المملكة العربية السعودية
المؤسسة العامة للتدريب المهني والتقني
الادارة العامة للتدريب المهني والأهلي
معهد التميز الكندي العالي للتدريب

برمجة الانترنت

المملكة العربية السعودية
المؤسسة العامة للتدريب التقني والمهني
الإدارة العامة للتدريب الاهلي
معهد التميز الكندي العالي للتدريب

برمجة الانترنت

اعداد: سارة ال سنان.

المحاضرة الاولى

لغة (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 وJava وPerl 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.

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.

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:\httpd\`

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:\httpd\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 المخرج 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.

المحاضرة الثانية

```
<”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 وكود وعلامات ولغة html وقد لاتحتوي على نصوص html.
لكي يعمل الكود يجب أن يكون إمتداد الملف php أو بأي إمتداد من إمتدادات الـphp
مثلاً php3 وphtml
--
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

بروتوكولات الانترنت :

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 آخرعندما 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

.

عندما تقوم بعملية طلب لصفحة من السيرفر هناك أمور إضافية ترسل مع عملية الطلب http request غير الـURL 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 الـHTTP request والـHTTP response لأن سكربت الـPHP 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 الـrequest 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 .

إذن كمثال على الـ Request Line أنظر إلى السطر التالى :
GET /testpage.htm HTTP/1.1

الـmethod يخبر السيرفر كيف يتعامل مع الطلب هناك ثلاثة أنواع شائعه من الـmethod

HTTP Header -2
البت الثاني من المعلومات هو الهيدر HTTP Header .الذي يحتوي على تفاصيل أو وثائق عـن العميـل مثـل 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 .

وهذا مثال :

* / * :Accept

.Accept language: Arabic-KSA
.Connection: Keep –Alive
Host : http://www.arabbuielder.com
Referer: http://www.arabbuielder.com/index.php?something=132
(.....;User –Agent :Iexploer (win98

مثلما ترى الـ HTTP Header عبارة عن إعداد يتكون من عدة سطور كل سطر يحتوي على قيم معينة .

هناك عدة سطور تشكل الـ HTTP header وأكثرها إختياري , يقوم الـHTTP بالإخبار عن إنتهاء معلومات الـ header بترك سطر فارغ (وهذا يكون في الـHTTP1.1) .

The HTTP Body -3 :
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 :
the Response Line -1
http header - 2
Http Body - 3

The Response Line - 1
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 .

مثال :

HTTP/1.1 200 OK

في هذا المثال يقوم الـresponse line بإرجاع القيمه 200 متبوعة بالكلمة OK هذه تشكل وتشير إلى نجاح الـ request ويكون الـresponse يحتوي على الصفحة المطلوبة والبيانات من السيرفر . ومثال آخر هو الشفرة 404 عندما تقوم بطلب صفحة ويفشل السيرفر في الحصول عليها .

HTTP Header - 2
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) .

كما قلنا سابقاً ، يتكون من عده سطور ويتم وضع سطر فارغ للإعلام عن إنتهاء الهيدر .

مثال :

HTTP/1.1 200 OK -the satus line
Date: Mon; 1st Nov 1999, 16:12:23 GMT -general header
Server : Apache/1.3.12 (Unix) (SUSE/Linux) PHP/4.0.2 -the response
Last-modified: Fri, 29 Oct 1999, 12:08:03 GMT -Entity Header

السطر الأول ناقشناه والسطر الثاني مفهوم من غير شرح ، السطر الثالث يقوم بتحديد البرنامج تبع السيرفر ونوعه ونظام التشغيل القائم عليه والسطر الأخير يقوم بتعريف آخر وقت تم فيه تعديل أو تجديد الصفحة .

ملاحظة : قد يحتوي الهيدر على أكثر من هذه المعلومات أو معلومات مختلفة وهذا يعتمد على نوع الشي المطلوب من السيرفر .

3 - Http Body
إذا تم معالجة الطلب بنجاح ، فإن الـHTTP response Body يحتوي على كود الـHTML 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 الـphp يتكون من ثلاثة أشياء : نص وكود php وكود html ، لانستطيع وصف الـhtml بأنها لغة برمجة بشكل جيد ونستطيع أن نقول أن الـphp لغة سكربتات Scripting Language لأنها تضيف قدرات html عليها مثل الجداول والفريمات بكود html بداخل كود الـphp هناك لغات تسمي لغات سكربتات قد تكون متآلفاً معها مثل الجافا سكربت والفجول بيسك سكربت بإستثناء أن الفرق بينها وبين الـphp هو أن الـphp لغة تعتمد على جهة المزود أي السيرفر ويمكنك تخصيص المتصفح الذي يستعرضها .
تجعلنا الـhtml نضمن سكربتات الـphp 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)

مفهوم الـparsing و الـExecution :

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 بالـParsing
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) .

المحاضرة الثالثة

التعليقات

لتنظيم العمل وتعديله 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 لافائدة له 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 :

```
$alfares = "How Are You Every Body?";
$اسم_المتغير = قيمه ;
```

لاحظ أن السطر السابق يتكون من خمسة أشياء :

1 / المتغير وهو alfares
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 :
1- اسماء المتغيرات حساسة لحالة الأحرف إذا كانت كبيرة وصغيرة

