الفرز (element) ولكن هذه الدالة تقوم بالاعتماد على رقم الفهرسه في الفرز (index)

مثال

```
<table border =1><tr><td>
asort()<br>
<?
"); أحمد", "A"=> "ماجد$NaNo=array ( "M"=>"
asort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "$e\t$r<br>";
}
?>
</td><td>
ksort()<br>
<?
"); أحمد", "A"=> "ماجد$NaNo=array ( "M"=>"
ksort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "$e\t$r<br>";
}
?>
<td></tr></table>
```

لقد اعتمد الـ php على index ولم يعتمد على الـelement في الفرز .

دوال المصفوفات الإضافيه

هناك الكثير من الدوال التي يمنحنا إياها الـPHP للتعامل مع المصفوفات والتي لايكفي الوقت لذكرها الآن سنقوم بشرح أهم دالتين والمستخدمة بكثره وهي array_push() و array_pop()

لنفرض أننا قمنا بإنشاء مصفوفة بالشكل التالى :

```
<?
$arr[ 5]=”majed”;
$ arr [ 85]=”khaled”;
$ arr [ 35]=”mohmed”;
$ arr [ 19]=”hajeer”;
?>
```

وأردنا أن نضيف عنصر جديد لها فقمنا بالتالى :

```
<?
$ arr [ 5]=”majed”;
$ arr [ 85]=”khaled”;
$ arr [ 35]=”mohmed”;
$ arr [ 19]=”ahmad”;
$ arr [ ]=”ali”;
?>
```

انظر إلى العنصر الأخير الذي سيعطيه الـPHP رقم الفهرسة (index) وسيكون رقم فهرسته هو 86 .

نريد أن نلفت نظرك بأننا نستطيع عمل إضافة لعنصر على المصفوفة بطريقة أخري وهي عن طريق الدالة array_push() كالتالى :

```
array_push (ArrayName اسم المصفوفه,Elemnt1, Elemnt2, Elemnt3,.....)
```

نضع في القسم الأول من الدالة اسم المصفوفة التي نريد إضافة العنصر لها ونضع في القسم الثاني عنصر واحد أو أكثر وهي التي سيتم إضافتها للمصفوفة .

مثال :

```
<?
$ arr [ 5]=”majed”;
$ arr [ 85]=”khaled”;
$ arr [ 35]=”mohmed”;
$ arr [ 19]=”hajeer”;
array_push ($arr,ali)
?>
```

مثال :

```
<?
$ arr [ 5]=”majed”;
$ arr [ 85]=”khaled”;
$ arr [ 35]=”mohmed”;
$ arr [ 19]=”hajeer”;
array_push ($arr,ali,salem,sameer,thamer)
?>
```

ولو أردنا حذف مثلاً عنصر من المصفوفة فإننا نقوم بتعريف المصفوفة من جديد أو يمكننا استخدام الدالة **array_pop** التي تقوم بحذف آخر عنصر من المصفوفة والتي تتطلب فقط اسم المصفوفة

```
)اسم المصفوفهArray_pop(ArrayName
```

مثال :

```
<?
$ arr [ 5]=”majed”;
$ arr [ 85]=”khaled”;
$ arr [ 35]=”mohmed”;
$ arr [ 19]=”hajeer”;
array_pop($arr)
?>
```

سيتم حذف العنصر hajeer من المصفوفة ولن يكون في المصفوفة غير ثلاث عناصر .

Implode و Explode

تقوم هذه الدالتين باقتصاص قيمة معينة من مصفوفة أو نصوص وتقوم بإضافة قيمة معينة على مصفوفة أو نصوص.

الدالة Implode

تقوم بإضافة قيمة بين عناصر المصفوفة .

مثال :

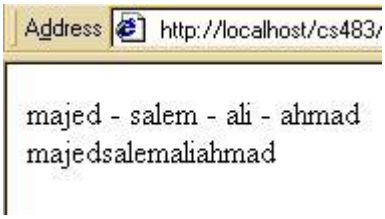
```
<?
$stng =array ("majed", "salem", "ali", "alfarsi");
$r =implode ("H",$stng);
echo $r;
?>
```

الدالة explode

تقوم بحذف قيمة من مصفوفة وذلك لايعني حذف عناصر من المصفوفة .

مثال :

```
<?
$stng =array ("\tmajed\t", "\tsalem\t", "\tali\t", "\tahmad\t");

$r =implode ("-",$stng);
echo $r;
echo "<br>";

for($i=0;$i<=4;$i++)
{
$r = explode ("\t",$stng[$i]);
echo $r[1];
echo $r[2];
echo $r[3];
echo $r[4];
}
?>
```



HTTP_GET_VARS و HTTP_POST_VARS

هذه ليست متغيرات بل مصفوفات ، نعم هذه مصفوفات ولكن في ماذا نستخدمها ولماذا ؟

في الواقع تحدثنا في الدرس السابق عن طريقة التعامل مع النماذج والحصول على البيانات من المستخدم وتكلمنا عن أسلوبين لنقل البيانات وهما GET و POST

عندما تصل البيانات محفوظة في متغيرات إلى صفحة الـ PHP فإنه يقوم بتعريفها تلقائياً ويمكنك طباعة المتغيرات وقيمها مباشرة من غير تعريف .... ولكن هذه الميزة في الـ PHP يمكن إلغاؤها عن طريق الملف PHP.INI وذلك بإغلاق ميزة register_globals

وذلك بوضع off بدلا من on

الوضع الافتراضي لها هو on ولكن تستطيع إغلاقها وقد تكون مستاجراً عند مزود خدمة ويب وسيط فيقوم بإغلاق هذه الميزة من باب الحماية ليس إلا .... لاتقلق يمكنك الحصول على البيانات فهي ما زالت موجودة ولكن يجب عليك أن تقوم بإستخدام هذه المصفوفتين لكي تستخرج البيانات .

لنفرض أنك اشتركت عند مزود ويب وكان قد أغلق ميزة (register_globals) حسناً لنفرض أنك قد صنعت نموذجاً يستخدم مربع نص ويحفظ قيمته في متغير اسمه Dorrah ثم بعد ذلك يقوم بإرسال هذه القيمة باستخدام الأسلوب GET إذاً سيكون جزء من الكود في الصفحة الأولى والتي تحتوي على النموذج كالتالى

```
<form method =get action = "try.php">
ما هو أول أيام الاسبوع !!
<br>
<input type=text name = "Dorrah">
<br>
```

في الملف الثاني(try.php) سنقوم بكتابة الجزء الذي سيقوم بطباعة القيمة كالتالى

```
<?
Echo HTTP_GET_VARS["Dorrah"];
?>
```

لاحظ أننا لم نستخدم $ ولكن إذا أردنا الإحتفاظ بقيمة المتغير في متغير آخر فيمكننا ذلك بشكل عادي كالتالى :

```
<?
$Dorrah = HTTP_GET_VARS["Dorrah"];
?>
```

طريقه بسيطة .... أليس كذلك ولكن .... لنفترض أن مزود خدمة الويب لديك حريص جداً ولذلك فقد ألغي أيضا ميزة استقبال هذه القيم في المصفوفات .... يمكنه ذلك في ملف الـ php.ini في اعدادات الـ track_vars الذي يقوم بمنع السيرفر من استخدام هذه المصفوفات (هذه الميزة يمكن إلغاؤها في php4) .... على ذلك انصحك بإرسال رسال تذمر وشكوي إلى مزود الخدمة لديك .. تعلن فيها أن الأمر اصبح لا يحتمل .

مصفوفه متعدده الابعاد

يمكنك صناعة مصفوفات بداخل مصفوفات على حسب ماتحتاجه في معلوماتك الرياضية فقد تحتاج مثلاً إلى إنشاء أشياء معقدة (ومقلقة نفسياً) نريد أن نخبرك على أية حال أنه يمكنك صناعة المصفوفات المتعددة الأبعاد ويمكنك استخدام حتي مائة مصفوفة متداخلة ولكن يجب أن تراعي حجم الذاكرة المستخدمة في السيرفر لديك (وعلى كل حال إن استطعت أن تقوم بالتركيز في صناعة عشر مصفوفات متداخلة بدون أي مشاكل أو مرض نفسي أو .... فأنت تستحق جائزة) .

يمكننا كتابة مصفوفة متداخلة كالتالى :

```
<?
$mon= array (1=>array ("sharkeh al-jafali",154786) ,2 => array ("majed
sa",1257) );
while (list($personnum) =each ($mon))
{
echo ("<br>$personnum<br>");

while (list(,$phone)=each ($mon[$personnum]))
   {
    echo ("$phone");
    }
}
?>
```

الشرح

هذا المثال قد يكون غامضاً جداً لكن فكرته بسيطة أولاً افترض أنك تعلم عن list..each جيداً وتعرف صيغة التكرار الذي يستخدمهما .

الآن لدينا مصفوفة تتكون من رقمين للفهرسة هذين الرقمين كل واحد منهما عنصره عبارة عن مصفوفة هذه المصفوفة تحتوي على عنصرين (ولنتناسي أنهما يحتويان على أرقام فهرسة) وهما اسم شخص ورقم هاتفه .

في أول خطوة :

```
while (list($personnum) =each ($mon))
{
echo (“<br>$personnum”);
```

قمنا بإخراج رقم الفهرسة الأساسي للمصفوفة والذي يعتبر هو الرقم التسلسلي للاشخاص أصحاب الهواتف ومن بعد ذلك يقوم بطباعة هذا الرقم التسلسلي ويبدأ من سطر جديد .

في الخطوة الثانية :

```
while (list(,$phone)=each ($mon[$personnum]))
   {
    echo ("$phone");
    }
```

نقوم بإخبار الـPHP بطباعة العناصر الذي تحتويها المصفوفة التى تم طباعة رقم فهرستها ، ولاحظ **(,$phone)** أنها تشير إلى عناصر مصفوفة وليس فهارسها لأننا تجاهلنا فهارس المصفوفة الداخلية .
لاتقلق الأمر سهل ولكنه يحتاج الى تدرب فقط ، وعليك أن تتدرب وصدقني أنني حاولت ان أبسط المثال من أجلك ... أتمني أن تكون قد فهمت .

تطبيق عملي

افتح محرر النصوص لديك واكتب الكود التالي :

```
<?
Echo "<form method =post action = 'exam2.php' " ;
");حسن","سعد", "خالد" , "أحمد$boy=array ("
while (list(,$Name) = each ($boy))
{
";؟$Name لـ ماهي السنة الدراسيةecho "
Echo "<select name = 'school[]'>
</option>اول ثانوي<option>
</option>ثاني ثانوي<option>
</option>ثالث ثانوي<option>
</select>";
echo "<br><br>";
echo "<input type =hidden name =boy[] value ='$Name'>";
}
echo "<input type =submit ></form>";
?>
```

احفظ الكود باسم **exam.php**

افتح محرر النصوص واكتب الكود التالي واحفظه في ملف باسم exam2.php

```
<html dir = "rtl">
<?
While (list($I,$V)=each($school))
{
    $friendschool[] = $school[$I].$boy[$I];
}
asort ($friendschool);
While (list ($I,$V)=each($friendschool))
{
echo "<br>$boy[$I]"."  ".$school[$I];
}
?>
```

قم بتشغيله بعد نقله لمجلد السيرفر

الشرح

الذي قمنا به في المثال السابق هو أننا قمنا بإنشاء مصفوفة لعده أشخاص ($boy) ونريد أن نعرف مرحلهم الدراسية في الثانوية فأنشأنا لكل طالب قائمة منسدلة بواسطة التكرار (list-each) بصناعة قوائم منسدلة وحقول مخفية يتم تخزين قيم الحقول (التي تحتوي على أسماء الأشخاص) في المصفوفة ($boy) وسيتم تخزين نتائج كل القوائم في مصفوفة ($school) وبعد أن يختار المستخدم الإجابات التي تناسبه وارسال البيانات سيتم استقبال المصفوفة التي فيها نتائج القوائم المنسدلة ($school) واستقبال المصفوفة التي فيها أسماء الأشخاص ($boy) ومن ثم يتم إنشاء مصفوفة جديدة باسم $friendschool[] ويؤخذ منها معلومات المصفوفتين ويتم دمجها فيها ومن ثم يتم بتكرار آخر طباعة عناصر المصفوفتين $boy و$school .

الدالة count

تقوم بحساب عدد العناصر الموجودة في المصفوفة

مثال :

```
<?
$c=array("a","b","c");
$v=count($c);
echo $v;
?>
```

ترتيب الكود البرمجي

تعلمنا في الدروس السابقة أساسيات من أساسيات البرمجة واعطينا مثال عن الروتين في الحياة اليومية وهو أن تقوم بعمل شي أكثر من مرة في الحياة اليومية مثل شرب الشاي أو شرب القهوة وغير ذلك ، درسنا اليوم يتكلم عن ترتيب الكود ويتكلم تقريباً عن نفس فكره الروتين اليومي فأنت في حياتك تكرر بعض الأعمال بشكل روتيني

وقد تكون مللت الروتين فأحضرت شي يساعدك على التخفيف من هذا الروتين ... فمثلاً عند استخدامك لبرنامج MS Word قد تكون مللت من تنسيق عدة نصوص بطريقة معينة فأنت عند ذلك تقوم بصناعة ماكرو يقوم بفعل العمل الذي كنت تفعله في عدة خطوات بخطوة واحد فقط !!

ولنقل أنك في حياتك اليومية وفي يوم إجازة وقررت أن تقوم بعمل تنظيف شامل (يا إلهي عليك غسيل أطباق الصحون وتنظيف الأثاث وتنظيف الأرضيه وترتيب المكتبة وترتيب غرفة النوم و ... و ... الخ) عند ذلك فإنك تبحث عن طريقة عملية لكي يتم انجاز هذه المهمة في أسرع وقت فتقوم بتقسيم هذه المهمة الكبيرة على عده أقسام (التنظيف ،الترتيب ، الغسيل ،.....) ثم تقوم باستدعاء أطفالك وفلذات اكبادك وتقسم على كل واخد منهم مهمة بسيطة يستطيع القيام بها .. هذا التقسيم يسمي في عالم البرمجة بالـfunction (دالة أو وظيفة)

Function

الدالة هي جزء من كود البرنامج يتم تعريفه عن طريق المبرمج ليتم تنفيذ شي معين بواسطتها ، تقوم الدالة بأخذ قيم وتسمي (arguments معطيات) كمدخلات ،، ثم تقوم بعمل بعض التعديلات على هذه المدخلات وتقوم بإخراج قيمة أخري في أكثر الأحيان تقوم الدالة بأخذ القيم ووضعها في متغيرات أخري تسمي بالـ(parameters) لكي يتم اجراء العمليات عليها داخل الدالة وهذه المتغيرات لاتعمل خارج الدالة أي أنها متغيرات خاصه بالدالة فقط ! ...في دروسنا السابقه قمنا باستخدام دوال عديده مثل دوال فرز المصفوفات ودوال ايجاد نوع البيانات ،،،، هذه المرة سنقوم ببناء دوالنا الخاصة بنا ،، ومن صنعنا نقوم باعطاءها المعلومات والبيانات وهي تقوم باجراء العمليات عليها ومن ثم اخراج الحلول ...

تعريف واستدعاء الدوال

لكي تقوم بتعريف دالة فإنك تقوم بكتابة الكلمة function متبوعة باسم الدالة والبارمترات الازمة والتي سيتم اجراء العمليات عليها بين قوسين ومن ثم تقوم بكتابة الكود الازم وسط { و }

الصيغه :

```
Function functionname (parameters)
{
function code
}
```

تقوم بكتابه اسم الدالة بدلاً من functionname ثم تقوم بتعريف المتحولات أو المتغيرات parameters ومن ثم تقوم بكتابه الكود الذي سوف يقوم بالمطلوب بين القوسين بدلاً من function code

دعنا الآن نقوم بكتابة دالة من إنشاءنا والتي تقوم باجراء عملية الجمع على متغيرين وسنقوم بتسمية الدالة باسم sumnoraml وهو اسم من تأليفنا ويدل على وظيفة وهدف الدالة ويمكن أن تقوم بتمسية الدالة باي اسم تريده ولست مجيراً بكتابه اسم معين

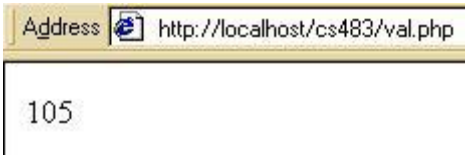
```
<?
Function sumnormal($a)
{
$a = $a + 100 ;
return $a;
}
?>
```

نقوم في هذه الدالة بإجراء عملية إضافة 100 على المتغير أو القيمة التي يتم تمريرها . سوف نحفظ هذا الملف بأسم fun.php

الان سوف نقوم بحفظ الملف التالي بأسم val.php ونحفظه في نفس المكان لملف الاول .

```
<?
include "fun.php";
echo sumnormal(5);
?>
```

وعند تنفيذ هذا الملف سوف نشاهد التالي :



Return

يجب أن نضعها في نهاية كل دالة ، نستخدم هذه الكلمة لكي نقوم بإعلام الداله ان وظيفتها انتهت وايضا نستخدمها إذا كان لدينا أكثر من قيمة ونريد أن نقوم بإخبار الـPHP ماهي القيمة التي سيتم اعتمادها ففي مثالنا هذا أردنا إخبار الـPHP بأن يقوم بأخذ المتغير $a بانه هو القيمة النهائية مع أنه لو لم نضع المتغير فسيتم اعتباره هو الناتج النهائي لانه لايوجد متغير اخر تم عليه أي عمليات

الذي اقصده أننا لو كتبنا الكود بالشكل التالى :

```
<?
Function sumnormal($a)
{
$a = $a + 100 ;
return ;
}
?>
```

فإنه لاضرر من ذلك لأنه لا يوجد لدينا إلا قيمة واحدة لن يتم اعتماد قيمة غيرها ولكن لو افترضنا أنه لدينا أكثر من قيمة كما في المثال التالى :

```
<?
Function sul($a,$b)
{
$a = $a + 100 ;
$b= $b*100;
return $a ;
}
?>
```

هنا يجب تحديد أي المتغيرين سيكون هو القيمة النهائيه للدالة .

```
<?
include "fun.php";
echo sul(10,2);
?>
```

في هذا المثال سوف سقوم بأخذ القيمه 10 ويجمعها على 100 ويظهر لنا الناتج والسبب هو تحديد المتغير $a المرجع فقط

شرح الداله (sumnormal)

تقوم الدالة التي صنعناها بأخذ قيمتين ومن ثم فإنها تقوم بزياده العدد الذي يتم تمريره 100

ولكي نقوم بإخراج نتيجة الدالة فإننا ببساطة نسطيع ذلك باجراء أحد الأمرين echo أو print .

مثال :

```
<?
Function sumnormal($a)
{
$a = $a + 100 ;
return ;
}
echo sumnormal(500);
?>
```

لقد قمنا بتمرير رقم بدلاً من المتغير ويمكننا أيضا تمرير متغير بدلاً من الرقم

مثال :

```
<?
Function sumnormal($a)
{
$a = $a + 100 ;
return ;
}
$f=100;
echo sumnormal($f);
?>
```

لاحظ أننا استخدمنا متغير في الدالة (مما يثبت كلامنا في الأعلى أن للدالة متغيرات خاصة بها) وليس معني ذلك أننا لانستطيع استخدام متغيرات بنفس الاسم المذكور في الدالة فيمكننا مثلاً كتابة نفس اسم المتغير بدون حصول أي مشاكل كالتالى :

```
<?
Function sumnormal($a)
{
$a = $a + 100 ;
return ;
}
$a=100;
echo sumnormal($a);
?>
```

يمكننا أيضا استدعاء دالة بشكل عادي إذا كانت هي تقوم بالطباعة

مثال :

```
<?
Function sumnormal($a)
{
$a = $a + 100 ;
print $a;
return ;
}

$a=100;
sumnormal($a);
?>
```

print

يقوم الأمر print بنفس عمل الدالة echo ولايوجد بينهما اختلاف سوى أن الدالة echo قديمة وهي الأصل أما الدالة print فقد تم إنشاؤها في php4 ولايوجد أي فرق بينهما اطلاقا .

مثال :

```
<?
"ماجدPrint "
?>
```

ويمكننا بها إخراج نتيجة دالة

```
<?
Function sumnormal($a)
{
$a = $a + 100 ;
return ;
}
$a=100;

print  sumnormal($a);
?>
```

اين يتم وضع الداله ؟

يمكنك وضع الدالة في أول الكود أو في آخرها أي أنه لافرق بين :

```
<?
لاحظ اننا قمنا بتعريف الداله اولا ثم استدعاءها//
Function majed($d)
{
print "admin@hotmail.com";
}

; majed($d)
?>
```

وبين :

```
<?
لاحظ اننا قمنا باستدعاء الداله اولا ثم تعريفها//
; majed($d)

Function majed($d)
{
print "admin@hotmail.com";
}
?>
```

يمكنك أيضا عدم وضع متغيرات في الدالة كالتالى :

```
Html_header ()
{
Print "<html><head><title>majed</title></head>";
Return ;
}
```

هذه الدالة تقوم بكتابة الطور الأول من صفحة html لاحظ أننا لم نقم بوضع أي متغيرات او عوامل او متحولات (سمها كما شئت) .

```
<?
include "fun.php";
echo Html_header();
?>
```

تمرير القيم الى الدالة

هناك نوعين من تمرير القيم

1 - تمرير القيمه مباشره الى الداله (passing by value)

وذلك أن نضع القيمة مباشرة بدون إدراجها في متغيرات .

مثال :

```
<?
Function majed($f)
{
$f=$f+$f;
return $f;
}
echo majed(100);
 ?>
```

لاحظ أننا قمنا بإدراج القيمة مباشرة للدالة من غير وضعها في متغيرات .

2 - تمرير القيمه عن طريق المرجع (passing by reference)

نقصد بهذا أننا نقوم بوضع القمية في متغير أولا ثم نضع هذا المتغير في الدالة لكي يتم اجراء العمليات عليه مثال :

```
<?
Function majed($f)
{
$f=$f+$f;
return $f;
}
$r =1000;
echo majed($r);
 ?>
```

اعداد قيمة افتراضيه للدالة

تستطيع أن تجعل الـ PHP4 يقوم بإدراج قيمة إفتراضية عند عدم تمرير متغيرات إليه

مثال :

```
<?
Function majed($f=40)
{
$f=$f+$f;
return $f;
}
echo majed();
?>
```

إذا لم يتم إعطاء قيمة للدالة فإنها ستفترض أن القيمة هي 40 مباشرة .

