

الفصل الثاني بروتوكولات الإنترنت

لأنريد 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 باستطاعته إعادة تمهيد طريقه للبيانات إذا حدث خلل في نقطة أو مكان أثناء نقلها ويتم ذلك بسرعة شديدة. عندما يطلب المستخدم من المستعرض أن يجلب له صفحة من الإنترنت 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 عنونها والبروتوكول الذي يقوم بعنونة البيانات يدعي HTTP يقوم هذا البروتوكول بوضع عنونة للبيانات لكي يعرف البروتوكول TCP أين سينقل البيانات (فهو لا يستطيع نقل البيانات إذا لم يكن لها هدف أو مكان) يستخدم البروتوكول HTTP عن طريق الويب في عملية نقل البيانات من كمبيوتر إلى آخر عندما ترى الصفحة متبوعة بـ http:// فإنك تعلم مباشرة أن الإنترنت تستخدم البروتوكول HTTP لإحضار هذه الصفحة يمكنك أن تأخذ صورة بأن الـ TCP عبارة عن ساعي البريد الذي يقوم بإيصال رسالة ، هذه الرسالة فيها طابع بريد وعنوان وهو مانسميه بالـ HTTP.

يتم تمرير الطلب من المستعرض إلى ملقم أو سيرفر الويب وهو ما يعرف بـ HTTP request ويقوم السيرفر برؤية مستودع البيانات لديه لكي يحصل على البيانات المطلوبة فإذا وجد الصفحة في المستودع قام بإرسالها على شكل حزم إلى الجهة التي قامت بالطلب باستخدام بروتوكول TCP ويعنون هذه الحزم لمستعرض الإنترنت لديك باستخدام بروتوكول http (ننبه دائماً إلى 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 فإنه يقوم بإرسال صفحة تحتوي على رسالة خطأ ٤٠٤ وهذه الصفحة التي أرسلت من ملقم الويب إلى المستعرض لديك تسمى HTTP response.

بروتوكول ال HTTP

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

عندما تقوم بعملية طلب لصفحة من السيرفر هناك أمور إضافية ترسل مع عملية الطلب http request غير ال URL وهي ترسل كجزء من http request. نفس الموضوع مع ال http response هناك أمور أخرى تصل معه كجزء منه.

الكثير من هذه المعلومات تولد تلقائياً في رسالة ال HTTP ولا يقوم المستخدم بالتعامل معها مباشرة، إذن لا يحتاج أن تقلق نفسك بشأن هذه المعلومات إذا أنت لم تتشأها في الأصل ويجب أن تأخذ أيضاً في معلوماتك أن هذه المعلومات ترسل كجزء من ال HTTP request وال HTTP response لأن سكربت ال PHP الذي نصنعه يمنحنا تحكماً إضافياً بهذه المعلومات.

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

١ - Request/response line.

٢ - Http header.

٣ - 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 الـ request على الأقل الـ request line (سطر الطلب) والـ HOST.

يرسل مستعرض الإنترنت طلبية (HTTP request) إلى ملقم الويب تحتوي على التالي:

١ - The Request Line

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

أ - أمر HTTP وهو ما يعني بـ method.

ب - المسار من السيرفر إلى المصادر المطلوبة (صفحات الإنترنت) المطلوبة من قبل العميل (المستعرض)

ج - إصدار الـ HTTP.

إذن كمثال على الـ Request Line أنظر إلى السطر التالي:

GET /testpage.htm HTTP/1.1

الـ method يخبر السيرفر كيف يتعامل مع الطلب ، هناك ثلاثة أنواع شائعة من الـ .method

٢- HTTP Header

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

الـ HTTP Header يحتوي على معلومات نستطيع تقسيمها إلى ثلاث فئات وهي:

أ - عامة GENERAL: تحتوي معلومات إما عن العميل أو السيرفر ولا تخصص إلى فرد أو مجموعة.

ب - شخصية Entity: تحتوي على معلومات عن البيانات التي أرسلت بين المتصفح والسيرفر.

ج - 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.arabbuilder.com

Referer: http://www.arabbuilder.com/index.php?something=132

(.....:User -Agent: Iexplorer (win98

مثلا 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 اختياري، يقوم الـ HTTP بالإخبار عن انتهاء معلومات الـ header بترك سطر فارغ (وهذا يكون في الـ HTTP1.1).

٣- The HTTP Body :

إذا تم استخدام الأمر POST في الـ HTTP Request Line عندما يقوم الـ HTTP بطلب المعلومات التي أرسلت في الـ body إلى السيرفر.

Http Response

يرسل من السيرفر إلى المستعرض ويحتوي على ثلاثة أشياء:

١- the Response Line .

٢- http header .

٣- Http Body .

١- The Response Line

الـ response line يحتوي فقط على نوعين من المعلومات:

١- رقم إصدار الـ HTTP .