```
<?
$Ahmed  = "salem";
$ahmed = "slmoon";
echo $ahmed;
echo  $Ahmed;
?>
```

المتغيرين الذين بالأعلي مختلفين بسبب حالة الأحرف.

2 - 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 الـPHP 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 "خطابسبب وجود علامة فى 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 المعامل (\) بنفسه ؟
الحل هو أن نتبعه بمثله ، وبالمثال يتضح المقال :

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

```
<?
$first = "منتدي" ;
$last = "المطور العربي";
$fullname = $first. $last
Echo $fullname ;
//ولكننا نريد وضع فراغ بين الكلمتين
$fullname=  $first . ' ' . $last  ;
Echo $fullname ;
?>
```

وأيضا يمكننا أن ضيف إلى متغير قيمة متغير آخر :

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

```
<?
//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 الآحرف

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

هناك ثوابت يستخدمها النظام مثل

PHP_OS

التي تقوم بعرض نظام التشغيل الذي يستخدمه السيرفر

مثال :

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

معرفه وتحويل انواع البيانات

إذا أردت أن تعرف نوع متغير ما يمكنك استخدام الدالة gettype

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 الدالة settype :

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 ولاترجع 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);
?>
```

الدالة 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 .

المحاضرة الرابعة

داوال الوقت التاريخ

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

مثال :

```
<?
Echo gmdate (m);
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 الـphp 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 :

```
<?
Echo gmdate ("D, d M Y H:i:s")
?>
```

المحاضرة الخامسة

النماذج

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 الـhtml 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 الـhtml 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;CLASS;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 تقوم 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 الطريقتين 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 كالتالي :

1- كتابه عنوان الصفحه المصدر .
2- اتباعها بعلامة استفهام .
3- 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 عناصـر (مربع علامة , مربع نص , زر اختيار) ولكل من هذه العناصـر عنوان خاص بها (name) ولكل منها قيمه خاصه بها (value) .
وهي مشـابهة للمتغيرات ويمكن أن يحتوي عنوان الصـفحة 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 اسـم متغير من المفترض تعريفه في الصـفحه الهدف (التي سـنكتبها بالـPHP).

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

URL Encoding

هناك بعض الأحرف لايسـتطيع المتصفح إضافتها لعنوان الصفحة بصيغتها الحقيقية بل يسـتخدم لغة التشـفير 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 الـــget ولكنها لاترسل المعلومات في عنوان صفحة الانترنت بل تقوم وضعها في الـــbody التابع لـhttp response .
بالإضافه إلى أنه يستطيع ارسال البيانات بكمية أكبر من الـGET .

أيهما تستخدم GET أم POST ؟

قد يكون العيب في الخاصـية GET عدم سـرية المعلومات التي تقوم بكاتبتها ومن الممكن أن تظهر للشـخص الذي يجلس الى جوارك ... خاصة عندما تريد الحفاظ على سرية معلوماتك .
أضف إلى ذلك أنها غير مفيدة في النصوص الكبيرة الحجم .
ولكنها مفيدة في أشـياء كثيرة فمثلاً محركات البحث يجب أن تسـتخدم هذه الخاصـية لكي يسـتطيع المسـتخدم أن يسـتخدم عنوان البحث ويحتفظ به لوقت آخر ولا يقوم من جديد بكتابة الكلمة التي يبحث عنها .

أيضا الـــPOST مفيدة في إخفاء المعلومات وإحتواء كميات كبيرة من البيانات ولكن لايمكن الاحتفاظ بعنوان الصـفحة .... مع ذلك فإنها أيضـا ليسـت جيدة في الحماية بحيث أن أي هاكر خبير يمكنه الحصـول على المعلومات إذا لم يكن لها تشـفير معين في نقلها ..لكن اذا اردت فعلاً ان تجعلها محمية فيجب عليك اسـتخدام اتصال محمي الى سـيرفر محمي .... او ماسـمونه (SCURE CONNECTION TO SCURE SERVER) .

أدوات التحكم في النماذج :

في الواقع أن أدوات التحكم عبارة عن مربعات النصـوص العاديه (التي يدخل فيها المسـتخدم اسـمه وعنوانه) وازرار الراديو (والتي يقوم المسـتخدم فيها باختيا شـي معين (مثل الوجبه المفضـلة لديه او المشـروب المفضـل 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 كود HTML 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 = “textbox.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 (textbox.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 كـ php . وقم بتسميته textbox.php .
7 - الآن قم بأخذ الملفين وضعهما في مجلد السيرفر لديك .
8-قم بتشغيل السيرفر واكتب في مستعرض الانترنت لديك
http://localhost/textbox.html
9 - قم بكتابة وجبتك المفضلة واضغط زر إرسال .
10- ستظهر النتيجة .

لاحظ كيف ظهر العنوان :
http://localhost/textbox.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="textbox.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 تسمية 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 وRESET

1- الـsubmit يقوم بإرسال المعلومات .