أما إذا تم تمرير قيمة أو متغير فإنه سيتم العمل بالقيمة التي تم تمريرها بدلاً من القيمة الإفتراضية

مثال :

```
<?
Function majed($f=40)
{
$f=$f+$f;
return $f;
}
echo majed(100);
?>
```

مدى المتغيرات (variable scope)

هناك متغيرات محلية (local) ومتغيرات عامة (global) ، نقصد بالمتغيرات المحلية التي تكون في داخل الدالة ونقصد بالعامة التي تكون في كود الـPHP بشكل عام

مثال

```
<?
هذا متغير عام//
$r= "Welcome";
function val($s)
{
هذا متغير محلى//
$s = "progrramer";
}
echo $r ;
val($s);
echo $s;
?>
```

مثال :

```
<?
هذا متغير عام//
$r= "majed";
function val($s)
{
هذا متغير محلى//
$s = "progrramer";
}
echo $r ;
$s=10;
echo $s;
?>
```

في المثال الأول استطعنا طباعة المتغير $r ولم نسطتع طباعة المتغير $s لأنه محلى (لايتم تنفيذه الا داخل الدالة) وعندما نريد طباعته فإننا يجب أن نطبع ناتج الدالة لكي نحصل عليه (أي أننا لانستطيع طباعته بشكل مباشر)

مثال :

```
<?
هذا متغير عام//
$r = "majed";
function val($s)
{
هذا متغير محلى//
$s = "programmer";
}
استطعنا طباعته بشكل مباشر//
echo $r ;
val($s);
يجب استخدام الداله لكي يتم طباعته//
echo val($s);
?>
```

لاحظ أننا حتي لو قمنا بعملية طباعة المتغير من نفس الدالة فالناتج يكون مختلف لأن لكل متغير عالمه الخاص به

لكي نقوم بجعل المتغير الذي بداخل الدالة متغيراً عاماً فيمكننا ذلك بإحدي الطريقتين التاليتين :

الطريقة الأولى :

```
<?
function val($y)
{
echo $y. "<br>";
global $s;
$s = "programmer";
return $s;
}
$f =10;
val($f);
echo $s;
?>
```

لاحظ أننا عندما استخدمنا **global** في داخل الدالة لكي يتم تعريف أن المتغير متغير عام وبعدما قمنا باستخدام الدالة قامت بطباعة المتغير المراد طباعته ومن ثم بعد ذلك قامت بتعريف متغير جديد ($s) وهذا المتغير متغير عام لأننا وضعنا قبله الكلمة **global** فاستطعنا طباعته بكل سهوله .

المتغيرات المستقره (static variable)

اقصد بالمتغيرات المستقرة هي التي تكون قيمتها ثابتة

مثال :

```
<?
Function  add($y)
{
$y;
$y=$y+1 ;
return $y;
}
echo add($y);
echo add($y);
echo add($y);
echo add($y);
?>
```


```
<?
Function  add($y)
{
static $y;
$y=$y+1 ;
return $y;
}
echo add($y);
echo add($y);
echo add($y);
echo add($y);
?>
```

لاحظ عندما عرفنا المتغير بأنه static فانه يحتفظ بقيمته حتي لو انتهت الدالة .

دوال متداخلة

يمكننا عمل تعشيش للدوال مثلما كنا نفعل مع بناء القرارات والتكرارات

مثال :

```
<?
Function sum($sa)
{
   $sa=$sa-1;
function goadd ($r)
{
$r = $r+$r;
return $r;
}
$sa= goadd ($sa);
return $sa;
}
echo sum (15);
?>
```

في مثالنا هذا لدينا دالتين الدالة الأولى هي sum والدالة الثانية هي goadd

وظيفة الدالة الأولى هي أن تقوم بالإنقاص من العدد الذي يمرر إليها واحد ثم تقوم بتطبيق دالة داخلية فيها هي goadd تقوم بزيادة العدد على نفسه .. ومن ثم قمنا بنداء الدالة الأولى (لأنها هي الاساس التي يوجد به الدوال الداخلية) وطباعة قيمتها .

اشتمال الملفات (include files)

قد يكون لديك في برنامجك متغير متكرر في أكثر من صفحة أو رسالة خطأ معينة أو تريد إدرج نص كبير الحجم في صفحات متعددة

هنا يمكنك اشتمال ملفات في داخل ملفات الـPHP . هذه الملفات قد تحتوي على نصوص أو كود html أو كود PHP .

إن الصيغة التي تستخدمها لاشتمال الملفات هي :

```
Include (filename);
```

مثال :

قم بفتح ملف نصي واكتب فيه ماتشاء ثم احفظه باسم a.txt
قم بانشاء ملف php واكتب فيه ومن ثم احفظه باسم b.php

```
<?
Include ("a.txt");
?>
```

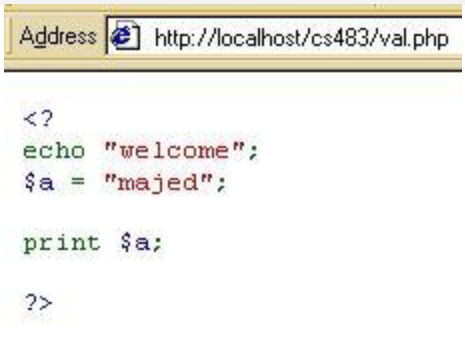
انقلهما الى مجلد السيرفر .. شغل ملف الـb.php وانظر النتيجة .
يمكنك أن تقوم بإنشاء ملف PHP وتحتفظ فيه بجميع الـfunction المطلوبة لبرنامجك وعند إرادتك لاستخدام أي واحدة منها تقوم فقط باشتمال الملف ومن ثم استدعاءها .

داله تلوين الكود

هل رايت مواقع تقوم بتلوين الكود بشكل مذهل مثل موقع zend ؟.... الأمر بسيط كل ماعليك أولا
قم بوضع الكود في ملف نصي وسمه باي اسم (مثلا file.txt) وبعد ذلك قم باستخدام الدالة
Show_source

مثال :

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



تتبع وتصيد ومنع الاخطاء

(avoiding and handling errors)

إن مصطلح الـ debbug هو من المصطلحات الشائعة والشيقة في عالم البرمجة ، هذا المصطلح يشير إلى كيفية إصلاح أخطاء البرنامج وتوقعها قبل حدوثها ، هناك أنواع من الأخطاء تحدث بسبب المبرمج وهناك أنواع من الأخطاء تحصل بسبب المستخدم ، في العادة يجب أن يكون المبرمج متآلفاً مع مصطلح تتبع الأخطاء وإصلاحها .

قد يكون من أهداف تتبع الأخطاء الحماية بقدر أهميه البرنامج الجاري العمل عليه أو الموقع فكلما كان الموقع مهماً كان وجوب حمايته أكبر .

قد يكون من الأسباب التي تسبب تدميراً للمواقع هو أن صاحب الموقع يغطي كل صغيرة وكبيرة عن برنامجه الذي يركبه في موقعه وقد يكون برنامجه هذا غير محمي بسبب كاف أو يكون مسير بعدة ملفات فيقوم شخص بحذف ملف من الملفات الأساسية بسبب عدم دقة في التراخيص المعطاة مما يؤدي إلى دمار الموقع نهائياً .

وقد يكون صاحب الموقع مهملاً في الحد ذاته فلا يحتفظ بالمعلومات السرية لموقعه مما يسبب مشاكل أكبر من التدمير مثل احتلال الموقع بشكل كامل .

رسائل الخطا في الـPHP لها طريقتها وتقنيتها الخاصة التي تسير عليها فهي ليست مثل الجافا وليست مثل cgi

فالـ PHP لا تقوم بإرسال الخطأ إلى السيرفر بل تقوم بكتابة رسالة خطأ في مكان الخطأ .

قد يكون هناك أخطاء يصعب تتبعها أو معرفة مكانها في الأصل ، وقد يكون هذا بسبب أنك تستخدم الـ PHP في صناعة موقع ديناميكي وتشرك معها الجافا سكربت وتضع علامات التعليق الخاصة التي تقوم بإخفاء الأخطاء في الجافا مما قد يجعلك تشعر بالحيره وتجن أين مكان الخطأ

```
<!- -

رساله الخطا

- - >
```

أنواع الأخطاء

هناك أنواع من الأخطاء منها الإملائية (Syntax Error) ومنها المنطقية ومنها أخطاء تحدث في وقت التنفيذ

ومثال الأخطاء الإملائية :

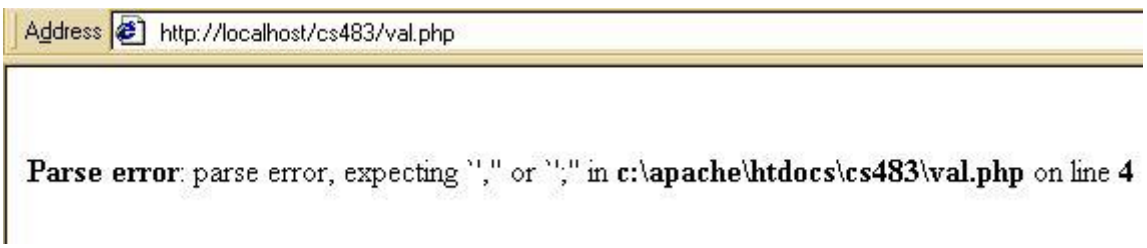
```
1 <?
2  Eco "1";
: من المفترض أن تكت التالي3  //
4  Echo "1";
5  ?>
```

هذا سيعطيك رسالة خطأ Parse error

ومن الأخطاء الإملائية نسيان الفاصلة المنقوطة (semi-colon) في نهاية الدالة :

```
1 <?
2  Echo "hello"
: من المفترض أن تكت التالي3  //
4  Echo "hello";
5  ?>
```

هنا سوف يعطيك الـ PHP رسالة خطأ لكن العجيب أنه لن يعطيك إياها بشكل صحيح فرسالة الخطأ تشير إلى أن السطر الرابع يحتوي على الخطأ بينما الخطأ هو في السطر الثاني .



وهناك خطأ آخر يحصل بسبب نسيان الـbrace (وهي الاقواس) :

```
1 <? Php
2 for ($loop = 0 ; $loop < 5 ; $loop ++ )
3 {
4 Echo "";
5 ?>
```

اذا كنت قد نسيت إغلاق القوس فهذا من الأخطاء الشائعة ، والأخطاء الإملائية لايمكن حصرها ، إنها أشبه بقواعد اللغة ، لكن أكثر الأخطاء الإملائية الشائعة في برامج الـPHP

1 – نسيان الأقواس .

مثال :

```
1  <?
2  for ($loop = 0 ; $loop < 5 ; $loop ++ )
3  {
4  for ($loop1 = 0 ; $loop1 < 10 ; $loop1 ++ )
5  {
6  for ($loop = 0 ; $loop < 5 ; $loop ++ )
7  {
8  code ....
9  }
10 }
11 ?>
```

في المثال السابق ينقصنا قوس إغلاق التكرار الأخير (})

2 - نسيان الفاصلة المنقوطة .

مثال :

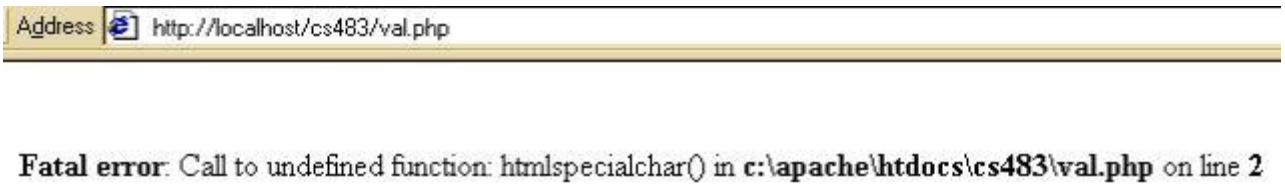
```
1 <?
2  Echo 10
3 <?
```

3 - خطأ إملائي في اسم function .

مثال :

```
1 <?
2  Htmlspecialchar($I);
3 ?>
```

سيعطيك رسالة خطأ :



وتصحيحها أن تكون :

```
1 <?
2  Htmlspecialchars($I);
3  ?>
```

4 - نسيان إغلاق النص .

مثال :

```
<?

Echo "PHPvillage;

?>
```

نسي الـ (") في نهاية الكلمة . وسيعطيك Parse error

الأخطاء المنطقية (Logical Errors)

إن الأخطاء المنطقية هي الأكثر صعوبة في التتتبع فقد تجد برنامجك يعمل بشكل صحيح وبكل سلامة ولكنه عند نقطة ما لايتم تنفيذها كما تريد أنت ، لنضرب مثلاً على خطأ منطقي بسيط جداً ، لنفرض أنك قمت بعمل نموذج مكون من مربع نص وزر ، عند ضغطك لهذا الزر فأنت تريد أن يتم كتابة كلمة كبير إذا كان الرقم أكبر من 30 وكلمة صغير إذا كان الرقم أصغر من 30 لنقم بكتابة الكود للمثال الأول :

```
<?
 :" ;ادخل عمرك echo "
echo '<br>
<form method = "post" action = "age.php">
<input type= "text" name = "age">
<br>
" >؟ هل أنا كبير أم صغير<input type= submit value = "
</form>' ;
?>
```

في ملف age.php اكتب الكود التالي :

```
<?
"; انت صغير If ($age<30) echo "
"; انت كبير If ($age>30) echo "
?>
```

سيعمل السيكريبت بشكل صحيح .. ولكن ربما تخطأ أنت في كتابة العلامات المنطقية (التي باللون الأحمر) فتأتي النتائج بشكا خاطئ

ومن الأخطاء المطنقية الأخطاء التي تقع في وقت التشغيل (Run times error) والتي تكون قد تقوم بإيقاف برنامجك بشكل كامل

مثال :

```
<?
$t=0;
$r=1;
$f=$r/$t;
?>
```

وعندها سينتج لك الرسالة التالية

هناك نوع آخر من الأخطاء المنطقية (unexpected) وهو لا يقوم بإيقاف البرنامج نهائياً بل يقوم بإخراج رسالة الخطأ في مكان الخطأ أو قد يقوم بتنفيذ البرنامج وإخراج البيانات بشكل غير صحيح أو قد لا يقوم بإخراج بيانات

وهو المثال الاول الذي ذكرناه سابقا (تقييم العمر) .

اخطاء التكرارات

قد يكون لديك أيضا تكرار فيه خطأ ولا يقوم بالتوقف نهائياً مثل هذا التكرار :

```
$c=1;
$t=true;
while ($t=true)
{
$c++;
}
```

لم نقم بعمل شي يوقف التكرار مثل أن تضع شرط يختبر قيمة المتغير ($c) ثم يقوم بإيقافه عند تعديه رقم معين وعلى ذلك فإن التكرار سيستمر بشكل غير متوقف ولن يعمل البرنامج .

عدم ارجاع قيمه من function

مثال :

```
<?
Function add($d)
{
$d =$d+$d;
}
```

الخطا هنا اننا لم نستخدم الـreturn لكي ننهي الدالة أو قد تكون الدالة تحتوي على أكثر من قيمة وننسي أن نقوم بتحديد القيمة النهائية للدالة

الخلط في المعاملات الحسابية والمنطقية

مثال :

```
If ($y=10) echo 12 ;
```

والمفترض أن تكون :

```
If ($y= =10) echo 12 ;
```

أفكار جيدة لتفادي الأخطاء

التعليقات

إن من الأفكار الجيدة للتقليل من الأماكن التي تبحث فيها عن الخطأ هو وضع تعليقات لوصف وظيفة دالة معينة .
مثال :

```
<?
//هذه الكود يقوم بطباعة كلمة ماجد
Echo " ماجد" ;
?>
```

الدوال

وأيضا من الأفكار الجيدة أن تقوم بتقسيم وظائف البرنامج على دوال بحيث أن لكل دالة وظيفتها المعينة :

```
<?
/*
+- ----------------------------------------------------+
|        هذه الداله تقوم بقسمه العدد علي 2        |
+- ----------------------------------------------------+
* /
function div($U)
{
$U=$U/2;
return $U ;
}
?>
```

هذه التقنية تساعدك على تفادي الأخطاء في صفحتك عند حدوثه مثل أن يقوم مستخدم ما بكتابة بريد الكتروني غير صحيح (مثال : a@y@.k.d) هذا البريد غير صحيح ولأجل أن تقوم بمنع حصول أي خطأ مثل ذلك وتقييد العبارات التي يدخلها المستخدم فإنك تقوم بإستخدام الـ RE (Regular Expressions) إنك بالأصح تجعل قواعد للكلمات التي يدخلها المستخدم فمثلاً تجعل المستخدم لايدخل سوي أرقام أو حروف فقط أو شكل معين من الكلمات ، تقوم أولاً بإنشاء نمط للكلمه التي تريد المستخدم أن يقوم بادخالها .

النمط (pattren)

ماهو النمط ؟ مارأيك إذا كتب المستخدم جملة في مربع نص تحتوي على عدة كلمات وتريد أن تتأكد من وجود كلمة معينة وسط هذه الجملة ، على حسب ما اخذناه من معلومات على المصفوفات سابقاً نستطيع فعل ذلك كالتالى :

```
<?
$words="one,two,three,four,five,";
$ty =explode (",",$words);
foreach ($ty as $w) {
if ($w ==  "five")
echo "found string 'five'";
}
?>
```

لقد كان المتغير $words يحتوي على جملة تتكون من عده كلمات وعندما أردنا فحصه قمنا باستخلاصه في مصفوفة ثم بعد ذلك قمنا بفحص المصفوفة باستخدام التكرار foreach ، ومع ذلك الذي فعلناه فإن هذا الاستخدام غير عملي بتاتاً وهنا تبرز قوه Regular Expressions لاحظ الآن كيف نستخرجه بواسطة الـ Regular Expressions :

```
<?
$words=”one, two, three, four, five,”;
if (ereg(“one”,$words))
‘ one’ “ ;  لقد وجدت العددecho “
?>
```

في هذا المثال قمنا باستخدام الدالة (ereg) ووضعنا في خانتها الأولى النمط (pattern) الذي نريد أن نتأكد من وجوده (أوالكلمة المراد البحث عنها) ووضعنا في الخانة الثانية المتغير الذي سيتم البحث فيه عن الكلمة أو النمط .

تقوم الدالة ereg بإعطاء القيمة true إذا تم العثور على الكلمة .

في الواقع هناك استخدامات أكثر فعالية للأنماط .

يمكننا مثلاً تخزين الكلمة إذا تم وجودها في مصفوفة خاصة كالتالى :

```
<?
$words=”one, two, one, four, five,”;
if (ereg(“one”,$words,$rok))  ;
echo $rok[0];
echo $rok[1];
?>
```

نقوم بوضع اسم المصفوفة التي نريد تخزين البيانات في الخانة الثالثة .. لاحظ مع أنه يوجد كلمتين في الجملة توافق النمط إلا انه أعطانا كلمة واحدة فقط إذ أن وظيفته أن يتأكد من وجود النمط في الجملة فقط فإذا تأكد من وجودها مرة واحدة استكفى واعتبر الموضوع قد انتهي .

ماذا لو أردنا من التأكد من عدة كلمات ، عند ذلك فإننا نفعل التالى :

```
<?
$words=”one, two, one, four, five,”;
if (ereg(“one”,$words,$rok))  echo $rok[0];
if (ereg(“two”,$words,$rok))  echo $rok[0];
?>
```

واريد أن أنبهك أن الـereg يقوم بإنشاء المصفوفة من جديد عند كل استعمال له فخذ حذرك من هذه النقط

أيضا فإن الـereg حساس لحالة الأحرف لاحظ هذا المثال :

```
<?
$words="one, two, vcx, four, five,";
if (ereg("One",$words,$rok))  echo $rok[0];
?>
```

لن يقوم بإخراج أي شي فقط لأن حرف الـO مختلف .

أيضا يمكنك البحث عن كلمة يسبقها فراغ مثلاً كالتالى :

```
<?
$words="one, two, vcxone, four, five,";
if (ereg("one",$words,$rok))  echo $rok[0];
?>
```

مثال آخر :

```
<?
$words="oned, two, vcxone, four, five,";
if (ereg("one",$words,$rok))  echo $rok[0];
?>
```

لاحظ في هذين المثالين أنه مع أن كلمة one غير موجودة بمفردها إنما موجودة كجزء من vcxone وoned ورغم ذلك فإن الدالة لم تأخذ اعتبارا لذلك بينما لو كتبنا كالتالى :

```
<?
$words="oned, two, vcxone, four, five,";
if (ereg(" one",$words,$rok))  echo $rok[0];
?>
```

فإنه سيبحث عن الكلمة مفصولة عن أي حرف ولن يجد كلمة كذلك فلن يقوم بكتابة أي شي .

يمكننا أن نفحص قيمة موجودة في متغير كالتالى :

```
<?
$reu = "one";
$words="one, two, vcxone, four, five,";
if (ereg($reu,$words,$rok))  echo $rok[0];
?>
```

هل لاحظت أننا فحصنا قيمة المتغير $rue بواسطة ereg مع $word ولم يتطلب منا ذلك أي شي إضافي غير اسم المتغير المراد البحث عن قيمته في الجمله .

يمكننا بالـ Regular Expression استعمال بعض الأحرف بشكل خاص التي لها استعمالها الخاص بواسطة الـ Regular Expressions

الأحرف الخاصة في الـ Regular Expression هي كالتالى :

```
. * ? + [ ] ( ) { } ^ $ | \
```

هذه الأحرف لها معناها الخاص في الـ Regular Expression

فقديماً مثلاً كنا نقول أنه لا يمكننا أن نستخدم علامتي تنصيص متداخلة من نفس النوع كالتالى :

```
<?
$r=”u\””;
?>
```

ولكي يتجاهل الـPHP هذا المعني فإننا نقوم بوضع (\) قبل علامة التنصيص .