٢- شيفره أو كود الـ http request التي تقوم بتحديد إذا كان الـ request ناجحاً أم فاشلاً.

مثال:

HTTP/1.1 200 OK

في هذا المثال يقوم الـ response line بإرجاع القيمة ٢٠٠ متبوعة بالكلمة OK هذه تشكل وتشير إلى نجاح الـ request ويكون الـ response يحتوي على

الصفحة المطلوبة والبيانات من السيرفر. ومثال آخر هو الشيفرة ٤٠٤ عندما تقوم بطلب صفحة ويفشل السيرفر في الحصول عليها.

٢ - HTTP Header

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

أ - عامة GENERAL: معلومات عن ال client أو السيرفر ولا تخصص إلى واحد منهما.

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

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

كما قلنا سابقاً، يتكون من عدة سطور ويتم وضع سطر فارغ للإعلام عن انتهاء الهيدر.

مثال:

HTTP/1.1 200 OK -the status line

16:12:23 GMT -general header، Date: Mon; 1st Nov 1999

Server: Apache/1.3.12 (Unix) (SUSE/Linux) PHP/4.0.2 -the response

12:08:03 GMT -Entity Header، 29 Oct 1999.Last-modified: Fri

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 الشيء المطلوب من السيرفر.

٣ - Http Body

إذا تمت معالجة الطلب بنجاح، فإن الـ HTTP response Body يحتوي على كود الـ HTML ويقوم مستعرض الإنترنت بتفسيرها وتحويلها إلى الصفحة النهائية التي تراها.

أين سكربت الـ PHP من ذلك كله ؟

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

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

تجعلنا الـ html نضمن سكربتات الـ php فيها ضمن قواعد لذلك لكي نستطيع تشغيلها ولكننا لانس أن امتداد الملفات يظل كما هو الـ php أو الـ php3 بدون تغيير فيه لكي يتم إرسال السكربت إلى مكتبة الترجمة (scripting engine) التي تقوم بترجمة السكربت إلى الـ html (كأنك تترجم من عربي إلى إنجليزي أو العكس)

مفهوم ال parsing و ال Execution:

ممکن أن نقسم عملية الترجمة الذي يقوم بها سيرفر php إلى قسمين أو عمليتين: العملية الأولى: هي أن السيرفر يقوم أولاً بفحص قواعد اللغة وهذا لا يضمن أن السكريبت صحيح مئة بالمئة ولكنه تدقيق في الأوامر وقواعد اللغة وهذا ما يسمونه بالParsing

العملية الثانية: هي تنفيذ السكريبت بعدها وإخراجه على شكل كود html وهذا ما يسمي بال Execution.

بقي أن نقول أمراً معروفاً وهو أن السكريبتات نوعين:

١ - وهو ما ينفذ من جهة المزود

Server –Side scripting

٢ - ما ينفذ من جهة المستعرض (صفحة انترنت).

التعليقات

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

يمكنك عمل تعليق من سطر واحد كالتالي:

<?

//هذا تعليق لفائدة له وليس له أي معنى

?>

مثال آخر:

```
<?  
//هذه الدالة تقوم بطباعة الكلمة تعليق  
"Echo تعليق";  
>
```

وأيضاً يمكنك استخدام تعليق من أكثر من سطر كالتالي:

```
<?  
/*تعليق يتكون من  
أكثر من سطر بعلامة السلاش والنجمة  
*/  
>
```

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 الـ PHP، المتغيرات تبدأ بعلامة الـ \$ ولكي تقوم بإدخال قيمة في المتغير فإنك تستخدم العامل (=) إذن لكي تقوم بإنشاء متغير يحتوي على قيمة يمكنك القيام بذلك كالتالي:

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

لاحظ أن السطر السابق يتكون من خمسة أشياء:

١ / المتغير وهو alfares

٢ / وقبله علامة الـ \$ لكي يعرف مترجم الـ PHP أنه متغير

٣ / المعامل (=)

٤ / الفاصلة المنقوطة (;)

٥ / القيمة وهي `How Are You Every Body?` وهي القيمة الموجودة في المتغير أو التي اقترحناها للمتغير أو التي وضعناها فيه (لأن الذي اقترح القيمة هو أنت (مبرمج الphp))

ملاحظات:

- أسماء المتغيرات حساسة لحالة الأحرف إذا كانت كبيرة وصغيرة

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

المتغيرين الذين بالأعلى مختلفين بسبب حالة الأحرف.

٢ - يمكنك استخدام المعامل (_)

`$First_name`

٣ - يمكنك استخدام ألف حرف في تسميه المتغيرات (وفي الواقع هي غير محددة).

علامات التنصيص

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

هناك أنواع للبيانات وهي:

١ - strings (حروف)

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

٢ - Integer (أرقام)

```
$Exam = 5;
```

٣ - Double (أرقام ذات فواصل)

```
$num = 5.4
```

٤ - array

يأتي تفصيلها فيما بعد

٥ - objects

تفصيلها في فصول أخرى

٦ - Unknown.