2- الـreset 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 الـURL 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

1- قم بفتح محرر النصوص لديك

2- 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
مربى
مكرونة
بيف برغر
سمبوسة
معصوب
مطبق
ماشادونا
ماخلص لو قعدت اكتب هاها
</TEXTAREA>
<br>
```

```
<INPUT TYPE = SUBMIT VALUE ="قم بإرسال الطلبات إلى الجرسون">
</FORM>
</html>
```

3- قم بحفظ الملف باسم TAREA.html .
4- الآن قم بفتح ملف جديد في محرر النصوص .
5- قم بكتابة الكود التالي :

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

6- قم بحفظ الملف باسم tarea.php
7- قم بوضعهما في مجلد السيرفر لديك .
8- قم بتشغيل البرنامج .

http://localhost/tarea.html

9- قم بضغط الزر لارسال البيانات .
10- شاهد النتيجه.

الشرح

لانضيف شيئاً على قولنا هنا سوى أننا نريدك أن تلاحظ كيف جهزنا القيمة الإفتراضية بكتابة نصوص بين وسومات الـــــtextarea
وأيضا أننا استخدمنا الأسلوب POST في ارسال البيانات مما جعلها لاتظهر في شريط العنوان .
وأن الـــNAME تحدد اسم المتغير التى ستذهب إليه القيمة واسم المتغير فى 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 = "swalif" value= "سوالف" checked>
```

نقوم بتحديد نوع الأداة بأنها مربع علامة في هذا الجزء

```
TYPE = "CHECKBOX"
```

نقوم بتحديد اسم المتغير في هذا الجزء

```
NAME = "swalif"
```

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

```
value= "سوالف"
```

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) :

1- 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>
```

2- قم بحفظ الملف باسم check.html .
3- قم بفتح ملف جديد في المفكرة وقم بكتابة التالى :

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

4- قم بحفظ الملف باسم check.php .
5- قم بنقل الملفين الى مجلد السيرفر .
6- اكتب في المتصفح

http://localhost/check.html

7- النتيجه

تطبيق عملي (2) :

1- 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>
```

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

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

3- قم بحفظه باسم check2.php
4- قم بتشغيل الملف .
5- النتيجه

تطبيق عملى (3)

1- 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>
```

2- قم بحفظه باسم check3.html وافتح محرر النصوص من جديد واكتب الكود التالي :

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

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

4- قم بتشغيل البرنامج

http://localhost/check.html

5- 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 :

1- 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>
```

2- قم بفتح محرر النصوص واكتب الكود التالي وقم بخفظه باسم radio.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 باذن الله .

تطبيق عملي :

1- 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="إرسال">
</html>
```

2- 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 معنى 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 الـPHP إذ أنها تسمح لنا أيضا بان تكون المعلومات المخفيه هي متغيرات PHP .

يتم صنع هذه الحقول المخفية كالتالى :

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

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

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

```
<?
$msg1= " هذه العبارة لن تظهر ";
```

```
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>
<?
$car1= "لكزس";
$car2= "ماكسيما";
$car3="لاندكرزور";
Echo "<form method =get action='hid2.php'>";
Echo "ماهي السيارة التي تتمنى أن تشتريها أو تحظي بها ؟";
Echo "
<select name= 'favcar'>
<option>$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'>
<input type = submit  value='إرسال'>
</form>";
?>
</body>
</html>
```

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

```
<html dir="rtl">
<head></head>
<body>
<?
Echo ": لقد قمنا بعرض السيارات التالية عليك<br>";
Echo "$hid1<br>";
Echo "$hid2<br>";
Echo "$hid3<br>";
Echo "<br>:ولقد قمت باختيار<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 الـphp 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>
<input type="text" name ="user">
<br>
كلمة المرور
<input type="password" name ="pass">
<br>
<input type = submit  value=’إرسال’>
</form>
<body>
</html>
```

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

```
<?
Echo “ : اسم المستخدم هو”;
Echo “<br>$user<br>”;
Echo “: وكلمه المرور هي”;
Echo “<br><br>$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
<input type="text" name = "sub">
<br>
الرساله
<textarea rows=10 cols=20 name = "msg" >
</textarea>
<input type="submit" value = "إرسال البريد الالكتروني">
</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 .

برامج عمليه

برنامج او سكربت ارسال بطاقات بسيط
يحتوي من ملفين الملف الأول به البطاقات وعنوان البريد الإلكتروني والملف الثاني هو الذي يقوم بعملية الإرسال

الملف الأول هو chcard.php وكوده كالتالى :

```
<html dir="rtl">
<form action =card.php method = "post">
اختر البطاقة التي تريد ارسالها
<br>
<br>
<INPUT TYPE = "radio" NAME = "card" value= " http://www.khalaad.f2s.com/MADINA9_small.JPG" checked>
البطاقه الاولى
<br>
<br>
<img src="http://www.khalaad.f2s.com/MADINA9_small.JPG" width="100" height="100" alt="" border=0>
<br>
<br>
<INPUT TYPE = "radio" NAME = "card" value= " http://www.khalaad.f2s.com/Haram3.jpg" >
البطاقه الثانيه
<br>
<img src="http://www.khalaad.f2s.com/Haram3.jpg" width="100" height="100" alt="" border=0>
<br>
اسمك
<br>
<input type="text" name = "myname">
<br>
بريدك الالكتروني
<br>
<input type="text" name = "you">
<br>
بريد صديقك
<br>
<input type="text" name = "to">
<br>
موضوع التهنئه
<input type="text" name = "sub">
<br>
```

```
الرساله
<br><br>
<textarea rows=10 cols=20 name = "msg" >
</textarea>
<br>
<INPUT TYPE = submit  value= "إرسال البطاقة" >
</form>
</html>
```

الملف الثاني يقوم بعملية ارسال البطاقة وتقوم بكتابة الكود التالي واحفظه في ملف باسم card.php

```
<?
$message = " لقد قام$myname بارسال بطاقه اليك " . "\n". " وهو يقول في نص رسالته لك:\n$msg"." وتجدها على الرابط
التالى" ."\n". "$card" ."\n";
mail("$to", "$sub", "$message","From:$you");
echo "<center> لقد تم ارسال الرساله بنجاح،،، مبارك</center>";
?>
```

ملاحظة :

الدالة \n تقوم فقط ببدا سطر جديد لاننا لانستطيع استخدام
 في نص الرسالة

المحاضرة السادسة

الأوامر الشرطية

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 الـphp 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 الـPHP 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 الأشـياء التي لاتحتمل أكثر من احتمالين وهما إما صـح وإما خطأ ، وهي نوع جديد من القيم غير التي كنت نعرفها سـابقا (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 والأصـــغر من في الرياضـــيات التي تتعلمها في المدرسة مما يجعل فهم هذا الأمثله بسيطاً .