ايضا مع الـ ereg فإن للـ(.) قداستها ولكي يتم تجاهلها فإننا نستخدم الـ(\)

تقوم الـ(.) بأخذ مكان حرف أو فراغ فمثلاً لاحظ المثال التالى :

```
<?
$P="I love yamen";
if (ereg ("love....",$P,$R)) echo $R[0];
?>
```

هل لاحظت الناتج ؟؟

ولكي يتم تجاهل قداسة الـ(.) في الـ Regular Expressions نقوم بوضع (\) قبلها . مثال :

```
<?
$P="I love yamen";
if (ereg ("love\.\.\.\.",$P,$R)) echo $R[0];
?>
```

في هذا المثال لن يتم طباعة أي شي لأنه لايوجد أي كلمة تطابق (love....) لأن الـ(.) فقدت قداستها وبدأ التدقيق في الكلمة حرفاً حرفاً .

صناعة فئة حروف [xyz]

أقصد بذلك أنني احدد نطاق معين من الكلمة من الممكن أن يكون في هذا النطاق أي حروف من الفئة التي أقوم بتحديدها أو الحروف التي أقوم بتحديدها .

مثال :

```
<?
$y=”how are you ? “ ;
if (ereg(“h[oe]” , $y)) echo “true”;
?>
```

هنا قام الـregular expression بالبحث عن أي كلمة تبدأ بالحرف h ومن ثم يتبعها أحد الحرفين o أو e مثال هذه الكلمات :

Hey – He – Hew - Homer

ولكنها لاتطابق :

Hty – Hnt - Hlay

أتمنى أن تكون فهمت ما أرمي إليه

يمكننا أيضا أن نقوم بإخبار الـregular expression بأن لا يقوم باختيار كلمات تحتوي على حروف معينة وذلك فقط بإضافة ^

```
<?
$y=”how are you ? “ ;
if (ereg(“h[^oe]” , $y))
echo “true”;
else
echo "false";
?>
```

نقوم هنا بإخبار الـre بأن يقوم بفحص الجملة فإذا وجد أي كلمة تبدأ بـh ولاتحتوي على o أو e فإنه يقوم بإعطاء true وإذا لم يجد يقوم باعطاء false

وهذا الكلام يطابق الكلمات التالية :

Hay - Hana - Hkg

ولايوافق هذه الكلمات :

Home – Hore - Here

يمكننا استعمال اختصارات لبعض الأمور فمثلاً إذا كنا نريد كلمة لاتحتوي على أي رقم كنا سنكتب كالتالى

[^123456789]

يمكننا أن نستعمل اختصار لهذا الموضوع كالتالى :

[^0-9]

وحتي إذا أردنا أن يتأكد من وجود رقم من واحد الى تسعه فقط علينا مسح الـ^

[0-9]

وأيضا الحروف الصغيرة من a الى z

[a-z]

وإذا نريد التأكد من عدم وجودها

[^a-z]

نفس القصة مع الحروف الكبيرة .

هناك اختصارات اخري لهذا الموضوع كالتالى :

الاختصار	المطابق له	معناه ووظيفته
\d	[0-9]	أي رقم من 0 الى 9
\D	[^0-9]	ممنوع الأرقام من 0 الى 9
\w	[0-9A-Za-z_]	أي رقم من 1-9 أو حروف A-Z او احرف صغيره او_
\W	[^0-9A-Za-z_]	عكس السابق
\s	[\t\n\r]	يقبل مسافة أو سطر جديد أو علامة جدولة (tab)
\S	[^\t\n\r]	عكس السابق

تحديد مكان الكلمة

يمكننا أن نقوم بتحديد مكان الكلمة ، اقصد بذلك أنه يمكنك تحديد مكان الكلمة إذا كانت في بداية أو نهاية النص ونستخد لهذا الأمر العلامتين (^) لتحديد المكان لبداية الجملة و ($) لنهاية الجمل .

مثال :

```
<?
$y="how are you ? " ;
if (ereg("^h",$y)) echo "true";
?>
```

هنا سيقوم الـphp بالبحث عن في الجملة فإذا وجد الجملة تبدا بحرف h كانت قيمة الـereg تساوي true وإذا لم يجد كانت قيمة الـereg تساوي false

```
<?
$y="how gone?" ;
if (ereg("^g",$y)) echo "true";
?>
```

في هذا المثال ستكون قيمة الـereg خطأ لأن العبارة لاتبدأ بحرف g
يمكننا فعل العكس بواسطة العلامه ($) التي عملها عكس (^) فهي تفحص إذا كان الحرف المراد فحصه موجود في نهاية الجملة

مثال :

```
<?
$y="how g" ;
if (ereg("g$",$y)) echo "true";
?>
```

يمكننا أيضا اختيار إذا ما كان واحد من نمطين صحيحاً بواسط العلامة (|)

```
<?
$y="how g" ;
if (ereg("^y | g$",$y)) echo "true";
?>
```

في هذا المثال سيقوم الـPHP بفحص الجملة فإذا وافقت أحد النمطين كانت قيمة الـereg عند ذلك true .

يمكننا أيضا تحديد إذا ما كان حرف أو جملة متكررة بعدد من المرات أو مره واحدة باستخدام أحد هذه الثلاث رموز (* ، + ، ?)

تقوم علامه الضرب بالتحقق من أن الحرف الذي يسبقها مكرر مرة أو أكثر أو غير موجود بتاتاً

مثال :

```
Bea*t
```

وتوافق :

Bet

Beat

Beaat

تقوم علامة الجمع (+) بالتأكد من وجود عنصر مرة أو أكثر :

```
Bea+t
```

وتوافق :

Beat

Beaat

Beaaaaat

أما علامة الاستفهام (?) فتقوم بالتأكد من وجود عنصر مرة واحده أو عدم وجوده بتاتاً :

```
Bea?t
```

وتوافق :

Bet

Beat

وتأكد دائماً أن هذه الثلاث علامات مسبوقه بحرف .

وعند إرادتك مثلاً التأكد من سبق حرفين أو ثلاث بشكل تحديدي يمكنك استخدام القوسين

مثال :

```
(wo)?man
```

ويوافق :

man

woman

يمكننا التأكد من تكرر حرف بشكل معين من المرات أو أكبر من عدد معين من المرات أو أصغر من عدد معين من المرات باستخدام القوسين {x,y}

فمثلاً لو أردنا أن نتأكد من أن حرف (d) مكرر مرتين إلى أربع مرات :

```
d{2,4}
```

أما إذا أردنا أن نتأكد من أنه مكرر أكثر من مرتين إلى عدد غير محدود من المرات :

```
d{2,}
```

أما إذا أردناه أن يتكرر 4 مرات على الأكثر :

```
d{,4}
```

أو إذا اردناه أن يتكرر بعدد محدود من المرات :

```
d{8}
```

أخيراً نريد أن نلفت النظر إلى الاختصار (\b) الذي معناه أي شي ولكن ليس حرفاً (الحروف التي بين \w وبين \W تقريباً)

ملخص ما أخذناه من القواعد تجدونه في الجدول التالى :

القاعده	المعني
[abc]	أي حرف كان a او b او c
[^abc]	أي حرف غير a و b و c
[a-z]	كل الحروف من a الى z
\d\D	\d للارقام و \D لغير الارقام
\w\W	\w للحروف جميعها و\W لغير الحروف
\s\S	\s للفراغ (space) و\S لغير الفراغ (no space)
\b	الحروف التي بين \w و\W
.	أي حرف
(abc)	تقوم باعتبار abc كمجموعه ..
؟	حرف او مجموعة حروف مكرره مره اوغير مكرره نهائيا
+	حرف او مجموعة حروف تتكرر مره او اكثر
*	حرف او مجموعة حروف تتكرر مره او اكثر او قد لاتتكرر نهائيا
{x,y}	تكرير بعدد معين من المرات ..
{,y}	تكرير بحد اقصي من المرات ..
{x,}	تكرير بحد ادني من المرات ...
{x}	تكرير بعدد معين من المرات
^	في بدايه النص
$	في نهايه النص

تعبير للتاكد من ايميل

```
^[_a-zA-Z0-9-]+(\.[_A-Za-z0-9-]+)*@[a-zA-Z0-9-]+(\.[a-zA-Z0-9-]+)*$
```

شرح التعبير

الرمز	الشرح
^	يجب ان يبدا النص
[_A-Za-z0-9-]	أي حرف من a-z كبيرا كان او صغيرا او _او ارقام
+	وقد يكون هذا الحرف متكررا اكثر من مره
(\.[_A-Za-z0-9-]+)	بالاضافه الى انه قد يتبع النقطه وحروف وارقام
*	وقد لايتبعه اوقد يتبعه ويتكرر اكثر من مره
@	وبعد ذلك يكون لديه حرف ال@
[a-zA-Z0-9-]+(\.[a-zA-Z0-9-]+)*$	وايضا نفس القواعد في النهايه

مثال :

```
<?
Function mailcheck($mail,$t)
{
$T="^[_a-zA-Z0-9-]+(\.[_A-Za-z0-9-]+)*@[a-zA-Z0-9-]+(\.[a-zA-Z0-9-]+)*$";
If (EREG($T,$mail))
{
$r="the mail is true";
echo $r;
}
else
{
$r="the mail is not true";
echo $r;
}
return ;
}
mailcheck("admin@hotmail.com",$t);
echo "<br>";
mailcheck("ad#min@hotmail.com",$t);
?>
```

the mail is true
the mail is not true

eregi()

الفرق بين هذه الدالة والدالة ereg أنه غير حساسة لحالة الأحرف كبيرة أو صغيرة أي أنه يمكننا كتابة المثال السابق كالتالى :

```
<?
Function mailcheck($mail,$t)
{
$T="^[_a-zA-Z0-9-]+(\.[_A-Za-z0-9-]+)*@[a-zA-Z0-9-]+(\.[a-zA-Z0-9-]+)*$";
If (eregi($T,$mail))
{
$r="the mail is true";
echo $r;
}
else
{
$r="the mail is not true";
echo $r;
}
return ;
}
mailcheck("admin@hotmail.com",$t);
echo "<br>";
mailcheck("ad#min@hotmail.com",$t);
?>
```

ereg_replace()

ماذا لو أردت تحرير عبارة ما من أحرف معينة وقد تكون متكررة في جملة أوغير ذلك

لنفرض أن لدينا العبارة التالية :

```
Majed love his game .....
```

ونريد أن نتخلص من النقاط التي في نهاية العبارة

أو لدينا مثلاً هذا المسار :

```
C:\windows\desktop
```

ونريد أن نستبدل العلامة (\) بـ (/)

كل ذلك ممكن بواسطة الدالة ereg_replace وقواعد الـ rgular expression التي أخذناها سابقاً

البنية التي نستخدمها للدالة كالتالى :

```
Ereg_replace(reg,string,var);
```

نضع في مكان reg القاعدة للـ regular expression ونضع مكان الـstring الحرف الجديد ونضع بدلاً من الـvar المتغير الذي نريد استخلاص الحروف منه .

مثال :

```
<?
$path =" C:\windows\desktop";
$tell= "Majed love his game .....";
$newpath= Ereg_replace("[\.]","/",$path);
$newtell= Ereg_replace("\.","",$tell);
echo $newpath;
echo "<br><br>";
echo $newtell;
?>
```

C:/windows/desktop

Majed love his game

أساليب أخري لتتبع الأخطاء

استخدام عباره echo

هو من أقدم الاساليب وكان يستخدم مثلاً في فحص بعض متغيرات نموذج فمثلاً أنت لديك نموذج يقوم بإرسال معلومات إلى النموذج وقد تستخدم في اختبار الأخطاء المنطقية التي يستصعب متابعتها في الكود

مثال :

```
<?
Echo "this is : $name";
Echo "<br>";
Echo "this is : $Email";
//كود يقوم بمعالجة معلومات المتغيرين
//طباعة المتغيرين بعد اداء عملية المعالجة ورؤية النتائج
Echo "this is after : $name";
Echo "<br>";
Echo "this is after: $Email";
?>
```

فحص كود الـhtml

قد تستخدم كود جافا سكربت ويتم إخفاء الأخطاء وسط علامات التعليقات فعليك حينئذ فحص كود الـhtml لرؤية إن كان هناك بعض الأخطاء المخفية أم لا .

تجاهل الأخطاء

لنفترض أنك تعلم أن الدالة التي صنعتها بها أخطاء ولكنك تريد تجاهل هذه الأخطاء فكل ماعليك أن تقوم بوضع @ أمام الدالة لكي يتم تجاهل الخطأ عند حدوثه .

مثلاً نحن نعلم أن القسمة على الصفر من الأشياء الغير مقبولة في الـPHP وأنت صنعت دالة تقوم بالقسمة على صفر ولن يتم تنفيذها لأنها بالأصل خطأ ولكنك تريد أن يقوم PHP بتجاهلها فكل ما عليك أن تفعله هو وضع @ أمام الدالة .

مثال :

```
<?
function amail ($y)
{
$y=$y/0;
return $y;
}
$s= @amail(44);
echo $s;
?>
```

التعامل مع العميل

كما رأينا في الدروس السابقة ، فإن الـ PHP يوفر رقم عظيم من المميزات عن الـ html لبناء مواقع الويب ، من الأشياء الأساسيه التي لم نتكلم عنها حتي الآن هي الموثوقية (أو الاستقرار) وهو بالمعني الصحيح والصريح :

القابيلة على الاحتفاظ بالمعلومات بين صفحتين منفردتين أو مختلفيتن في المستعرض ...

بدون أي إضافات ، HTTP لايوفر أي ميكانيكيه للحفاظ على البيانات وجعلها مستقرة لمعالجة تتم بين صفحتين ، كل طلب لصفحة في الانترنت (request) ليس له أي علاقة بأي طلب آخر ... مثلاً عندما تتطلب موقع المطور العربي ومن ثم منتدي المطور العربي فان كل الطلبين ليس لهما علاقه ببعضهما ...

بمصطلح آخر يمكننا أن نقول أن الـHTTP فاقدة لحالتها (stateless) أي أنها لاتعرف أي أن أمر طلب الصفحة ينتهي عند انتهاء الطلب ، فهي عندما تقوم بنقل بيانات صفحة من السيرفر الى المستخدم فهي تعرف من هو المستخدم الذي يطلب البيانات وعلى أي نافذه سيتم نقل البيانات وعند انتهاء ذلك فان كل هذا الموضوع ينتهي وإذا عاد المستخدم فطلب صفحة أخري فإنها لاتعرف إن كان هو نفس المستخدم أو لا !

إن القدرة على الحفاظ على وجود البيانات ليست وسيلة أو ميزة او قوة مقتصرة على الـPHP فقط .

فلقد رأيت كيف استطعنا ارسال معلومات من صفحة إلى صفحة بدون خسران أي معلومات وذلك عن طريق الـ html وبالرغم من ذلك فإن المستخدم عندما يقوم بإغلاق الصفحة عند استقبالها للبيانات فان ذلك يعني فقدانها للأبد ، عن طريق استخدام الـ PHP يمكننا اخبار السيرفر بأن يقوم بارجاع البيانات بطريقة تمكننا من الحفاظ عليها ، مثلما سنري في هذا الدرس ، هناك ثلاث طرق لعمل ذلك ....

التميز الحقيقي في قوة الفهم للـ PHP ، يتطلب منا مفهومية جيدة في كيفيه استعمال الـ PHP في التفاعل مع المستخدم والمتصفح الذي يستخدمه لكي نتغلب على نقاط الضعف التي في الـ http .

هذا هو موضوعنا لهذا اليوم والذي سنتكلم فيه عن :

1 - الـHTTP والـ html ومحدودية قدراتهم ، وكيف يستطيع الـPHP التغلب علي القصور فيهم .

2 - الاحتفاظ بالمعلومات التي نريد أن نستخدمها بين طلب لصفحتين مختلفيتن .

3 - مكنكة الحفاظ على البيانات .

4 - الكعكات (cookies) وكيفية استخدامها .

5 - PHP4 والـ native session – المكنكة الداخلية للحفاظ على وجودية البيانات .

هذا الدرس مفيد بشكل ظاهري لمن هو جديد على انشاء مواقع متفاعلة متوسطة – كبيرة الحجم بواسطة الـPHP ..

إنه يحتوي على الكثير من بعض الأمثله التي تفيدك .

الهدف من هذا الدرس هو أن تتعرف على كيفية الحفاظ على معلومات المستخدم عبر متغير أو أكثر بين أكثر من صفحة ، مثل أن تجعل اسم المستخدم ظاهر في كل صفحة يقوم بالولوج إليها ... مما يؤكد استمرارية وجود البيانات .

لنفرض أن لدينا موقعاً على الانترنت هذا الموقع يهتم ببيع وتسويق مواد غذائية أو أن هذا الموقع يقدم مسابقات ثقافية ، في العادة عندما يقوم المستخدم بطلب شراء سلعة معينة أو عندما يختار الدخول في مسابقة من المسابقات الثقافية فإنه يقوم بدخول أكثر من صفحة بالتتابع

يختار السلعة في الصفحة الأولى وبعد ذلك يقوم برؤية معلومات السلعة في الصفحة الثانية والصفحة الثالثة يقوم فيها بتعبئة معلوماته للشراء أو غير ذلك إلى أن ينتهي من كافة المعلومات وبعد ذلك تنتج له في النهاية صفحة فيها معلوماته والسلعة التي قام باختيارها وفاتورة شراء !!

أو يقوم باختيار نوع المسابقة الثقافية في الصفحة الأولى وبعد ذلك يقوم بالحصول على عدة أسئله مقسمة على عدة صفحات إلي أن ينتهي من المسابقة فتخرج له في النهاية مجموع الدرجات للأسئلة ومعلوماته وهل هو فائز أم خاسر!!

في الواقع هذا مايسمونه بالمحافظة على الجلسة (maintain session) وأقصد بذلك دخول المستخدم إلى صفحة وانتقاله من صفحة إلى صفحة مع المحافظة على معلوماته وغير ذلك من البيانات ، لكي نستطيع متابعته أولاً بأول .

في بروتوكول الـhtml والـhttp لانستطيع معرفة إذا ما كان الشخص عندما يطلب صفحة ما هو نفسه عندما يذهب إلى الصفحة الثانية إذ أن المستخدم عندما يطلب صفحة ما (request) من السيرفر فإن السيرفر يقوم بمعرفة من

أي مكان بالعالم يتكلم هذا الشخص ويقوم بارسال استجابه إليه باعطاءه الصفحة التي كان يطلبها (response) ولكن بعد ذلك فإن السيرفر لا يعرف إذا كان هذا الشخص هو نفسه الذي يقوم بطلب الصفحه الثانية أو الثالثة في السيرفر .

هنا تأتي ميزة الـ PHP وغيره من لغات برمجة الانترنت لصناعة ميكانيكية إبقاء تفاعل مستمر بين المستخدم والسيرفر عن طريق الـsession و الـcookie ، ولكي لا نعقد الموضوع دعونا نتكلم عن ذلك عملياً فذلك أفضل لفهم الموضوع من الثرثرة التي لا فائدة منها .

استخدام الحقول المخفيه

سنقوم الآن بإنشاء ثلاث صفحات ، الصفحة الأولى تطلب من المستخدم ادخال اسمه ، والصفحة الثانية تقوم بالترحيب به واعطاءه ثلاثة أسئلة ، والصفحة الثالثة تقوم باعطاءه النتيجة .

افتح محرر نصوص لديك واكتب الكود التالي :

```
 </p>ادخل اسمك الكريم<p dir="rtl" align="center">

<form method="POST" action="quiz2.php">

<hr>

<input type="text" name="name" size="20"><br>

" ></p>إرسال<input type="submit" value="

</form>
```

احفظها باسم quiz.php

قم بفتح محرر النصوص واكتب الكود التالي :

```
<html dir ="rtl">
<?
If  (isset($name)) {
" . $name ;مرحبا بك يا Echo "
 Echo '
<br>
<form method="POST" action="quiz3.php" dir="rtl">
<input type="hidden" name = "thename" value = "'.$name.'">"
 </p>؟من هو أول الخلفاء الراشدين
" أبوبكر الصديق  <p dir="rtl"><input type="radio" value="
أبوبكرname="khlifa">
 .</p>الصديق
" عمر بن الخطاب  <p dir="rtl"><input type="radio" value="
عمرchecked name="khlifa">
</p>بن الخطاب
</p>؟من هو الفاروق  <p dir="rtl">
عمر بن  <p dir="rtl"><input type="radio" name="faroq" value="
الخطاب">عمر بن
</p>الخطاب
" سالم  <p dir="rtl"><input type="radio" name="faroq" value="
سالم بنchecked>
</p>عامر
" dir="rtl">إرسال<input type ="submit" value = "
</form>' ;
}
else
{
" ;غير مصرح لك بدخول هذه الصفحةecho "
}
?>
```

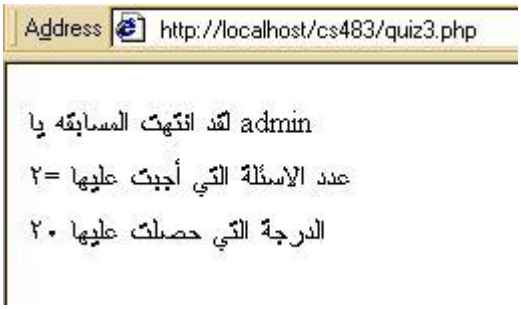
احفظها باسم quiz2.php

قم بفتح المفكرة واكتب الكود التالي :

```
<?
If  ((isset($thename)) && (isset($khlifa)) && (isset($faroq)))
{
 '. $thename ;يا لقد انتهت المسابقه echo '
$range=0;
$co = 0;
") {أبوبكر الصديق        if ($khlifa =="
        $range=$range+10;
        $co = $co +1;
        }
")عمر بن الخطاب   if ($faroq =="
             {
             $range=$range+10;
             $co=$co+1;
             }
          if ( $range < 10)
          {
";ليس هناك أي إجابة صحيحة          echo "
          }
          else
          {
 =" . $co ;عدد الاسئلة التي أجبت عليها          echo "<br>"."
 " . $range ;الدرجة التي حصلت عليها          echo "<br>" . "
          }
}
?>
```

احفظها باسم quiz3.php

قم بوضع الملفات الثلاثه السابقه في مجلد السيرفر ثم قم بتشغيلها



الشرح

قمت في هذا المثال بمحاولة صنع مكنكة تواصل للبيانات ، بمعني أنني أحاول أن أقوم بالاختفاظ بالبيانات عبر الثلاث صفحات بشكل متواصل ، لاحظ أنني كنت اختبر في quiz2 و quiz3 باختبار المتغيرات قبل طباعة أي شي فقد يقوم المستخدم مثلاً بالاحتفاظ بالصفحة التي وصل إليها في المفضلة ثم يقوم باكمال المسابقة في وقت آخر ولكني لا اريد ذلك بل أريد ان أجعل وقتها محدوداً (طبعاً هذا الكلام سيحصل إذا كانت المسابقة طويلة) لذلك فإنني في كل عند الانتقال من صفحة إلى صفحة اقوم باختبار إن كانت جميع هذه القيم موجودة ولاحظ أنني كنت اجتفظ دوماً بقيم المتغيرات في متغيرات جديدة في حقول مخفية وكلما كان عدد المعلومات أكبر في كل مره كان عدد الحقول المخفية أكثر ، إن لهذه الطريقة أيضا مشاكلها فقد يفتح المستخدم كود الـhtml ويقوم بتفحص كيفية ملاحقته عبر المسابقة وقد يصنع هو الكود في وقت لاحق لكي يستطيع اكمال المسابقة بهذه الخدعة الماكرة ... لذلك يفضل أن لا تقوم بذلك وتقوم بجعل المسألة السابقة اكثر تعقيداً باستخدام الـ regular expression بمحاولة تلغيم البيانات بواسطته ومن ثم فك هذا التلغيم في الصفحات التي تصل اليها البيانات .