يأتي تفصيلها في فصل آخر.

المتغيرات لا يتم تعريف نوعها من قبل المبرمج إنما مترجم الـ PHP يقوم بالتعرف عليها لكي يتم إتمام العمليات المختلفة عليها.

البيانات الحرفية /

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

أمثلة:

“هذا النص بين علامتي تنصيص عادية أو مزدوجة”

‘هذا النص بين علامتي تنصيص مفردة أو وحيدة’

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

```
<?  
$d="غلط"  
echo "خطأ"  
>
```

لا يمكنك أيضاً أن تقوم بوضع علامة تنصيص من نفس النوع الذي تستخدمه القيمة
الحرفية في وسط العبارة الحرفية أو النص.

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

وتصحيحه

```
<?  
$variable = "صحيح هذا النص";  
>
```

وأيضاً مثال آخر

```
<?  
$r = "This is"BAD"; // خطأ  
$t = "This is 'good"; // صحيح  
?>
```

أما إذا كنت مصرّاً على ذلك أو تحتاج إليها في عمليات ضرورية (كما سوف نرى 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";  
?>
```

حسناً ما رأيك لو أردنا أن نطبع المعامل (\) بنفسه؟
الحل هو أن نتبعه بمثله، وبالمثال يتضح المقال:

```
$file = "c:\windows\system.ini";  
c:windowssystem.inineتجئةecho $file; //  
  
$file = "c:\\windows\\system.ini";  
c:\windows\system.inineتجئةecho $file; //
```

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

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

وأيضاً يمكننا أن نضيف إلى متغير قيمة متغير آخر:

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

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

الأرقام

العدد الفردي والمزدوج

الاختلاف المعروف لدي أنا حتى الآن هو أن الفرق بينهما هو الفاصلة العائمة (والله حتى إعطاؤها هذا الاسم يجعلنا نشعر بالإحباط والخوف)
لاحظ أننا لا نستخدم علامات التنصيص وذلك ليعرف الـ PHP 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 الحسابية

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

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

مثال آخر:

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

مثال لعملية حسابية نستخدم فيها متغير حريفي

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

مثال لعملية أخرى لكنها لم تعمل عليك استتباط السبب بنفسك:

```
<?  
$W="L10";  
$E= 2;  
$F = $W * $E;  
echo $W.' '.$E.' '.$F;  
?>
```

يمكننا إضافة رقم واحد إلى متغير بثلاث طرق متنوعة:

مثال

```
$j++
```

أو

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

أو

```
$j += 1
```

ويمكننا على ذلك إضافة 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 إنها قيم ثابتة لا تتغير ونعرفها عن طريق الدالة

`define`

الثوابت حساسة أيضاً لحالة الحروف

```
<?
"alfarees"); Define ("author"
Echo "author is ". author ;
?>
```

هناك ثوابت يستخدمها النظام مثل:

`PHP_OS`

التي تقوم بعرض نظام التشغيل الذي يستخدمه السيرفر

مثال:

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

معرفة وتحويل أنواع 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>";
"string");, settype ($n
echo "After That is go ". gettype ($n);
?>
```

الدالة `isset`

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

مثال:

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

الدالة unset

تقوم بحذف المتغير إذا كان موجوداً وتحرير الذاكرة منه (لذلك تأكد جيداً قبل استخدام هذه الدالة من إعطاء دموعة 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 ("") ولا تقوم بإرجاع أي شيء إذا كان المتغير منشأ وفيه قيم غير المذكورة.

داوال الوقت والتاريخ

نستطيع إيجاد الوقت والتاريخ عن طريق دوال في الـ PHP من تلك الدوال الدالة gmdate ()

مثال:

```
<?
Echo gmdate (m);
Echo gmdate (M);
?>
```

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

تحتجز الـ php بكثير من الدوال والكلمات المحجوزة التي تقوم بعمليات مختلفة مثل العمليات الحسابية المعقدة والقيام بإيجاد الوقت والتاريخ وإرسال الرسائل البريدية وإيقاف السكريبتات 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 أيضاً 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:

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

```

النماذج

النماذج في الويب أو صفحات الإنترنت 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 إنك مثلاً أردت شراء كتاب من الإنترنت 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 باختيار الكتاب الذي تريد وتكتب اسمك ورقم هاتفك وصندوق بريدك (ربما) في فراغات أو عن طريق الإشارة إلى مجموعة من الخيارات. يتم تخزين هذه القيم في المتغيرات التي يتم كتابتها في الخاصية name (نتكلم عنها في هذا الفصل) ويتم إرسالها عند ضغط زر - إرسال البيانات - (submit) إلى (البرنامج) الصفحة التي سوف تقوم بمعالجة هذه البيانات (والتي يتم تحديدها في الخاصية ACTION) وإجراء العمليات عليها مثل تخزينها مثلاً في قاعدة البيانات أو إرسالها إلى البريد الإلكتروني وذلك عن طريق php.

ماذا يعمل العميل في النماذج ؟

إنه باختصار يقوم بتعبئة مربعات نصوص (textBox) ويقوم بوضع علامة صح في مربعات الاختيار (check boxes) أو يقوم بالتصويت أحياناً لشئ معين فيختر زر اختيار (أزرار الراديو).

هذه الأشياء كلها يتم إنشاؤها بواسطة الhtml وفصلنا يناقش كيفية إنشاؤها وكيفية التعامل والحصول على البيانات منها ، بقي علينا كبدائية أن نعرف أن هذه الأدوات تنشأ في الواقع بين وسمين من وسوم لغة الhtml وهي الوسمين

<form> و </form>

خصائص النماذج

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

ACTION

وظيفة هذه الخاصية أن تخبر السيرفر مكان الصفحة التي يقوم بإرسال معلومات النموذج إليها أو عنوانها أيضاً كان نوعها، وطبعاً في حالتنا ستكون الصفحة الثانية هي الصفحة التي تحتوي على سكربت الـ.php. ليس مهماً أن تكون الصفحة php فقد تكون html ولكنها تحتوي على كود يختص بالتعامل مع برنامج تفاعلي لصفحات الويب مثل الجافا. ولا نريد أن نخرج عن نطاق الموضوع فدعنا نعطي مثالاً على هذه الخاصية:

```
<FORM ACTION ="TEST.PHP">  
.....  
</FORM>
```

METHOD

هذه الخاصية تقوم بإخبار النموذج طريقة إرسال المعلومات إلى الصفحة الهدف وفي الواقع هناك طريقتين مشهورتين ومعروفتين لإرسال المعلومات هما GET و POST.

```
<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 لا
تستخدم إلا بشكل نادر.

دعنا الآن نفصل هاتين الطريقتين بشكل أوسع:

GET

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

- كتابه عنوان الصفحة المصدر.
- اتباعها بعلامة استفهام.
- كتابة العناوين والقيم.

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

قد تكون النقطيتين الأخيرتين غير مفهومتين بشكل جيد بسبب أنك لم تتعامل مع
النماذج من قبل.

لكن الحقيقة أن النموذج يتكون من عناصر (مربع علامة، مربع نص، زر اختيار)
ولكل من هذه العناصر عنوان خاص بها (name) ولكل منها قيمة خاصة بها
(value).

وهي مشابهة للمتغيرات ويمكن أن يحتوي عنوان الصفحة على أكثر من عنوان
(name) وأكثر 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) نتيجة الاستعلام الحرفية.

العنوان دائماً يكون باللغة الإنجليزية (name) ونعامله كأنه اسم متغير من المفترض تعريفه في الصفحة الهدف (التي سنكتبها بال-PHP).

قد تحتوي القيم على فراغات أو معاملات مثل (+, -, \, #, %). يقوم المتصفح باستخدام لغة تشفير الصفحات URL ENCODING. أيضاً يستخدم الـ URL ENCODING مع الأحرف العربية أو اللغات الأخرى غير الإنجليزية في كتابة الحرف.

URL Encoding

هناك بعض الأحرف لا يستطيع المتصفح إضافتها لعنوان الصفحة بصيغتها الحقيقية بل يستخدم لغة التشفير في التعريف عنها وهذه جداول بالرموز الذي يستخدم المتصفح كود بدلاً 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 بالعملية كلها بدلاً عنك.

في الواقع وظيفتها هي نفس وظيفة الـ get ولكنها لا ترسل المعلومات في عنوان صفحة الإنترنت بل تقوم وضعها في الـ body التابع لـ http response. بالإضافة إلى أنه يستطيع إرسال البيانات بكمية أكبر من الـ GET.

أيهما تستخدم GET أم POST ؟

قد يكون العيب في الخاصية GET عدم سرية المعلومات التي تقوم بكتابتها ومن الممكن أن تظهر للشخص الذي يجلس إلى جوارك... خاصة عندما تريد الحفاظ على سرية معلوماتك.

أضف إلى ذلك أنها غير مفيدة في النصوص الكبيرة الحجم. ولكنها مفيدة في أشياء كثيرة فمثلاً محرركات البحث يجب أن تستخدم هذه الخاصية لكي يستطيع المستخدم أن يستخدم عنوان البحث ويحتفظ به لوقت آخر ولا يقوم من جديد بكتابة الكلمة التي يبحث عنها.