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

سيقوم الـــPHP 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)
{
echo (" رقم الحظ أصغر من الرقم ستة");
}
?>
```

مثال (2)

```
<html dir="rtl">
<?
$f=5;
$r=10;
If ($f >$r)
{
echo " المتغير $f أكبر من المتغير r";
}
?>
```

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

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

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)
{
Echo " 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 الـــphp4.01 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)
{
Echo " القيم متساوية";
}
?>
```

مثال (2) :

```
<?
$m="11";
$u=11;
If ($m===$u)
{
Echo " القيم متساوية";
}
?>
```

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) echo "القيم غير متساوية";
?>
```

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 الـــ(<>) وهو يقوم بارجاع قيمة (true) إذا كانت القيمتين مختلفتين عن بعضهما أي أنه مثل علامة =! تقريباً .

مثال :

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

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>
<input type ="radio" name = "man" value="عمر">
عمر بن الخطاب رضي الله عنه
<br>
<input type ="radio" name = "man" value="أبوبكر">
أبو بكر الصديق رضي الله عنه
<br>
<input type ="radio" name = "man" value="عثمان">
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>
<?
If ($q==" عمر") echo "الإجابة صحيحة";
If ($q!="عمر") echo "الإجابة خاطئة";
?>
```

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 السـابقين قمنا بعملية التحقق من أكثر من شـرط باسـتخدام المعاملين (&& و and) فعندما تحققت جميع الشـروط 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 أصـلاً 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 .

```
<?
$F="الفارس";
IF !($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 بالـــphp 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 ) echo ("ممتاز" . "<br>");
$t = 5;
If ($t <= 9 ) echo ("جيد جدا");
?>
```

تجميع المعاملات

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)) echo "تحققت جميع الشروط";
?>
```

سيتم طباعة 18 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)
 {
echo "مرحبا بك في أكبر موقع تجاري إلكتروني";
}
else
{
echo "ممنوع دخول الأطفال الموقع لأنهم لايملكون المال";
}
?>
```

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)
 {
echo "مرحبا بك في أكبر موقع تجاري إلكتروني";
}
elseif ($y >= 44);
{
echo " مافي مشكلة برضه إذا كنت كبير";
```

```
}
else
{
echo "ممنوع البقية";
}
?>
```

تعشيش العبارات الشرطية

يمكنك تعشيش العبارات الشرطية ، ونعني بتعشيش العبارات الشرطية هي أن تقوم بعملية تعشيش الشروط فمثلاً إذا كان شرط ما صحيحاً فإنه يجب أن يكون شرط آخر صحيحاً لكي يتم حصول شي معين وغير ذلك .

مثال :

```
<?
$h="ahmed";
$f=45;
If  ($h = = "ahmed" )
{                    If ( $f= = 45)
          {
          echo "الاسم والرقم صحيحان";
                    }
                     else
                    {
                    echo (" الرقم غير صحيح" );
          }          }
else      {
echo " اسم تسجيل الدخول غير صحيح" ;
}
        ?>
```

هذا مجرد مثال بسيط جداً لتعشيش 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

1- قم بانشاء ملف Msabqa.html .
2- قم بكتابه الكود التالي فيه :

```
<html>
<body>

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

</p>

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

</body><html>
```

قم بفتح ملف وقم بتسميته msabqa.php

```
<?
<html dir = "rtl">
If $s == "3mar"{
 الإجابه صحيحة
}
else
{
echo "الإجابه خاطئة";
}
?>
```

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;

تقوم بعملية الخروج من الكود نهائياً ولاتطبق أي أوامر بعدها ، وفي الأمثله التوضيحية التالية ستجد أن break; تخرج من العبارة فقط (Statement) بينما الـــexit; 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”:
                Echo “مسموح “;
                Break ;
 Case “khaled “ :
                Echo “ممنوع “ ;
                Break ;
 Case “salem“ :
                Echo “ممنوع “ ;
                Break ;
  Case “Mohmed “ :
                Echo “مسموح “ ;
                Break ;
Default ;
        Echo “لقد ادخلت اسم غير صالح”;
}
?>
```

مثال (2)

```
Switch ($g) {
        Case $g>50:
                Echo “كبير “;
                Break ;
 Case  40 :
                Echo “لاباس “ ;
                Break ;
 Case ($g<15) :
                Echo “أطفال ممنوع “ ;
                Break ;
  Case 30 :
                Echo “مسموح “ ;
                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”>
<input type=submit value=”ارسال”>
</form>
</html>
```