ارسال بيانات بواسطة query strings

نستطيع ارسال بيانات بسيطة بواسطة الاستعلامات التي نقوم بإضافتها الى اسم الصفحة في الأعلى متبوعة بـ(؟) علامة استفهام ثم اسم متغير وقيمته وإذا كان هناك أكثر من متغير يتم الربط بينهم بعلامة & وراجع درس النماذج لمزيد من المعلومات .

قم بعمل صفحة وسمها ask.php وقم بكتابة الكود التالى فيها :

```
<?
If (isset($ask)) {
 If ($ask == login) {
";الصفحة إلى الدخول تسجيل تم Echo "
 }
}
if (!isset($ask)) {
" . "<br>";الصفحة إلى الدخول تسجيل يتم لم echo "
 دخولك تسجيل ليتم هنا اضغط Echo "<A HREF=$PHP_SELF?ask=login>
</a><br>";
}
?>
```

قم بتجربة هذا المثال على موقع يدعم PHP على نظام تشغيل لينوكس إذا لم يعمل بشكل جيد على الوندوز

لاحظ أننا في أول الولوج الى الصفحة لم نستخدم أي استعلامات وعند الضغط على الرابط قام الرابط بارسال قيمه المتغير الذي يقوم الPHP باختبارها فاذا وجد انه قد تم ارسالها (بواسطه الرابط الذي تم الضغط عليه) قام بطباعة (تم تسجيل الدخول) واذا لم يجدها قام بطباعة (لم يتم تسجيل الدخول) بالإضافة إلى طباعة الرابط الذي يحتوي على المتغير في طياته

الكوكيز أو الكعكعات (cookies)

إذاً ماهي الكوكيز ، الكوكيز هي عبارة عن بعض المعلومات أو القطع الصغيرة من البيانات يتم الاحتفاظ بها في جهاز العميل لكي يتم الاحتفاظ بها عند الزيارات المختلفة للمستخدم (العميل) ، أنت لا تقوم بالاحتفاظ فيها بقيم ضخمة لكنك تستفيد منها في أشياء أخري مثل :

1 - جعل لكل مستخدم الألوان الخاصة التي يري فيها صفحتك (أي أن تجعل للمستخدم مثلاً إعدادات الألوان الخاصة لرؤية موقعك)

2 - جعل مفتاح للمستخدم لكي يستطيع به التحكم في بياناته الخاصة عند زياراته لموقعك في مرات اخري.

الكوكيز مفيد للاستخدام في الأشياء البسيطة والغير خطيرة ، لكنه الآن يستخدم بشكل سئ ، مثل استخدامه مثلاً في معرفة معلومات عن المستخدم بدون علم منه ، أو تخزين كميات كبيرة من البيانات فيه والتي من الأجدر أن يتم حفظها في ملف على السيرفر .

ويكون استخدامه مفيدا عندما تضمن أن جميع زوار موقعك تسمح متصفحاتهم بالكوكيز (مثل طلبة المدارس أو شبكات انترانت) .

عندما يكون فقط لأشياء بسيطة لاضرر منها عند عدم السماح بالكوكيز بجهاز العميل .

بدايتك مع الكيوكيز

قبل أن نبدأ علينا معرفة بعض الأساسيات عن الكيوكيز

الكوكيز عبارة عن قطعة صغيرة من البيانات التي تستخدم لتخزين اسم متغير وقيمته مع معلومات حول الموقع التي أتت منه وتاريخ انتهاءها .

الكوكيز عباره عن تقنية للتخزين من جهة العميل (client-side storage) تتخزن في ملفات في جهاز العميل

يتم العبور إلى هذه الكوكيز ومسحها من المكان التي ارسلت منه .

عندما يطلب المستعرض صفحة من السيرفر وهذه الصفحة تقوم بتتخزين كوكيز فإن السيرفر يقوم باخبار المستعرض بأنه سيقوم بوضع كوكيز للاستعمال لاحقا .

عندما يتم طلب الصفحة في مرة آخري يقوم المستعرض بارسال البيانات التي تم إنشاؤها سابقاً عند طلب الصفحه .

يتم انتهاء مده الكوكيز بإنتهاء وقت صلاحيتها االمحدد من قبل السيرفر ويتم مسحها فورياً عند اغلاق الصفحة إذا كان وقت صلاحيتها صفراً من الثواني .

بإختصار عندما يعطي السيرفر الكوكيز للمستعرض فإنه يقول لك هذا شي اتذكرك به في وقت لاحق (قد يكون هذا الوقت من ضغط رابط آخر في الصفحة التى زرتها حتي بعد أسبوع أو أكثر) .

يقوم السيرفر بإرسال الكوكيز عبر الـHTTP Headers الذي يتم إرساله قبل أي مخرج من مخرجات الـhtml

والمستعرض أيضا يقوم بإرسال الكوكيز عبر الـ HTTP Header بالإضافه إلى أن المستعرض يتعرف على من سيقوم بإرسال الكوكيز فلو كانت الكوكيز مثلاً مرسلة من قبل الموقع www.php.net فإنه لن يقوم بإرسالها إلى موقع www.phpbuilder.com .

باستطاعتك عند إنشاء الكوكيز تحديد مسار يتم ارسال الكوكيز لكي يتم اقتصار عملية العبور إلى الكوكيز إلى أماكن معينة .

قبل أن نقوم بوضع كود بسيط سنقوم الآن بتعريف كيفية تخزين الكوكيز وكيفية قراءتها :

كون الـPHP لغة حديثه لعمل سكربتات ويب فإنها تأتي بدعم كامل للكوكيز بواسطة الدالة setcookie() باستثناء أنك عند استعمالها يجب استعمالها قبل طباعة أي مخرجات html .

تاخذ الدالة setcookie() ثلاث معاملات ، الثلاثة الأولى هي الأهم والأمثل استخداماً وهي بالترتيب :

- ❖ قيمة حرفية يتم تخزينها كاسم للمتغير
- ❖ قيمة حرفية يتم تخزينها كقيمة لذلك المتغير
- ❖ Unix timestamp الذي يقوم بالإشارة إلى تاريخ إنتهاء الكوكيز

Unix timestamp عبارة عن رقم صحيح لا يحتوي على فواصل عشرية يقوم بحساب الثواني من منتصف ليلة 01/01/1970 . وإذا كنا نريد مثلاً أن نقوم بمسح الكوكيز بعد ساعة من تخزينه فإننا نقوم باستعمال الدالة time() التي تقوم بحساب الـ timestamp ثم نضيف عليه الوقت الذي نريده وفي حالتنا الساعه تساوي 3600 ثانية وعلى ذلك سنقوم بإضافة ناتج الدالة time على 3600 لكي يتم مسح الكوكيز بعد ساعة واحدة !

الثلاث العوامل الأخري التي يتم ستخدامها أيضا في الكوكيز ولكنها نادرة الاستخدام ولن نناقشها في موضوعنا هذا هي :

- ✓ المسار الذي يتم إرسال الكوكيز إليه فلو تم فتح نفس الصفحة من نفس الموقع ولكن من مسار اخر (مثلاً المسار كان pag\url\one وتم تغييره إلى page\url\two فان المستعرض لن يقوم بارسال البيانات إلي الصفحة لأنه تم تحديد المسار الذي سيتم ارسال الكوكيز اليه)
- ✓ الدومين الذي سيتم إرسال البيانات إليه وهو مفيد في حالة ما إذا كان هناك أكثر من دومين تريد ارسال الكوكيز اليه
- ✓ متغير من نوع integer يتم الإشارة إليه بـ secure يتم في حالة استخدام عمليات تشفير بالـ SSL

العبور الى الكوكيز بسيط جداً فالمتغير الذي يتم ارساله يتم تخزينه ضمن المتغيرات العامة (global) وعندئذ فإنه لو كان لدينا كوكيز اسمه ahmed فإن قيمته توضع مباشرة في متغير اسمه $ahmed !!

يمكننا مسح الكوكيز بأكثر من طريقة ، بالطبع فإن المستخدم يستطيع مسح الكوكيز وتغيير محتوياتها بنفسه ولكن في حالة ما إذا أردنا أن نجعل السيرفر يقوم بمسحها فإننا نستخدم إحدي هاتين الطريقتين

إما أن نقوم بإخبار السيرفر بوقت قديم :

```
<?

Set cookie (“majed” , “0”, time()-999);

?>
```

وإما القيام بمسح الكوكيز بكتابة اسمه فقط :

```
<?
Setcookie (“majed”);
?>
```

مثال لتخزين وقراءة كوكيز

قم بفتح المفكرة واكتب الكود التالي :

```
<?
If  ($thename)  setcookie ("rname", $thename, time()+3600);
Echo '<form method="post">
<input type ="text" name="thename">
">تسجيل<input type="submit" value="
</form>';
". " ". $thename ."<br><br>";قيمه المتغير الذي لديك echo "
 =" . $rname ;قيمة الكوكيز echo "
?>
```

الشرح

عند تشغيل الصفحه لاول مره

عند تشغيلك للصفحة سيتم اختبار ما إذا كان هناك متغير بالاسم $thename فإذا تم الحصول عليه فسيتم وضع قيمته في كوكيز باسم (rname) (وطبعا لن يتم الحصول عليه في أول مرة لأننا لم نقم بارسال أي بينات بعد) وبعد ذلك طباعة نموذج من مربع نص واحد وزر لإرسال المعلومات .

ويتم طباعة قيمة المتغير إذا كان هناك أي متغير تم ارساله باسم $thename ويتم فحص قيمة الكوكيز $rname وطباعتها وبالطبع لا يوجد حتي الآن أي كوكيز .

المرحلة الثانية

الآن قم بكتابة أي شي في مربع النص (اكتب اسمك مثلاً) ثم قم بضغط زر الارسال سيتم ارسال البيانات الى نفس الصفحة ولكن هذه المرة سيتم تسجيل قيمة المتغير الذي يحمل البيانات في الكوكيز (rname) وبعد ذلك سيتم طباعة النموذج بشكل عادي وسيتم طباعة قيمة المتغير $thename ولكن لن يتم طباعة قيمة المتغير $rname لأننا فقط قمنا بتسجيله ولم يتم ارساله عند طلب الصفحة (لأننا نعرف أنه يتم ارسال الكوكيز عند طلب الصفحة وهذه المرة عندما طلبنا الصفحة لم يكن الكوكيز موجوداً بالأصل فلم يرسله السيرفر وقمنا نحن بتسجيله استعداداً للمرحلة القادمة) .

المرحلة الثالثة

في هذه المرة سيكون الكوكيز موجوداً فسيتم ارساله على هيئة متغير ويتم ارساله ومن ثم طباعة النموذج وقيمه المتغير $thename وقيمة الكوكيز الذي يوجد بجهازك !

بدايتك الى الـsession

الـ session هي عبارة عن تقنية للترابط مع المستخدم وهي موجودة ضمن الـ PHP4 ولم تكن موجودة ضمن الإصدارات التي قبله بل كان يجب أن تقوم بتركيب مكتبة لكي تستطيع استخدام هذه التقنية ، يعتمد فهمنا للـsession على فهمنا للكوكيز وكيفية استعمالها ولقد تكلمنا عن الكوكيز بشكل جيد في الدرس السابق ، يستخدم الـsession لعمل ميكانيكية تواصل بين المستخدم والسيرفر ، فلقد قلنا أن الـ http لا يوفر لنا ميكانيكية لعمل تواصل ، فإذا طلب المستخدم صفحة من السيرفر فإن السيرفر يقوم بإعطائه ما أراد وينتهي عند ذلك فلا يعرف إن كان هو نفس المستخدم أو ليس هو ... لأجل ذلك تم انشاء تقنية الـsession لأجل عمل تقنية تواصل بين المستخدم والموقع ، فبإستطاعتك مثلاً أن تقوم بتحديد عدد زيارات مستخدم معين لصفحتك ليوم واحد أو لأسبوع أو لمدة معينة من الوقت .... أو يمكنك عمل متجر إلكتروني بسيط يستطيع المستخدم شراء عدة أشياء دفعة واحدة من الموقع ويكون على تواصل بينه وبين الموقع عندما يقوم بإضافة مشترى إلي سلة التسوق أو حذف مشتريات.

قبل أن أتكلم عن كيفية استخدام الـ Session وإعطاء بعض الأمثلة البسيطة ، سأقوم بالتكلم عن كيفية إعداد الـsession مع الـPHP .

اعدادات الـsession في الـPHP

لكي تستطيع التعامل مع الـ session بشكل جيد يجب عليك أن تتعرف على بعض الإعدادات التي في ملف الـphp.ini

عندما تفتح الملف ستجد قسماً خاصاً فيه بالـsession هناك حوالى 19 إعداد ولكن لن نتطرق إليها كلها بل سنتكلم عن الأساسية والمهمة منها فقط كبداية لنا للتعرف على الـsession وكيفية عمله .

اعداد طريقة التخزين

```
session.save_handler (files | mm | user)
```

ستجد هذه العبارة مكتوبة في الملف كالتالى بشكل افتراضي :

```
session.save_handler = files
```

وهذا الإعداد يقوم بتحديد طريقة التخزين للـsession وهناك ثلاث حالات للتخزين :

1 - التخزين في ملفات عادية على السيرفر :

```
session.save_handler = files
```

2 - التخزين على ذاكرة السيرفر :

```
session.save_handler = mm
```

3 - التخزين بطريقة أخري معرفة ومعينة من قبل المستخدم مثل التخزين في قواعد البيانات وهذا ما سوف نقوم بالتفصيل عنه بعد الكلام عن قواعد البيانات :

```
session.save_handler = user
```

يجب أن تأخذ في اعتبارك عدد الملفات التي سيقوم الـsession بتخزينها عند استخدامك للأعداد الأول والإفتراضي خاصة عندما يكون عدد الزوار بالمئات أو الآلاف .

قد يكون استعمال الذاكرة أسرع ولكن المشكلة أنه من السهل مسح البيانات منها ببساطة .

الطريقة الثالثة قد تكون أكثر الطرق مرونة ، ولكنها معقدة وصعبة جداً ، وهي تعطيك مرونة لتخزين البيانات في أي وسائط مدعومة من قبل الـPHP مثل قواعد بيانات mysql و oracle .

الذي افترضه الآن أنك قمت بوضع قيمة هذه الخاصية إلى files

إعداد مكان التخزين

```
session.save_path (path/to/directory)
```

هذه الخاصية مفيدة إذا كنت قد ضبط الإعداد السابق إلى files

تقوم هذه الخاصية بتحديد مكان التخزين على السيرفر ومن الأفضل أن تقوم بتحديد مكان التخزين بعيداً عن مجلد السيرفر لكي تمنع تصفح هذه الملفات .

الإنشاء التلقائي للـsession

```
session.auto_start (0 | 1)
```

هذا الإعداد يقوم بتحديد إذا ما كان الـ session سيتم إنشاؤه تلقائياً عند كل زيارة للموقع أو لأي صفحة من صفحاته بدون إدراج كود الـsession في كل صفحة ... وعلى ذلك فإنك تقوم بوضع القيمة إلى (1) إذا أردت ذلك .

وعلى إفتراض أنك لا تحتاج إلى أن تجعل الـPHP يقوم بعمل session لكل صفحة تلقائياً ومن غير طلب فستقوم بوضع قيمة هذا الإعداد الى (0)

الـSID

عندما يقوم الزائر بزيارة صفحتك فإن الـ session يستطيع تتبع هذا الزائر وعدد المرات التي قام فيه الزائر بالدخول لليوم الواحد ، يقوم الـPHP بعمل SID (session identefier) أو رقم معرف تلقائي بشكل افتراضي عندما تقوم بطلب إنشاء session بالزائر ، وكل رقم معرف يختلف عن الآخر تماماً ، إن رقم المعرف الذي ينشئه الـPHP شبيه للشكل التالى :

```
fc94ad8b1ee49ef79c713ee98ac1fcc4
```

هناك طريقتين يستطيع بها الـPHP متابعة الـSID للمستخدم :

1 - عن طريق المتابعة والتخزين بتسلسل في الكوكيز .

2 - عن طريق اتباع رقم المعرف بعنوان الصفحة في الانترنت .

سنأخذ أمثلة عن كلا الطريقتين :

1 - استخدام الكوكيز

بالطبع هذه هي أكثر الطرق شيوعاً للحصول على ترابط بين المستخدم والموقع وهي الأسهل ، ولكن يجب أن تضع في اعتبارك أن المستخدم قد يكون قد ألغي أو منع ميزة الكوكيز في المتصفح أو قد يكون متصفحه لا يدعم الكوكيز .

خذ في اعتبارك أن بعض المتصفحات لا تسمح بأن يزيد حجم الكوكيز عن 5 كيلوبايت .

هناك بعض الإعدادات البسيطة في ملف php.ini التي يجب معرفة معلومات عنها قبل البدء باستخدام الـ session مع الكوكيز :

```
session.use_cookies (0 | 1)
```

هذه الخاصية تحدد ماذا كان يمكنك استخدام الكوكيز مع الـ session أو لا وعند وضع القيمة (0) فهذا يمنعك من استخدام الكوكيز مع الـ(session) وأما اذا كانت قيمته (1) فهذا يسمح باستخدام الكوكيز مع الـsession

```
session.name (Default: PHPSESSID)
```

هذا الإعداد يقوم بتحديد اسم الكوكيز الذي سيحتفظ برقم المعرف (SID) والإعداد الافتراضي هو PHPSESSID ولن اقوم بتغيير هذا الإعداد لكي تستطيع فهم المثال الذي سأطرحه بعد قليل

```
session.cookie_lifetime (Default: 0)
```

يقوم هذا الإعداد بتحديد المدة التي سيبقي فيها الكوكيز الذي يحتفظ بقيمة الـ(SID) والإعداد الافتراضي هو صفر ، أي أنه سيتم مسح الكوكيز تلقائياً بعد اغلاق المستخدم لنافذة المتصفح مباشرة

```
session.cookie_path (Default: /)
```

يقوم هذا الإعداد بتحديد مسار دومين يتم تخزين الكوكيز له .. لا تقم بتغيير قيمته ودعه كما هو

```
session.cookie_domain (Default: null)
```

يقوم هذا الإعداد بتعريف اسم دومين يتم تخزين الكوكيز لصالحه .. والقيمة الافتراضية هي null ، لا تقم بتغييرها

ضع في اعتبارك انه اذا كانت قيمه الاعداد (session.use_cookies) تساوي واحد فان لا داعي لاستدعاء الدالة set_cookie() لإعداد الكوكيز بل سيتم اعدادها تلقائياً بواسطة الـPHP

2 - الإضافة أوالكتابة إلى عنوان الصفحة

إن إضافة عنوان الـSID إلى عنوان الصفحة يعتبر من الأشياء البشعة جداً رغم أن طريقته سهلة ومفيدة في حالة ما إذا كان الكوكيز غير مدعوم في المتصفح بشكل جيد

مثال :

```
<a href="configure.php?<?=SID?>">Go to the configuration page</a>
```

بهذه الطريقة نقوم بإضافة المتغير المرجعي SID الذي سيقوم بإعطاء رقم معرف للمستخدم .

متابعة الـsession

لقد أخذنا حتي الآن معلومات تجعلنا ندخل عالم البرامج المسيره بالـ session بدون خوف ، سأبدأ الآن في طرح بعض الأمثلة البسيطة التي تثبت لديك بعض المفاهيم الأساسيه في الـ session ... سأشرح في هذا المثال كيفية إنشاء الـSID وتخزينه لاستعماله لاحقاً ، وخلاصة السيناريو للصفحة أننا نريد من المستخدم أن يفهم أنه يستطيع تخصيص لون الخلفية الذي يريد أن يشاهد به صفحات موقعنا ... سأقوم بتخزين قيمة مبدئية في المتغير الذي يقوم بتحديد لون الصفحة ، أنا افترض طبعاً أن المتصفح يدعم الكوكيز :

سكربت يقوم بانشاء وتسجيل متغير session

```
<?
session_start();
session_register("zx");
session_register("co");
$zx=10;
$co++;
echo 'مرحبا بك في موقعنا أيها الزائر الكريم '."<br>";

echo "عدد زياراتك لهذه الصفحة=".  $co ;
echo "<br>";
echo '<a href=" php2.php ">الصفحة الثانية</a>';
?>
```

اقصد بالجلسة هي الـ(session) وإن كانت الترجمة غير صحيحة ولكن فقط نأخذه كمصطلح .
متغير الجلسة هو الـ(session-variable) أو متغير الـsession أو سمه ما شئت .

الشرح

يقوم هذا السكربت في البداية بإنشاء متغير اسمه (zx) ومتغير اسمه (co) وقمنا بإعطاء القيمة (10) للمتغير (zx) وقمنا بزيادة القيمة الموجودة (وهي الصفر) في (co) بواحد وكتبنا مرحبا بك ايها الزائر الكريم في موقعنا ، ثم قلنا له إن عدد زياراتك لهذه الصفحة هي قيمة المتغير(co) ثم اعطيناه رابط للصفحة الثانية .