أيضاً الـ POST مفيدة في إخفاء المعلومات واحتواء كميات كبيرة من البيانات ولكن لا يمكن الاحتفاظ بعنوان الصفحة.... مع ذلك فإنها أيضاً ليست جيدة في الحماية بحيث أن أي هاكر خبير يمكنه الحصول على المعلومات إذا لم يكن لها تشفير معين في نقلها.. لكن إذا أردت فعلاً أن تجعلها محمية فيجب عليك استخدام اتصال محمي إلى سيرفر محمي.... أو مايسمونه (SCURE CONNECTION TO SCURE SERVER).

أدوات التحكم في النماذج:

في الواقع إن أدوات التحكم عبارة عن مربعات النصوص العادية (التي يدخل فيها المستخدم اسمه وعنوانه) وأزرار الراديو (والتي يقوم المستخدم فيها باختيار شيء معين مثل الوجبة المفضلة لديه أو المشروب المفضل) ومربعات الاختيار (التي تتيح للمستخدم أن يختار ما يشتهي ويحب من الخيارات المعروضة) وأيضاً القوائم التي تساعدك على اختيار أكثر من شيء أو شيء واحد.

في أغلب هذه الأشياء يتم استعمال الوسم

<INPUT>

وتلخيص تفصيله كالتالي:

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

الشرح:

١ - TYPE= type

نحدد نوع الكائن إذا كان زر راديو أو مربع نص عادي أو مربعات الاختيار.

٢ - NAME= name

تقوم فيها بإعطاء اسم لمتغير يتم حفظ القيمة فيه.

٣ - VALUE= value

ستتضح وظيفته أكثر عندما ندرج عليه أمثلة إذ إن عمله يختلف من أداة إلى أخرى.

تطبيقات عملية

سنقوم في هذه التطبيقات بصنع برامج بسيطة تتكون من ملفين، الملف الأول يحتوي على كود HTML يقوم بتكوين النموذج والملف الثاني يقوم باستقبال النتائج وطباعتها.

مربعات النصوص [TEXT Box]:

نقوم بعمل ذلك كالتالي:

١ - قم بتشغيل محرر النصوص لديك.

٢ - اكتب الكود التالي:

```
<html dir = "rtl" >
<FORM METHOD = "GET" ACTION = "textbox.php" >
ماهي وجبتك المفضلة في الصباح؟

<br >
" > <INPUT TYPE = "text" NAME = "food" value = "
<br >
" > <INPUT TYPE = submit VALUE = "إرسال"
" > <INPUT TYPE = reset VALUE = "مسح"
</form >
</html >
```

٣ - قم بحفظ الملف كصفحة HTML. وسميته (textbox.html).

٤ - افتح محرر النصوص إذا كنت أغلقته.

٥ - اكتب الكود التالي:

```
<?
" Echo وجبتك المفضلة والتي تموت في حبها هي $food ; " . " .
?>
```

٦ - قم بحفظ الملف كـ php. وسميته textbox.php.

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

٨-قم بتشغيل السيرفر واكتب في مستعرض الإنترنت لديك

<http://localhost/textbox.html>

٩ - قم بكتابة وجبتك المفضلة واضغط زر إرسال.

١٠ - 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 و زر يقوم بعملية إرسال البيانات

قمنا بصناعة بداية النموذج بواسطة الوسم <FORM> و قمنا بتحديد المكان الذي سيتم إرسال البيانات إليه بواسطة = "textbox.php" ACTION

و قمنا بصنع مربع النص بواسطة الوسم INPUT واخترنا الـ TYPE="text"

كما قمنا بوضع القيمة الافتراضية فيه بواسطة القيمة " Value= "جبنه ومربى"
و قمنا بوضع الناتج الذي يضعه المستخدم في مربع النص في المتغير food.

(لاحظ أن تسمية المتغيرات حساسة لحالة الأحرف في PHP وأننا لم نقم بوضع \$ في صفحة المتغير في كود الـ html).

وأيضاً لقد قمنا بإضافة زر بواسطة TYPE=SUBMIT

و قمنا بوضع كلمة على الزر وهي كلمة (إرسال) " VALUE=إرسال"

أيضاً قمنا بصنع زر آخر Type =reset

و قمنا بجعل العبارة التي عليه (مسح) " Value=مسح"

هناك نوعين من الأزرار هي SUBMIT و RESET

١- ال submit يقوم بإرسال المعلومات.

٢- ال reset يقوم بمسح البيانات في جميع الأدوات في النموذج لإعادة إدخالها من جديد.

بعدما قمنا بإدخال البيانات وضغط زر الإرسال قام النموذج بإرسال البيانات إلى الصفحة المحددة في الخاصية ACTION وقامت الصفحة المحددة باستقبال النتائج الموجودة في النموذج وهي نتيجة واحدة في مربع نصوص تم حفظ قيمته في المتغير .food

وقامت بطباعتها بواسطة الدالة echo.

نظراً لأننا استخدمنا الأسلوب GET فقد تم إعطاؤنا عنوان الصفحة بالإضافة إلى (?) وأيضاً المعلومات المسجلة في المتغيرات والتي تم استخدام ال URL ENCODING فيها لأنها تستخدم حروف عربية.