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

```
<?
 Switch ($g) {
          Case $g>50:
                    Echo “كبير “;
                    Break ;
  Case  40 :
                    Echo “لاباس “ ;
                    Break ;
  Case ($g<15) :
                    Echo “أطفال ممنوع “ ;
                    Break ;
   Case 30 :
                    Echo “مسموح “ ;
                    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 .

التخلص من وسوم الـ html

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">
<input type=submit value="ارسال">
</form>
```

```
</html>
```

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

```
<?
Echo " هذا هو الشكل الطبيعي للعبارة عند طباعتها";
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 كـــــ 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);
Echo " هذا هو الشكل بعد استخدام الدالة";
Echo "<br>" . $fname;
 ?>
```

المحاضرة السابعة

التكرارت والمصفوفات

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++;
}
?>
```

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
while (condition شرط);
```

مثال :

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

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;
}
 ?>
```

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 $u ويتم تنفيذ الكود حتي يصبح المتغيرu$ 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[ ] = "alfareees";
$A[ ] = 13;
?>
```

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

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

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

مثال :

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

هل لاحظت أيضا أننا لم نقم بتعريف نوع متغيرات المصفوفة وقام الـــPHP بتعريفها تلقائياً بدلاً منا فمرة استخدمنا قيمة حرفية ومرة استخدمنا رقماً ورغم ذلك فلم يقم الـPHP بعمل أي اعتراض إضافة إلى ذلك فإن الـPHP 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" ] = "alfareees";
$A["b" ] = 13;
?>
```

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

مثال :

```
<?
$r ["aa"] = "ahmed ali";
$r [1] = 13273;
$r [20] = 13273;
echo $r[aa];
echo $r[20];
echo $r["aa"];
?>
```

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

يمكننا تعريف المصفوفات أيضا بطريقة أخري

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

مثال :

```
<?
$t =array ("ahmed", "ali", "salem", "alfarsi");
echo $t [0];
?>
```

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

الفهرسه Index	العنصر Element
0	Ahmed
1	Ali
2	Salem
3	alfarsi

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

```
<?
$r = array (1=>"ahmed", "ali","salem", "alfarsi");
?>
```

عند تعريفك لرقم الفهرسة للقيمة الأولى سيقوم الـPHP بإعطاء أرقام فهرسة بشكل تسلسلي ،
عندئذ ستصبح الفهرسة كالتالى :

الفهرسه Index	العنصر Element
1	ahmed
2	Ali
3	salem
4	alfarsi

هناك طريقة لتكون أيضا الفهرسة هي عبارة عن حروف :

```
<?
$r = array (“ss”=>”ahmed”, “sf”=> “ali”, “da”=>”salem”, “bv”=> “alfarsi”);
?>
```

عندئذ ستصبح الفهرسة كالتالى :

الفهرسه Index	العنصر Element
Ss	Ahmed
Sf	Ali
Da	Salem
Bv	Alfarsi

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

مثال :

```
$r [ss]= “لمياء”;
```

لاحظ أننا قمنا بتغيير القيمة من (ahmed) 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 (“ahmed”, “ali”, “salem”, “alfarsi”);
?>
```

واردت أن تطبع أسماء جميع الاشخاص المتواجدين فيها

أولاً نحن نعرف أن المصفوفة إذا لم نقم بتعريف رقم فهرسة لها فإن الـــPHP 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 :

```
<?
$people =array (“ahmed”, “ali”, “salem”, “alfarsi”);
echo “$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 :

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

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

```
<?
$people =array ("ahmed", "ali", "salem", "alfarsi");

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

```
?>
```

لاحظ أننا بدأنا العداد بالقيمة صـفر ثم اشـترطنا أن يكون أقل من 4 لأن آخر عنصـر في المصـفوفة رقم فهرسـته 3 ثم قمنا بجعله يزداد بقيمة 1 لأننا نريد طباعة جميع عناصر المصفوفة وقمنا بوضع رقم العداد في خانة الفهرسـة وعلى ذلك سـيتم 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>
<input type=submit value = "لذيذ" >
</form>
```

في ملف الـarray.php اكتب :

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

لقد عرضنا في القائمة خمسـة عناصر ... لاحظ أننا وضعنا في اسـم المتغير للقائمة قوسـين [ ] لكي يتعرف الـــhtml على أنه سـيتم تخزين البيانات تلقائياً بعد ذلك قام الـــPHP بفهرسـة العناصـر التي تم إرسـالها من قبل العميل سـواء كانت ثلاثة أو أربعة ولكنها بالطبع لن تزيد على خمسـة .... على ذلك سـيكون 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" >
<?
$shrab =array("شاي","قهوة","كابتشينو","توت","برتقال");
For ($k=0;$k<4;$k++)
{
echo "<option>".$shrab[$k]."</option>";
}
?>
</select>
</form>
```

عند اختيار المســتخدم للقيمة ســيتم وضــعها في المتغير $s يمكنك مراجعة درس النماذج لكي تفعل ذلك ، هذا المثال يقوم بصــناعة مصــفوفة للمشــروبات 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 .