في الواقع إن هذه المتغيرات وقيمها يتم الاحتفاظ بها في كوكيز له اسم خاص قمنا بتحديده سابقاً من ملف PHP.ini ، وهذا الكوكيز يحتفظ بقيمة الـSID للـsession .

نحن لا نقوم بإخبار الـPHP أين سيحتفظ بقيمة المتغيرات لأننا بدأنا بكلمة الـ:

```
session_start();
```

وعلى هذا فإن الـPHP سيفهم أنه سيقوم بتخزين القيمة في الكوكيز الخاص بالـsession .

قمنا بجعل المتغير co كعداد بسيط لعدد المرات التي سوف نقوم بها بزيارة الصفحة فعند عمل تحديث للصفحة سيتم زيادة العداد بمقدار واحد

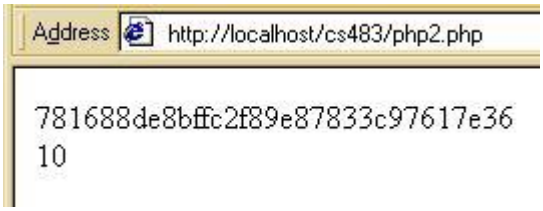
```
$c++;
```

وطبعاً قبل زيادة العداد بقيمة واحد فإنه يتم حساب القيمة السابقة للمتغير عند إنشاءه تلقائياً ... ومن ثم يتم الزيادة وبعد ذلك طباعة القيمة .

كتابة رقم الـSID

اكتب الآن الكود التالي واحفظه باسم php2.php

```
<?
session_start();
echo $PHPSESSID ."<br>";
echo $zx;
?>
```



في هذه الصفحة نقوم بطباعة قيمة الـ SID وذلك بطباعة قيمه المتغير $PHPSESSID (الذي هو اسم الكوكيز الخاصة بالـsession) .

بعد ذلك قمنا في النهاية بطباعة قيمة المتغير $zx لكي ألفت نظرك بأن الكوكيز ما زال يحتفظ بها ولم يفقدها لأننا قد حددنا الإعداد في ملف php.ini الخاص بوقت الكوكيز الـ 3600 أي لمده ساعة ثم بعد تلك الساعة سيتم مسح الكوكيز ولن يمكنك استرجاع قيمة أي متغير :

```
session.cookie_lifetime = 3600
```

واضف إلى معلوماتك أنه لا يمكنك قراءة القيم للكوكيز الخاصة بالـsession إلا عن طريق إضافه الأمر

```
session_start();
```

يجب أن تبدأ بهذا الأمر دائماً إذا أردت قراءة قيم المتغيرات التي يحتفظ بها الكوكيز الخاص بالـsession .

مسح متغير من الـsession

كل ما عليك فعله هو استخدام هذه الدالة :

```
session_unregister(variable name);
```

تقوم بوضع اسم المتغير في مكان الـ(variable name)

مثال :

```
session_unregister("brn");
```

سيقوم هذا الأمر بمسح المتغير (brn) من الـكوكيز الخاصة بالـ(session)

قراءة قيم المتغيرات في الـكوكيز الخاصة بالـsession

كل ما عليك فعله هو استخدام الدالة :

```
session_encode();
```

مثال :

```
<?
session_start();
session_register("bgcolor");
session_register("name");
session_register("email");
$bgcolor = "#8080ff";
$name = "majed sa";
$email = "php@php.com";
$e = session_encode();
print "The encoded string is: $e";
?>
```

بهذا السكربت نكون قد أنهينا درسنا عن مقدمة بسيطة للـsession . هذه مجرد مقدمة ولكي نستطيع أن نتعمق بالـsession فيجب علينا أن نتعلم شيئاً عن قواعد البيانات .

قراءة وكتابة معلومات في ملفtxt

عندما لا يكون لدينا قاعدة بيانات، يجب أن نستخدم ملفات txt عادية لحفظ المعلومات .. في الـPHP ، إنشاء أو قراءة معلومات من ملف، أمر سهل! .. يوجد عدد من الدوال - سنقوم بدراستها اليوم - تساعدنا على عمل ذلك ..

سنقوم بإنشاء سكربت بسيط، يحفظ (الاسم) والـ (بريد) لمستخدمين، ومن ثم نقوم بعرضها ..

---

1 - الدالة fopen()

الدالة الأساسية هيfopen() ، وهي التي تسمح بفتح ملف؛ للقراءة، أو لإنشاءه إن لم يكن موجوداً، أو للكتابة .. وتستخدم بهذا الشكل تقريباً:

```
fopen("File name & extention","mode");
```

File name & extention = اسم الملف والمراد الكتابة فيه وامتداده ..
mode = الطور ..

**جدول الأطوار المتاحة **

r = فتح وقراءة فقط ..
w = فتح وكتابة فقط (الدالة تقوم بإنشاء الملف إن لم يكن موجوداً) ..
a = فتح وقراءة فقط مع إضافة المحتويات في نهاية الملف (الدالة تقوم بإنشاء الملف إن لم يكن موجوداً) ..

r+ = فتح للقراءة والكتابة ..
w+ = فتح للقراءة والكتابة (الدالة تقوم بإنشاء الملف إن لم يكن موجوداً) ..
a+ = فتح للقراءة والكتابة مع إضافة المحتويات في نهاية الملف (الدالة تقوم بإنشاء الملف إن لم يكن موجوداً) ..

أمثلة:

```
$fp = fopen("../file.txt","r");

$fp = fopen("ftp://localhost/pub/file.txt","w");

$fp = fopen("http://localhost/file.txt","a");
```

الدالة fopen إذن تسمح بفتح ملفات على الويب.....

2-إنشاء (أو تحديث) ملف

قبل كل شيء، يجب وضع معلومات في ملف .. يجب إذن فتح ملف للكتابة وإنشاءه إن لم يكن موجوداً .. إذن لدينا الاختيار بين طور "w" وطور"a" ، ولكن نفضل الثانية، لأنها تبدأ بالكتابة في آخر الملف، بمعنى آخر؛ تتم كتابة المعلومات بعد المعلومات المكتوبة سابقاً في الملف ..

ملاحظة: شيء جيد التأكد من ان الملف تم فتحه بنجاح

```
if($fp = fopen("file.txt","a")){

  fputs($fp, "\n");

  fputs($fp, "$name|$email");

  fclose($fp);

}else{

  echo "not file";

  exit();

}
```

الدالة fputs() تسمح بالكتابة في ملف .. وهي مرادفة للدالةfwrite()، أي أن لهما نفس العمل بالضبط !

وتستخدم بالشكل التالي :

fputs(...);

---

3 - قراءة من ملف

يمكنك قراءة المعلومات الموجودة في الملف بفتحها بطور القراءة:

```
if (file_exists($file))

$fp = fopen("file.txt", "r");

else{

echo "not file.";

exit();

}

while (!feof($fp)){

$line = fgets($fp,4096);

$liste = explode("|",$line);

$name = $liste[0];

$email = $liste[1];

echo "Name: $name     email: $email<br>";

}


fclose($fp);
```

نستخدم الدالة fgets() لاسترجاع المعلومات الخاصة بالملف، واحدة واحدة ..

الدالة feof() تسمح لنا بفحص إذا لم نصل إلى نهاية الملف ..

ملاحظة :

يمكنك وضع الكم الذي تريده من المعلومات، ولكن يجب فصلها بـ| ، للبدأ وللنهاية وبين المعلومات..

مثال الاضافه والاستعراض

نقوم بفتح ملف ونكتب به الاتي ونحفظه بالاسم التالي index.html

```
<head>
<title/>المعلومات الشخصية<title>
<head/>
<body>
<"form method="post" action="add.php">
<"div align="center>
<center>
table bordercolor="#944E6D" border="2" cellpadding="0" cellspacing="2" >
<"width="53%" height="1
<tr>
<"td width="200%" align="center" height="42" bgcolor="#E1F0FF>
<b></font></td/>المعلومات الشخصية<font color="#000080"><b>
<tr/>
<tr>
td width="200%" align="center" height="109" valign="top" >
<"bgcolor="#FFFFFF
table border="1" cellpadding="0" cellspacing="0" style="border-collapse: >
<"collapse" width="100%" height="103
<tr>
<"td width="100%" height="103" valign="top>
table border="0" cellpadding="0" cellspacing="0" style="border->
<"collapse: collapse" bordercolor="#111111" width="100%
<tr>
<td/>الأسم الأولى<"td width="46%" align="left>
<"td width="154%" align="right>
<input type="text" name="T1" size="20"></p>
<td/>
<tr/>
<tr>
```

```
<td/>الأسم  الأخير<"td width="46%" align="left>
<"td width="154%" align="right>
<input type="text" name="T2" size="20"></p>
<td/>
<tr/>
<tr>
<td/>الـدولـــــــة<"td width="46%" align="left>
<"td width="154%" align="right>
<input type="text" name="T3" size="20"></p>
<td/>
<tr/>
<tr>
<td/>المديـنـــــة<"td width="46%" align="left>
<"td width="154%" align="right>
<input type="text" name="T4" size="20"></p>
<td/>
<tr/>
<tr>
<td/>الـعـمـــــــر<"td width="46%" align="left>
<"td width="154%" align="right>
<input type="text" name="T5" size="20"></p>
<td/>
<tr/>
<table/>
<td/>
<tr/>
<table/>
<td/>
<tr/>
<tr>
td width="200%" align="center" height="30" valign="top" >
```

```
<bgcolor="#E1F0FF"><p
name="submit"> <input   "أرسل"=input type="submit" value>
<name="reset"></p  "مسح"=type="reset" value
<td/>
<tr/>
<table/>
<center/>
<div/>
<form/>
<body/>


<html/>
```

الان نقوم بفتح ملف أخر ونقوم بكتابة الكود التالي به ونحفظه بأسم add.php

```
<?
// لفتح الملف المحدد
$fp = fopen("pro.txt","a+");
if(!$fp)
{
echo " لايمكن فتح الملف";
exit;
}
$date = $T1."\t".$T2."\t".$T3."\t".$T4."\t".$T5."\n";
// للكتابة في الملف
fwrite($fp,$date);
echo "<center><font color='#944E6D'><b>  لقد تم تخزين البيانات في الملف بي
نجاح</b></font><center>";
fclose($fp);
// الكود القادم ماهو الى كود للانتقال أتوماتيكي الى الصفحه المحدد بثواني محدد
echo "<META HTTP-EQUIV='Refresh' Content=3;URL='index.html'>";
?>
```

وعند تنفيذ الملف add.php وتعبئة البيانات سوف يظهر لنا الشكل التالي الذي يأكد حفظ البيانات في الملف بنجاح

الان قوم بي الذهاب الى نفس الدليل سوف تشاهد أنه تم أنشاء ملف pro.txt ووضع به البيانات التي وضعتها

ولاستعراض البيانات الموجوده في هذا الملف نقوم بكتابة الكود التالي ونحفظه بالاسم التالي view.php

```
<html dir="rtl">
<head>
  <title>ترتيب المصفوفات</title>
</head>
<body><center>
<h1>جميع البيانات</h1>
<?

   $orders= file("pro.txt");
  // لحساب عدد البيانات المدخله كل مره
   $number_of_orders = count($orders);
   if ($number_of_orders == 0)
   {
     echo "<p><strong>لا يوجد أي بيانات في الملف</strong></p>";
   }
   echo "<table border=1>\n";
   echo "<tr><th bgcolor = \"#CCCCFF\">الاسم الاول</td>
            <th bgcolor = \"#CCCCFF\">الاسم الثاني</td>
            <th bgcolor = \"#CCCCFF\">الدولة</td>
            <th bgcolor = \"#CCCCFF\">المدينة</td>
            <th bgcolor = \"#CCCCFF\">العمر</td>
         <tr>";
   for ($i=0; $i<$number_of_orders; $i++)
   {
      $line = explode( "\t", $orders[$i] );

      echo "<tr><td>$line[0]</td>
               <td align = center>$line[1]</td>
               <td align = center>$line[2]</td>
               <td align = center>$line[3]</td>
               <td align = center>$line[4]</td>
            </tr>";
   }
   echo "</table>";
?>
</center></body>
</html>
```

جميع البيانات

الاسم الاول	الاسم الثاني	الدولة	المدينة	العمر
majed	SA	Saudi	Jaddeh	٢٥

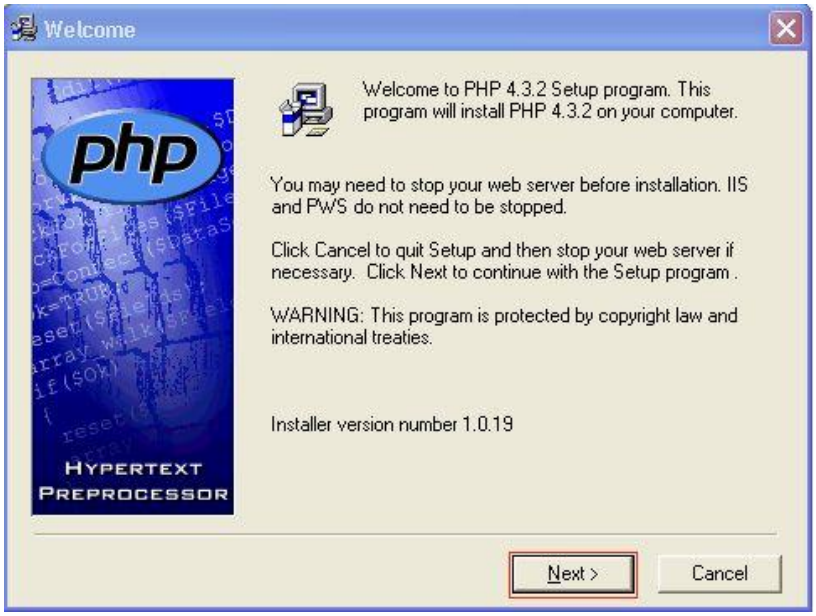
ملحق

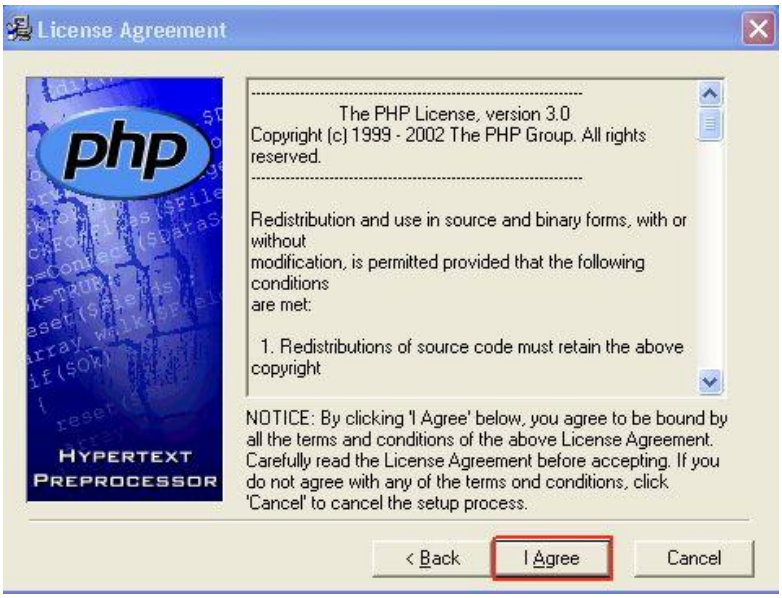
طريقة أخر لتنصيب PHP على IIS 5 بكل سهولة

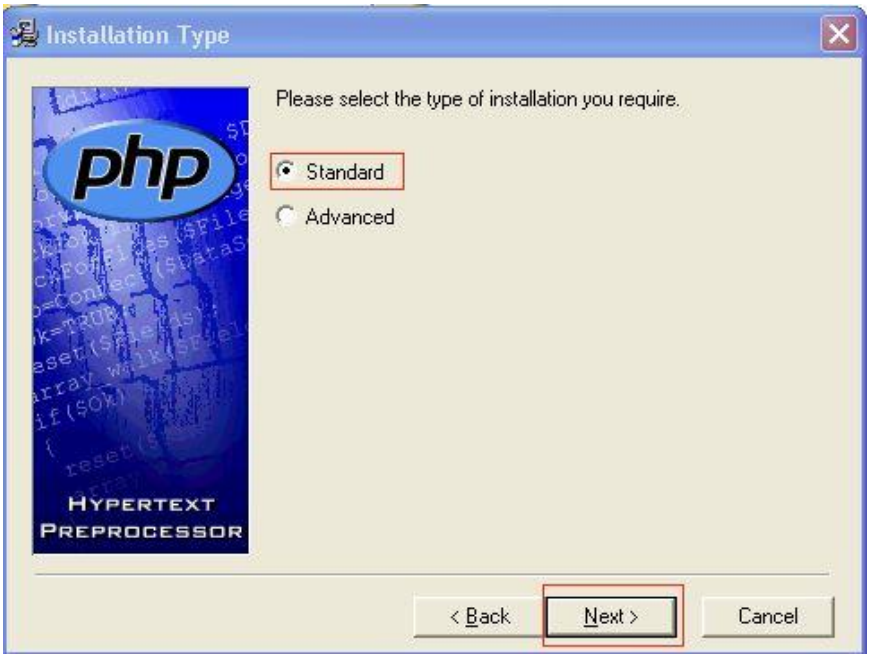
أولاً قم بالذهاب الى موقع http://www.php.net/download.php

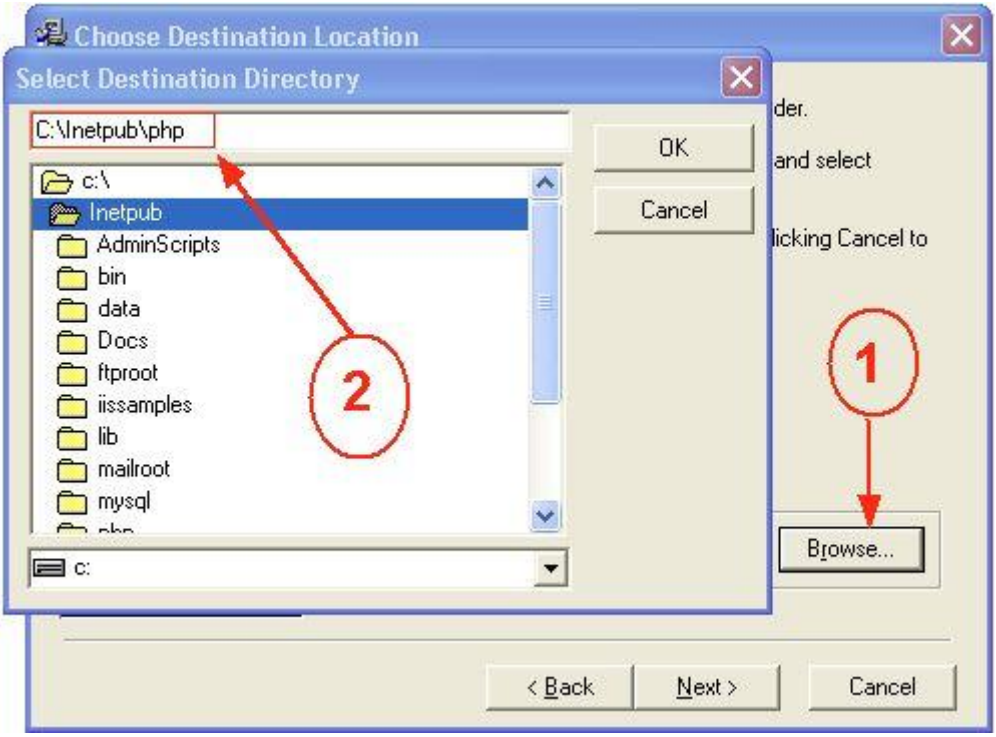
أختر المكان الذي سوف تحمل منه أنا أخترت USA