مربعات النصوص الكبيرة (text area) طلبات أكبر للطعام الشهية !
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 الأسطر.

تستخدم هذه الأداة وسم فتح ووسم إغلاق

<TEXTAREA>

</TEXTAREA>

ويمكنك تحديد حجمها بواسطة تحديد الصفوف بالخاصية ROWS والأعمدة بالخاصية COLS.

تمرين عملي

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

```
<html dir="rtl" >
METHOD="POST"> <FORM ACTION = "TAREA.PHP"
ما هي وجبتك المفضلة ؟
<br>
<TEXTAREA NAME = "food" ROWS="10" COLS = "50" >
جينة
مرى
معكرونة
بيف برغر
</TEXTAREA>
<br>
" > <INPUT TYPE = SUBMIT VALUE = "
</FORM>
</html>
```

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

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

- قم بحفظ الملف باسم tarea.php
- قم بوضعهما في مجلد السيرفر لديك.

- قم بتشغيل البرنامج.

<http://localhost/tarea.html>

- قم بضغط الزر لإرسال البيانات.

١٠- شاهد النتيجة.

الشرح

لا نضيف شيئاً على قولنا هنا سوى أننا نريدك أن تلاحظ كيف جهزنا القيمة الافتراضية بكتابة نصوص بين وسومات الـ `textarea` وأيضاً أننا استخدمنا الأسلوب `POST` في إرسال البيانات مما جعلها لا تظهر في شريط العنوان. وأن الـ `NAME` تحدد اسم المتغير التي ستذهب إليه القيمة واسم المتغير في الكود لا يحتوي على `$` لأنه كود `HTML` وليس `PHP`.

مربعات الاختيار (Check Box) أكثر من خيار في وقت واحد !

في الواقع قد نرى مربعات الاختيار في صفحات الويب عندما نريد الاشتراك في موقع معين لرؤية محتوياته أو عندما نريد تسجيل بريد إلكتروني أو حجز مساحة عند موقع.

وفائدتها هي إتاحة فرصة للمستخدم لتحديد أنواع الأشياء التي يريد أن يشترك فيها مثلاً أو إتاحة فرصة له لقبول إتفاقية أو غير ذلك أو رفض الجميع أو قبول الجميع.

يمكننا صنع مربع العلامة بواسطة الوسم `INPUT`

```
" checked" <INPUT TYPE = "CHECKBOX" NAME = "swalif" value = "
```

نقوم بتحديد نوع الأداة بأنها مربع علامة في هذا الجزء

```
TYPE = "CHECKBOX"
```

نقوم بتحديد اسم المتغير في هذا الجزء

NAME = "swalif"

ونقوم بتحديد القيمة التي يتم وضعها في المتغير إذا قام المستخدم باختيار مربع

العلامة في هذا الجزء:

value="value" محادثة "

إذا لم تقم بوضع الخيار value فستكون القيمة الافتراضية هي on عند اختيار

المستخدم مربع العلامة وستكون فراغ إذا لم يقم المستخدم باختيار المربع.

ونقوم بوضع القيمة الافتراضية بإضافة الكلمة checked فإذا تم وضع هذه

الكلمة يكون مربع العلامة مختار تلقائياً أما إذا لم نكتبها فسيكون بدون علامة

الاختيار.

Checked

تطبيق عملي (١):

- قم بفتح المفكرة و قم بكتابة الكود التالي:

```
<html dir="rtl">
<FORM ACTION="CHECK.PHP" METHOD = "POST">
ما الذي تريد أن تفعله في الحياة ؟ (يمكنك اختيار أكثر من إجابة)
<br>
<INPUT TYPE="CHECKBOX" NAME = "WIFE" CHECKED>
الذي أريد أن أفعله في الحياة هو أن أعمل.
<br>
"><input type= submit value = "
</FORM>
</html>
```

- قم بحفظ الملف باسم check.html.

- قم بفتح ملف جديد في المفكرة و قم بكتابة التالي:

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

- قم بحفظ الملف باسم .check.php

- قم بنقل الملفين إلى مجلد السيرفر.

- اكتب في المتصفح

<http://localhost/check.html>

- النتيجة.

تطبيق عملي (٢):

- افتح المفكرة واكتب الكود التالي وقم بحفظه في ملف جديد باسم

check2.html

```
<html dir="rtl">
<FORM ACTION="CHECK2.PHP" METHOD = "POST">
ما الذي تريد أن تفعله في الحياة ؟ (يمكنك اختيار أكثر من إجابة)
<br>
" CHECKED>عمل< INPUT TYPE="CHECKBOX" NAME = "WIFE" value= "
الذي أريد أن أفعله في الحياة العمل.
<br>
" >أتعلم< INPUT TYPE="CHECKBOX" NAME = "jihad" value= "
أن أتعلم أكثر
<br>
" CHECKED>أولادي< INPUT TYPE="CHECKBOX" NAME = "qran" value= "
أن أربي أولادي بشكل صالح
<br>
">إرسال<input type= submit value = "
</FORM>
</html>
```

- قم بفتح ملف جديد و قم بوضع الكود التالي فيه :

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

- قم بحفظه باسم check2.php

- قم بتشغيل الملف.