دوال المصفوفات

الدالة key

لنفرض أن لدينا مصفوفة مكونة من عنصرين :
مثال :

```
$s= array ("على","محمد");
```

الآن لنضف إليها هذه السطور

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

الدالة current()

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

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

```
<?
$s= array ("ع"=>"على","م"=>"محمد","ا"=>"احمد");
echo key($s)."<br>";
echo current($s) ."<br>";
?>
```

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

مثال :

```
<?
$s= array ("ع"=>"على","م"=>"محمد","ا"=>"احمد");
next($s);
echo key($s)."<br>";
echo current($s) ."<br>";
?>
```

```
<?
$s= array ("ع"=>"على","م"=>"محمد","ا"=>"احمد");
next($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 :

```
<?
$s= array ("ع"=>"على","م"=>"محمد","ا"=>"احمد");
next($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 :

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

سيقوم الـــPHP 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 :

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

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 "<br> $e<br> $r";
}
```

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 "<br> $e<br> $r";
}
```

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 :

```
<?
```

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

لاحظ معH اننا 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="100%" cellspacing="0" bordercolorlight="#000000"
bordercolordark="#000000" bordercolor="#000000">
<tr>
   <td align='center'>الاسم</td>
   <td align='center'>رقم التلفون</td>
  </tr>
<?
$s = array (658=>"465873 , "سالم"<=456546, "عادل);
While (list($e,$r) = each ($s))
{
echo "<tr><td align='center'>". $r . "</td><td align='center'>" .$e . "</td></tr>";
}
?>
</table>
```

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 :

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

فرز المصفوفات

هناك العديد من الدوال التي يوفرها لنا الـPHP 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","salem","hythem","Khaled","Ammar","Hesham");
```

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

```
<?
$NaNo=array (“ali”,“salem”,“hythem”,“Khaled”,“Ammar”,“Hesham”);
sort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo “<br> $e<br> $r”;
}
?>
```

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”=>”ahmed”, “kh”=> “khaled”);
```

وأردنا فرزها وطباعة الفهارس والقيم كما في المثال التالى :

```
<?
$NaNo=array ( “ad”=>”ahmed”, “kh”=> “khaled”);
sort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo “<br> $e<br> $r”;
}
?>
```

قارن ناتج المثال السابق مع هذا المثال :

```
<?
$NaNo=array ( “ad”=>”ahmed”, “kh”=> “khaled”);
asort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo “<br> $e<br> $r”;
}
?>
```

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 :

```
<?
$NaNo=array ( “ad”=>”ahmed”, “kh”=> “khaled”);
rsort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo “<br> $e<br> $r”;
}
?>
```

مثال :

```
<?
$NaNo=array ( “ad”=>”ahmed”, “kh”=> “khaled”);
arsort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo “<br> $e<br> $r”;
}
?>
```

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 :

```
RSORT()
<?
$NaNo=array ( "ad"=>"سالم", "kh"=> "احمد");
rsort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "<br> $e<br> $r";
}
?>
<br>--------<br>
ARSORT()
<?
$NaNo=array ( "ad"=>"احمد", "kh"=> "أحمد");
arsort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "<br> $e<br> $r";
}
?>
<br>--------<br>
ASORT()
<?
$NaNo=array ( "ad"=>"هاشم", "kh"=> "جمال");
asort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "<br> $e<br> $r";
}
?>
<br>--------<br>
SORT()
<?
$NaNo=array ( "ad"=>"هاشم", "kh"=> "جمال");
sort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "<br> $e<br> $r";
}
?>
```

الدالة ksort

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

مثال :

sort

```
<br>--------<br>
asort()
<?
$NaNo=array ( "ad"=>"هاشم", "kh"=> "جمال");
asort($NaNo);
While (list($e,$r) = each ($NaNo))
{
echo "<br> $e<br> $r";
}
?>
<br>--------<br>
ksort()
<?
$NaNo=array ( "ad"=>"هاشم", "kh"=> "جمال");
ksort($NaNo);
While (list($e,$r) = each ($NaNo))
{
```

```
echo "<br> $e<br> $r";
}
?>
```

لقد اعتمد الـphp 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 :

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

وأردنا أن نضيف عنصر جديد لها فقمنا بالتالى :

```
<?
$saher[ 5]=”salem”;
$saher[ 85]=”khaled”;
$saher[ 35]=”mohmed”;
$saher[ 19]=”hajeer”;
$saher[ ]=”Alfarees”;
?>
```

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 (ArrayNameاسم 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 :

```
<?
$saher[ 5]=”salem”;
$saher[ 85]=”khaled”;
$saher[ 35]=”mohmed”;
$saher[ 19]=”hajeer”;
array_push ($saher,Alfarees)
?>
```

مثال :

```
<?
$saher[ 5]=”salem”;
$saher[ 85]=”khaled”;
$saher[ 35]=”mohmed”;
$saher[ 19]=”hajeer”;
array_push ($saher,Alfarees,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

```
Array_pop(ArrayName اسم المصفوفه)
```

مثال :

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

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 على بين عناصر المصفوفة .

مثال :

```
<?
$stng =array ("ahmed", "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 ("ahmed", "salem", "ali", "alfarsi");
$r =implode ("-",$stng);
echo $r;
$r= explode ("-",$stng);
echo $r;
?>
```

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 الـPHP فإنه يقوم بتعريفها تلقائياً ويمكنك طباعة المتغيرات وقيمها مباشرة من غير تعريف .... ولكن هذه الميزة في الـPHP 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 الحصـول على البيانات فهي ما زالت موجودة ولكن يجب عليك أن تقوم بإسـتخدام 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">
ماهو اسم الطفل الذي استيقظ به العالم الاسلامي من غفلته قبل عده شهور !!
<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 الـphp.ini في اعدادات الـtrack_vars الذي يقوم بمنع السيرفر من استخدام

هذه المصـفوفات (هذه الميزة يمكن إلغاؤها في php4) .... على ذلك انصـحك بإرسـال رسـال تذمر وشـكوي إلى مزود الخدمة لديك .. تعلن فيها أن الأمر اصبح لايحتمل .