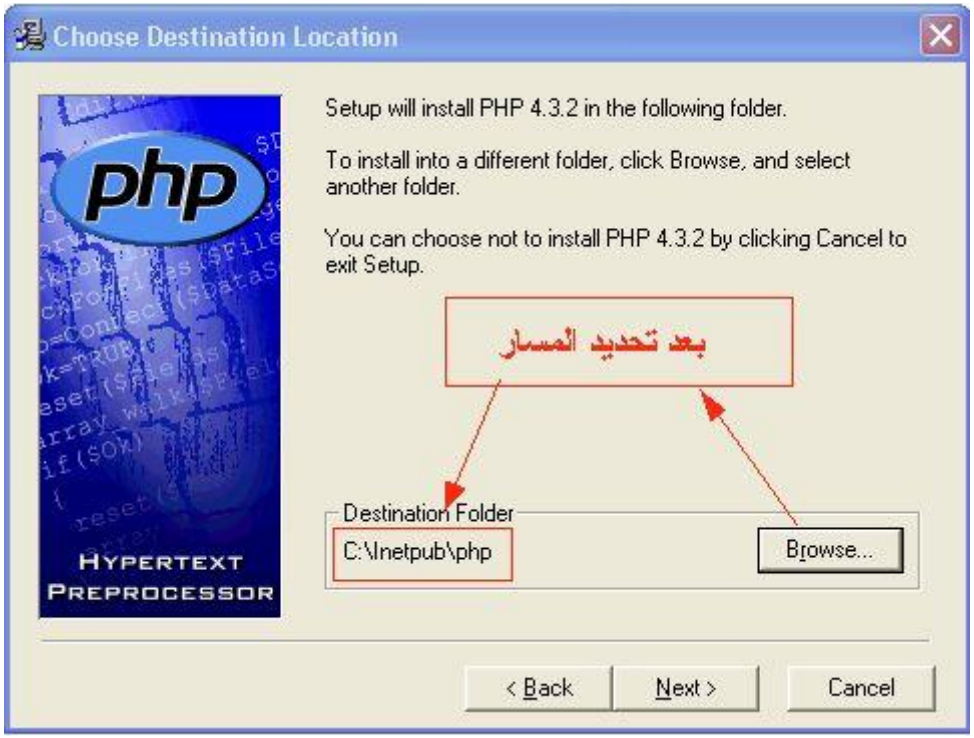
بعد تحميل ملف التنصيب

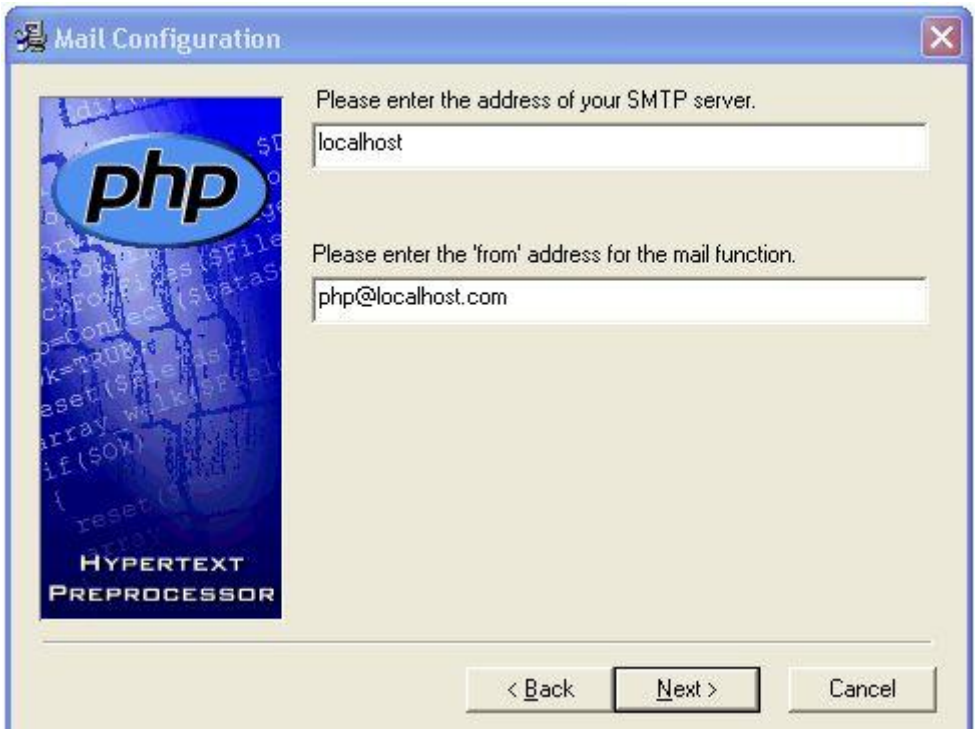


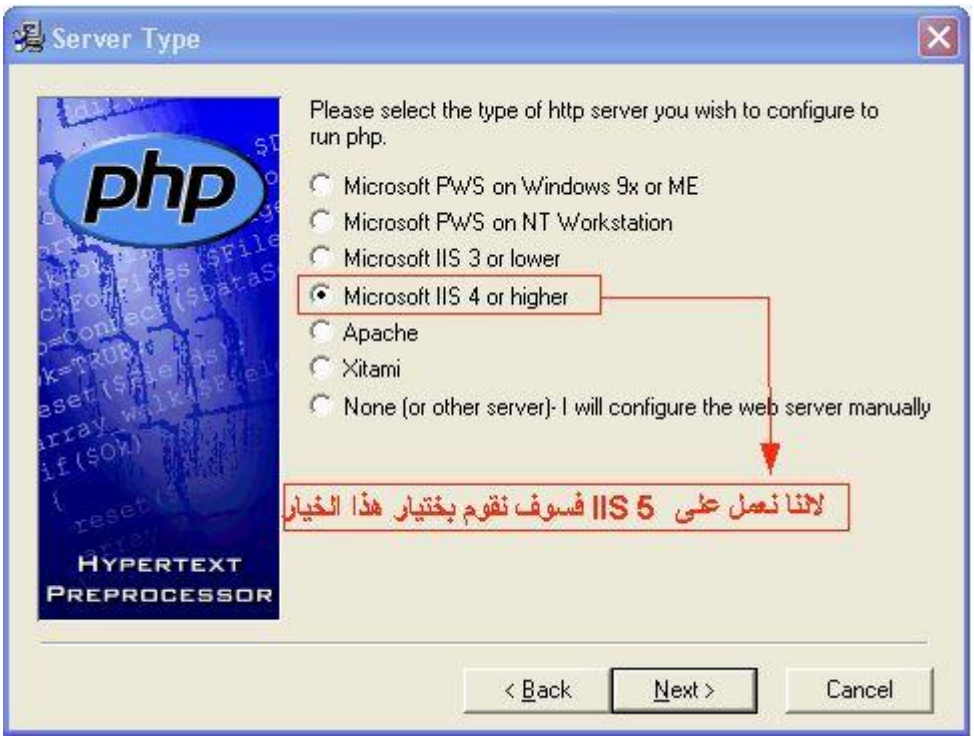


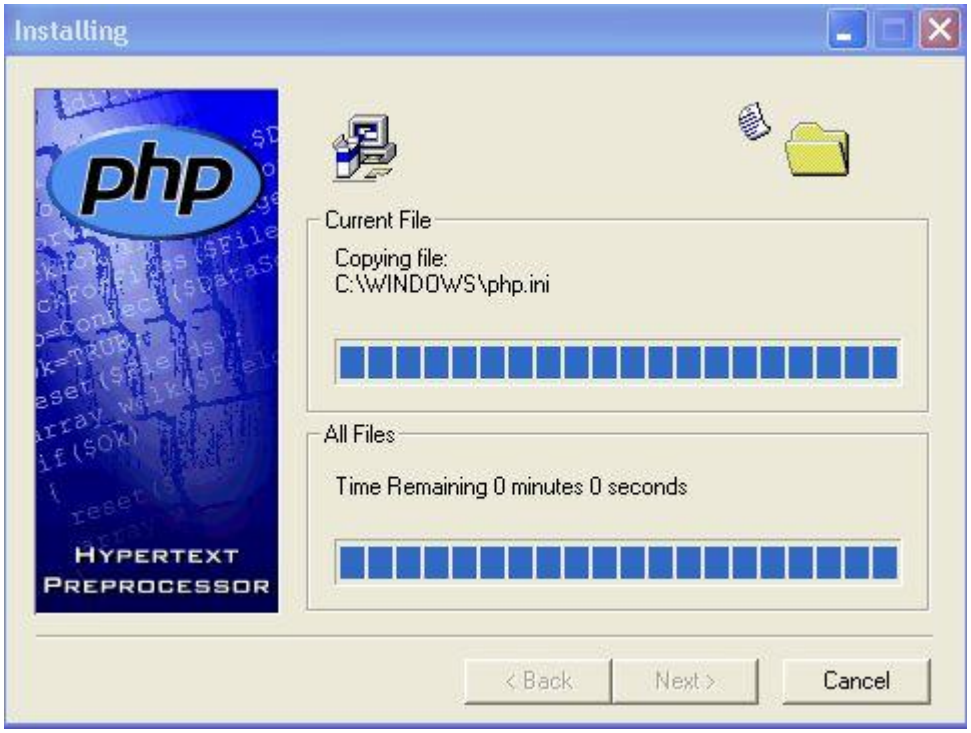






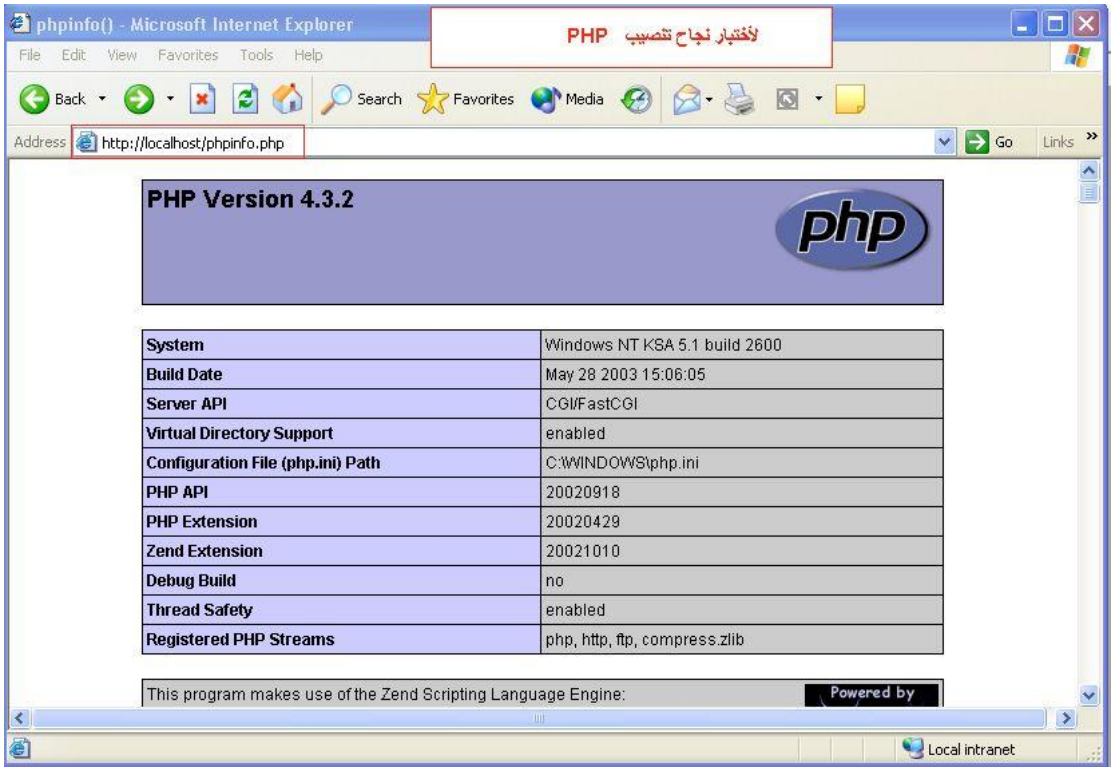

Mail Configuration
Please enter the address of your SMTP server.
localhost
Please enter the 'from' address for the mail function.
php@localhost.com
HYPERTEXT
PREPROCESSOR
< Back
Next >
Cancel

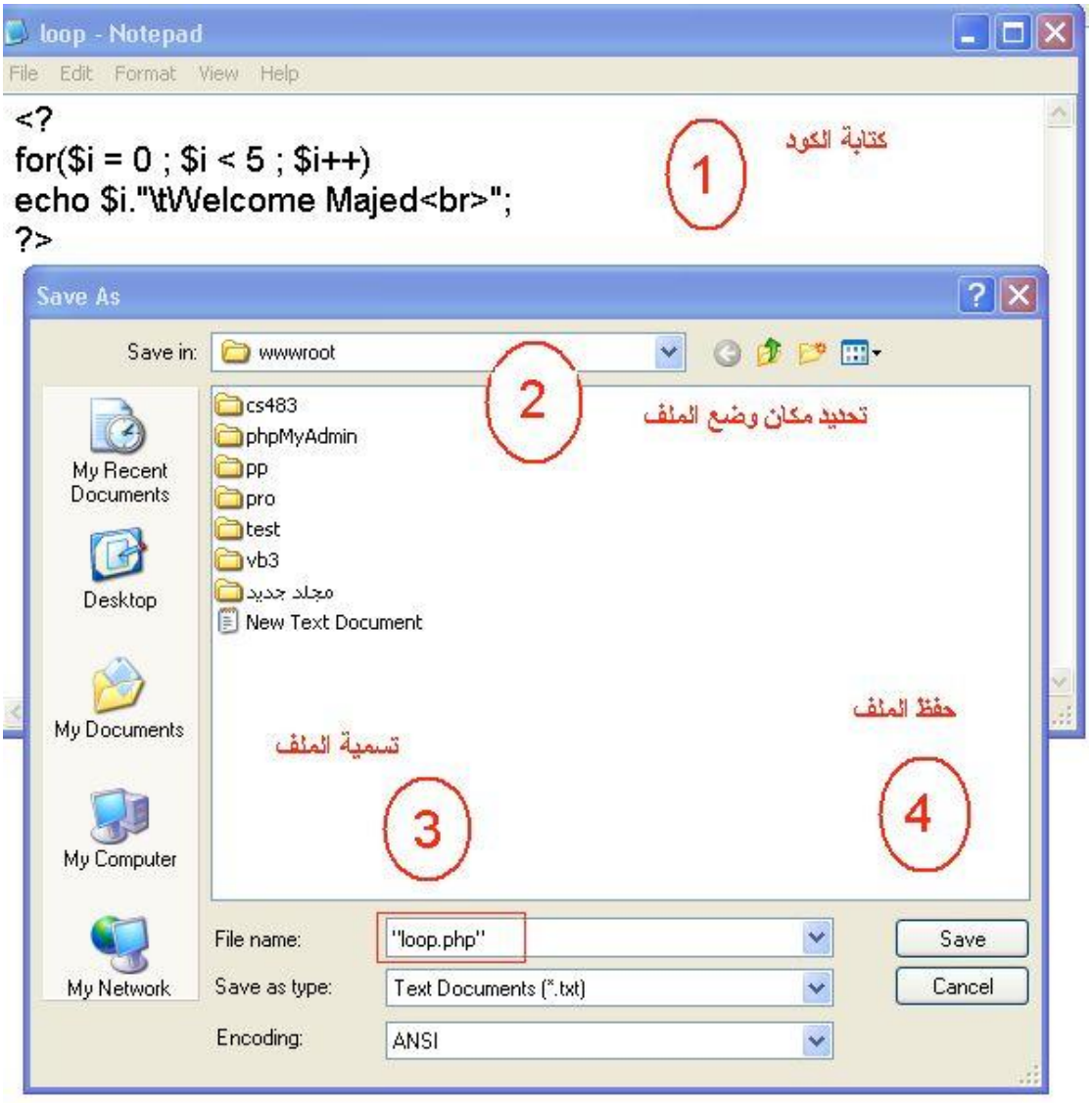

Server Type
Please select the type of http server you wish to configure to run php.
Microsoft PWS on Windows 9x or ME
Microsoft PWS on NT Workstation
Microsoft IIS 3 or lower
Microsoft IIS 4 or higher
Apache
Xitami
None (or other server)- I will configure the web server manually
لاننا نعمل على IIS 5 فسوف نقوم بختيار هذا الخيار
HYPERTEXT
PREPROCESSOR
< Back
Next >
Cancel




Installation complete
PHP 4.3.2 has been successfully installed.
Press the OK button to exit this installation.
NT users may need to set appropriate permissions for the various php files and directories. Usually IUSR_MachineName will need read write access to the uploadtmp and session directories, and execute access for php.exe and php4ts.dll.
OK


New Text Document - Notepad
File Edit Format View Help
<?
Echo phpinfo();
?>
1
كتابة الكود
Save As
Save in: wwwroot
cs483
phpMyAdmin
pp
pro
test
vb3
مجلد جديد
New Text Document
2
تحديد المسار لوضع الملف
My Recent Documents
Desktop
My Documents
My Computer
My Network
تسمية الملف
حفظ الملف
3
4
File name: "phpinfo.php"
Save as type: Text Documents (*.txt)
Encoding: ANSI
Save
Cancel


phpinfo() - Microsoft Internet Explorer
لأختبار نجاح تنصيب PHP
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/phpinfo.php
Go Links
PHP Version 4.3.2
php
System Windows NT KSA 5.1 build 2600
Build Date May 28 2003 15:06:05
Server API CGI/FastCGI
Virtual Directory Support enabled
Configuration File (php.ini) Path C:\WINDOWS\php.ini
PHP API 20020918
PHP Extension 20020429
Zend Extension 20021010
Debug Build no
Thread Safety enabled
Registered PHP Streams php, http, ftp, compress.zlib
This program makes use of the Zend Scripting Language Engine:
Powered by
Local intranet


loop - Notepad
File Edit Format View Help
<?
for($i = 0 ; $i < 5 ; $i++)
echo $i."\tWelcome Majed
";
?>
1
كتابة الكود
Save As
Save in: wwwroot
2
تحديد مكان وضع الملف
cs483
phpMyAdmin
pp
pro
test
vb3
مجلد جديد
New Text Document
My Recent Documents
Desktop
My Documents
My Computer
My Network
تسمية الملف
3
حفظ الملف
4
File name: "loop.php"
Save as type: Text Documents (*.txt)
Encoding: ANSI
Save
Cancel


http://localhost/loop.php - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/loop.php
Go Links
0 Welcome Majed
1 Welcome Majed
2 Welcome Majed
3 Welcome Majed
4 Welcome Majed
تجربة ملف PHP
Done
Local intranet

تركيب Apache مع PHP

متطلبات العمل :

- **برنامج أباتشي (Apache) يفضل أخر أصدار 2.0.47 . ملاحظة : يجب أن يكون من نوع win32**
- **ملفات PHP ويفضل أخر أصدار 4.3.2**

هذا الدرس عمل على نظام XP بخصوص الانظمه الأخرى الايختلف كثيراً وقد تم التوضيح ما يلزم تغييره في الانظمه الأخرى .

للحصول على أخر أصدارات Apache المتوافقه مع Windows تفضل بزيارة هذا الموقع

/http://nagoya.apache.org/mirror/httpd/binaries/win32

للحصول على أخر أصدارات PHP المتوافقه مع Windows تفضل بزيارت هذا الموقع

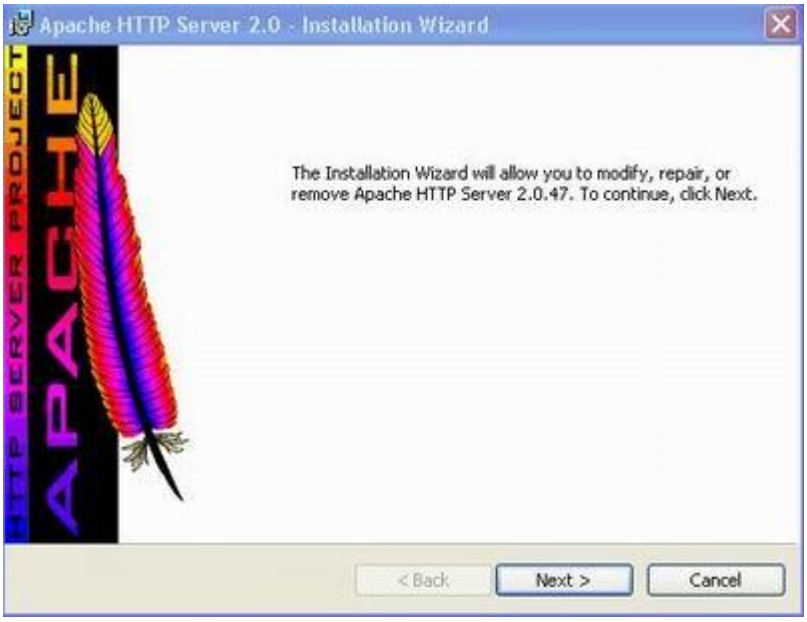
http://www.php.net/get/php-4.3.1-Win32.zip/from/a/mirror

خطوات التركيب :

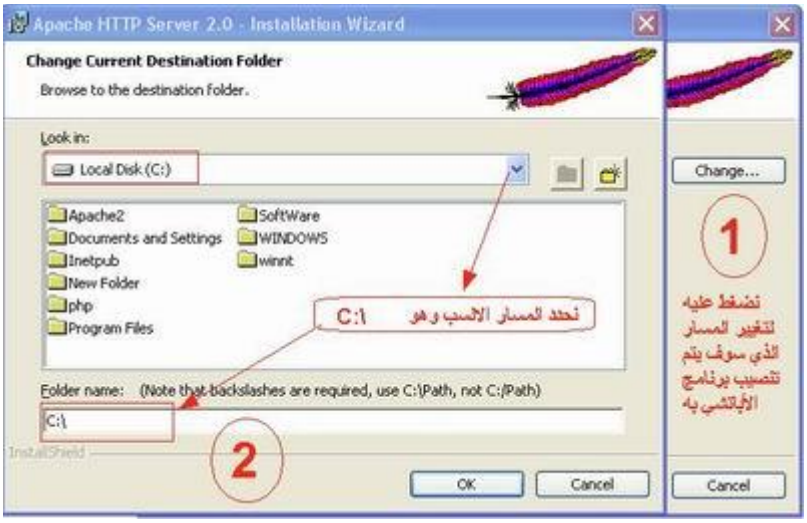
- **تنصيب برنامج Apache وهو كتالي**

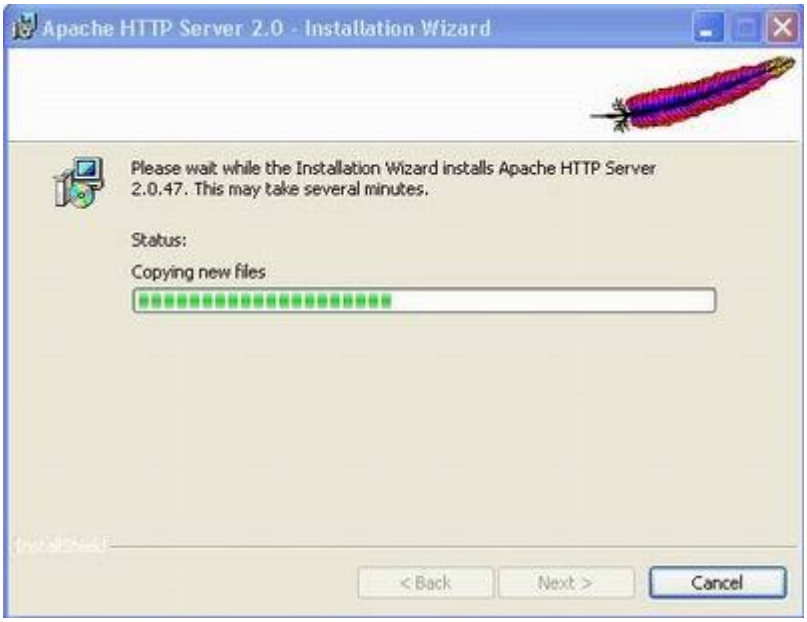
قم بتنصيب البرنامج بضغط على البرنامج

الان سوف تشاهد الشاشات الخاصه بي التنصيب كتالي




Apache HTTP Server 2.0 - Installation Wizard
Setup Type
Choose the setup type that best suits your needs.
Please select a setup type.
Typical
Typical program features will be installed. (Headers and Libraries for compiling modules will not be installed.)
Custom
Choose which program features you want installed and where they will be installed. Recommended for advanced users.
InstallShield
< Back
Next >
Cancel