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 المسـتخدمة في السـيرفر لديك (وعلى كل حال إن اسـتطعت أن تقوم بالتركيز في صناعة عشر مصفوفات متداخلة بدون أي مشاكل أو مرض نفسي أو .... فأنت تستحق جائزة) .

يمكننا كتابة مصفوفة متداخلة كالتالى :

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

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

الشرح

هذا المثال قد يكون غامضـــاً جداً لكن فكرته بســـيطة أولاً افترض أنك تعلم عن list..each جيداً وتعرف صـــيغة التكرار الذي يستخدمهما .
الآن لدينا مصفوفة تتكون من رقمين للفهرسـة 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 .

echo
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 الـــPHP 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 سـهل ولكنه يحتاج الى تدرب فقط ، وعليك أن تتدرب وصـدقني أنني حاولت ان أبسـط المثال من أجلك ... أتمني أن تكون قد فهمت .

تطبيق عملي

افتح محرر النصوص لديك واكتب الكود التالي :

```
<?
Echo "<form method =post action = 'exam2.php' " ;
$boy=array ("أحمد" , "خالد", "سعد","حسن");
```

```
while (list(,$Name) = each ($boy))
{
echo " ماهى السنة الدراسية لـ $Name ؟";
Echo "<select name = 'school[]'>
<option>اول ثانوي</option>
<option>ثاني ثانوي</option>
<option>ثالث ثانوي</option>
</select>";
echo "<br><br>";
echo "<input type =hidden name =boy[] value ='$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 .

تكرار foreach

هذا التكرار هو من الأشياء الجديدة في الـphp4 وهو يساعدك على معرفة عناصر مصفوفة معينة أو طباعة محتوياتها .

```
Foreach ($ArrayName As $ArrayItem)
{
        code شفره
}
```

مثال :

```
<?
 $T= array (a=>"ahmed " , b => "basem", c=>"car")

Foreach ($T As $A => $r)
{
     echo $A ."-----". $r;
}
?>
```

الدالة count

تقوم بحساب عدد العناصر الموجودة في المصفوفة

مثال :

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

المحاضرة الثامنة

ترتيب الكود البرمجي

Function

الدالة هي جزء من كود البرنامج يتم تعريفه عن طريق المبرمج ليتم تنفيذ شـي معين بواسـطتها ، تقوم الدالة بأخذ قيم وتسـمي (arguments معطيات) كمـدخلات ،، ثم تقوم بعمـل بعض التعديلات على هـذه المـدخلات وتقوم بإخراج قيمـة أخري في أكثر الأحيان تقوم الدالة بأخذ القيم ووضعها في متغيرات أخري تسمي بالــ(parameters) لكي يتم اجراء العمليات عليها داخل الدالة وهذه المتغيرات لاتعمل خارج الدالة أي أنها متغيرات خاصه بالدالة فقط ! ...في دروسـنا السـابقه قمنا باسـتخدام دوال عديده مثل دوال فرز المصـــفوفات ودوال ايجاد نوع البيانات ،،، هذه المرة ســـنقوم ببناء دوالنا الخاصـــة بنا ،، ومن صـــنعنا 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 باســـم الدالة والبارمترات الازمة والتي ســـيتم اجراء العمليات عليها بين قوسين ومن ثم تقوم بكتابة الكود الازم وسط { و }

الصيغه :

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

تقوم بكتابه اســم 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 وسـنقوم بتسـمية الدالة باسـم sumnoraml وهو اسـم من تأليفنا ويدل على وظيفة وهدف الدالة ويمكن أن تقوم بتمسـية الدالة باي اسـم تريده ولسـت مجيراً بكتابه اسـم معين

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

نقوم في هذه الدالة بإجراء عملية إضافة 100 على المتغير أو القيمة التي يتم تمريرها .

Return

يجب أن نضـعها في نهاية كل دالة ، نسـتخدم هذه الكلمة لكي نقوم بإعلام الداله ان وظيفتها انتهت وايضـا نسـتخدمها إذا كان لدينا أكثر من قيمة ونريد أن نقوم بإخبار الــPHP ماهي القيمة التي سيتم اعتمادها ففي مثالنا هذا أردنا إخبار الــPHP 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 لاضـرر من ذلك لأنه لا يوجد لدينا إلا قيمة واحدة لن يتم اعتماد قيمة غيرها ولكن لو افترضـنا 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 .

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 :

```
<?
Print "احمد";
?>
```

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 fares($d)
{
print "alfareees@hotmail.com";
}

fares($d) ;
?>
```

وبين :

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

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

يمكنك أيضا عدم وضع متغيرات في الدالة كالتالى :

```
Html_header ()
{
        Print "<html><head><title>alfareees</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) .

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

اعداد قيمة افتراضيه للدالة
تستطيع أن تجعل الـPHP4 يقوم بإدراج قيمة إفتراضية عند عدم تمرير متغيرات إليه
مثال :

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

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

مدي 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
$r= "salem";
function ala($s)
{
//هذا متغير محلى
$s = "progrramer";
}
echo $r ;
ala($s);
echo $s;
?>
```

مثال :

```
<?
// هذا متغير عام
```

```
$r= “salem”;
function ala($s)
{
//هذا متغير محلى
$s = “progrramer”;
}
echo $r ;
$s=10;
echo $s;
?>
```