Apache HTTP Server 2.0 - Installation Wizard
Change Current Destination Folder
Browse to the destination folder.
Look in:
Local Disk (C:)
Apache2
Documents and Settings
Inetpub
New Folder
php
Program Files
SoftWare
WINDOWS
winnt
C:\ نحدد المسار الاسب و هو
Folder name: (Note that backslashes are required, use C:\Path, not C:/Path)
C:\
2
InstallShield
OK
Cancel
Change...
1
نضغط عليه لتغيير المسار الذي سوف يتم تنصيب برنامج الإبانشي به
Cancel



بعد تنصيب البرنامج بنجاح نقوم بأختبار السيرفر كتالي

ضع هذا العنوان http://localhost **أو** http://127.0.0.1

في عنوان المتصفح وسوف تشاهد الصوره التالية وهي دليل على عمل البرنامج

•أعداد ملفات PHP وهو كتالي

أولاً قم بي فك ضغط الملف ثم قم بتغيير أسم المجلد الى php (حروف صغيره)

قم بنقل الملف تحت C:\ مباشرتاً كتالي C:\php

الان في داخل مجلد PHP سوف تجد ملف بأسم `php4ts.dll` **قم بنسخ هذا الملف الى المسار التالي**

C:\WINDOWS\system32 **أذا كنت على نظام** Windows NT/2000/XP

أما أذا كنت على نظام Windows 95/98/ME **فضعه على المسار** C:\WINDOWS\system

الأن في داخل مجلد PHP سوف تجد ملف بأسم `php.ini-recommended` **قم بتغيير أسمه الى** `php.ini` **كما في الصوره التالية**

الأن قم بتحرير الملف بأي محرر ثم قم بالتعديل الاتي

1 - أبحث في داخل الملف عن هذه الكلمة

`extension_dir =` **وقم بتغييرها الى** `extension_dir = "c:\php\extensions\"`

كما في الصوره التالية

```
440 user_dir =
441
442 ; Directory in which the loadabl
443 extension_dir = "./"
444
445 ; Whether or not to enable the d
```

الى هذه الصوره

```
441
442 ; Directory in which the loadable extension
443 extension_dir = "c:\php\extensions\"
444
445 ; Whether or not to enable the dl() functio
```

2 - أبحث عن التالي `cgi.force_redirect` **وقم بتغيير قيمته من 1 الى 0**

كما في الصوره التاليه

```
452 ; turn it off here AT YOUR
453 ; **You CAN safely turn thi
454 ; cgi.force_redirect = 1
455
```

الى

```
453 ; **You CAN safely turn thi
454 ; cgi.force_redirect = 0
455
```

الأن قم بنسخ هذا الملف وهو `php.ini` **وضعه في هذا المسار**

`C:\WINDOWS\system32` **أذا كنت على نظام** Windows NT/2000/XP

أما أذا كنت على نظام Windows 95/98/ME **فضعه على المسار** `C:\WINDOWS\system`

- **أعداد ملف httpd.conf**

سوف تجد هذا الملف على المسار التالي `C:\Apache2\conf`

قم بتحرير هذا الملف بأي محرر وقم بتعديل الأتي

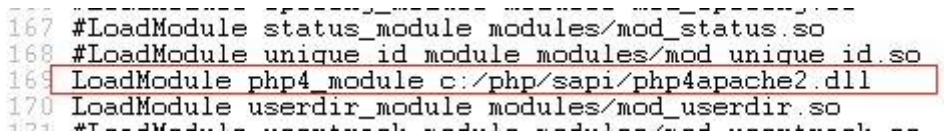
1 - أبحث عن هذه الجملة

```
#LoadModule unique_id_module modules/mod_unique_id.so
```

تحت هذه الجمله قم بوضع الجمله الأتيه

```
LoadModule php4_module c:/php/sapi/php4apache2.dll
```

كما في الصوره التاليه



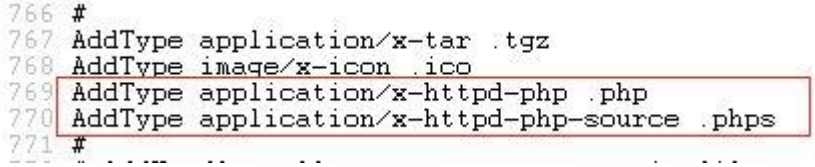
2 - أبحث عن هذه الجمله

```
AddType image/x-icon .ico
```

وضع تحتها مباشرتاً هذه الجملتين

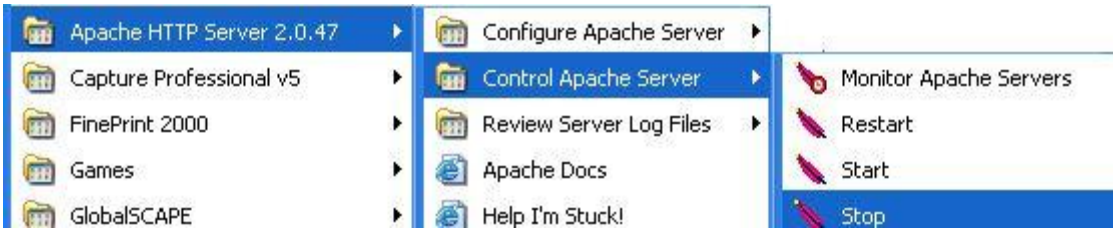
```
AddType application/x-httpd-php .php

AddType application/x-httpd-php-source .phps
```

كما في الصوره الاتية

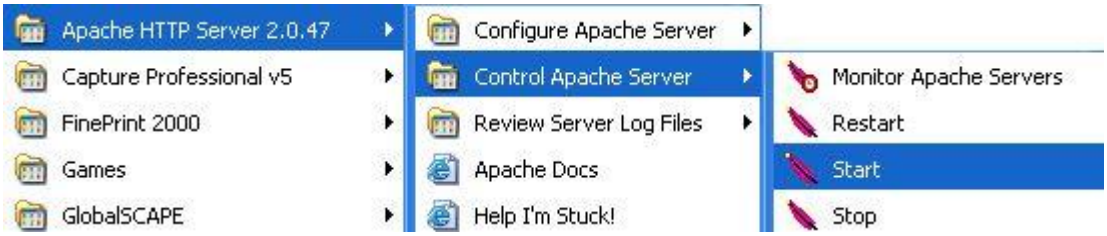


أحفظ التغييرات التي عملتها والان سوف نقوم بأختباره

أولاً قم بي أغلاق السيرفر



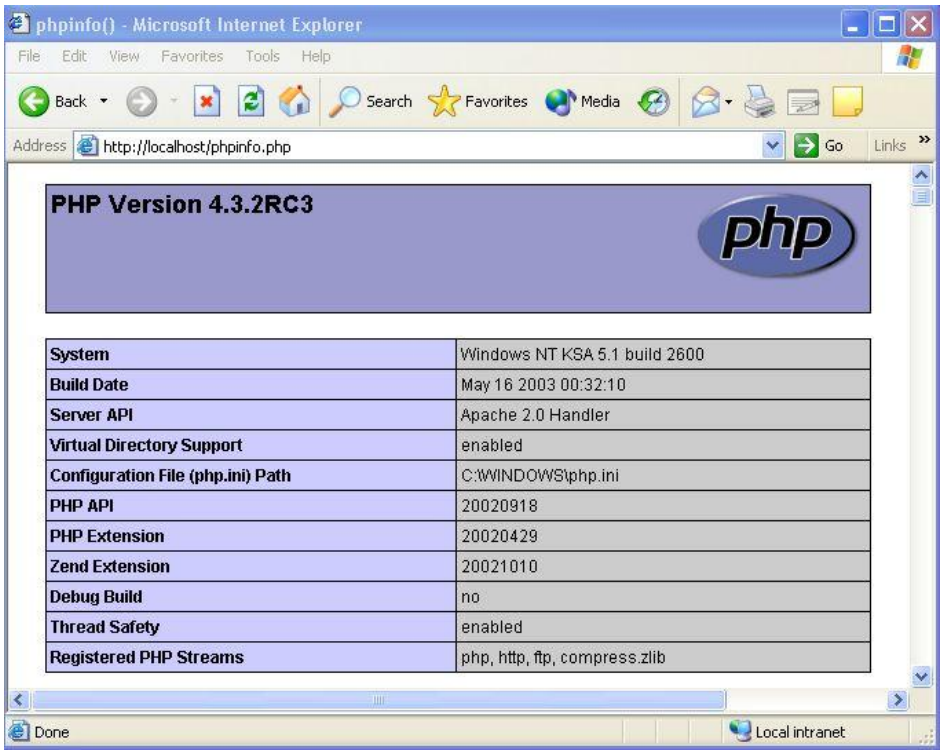
ثم قم بتشغيله من جديد كي يتعرف على الاعدادات الجديده



الأن هذه الخطوه ظروريه للذين يتعاملون مع session قم بنشاء مجلد جديد داخل مجلد Apache2 وقم بأعطاه الأسم التالي tmp

الأن قم بعمل ملف PHP كما في الصوره التاليه لكي نختبر عمله على السيرفر

بعد ذالك أكتب في المتصفح التالي http://localhost/phpinfo.php



مبروووووووووووك الان كمبيوتر به سيرفر Apache ويدعم لغة PHP

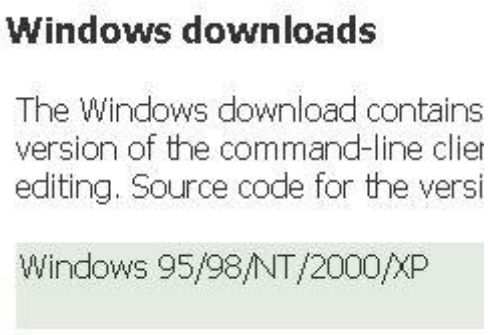
تركيب MySQL مع Apache

المتطلبات كتالي :

- **فقط برنامج MySQL أخر أصدار يفضل وهو 4.0.12**

للحصول على البرنامج أذهب الى الموقع التالي (لاحظ أن النظام لديك هو Windows)

http://mysql.progen.com.tr/downloads/mysql-4.0.html



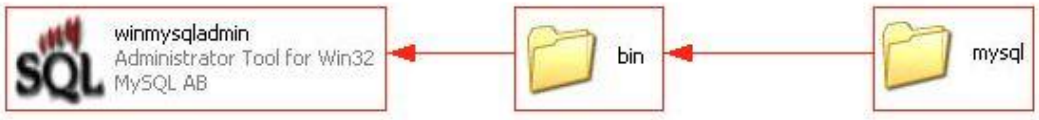
ولتحميل المباشر

http://mysql.progen.com.tr/Downloads/MySQL-4.0/mysql-4.0.12-win.zip

عند فك الضغط عن البرنامج قم بتنصيب البرنامج

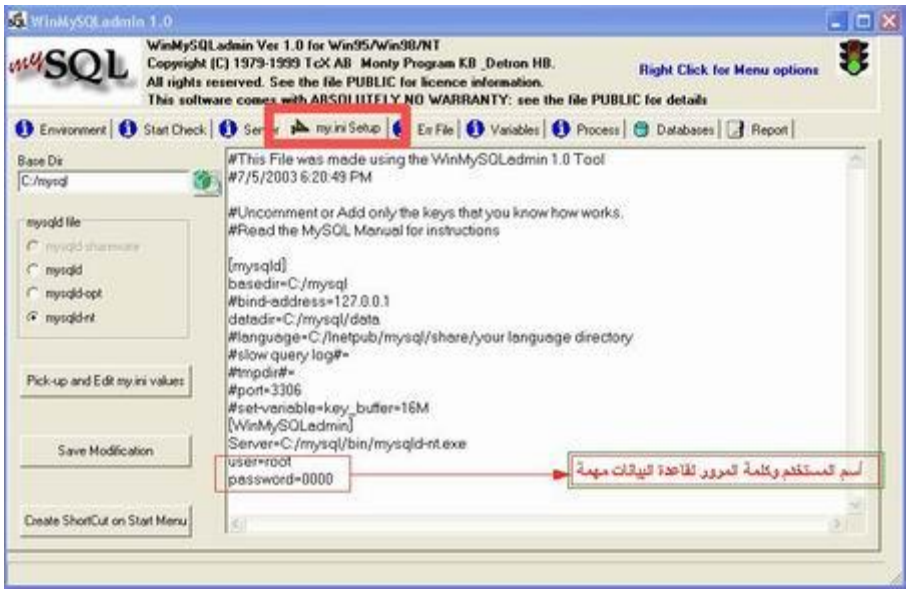
سوف تظهر لك الشاشات الاتيه

الان أتجه الى Winmysqladmin وقم بتشغيله بالنقر عليه مرتين



سوف تظهر لك هذه الشاشة للمره الأولى فقط وهي لوضع أسم المستخدم وكلمة المرور لقاعدة بياناتك أذا لم تقم بأختيار كلمة المرور فأنك في خطوات قادمه لن تحتاج لوضعها من الافضل لك هو وضع أسم مستخدم root وعدم وضع كلمة مرور

ولمشاهدة أسم المستخدم وكلمة المرور



ولتأكد من عمل MySQL سوف تشاهد بجوار الشاعه مثل أشارة المرور تعطي الون الأخضر لدلاله على عملها .

لايقاف MySQL أضغط بزر الفاره الايمن على الايقونة كما في الصوره الاتيه

سوف يظهر لك مربع أختر Yes لايقاف MySQL

الان تشاهد أيقونة البرنامج قد أصبحت حمراء وهذا يدل على أيقاف MySQL

الان كمبيوتر يدعم MySQL + PHP + Apache

الان نحتاج الى تركيب برنامج لادارة قواعد البيانات وهو PHPMyAdmin

●تركيب برنامج PHPMyAdmin

نحتاج الى برنامج PHPMyAdmin ويمكننا الحصول عليه من الموقع التالي أخر أصدار

http://phpmyadmin.net/

ولتحميل المباشر

http://www.phpmyadmin.net/index.php?dl=3

الان نقوم بفك الضغط عن الملف ونقوم بتغيير أسم الملف الى phpmyadmin

ملاحظة مهمة : في مجلد phpmyadmin يجب أن يكون تحته الملفات على طول وليس مجلد أخر بنفس الاسم وبه الملفات . أرجوا أن تكون المعلومه وصلت

والان نقوم بنقل مجلد phpmyadmin الى المسار التالي C:\Apache2\htdocs

الان في داخل مجلد phpmyadmin يوجد ملف بأسم config.inc قم بتحرير هذا الملف

قم بتعديل هذا السطر

```
37  * $cfg['PmaAbsoluteUri_DisableWarning'] variable below.
38  */
39 $cfg['PmaAbsoluteUri'] = 'C:\Apache2\htdocs\phpMyAdmin';
40
41
```

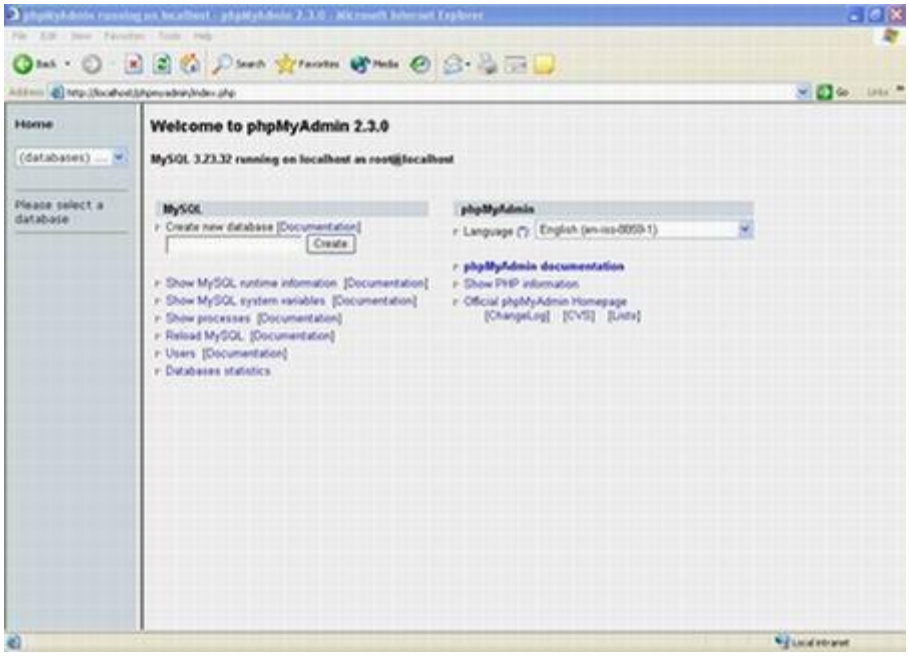
في بعض الاجهزة أذا ما نفع الامتداد الاول أكتب التالي : http://localhost/phpmyadmin

وايضاً قم بأضافة كلمة المرور أذا كنت قد وضعتها عند تنصيب برنامج MySQL وأيضاً قم بتغيير أسم المستخدم أذا غيرته في البرنامج من قبل أو أتركه كما هو أذا لم تقم بتغييره مع ملاحظة أنه سوف يتكرر ثلاث مرات أي قم بتعديله ثلاث مرات . والصوره سوف توضح لك ما يجب تغييره فقط

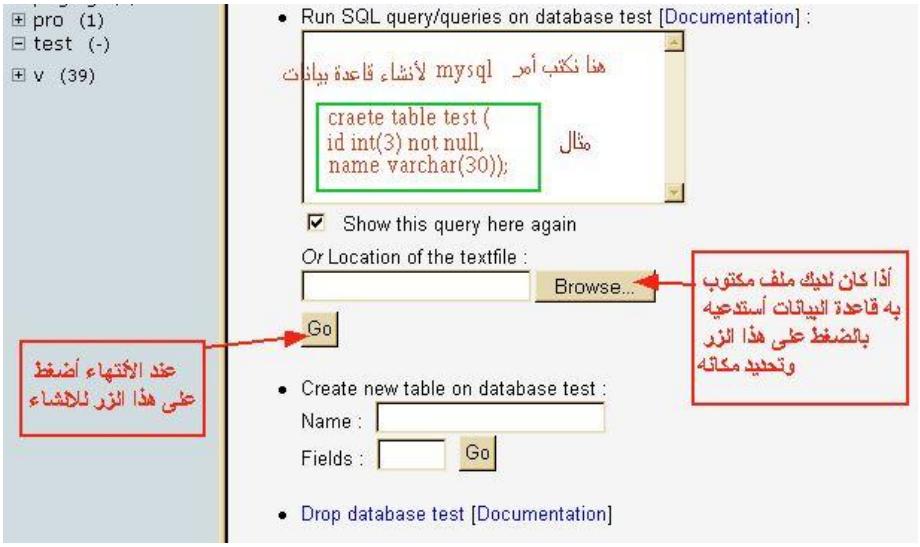
```
71 $cfg['Servers'][$i]['auth_type']     = 'config';
72 $cfg['Servers'][$i]['user']          = 'root';
73 $cfg['Servers'][$i]['password']      = '0000';
74
75 $cfg['Servers'][$i]['only_db']       = '';
```

الأن قم بكتابة هذا في متصفحك http://localhost/phpmyadmin/index.php

سوف تشاهد الاتي



لأنشاء قاعدة بيانات أتبع هذه الصورة



هذا الكود عباره عن تجربه لك في أنشاء جدول

```
) CREATE TABLE info
،id int(3) NOT NULL auto_increment
،'' fname varchar(15) NOT NULL default
،lname varchar(15) default NULL
،'' address1 varchar(30) NOT NULL default
،address2 varchar(30) default NULL
،address3 varchar(30) default NULL
،'postcode int(5) unsigned NOT NULL default '0
،'' country varchar(15) NOT NULL default
،'prim_tel int(10) unsigned NOT NULL default '0
،sec_tel int(10) unsigned default NULL
،'' email varchar(20) NOT NULL default
،brithday date default NULL
(PRIMARY KEY  (id
؛TYPE=MyISAM (
```

الان لاختبار الاتصال مع قاعدة البيانات أفتح النوت باد وقم بوضع هذا الكود به

```
php?>
;'dbServer='localhost$
// ضع أسم المستخدم وكلمة المرور لقاعدة البيانات
;'dbUser='root$
;'dbPass='0000$
// ضع أسم قاعدة البيانات
;'dbName='test$
link = mysql_connect("$dbServer", "$dbUser", "$dbPass") $
or die("<font color=#ff0000><center>لايمكن الاتصال بقاعدة
;("<center></font/>البيانات
print "<h2><center><font color=#008000>تم الأتصال بنجاح مع
;"<font></center><h2><br/>قاعدة البيانات
mysql_select_db("$dbName") or die("<font
color=#ff0000><center>لايمكن العثور على قاعدة
;("<center></font/>بيانات
print "<h2><center><font color=#0000ff>تم العثور على قاعدة
;"<font></center></h2><br/>بيانات
;(mysql_close($link
<?
```

الان أعطي هذا الملف أي أسم المهم يكون أمتداده هو php لنفرض أنك أعطيته الاسم التالي db.php
الان أكتب هذا في المتصفح http://localhost/db.php

سوف تشاهد الصوره الاتيه وهي التي تخبرك أذا كان الاتصال صحيح أم لا مع قاعدة البيانات

تركيب Perl + CGI مع Apache

المتطلبات كتالي :

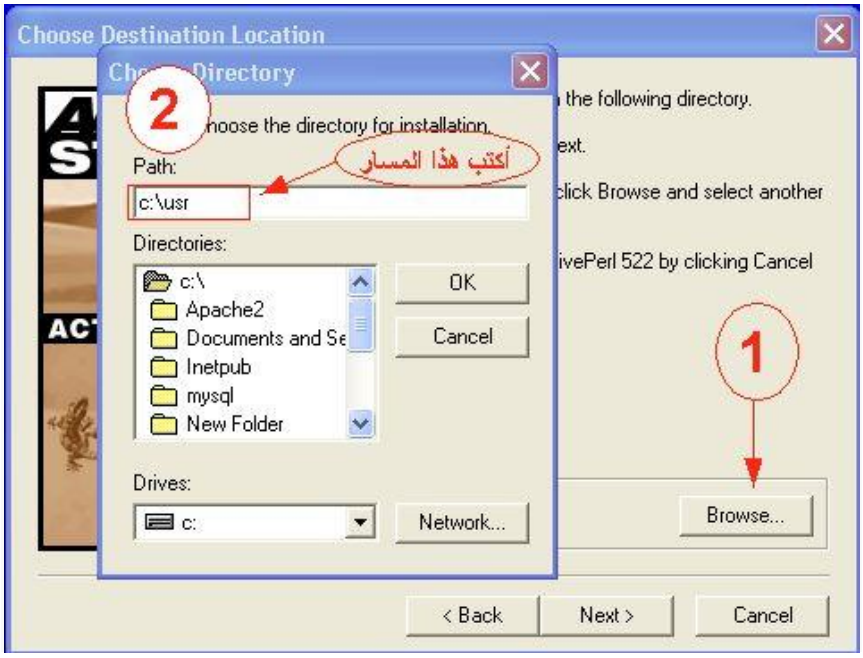
•برنامج Perl ويمكنك الحصول عليه من الموقع التالي

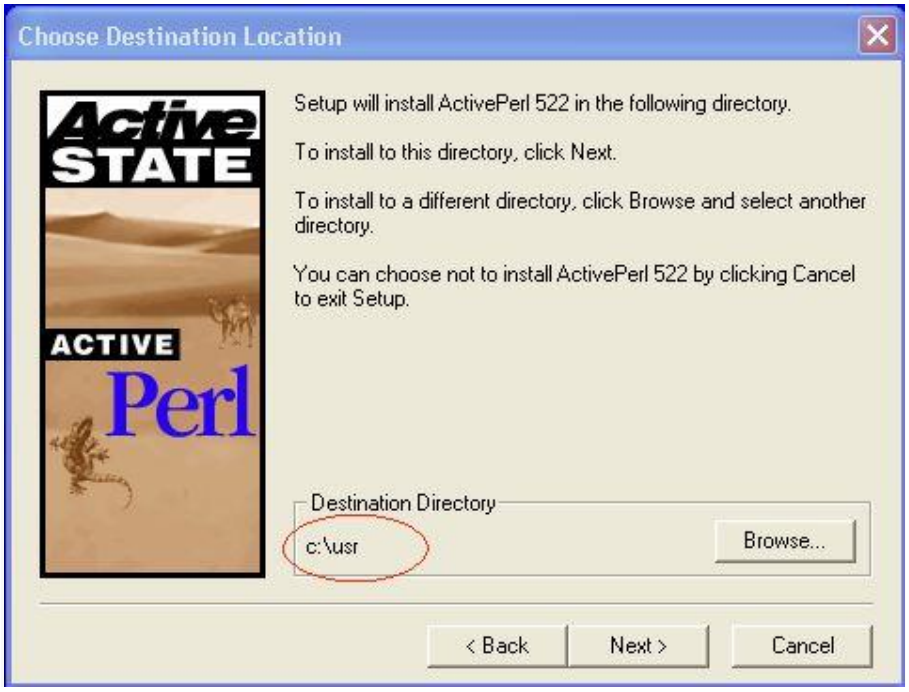
http://www.activestate.com/ActivePerl/download.htm

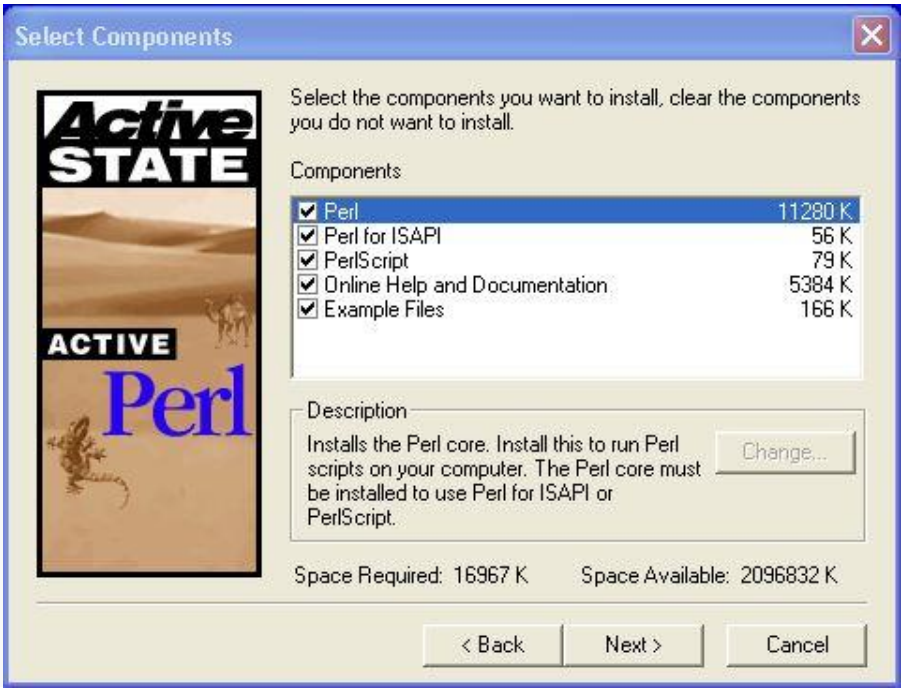
يفضل أن تستخدم أصدار جديد . أنا في هذا الدرس أستخدم أصدار أقدم النه الموجود حالياً لدي

بعد الحصول على البرنامج قم بتنصيبه

في هذه الخطوه يجب تغيير مسمى المسار الى usr وهذا مهم جداً لانه يوفر عليك الكثير من العناء







قم بالضغط على Next حتى النهاية

الان أذهب الى الملف الاتي httpd.conf **وسوف تجده على هذا المسار** C:\Apache2\conf

وقم بتحرير هذا الملف لاجراء بعض التعديل به

1 - أبحث عن هذه الجمله Options Indexes FollowSymLinks وقم بتغييره الى الاتي

Options Indexes FollowSymLinks ExecCGI Includes كما تلاحظ في الصوره التالية

```
# for more information.
#
    Options Indexes FollowSymLinks ExecCGI Includes

#
# AllowOverride controls what directives may be placed in .h
```

2 - أبحث عن هذه الجملة #AddHandler cgi-script .cgi وقم بأزالت علامة # كما في الصوره التالية

```
# (You will also need to add "ExecCGI" to th
#
AddHandler cgi-script .cgi

#
# For files that include their own HTTP hea
```

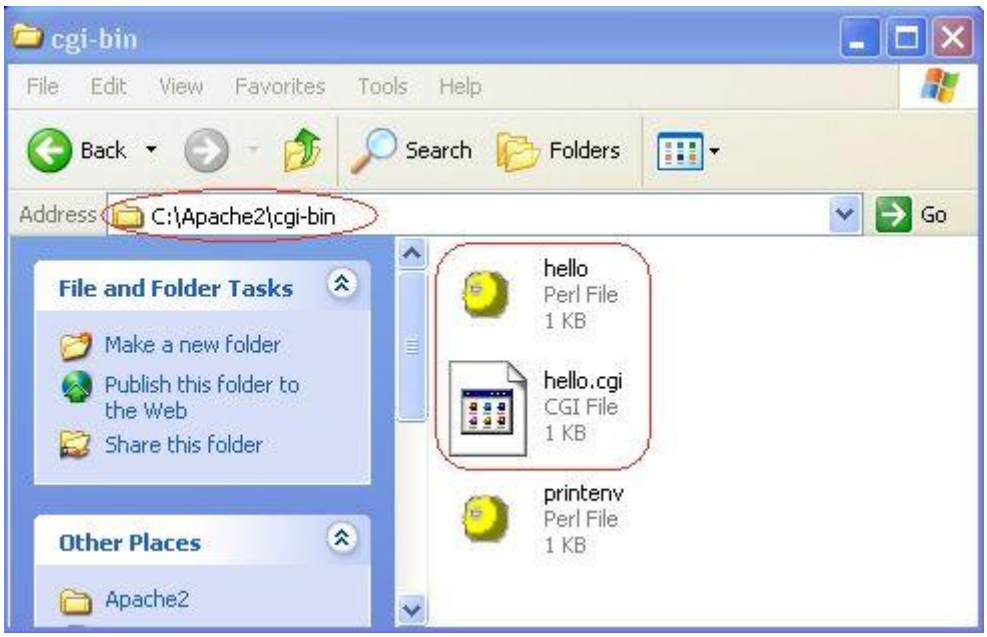
الان سوف نختبر عمل CGI + Perl على Apache

قم بفتح أي محرر وأكتب به الاتي

```
#!/usr/bin/perl
print "Content-type:text/html\n\n";
print "hello Majed";
```

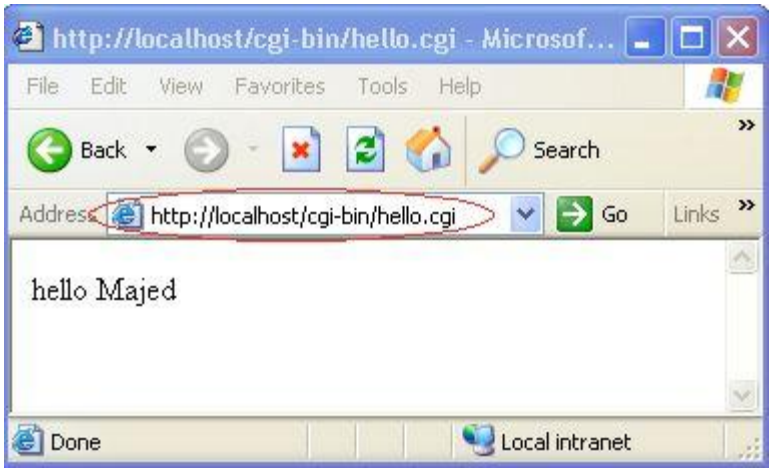
وقم بحفظه مرتين مره بأسم hello.cgi ومره بأسم hello.pl

على هذا المسار C:\Apache2\cgi-bin



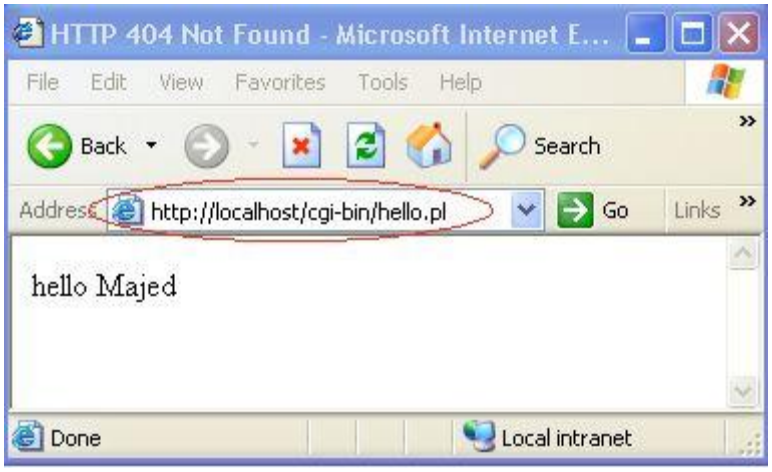
الان في المتصفح أكتب هذا المسار http://localhost/cgi-bin/hello.cgi

سوف تشاهد الصوره الاتيه



الان في المتصفح أكتب هذا المسار http://localhost/cgi-bin/hello.pl

سوف تشاهد الصوره الاتيه



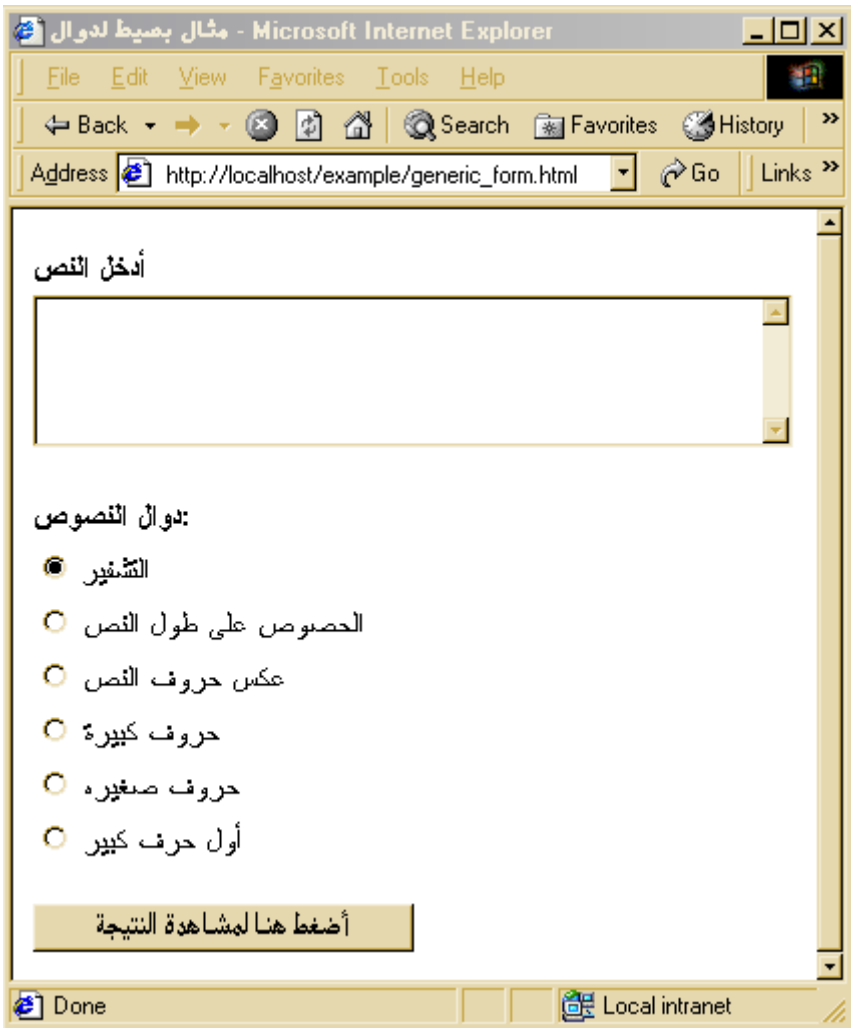
الان جهازك به سيرفر Apache ويدعم الاتي PHP + Mysql + CGI + Perl وبه أدارة قواعد البيانات PHPMyAdmin

مثال بسيط :

أولاً قم بكتابة هذا الكود وأحفظه بأسم : generic_form.html

```
<HTML>
<HEAD>
<TITLE/>مثال بسيط لدوال<TITLE>
<HEAD/>
<BODY>

<FORM METHOD="post" ACTION="display_input.php">

<strong><br/>أدخل النص<P><strong>
<TEXTAREA NAME="text1" COLS=45 ROWS=5 WRAP=virtual></TEXTAREA>
<p/>

<strong><br/>:دوال النصوص<P><strong>
<br>التشفير <INPUT TYPE="radio" NAME="func" VALUE="md5" checked>
<br>الحصوص على طول النص <"INPUT TYPE="radio" NAME="func" VALUE="strlen>
<br>عكس حروف النص <"INPUT TYPE="radio" NAME="func" VALUE="strrev>
<br>حروف كبيرة <"INPUT TYPE="radio" NAME="func" VALUE="strtoupper>
<br>حروف صغيره <"INPUT TYPE="radio" NAME="func" VALUE="strtolower>
<p/>أول حرف كبير <"INPUT TYPE="radio" NAME="func" VALUE="ucwords>

<p/><"أضغط هنا لمشاهدة النتيجة"=P><INPUT TYPE="submit" NAME="submit" VALUE>

<FORM/>

<BODY/>
<HTML/>
```

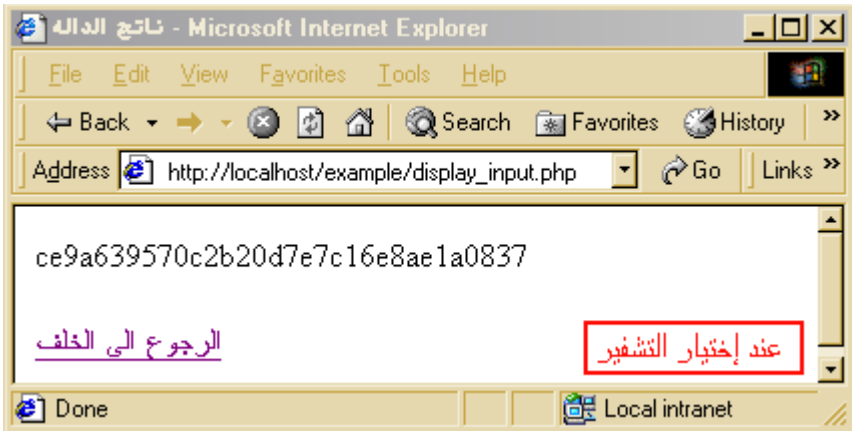
وعند كتابة العنوان التالي في المتصفح : http://localhost/example/generic_form.html



ثم قم بكتابة هذا الكود وحفظه بأسم : display_input.php

```
<?
If ($func == "" )
{
header("Location: http://localhost/generic_form.html");
exit;
}
$result = $func($text1);
?>
<HTML>
```

```
<HEAD>
<TITLE/>ناتج الداله<TITLE>
<HEAD/>
<BODY>

<? Echo "$result"; ?>

<p><a href="generic_form.html>الرجوع الى الخلف</a></p>

<BODY/>
<HTML/>
```



أي ملاحظات أو أستفسار بخصوص هذا الدرس أو أضافه أرجوا مراسلتي على العنوان التالي

phpvillage@yahoo.com

www.phpvillage.com

شرح لبعض وظائف دوال قاعدة البيانات

```
mysql_connect("localhost","User Name","Password")
```

تحتوي هذه الدالة على ثلاثة براميتر هما

Localhost : وهذا هو الخادم في السيرفرات الشخصيه

User Name : أسم المستخدم لقاعدة البيانات أذا كنت قد وضعت أسم مستخدم لقاعدة البيانات

Passowrd : كلمة المرور لقاعدة البيانات أذا كنت قد وضعت كلمة مرور لقاعدة البيانات

ملاحظة : المنفذ المعرف أتوماتيكياً لقاعدة البيانات هو : 3306

وظيفتها : هي الأتصال مع قاعدة البيانات

.....

```
mysql_select_db("Name DataBase",link_identifier)
```

تحتوي هذه الدالة على براميترين لكن الثاني أختياري

Name DataBase : أسم الجدول المنشاء في قاعدة البيانات

Link identifier : يعتبر مرجع للاتصال بي قاعدة البيانات

مثال على Link identifier :

```
$link = mysql_connect("localhost","root","123")

mysql_select_db("Table",$link)
```

وظيفتها : هي ربط بي جدول قاعدة البيانات

.....

mysql_query("Query", link_identifier)

تحتوي هذه الدالة على براميترين لكن الثاني أختياري

Query : هو الاستعلام في قاعدة البيانات

Link identifier : يعتبر مرجع للاتصال بي قاعدة البيانات

ماذا نقصد بي الأستعلام ؟

هي أوامر SQL عاديه ومثال عليها كتالي

```
select *
from table1
where id = '".$id"'
```

مثال على Link identifier :

```
$link = mysql_connect("localhost","root","123")
mysql_query ("Table",$link)
```

وظيفتها : هي أظهار أو أخراج نتائج من قاعدة البيانات على حسب الاستعلام

.....

mysql_num_rows(result_identifier)

تحتوي هذه الدالة على براميتر واحد

Result Identifier : هو معرف يحتوي على ناتج تنفيذ أمر [mysql_query]

مثال :

```
$result = Mysql_query("select * from Table1")

mysql_num_rows($result)
```

وظيفتها : هي معرفة عدد الصفوف (أي عدد البيانات في قاعدة البيانات) وينتج لنا عدد بعددها

.....

mysql_fetch_array(result_identifier,result_type)

تحتوي هذه الدالة على براميترين

Result Identifier : هو معرف يحتوي على ناتج تنفيذ أمر [mysql_query]

Result type : هو معرف ثابت لنوع المصفوفة التي سوف تنتج وهي ثلاثة أنواع كتالي

MYSQL_NUM : وهي تقوم بي أرجاع أرقام الفهارس للمصفوفة

MYSQL_ASSOC : وهي تقوم بي أرجاع قيمة للمصفوفة

MYSQL_BOTH : وهي تقوم بي أرجاع أرقام الفهارس أ و قيم للمصفوفة

حتى تتضح اليك الصوره سوف نضرب لك مثال على MYSQL_NUM

```
$result = mysql_query("select * from table1)

$row = mysql_fetch_array($result,MYSQL_NUM)


print $row[1];

print $row[2];
```

وفي هذا المثال نعرف طريقة التعامل مع MYSQL_ASSOC

```
$result = mysql_query("select * from table1)

$row = mysql_fetch_array($result,MYSQL_ ASSOC)


printf ("ID: %s  Name: %s", $row[0], $row[1]);
```

لاحظ أن القيمة التي سوف نحصل عليها من $row[0] سوف تنتقل الى %s الاول المجاوره لـ ID

و $row[1] سوف تنتقل الى %s المجاوره لـ Name

وفي هذا المثال سوف نعرف طريقة التعامل مع MYSQL_BOTH

```
$result = mysql_query("select * from table1)

$row = mysql_fetch_array($result,MYSQL_ ASSOC)


printf ("ID: %s  Name: %s", $row[0], $row["name"]);
```

ملاحظة مهمة : في الطريقتين الاخيره يجب أن تستعمل لطباعة printf غير هذا سوف يعطيك خطاء

وظيفتها : هي وضع كل عامود في مصفوفة لاظهار نتيجتها

ملاحظة : في الغالب عندما نريد أظهار كل البيانات نستعمل LOOP معها . أذا لم نستعمل LOOP سوف يظهر لنا ناتج صف واحد فقط

.....

mysql_free_result(result_identifier)

تحتوي هذه الدالة على براميتر واحد فقط

Result Identifier : هو معرف يحتوي على ناتج تنفيذ أمر[mysql_query]

مثال على كيفية أستعمالها

```
$result = mysql_query("select * from table1)

mysql_free_result($result)
```

وظيفتها : تحرير (أي تفريغ) كل البيانات من الذاكره

نحتاج الى هذه الداله عندما نستعلم عن أشياء كثيره ولا نريد ضغط الذاكره لدينا ففي نهاية كل برنامج نضع هذه الداله لتحرير الذاكره من البيانات التي تم الاستعلام عنها

.....