في المثال الأول استطعنا طباعة المتغير $r ولم نستطع 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
$r = “salem”;
function ala($s)
{
//هذا متغير محلى
$s = “programmer”;
}
//استطعنا طباعته بشكل مباشر
echo $r ;
ala($s);
// يجب استخدام الداله لكي يتم طباعته
echo ala($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 الطريقتن التاليتين :

الطريقة الأولى :

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

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

الطريقة الثانيه : هي أن نستخدم المصفوفة $GLOBALS التي تستخدم في PHP لتعريف المتغيرات العامة أيضا

مثال :

```
<?
function ala($y)
{
echo $y. “<br>”;
$GLOBALS[“s”] ;
$s = "programmer";
return ;
}
$f =10;
ala($f);
echo $s;
?>
```

المتغيرات المستقره (static variable)

اقصد بالمتغيرات المستقرة هي التي تكون قيمتها ثابتة

مثال :

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

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

المحاضرة التاسعة

تتبع وتصيد ومنع الاخطاء
(avoiding and handling errors)

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 .

أنواع الأخطاء

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

ومثال الأخطاء الإملائية :

```
<?
Eco “1”;
// : من المفترض أن تكتب التالي
Echo “1”;
?>
```

هذا سيعطيك رسالة خطأ Parse error

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

```
<?
Echo “hello”
// : من المفترض أن تكت التالي
Echo “hello”;
?>
```

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

```
<? Php
for ($loop = 0 ; $loop < 5 ; $loop ++ )
{
Echo “”;
?>
```

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 :

```
<?
for ($loop = 0 ; $loop < 5 ; $loop ++ )
{
for ($loop1 = 0 ; $loop1 < 10 ; $loop1 ++ )
{
for ($loop = 0 ; $loop < 5 ; $loop ++ )
{
code ….
}
}
```

في المثال السابق ينقصنا قوس إغلاق التكرار الأخير (})

2 - نسيان الفاصلة المنقوطة . مثال :

```
<?
Echo 10
<?
```

3 - خطأ إملائي في اسم function . مثال :

```
<?
Htmlspecialchar($I);
```

```
?>
```

سيعطيك رسالة خطأ :

```
Fatal error : call to Undefined function : htmlspecialchar().
```

وتصحيحها أن تكون :

```
<?
Htmlspecialchars($I);
?>
```

4 - نسيان إغلاق النص . مثال :

```
<?
Echo "arabbuilder;
?>
```

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

الأخطاء المنطقية (Logical Errors)

إن الأخطاء المنطقية هي الأكثر صعوبة في التتبع 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 :

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

في ملف age.php اكتب الكود التالي :

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

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

Warning : Division by zero in (path) on line (line number)

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 ($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 ;
```

المحاضرة العاشرة

التعامل مع العميل

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 .

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 .

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 الـPHP 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 dir="rtl" align="center">ادخل اسمك الكريم </p>
<form method="POST" action="quiz2.php">
<hr>
<input type="text" name="name" size="20"><br>
<input type="submit" value="إرسال" ></p>
</form>
```

احفظها باسم quiz.php

قم بفتح محرر النصوص واكتب الكود التالي :

```
<html dir ="rtl">
<?
If  (isset($name)) {
Echo "مرحبا بك يا" . $name ;
 Echo '
<br>
<form method="POST" action="quiz3.php" dir="rtl">
<input type="hidden" name = "thename" value = "'.$name.'">"
  من هو أول الخلفاء الراشدين ؟ </p>
  <p dir="rtl"><input type="radio" value="أبوبكر الصديق" name="khlifa">أبوبكر
  الصديق .</p>
  <p dir="rtl"><input type="radio" value="عمر بن الخطاب" checked name="khlifa">عمر
  بن الخطاب</p>
  <p dir="rtl">من هو الفاروق ؟</p>
  <p dir="rtl"><input type="radio" name="faroq" value="عمر بن الخطاب">عمر بن
  الخطاب</p>
  <p dir="rtl"><input type="radio" name="faroq" value="سالم" checked>سالم بن
```

```
  عامر</p>
<input type ="submit" value = "إرسال" dir="rtl">
</form>' ;
}
else
{
echo "غير مصرح لك بدخول هذه الصفحة" ;
}
?>
```

احفظها باسم quiz2.php

قم بفتح المفكرة واكتب الكود التالي :

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

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

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 الـــHTTP 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 الـــــPHP لغة حديثه لعمل ســكريبتات ويب فإنها تأتي بدعم كامل للكوكيز بواســطة الدالة setcookie() باســتثناء 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 بمســح الكوكيز بعد ســاعة من تخزينه فإننا نقوم باســتعمال الدالة time() التي تقوم بحســاب الـــــtimestamp 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 المســتعرض لن يقوم بارســال البيانات إلي الصــفحة 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 بســيط جداً فالمتغير الذي يتم ارســاله يتم تخزينه ضــمن 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 (“ahmed” , “0”, time()-999);
?>
```

وإما القيام بمسح الكوكيز بكتابة اسمه فقط :

```
<?
Setcookie (“ahmed”);
?>
```

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">
<input type="submit" value="تسجيل">
</form>';
echo "قيمه المتغير الذي لديك". " ". $thename ."<br><br>";
echo "قيمة الكوكيز =" . $rname ;
?>
```

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 بيناات 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 !