- النتيجة.

تطبيق عملي (٣)

- افتح محرر النصوص واكتب الكود التالي :

```
<html dir="rtl">
<FORM ACTION="CHECK3.PHP" METHOD = "POST">
ما الذي تريد أن تفعله في الحياة ؟ (يمكنك اختيار أكثر من إجابة)
<br>
" CHECKED>عمل<INPUT TYPE="CHECKBOX" NAME = "alswalif[]" value= "
الذي أريد أن أفعله في الحياة هو أن أعمل
<br>
" >أتعلم<INPUT TYPE="CHECKBOX" NAME = "alswalif[]" value= "
أن أتعلم أكثر
<br>
" CHECKED>أولادي<INPUT TYPE="CHECKBOX" NAME = "alswalif[]" value= "
أن أربي أولادي بشكل صالح
<br>
">إرسال<input type= submit value = "
</FORM>
</html>
```

- 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 "$salswalif[2] <br>" ;  
?>  
</html>
```

- قم بحفظه باسم `check3.php` و قم بنقلهما إلى ملف السيرفر.

- قم بتشغيل البرنامج

<http://localhost/check.html>

- قم بضغط زر إرسال وانظر للنتيجة

الشرح

في الواقع لقد قمنا بتطبيق ثلاث تمارين التمرين الأول 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 بإضافة قيم يتم وضعها في المتغيرات عند اختيار المستخدم لها كما في التمرين الثاني وأردنا لفت النظر في التمرين إلى شيء يسمى بالمصفوفات فإذا أردنا مثلاً أن نجعل اسم المتغير متشابهاً وإجراء عمليات تكون أسرع عليه نستخدم المصفوفات ولن نتطرق إلى المصفوفات حالياً ولكن أردنا لفت نظرك فقط وسنقوم بالتكلم عن المصفوفات بالتفصيل لاحقاً هي والتكرارات بعد التكلم عن العبارات الشرطية في الـ PHP.

أزرار الراديو [RADIO BUTTONS] [اختر المشروب المفضل !]

ماهو اختيارك المفضل ؟ علماً بأنه لا يمكنك اختيار أكثر من خيار واحد !!

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

يتم استخدام أزرار الراديو باستخدام العبارة `<INPUT >` كالتالي:

```
<INPUT TYPE = "radio" NAME = "name" value= "value" checked>
```

نقوم بتحديد نوع الكائن بأنه 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 أزرار 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:

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

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

</form>
</html>
```

- قم بفتح محرر النصوص واكتب الكود التالي وقم بحفظه باسم radio.php

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

٣ - قم باختيار المشروب المفضل واختر إرسال.

الشرح:

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

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

وأيضاً لاحظ أننا استخدمنا متغيراً واحداً فقط لجميع الاختيارات بحيث أن جميع الأزرار قيمتها تعود إلى هذا المتغير.

القوائم [Lists Or drop down menus] اختر مواصفات

زوجك للمستقبل واسمها:

تستخدم القوائم في الـ html بشكل مختلف قليلاً عن الأدوات السابقة إذ إننا نستخدم وسمين من وسوم لغة html وهما: <select> لنقوم بإنشاء القائمة و <OPTION> ونستخدم الخاصية MULTIPLE إذا كنا نريد إتاحة الفرصة للمستخدم أن يختار أكثر من قيمة ونقوم بوضع القيمة التي يختارها المستخدم في متغير بواسطة الخاصية NAME أو في مصفوفة متغيرات (وسيتضح مفهوم المصفوفات لديك جيداً في فصل المصفوفات بإذن الله).

تطبيق عملي:

- 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 أن يكون اسم زوجة المستقبل(لغير المتزوجين) ؟
<br>
<select name = "wife" >
</option>هنا<option>
</option>جمانة<option>
</option>رزان<option>
</option>سحر<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>شقرء<option>
</option>جعداء الشعر<option>
</option>سوداء<option>
</option>سمرء<option>
</option>بيضاء<option>
</select>
<br>
">إرسال<INPUT TYPE=SUBMIT VALUE="
</html>

```

- 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 ما تريد ثم اضغط زر إرسال

الشرح:

لقد قمنا بصناعة قائمة تسمح باختيار قيمة واحدة منها ثم تذهب هذه القيمة إلى المتغير wife وصنعنا قائمة ثانية تسمح باختيار أكثر من عنصر واحد وقمنا بوضع هذه القيم في مصفوفة متغيرات (سيوضح معني المصفوفات في فصول قادمة إن شاء الله).

الأداة الخفية [والمعلومات السرية!] [hidden control]

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

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

هذه الأداة تلعب دوراً مختلفاً ومتميزاً عن بقية الأدوات وهي إخفاء المعلومات التي تم إدخالها كما شرحنا في السابق وهي مفيدة جداً مع النماذج المصنوعة بواسطة الـ PHP إذ إنها تسمح لنا أيضاً بأن تكون المعلومات المخفية هي متغيرات الـ PHP.

يتم صنع هذه الحقول المخفية كالتالي:

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

نقوم بوضع HIDDEN لكي يعرف المتصفح أن هذه المعلومات خفية (لا تظهر للمستخدم) ونضع اسماً للمتغير الذي يقوم بالاحتفاظ بالمعلومات والذي يتخزن اسمه في الـ NAME ونقوم بوضع المعلومات التي نريد إخفاءها في الـ VALUE.

نستطيع الاستفادة أيضاً منها عن طريق الـ php وذلك عن طريق كتابة كود الـ HTML بواسطة الأمر echo() في الـ PHP كما في المثال التالي:

```
<?
$msg1 = "هذه العبارة لن تظهر ";
Echo "<form>";
Echo "<input type=hidden name =secret value= '$msg1'>";
Echo "<input type=submit>";
Echo "</form>";
?>
```

هذا الكود الذي تراه عبارة عن كود HTML تمت كتابته بالـ PHP عن طريق الأمر echo() ولقد استطعنا تخزين قيمة متغير php (\$msg) في متغير html (secret).

تطبيق عملي:

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

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

```

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

```

<html dir="rtl">
<head></head>
<body>
<?
<br>">Echo لقد قمنا بعرض السيارات التالية عليك: "
Echo "$hid1<br>";
Echo "$hid2<br>";
Echo "$hid3<br>";
<br>">Echo "ولقد قمت باختيار: "<br>
Echo $favcar;
?>
</body>
</html>

```

٣- قم بنقل الملفين إلى مجلد السيرفر ثم قم بتشغيل السكريبت:

<http://localhost/hid.php>

الشرح:

لقد قمنا بعمل نموذج بسكريبت الـ php لاحظ أننا استخدمنا الـ (') بدلاً من (") كما كنا نعمل في الـ html وذلك لأننا قلنا سابقاً أن القيم الحرفية (راجع فصل المتغيرات) ولقد قمنا بإدراج قيم متغيرات الـ php في كود الـ html مما يوفر علينا الكثير من إعادة الكتابة (في حال كان النص المستخدم طويلاً).
اقرأ المثال أكثر من مرّة وسيوضح لك المقال أكثر باذن الله.

استخدام حقول 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 فإن هذه الطريقة جيدة قليلاً في أن لا يرى أي شخص معلومات الآخر السرية. في الواقع مع ذلك فإنك لا تكون قد أدت حماية إذا كان الأسلوب المستخدم في إرسال بيانات المستخدم هو الأسلوب get إلا إذا كنت تستخدم تشفير البيانات ويكون أكثر جودة إذا استخدمت الأسلوب post وأيضاً لن يكون محمياً من الهاكر إذا لم تكن تستخدم SSL (Secure Socket Layer) لكي تقوم بتشفير البيانات.

تطبيق عملي

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

```
<html dir="rtl">
<body>
<form method=post action="pass1.php">
اسم المستخدم
<br>
>"user"<input type="text" name =
<br>
كلمة المرور
>"pass"<input type="password" name =
<br>
'><input type = submit value='
</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 إلى مجلد السيرفر لديك.
قم بتشغيل البرنامج ولاحظ النتيجة.

إرسال البريد الإلكتروني بواسطة الphp:

البريد الإلكتروني هو الحياة التي تنبض بها السكريبتات فمثلاً هناك سكريبتات إرسال بريد إلى صاحب الموقع تخبره بشي معين أو ملحوظة أو غير ذلك ويمكن استخدامها في أكثر من مجال.

والدالة التي تستخدم في ذلك هي الدالة mail()

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

وتقوم بوضع بريد الذي ستصله الرسالة في الخانة \$to وموضوع الرسالة في الخانة \$sub والرسالة في الخانة \$msg وبيدك 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>
الرسالة
<textarea rows=10 cols=20 name = "msg" >
</textarea>
">إرسال البريد الإلكتروني<input type="submit" value = "
</form>
</body>
</html>

```

قم بإنشاء ملف آخر وسم بكتابة الكود التالي وسم بإعطائه الاسم mail.php.

```

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

```

قم بوضع الملفين في مجلد السيرفر وسم بتشغيل البرنامج واملأ البيانات واضغط زر الإرسال وسترى ان الرسالة تم إرسالها بنجاح.

برامج عملية

برنامج أو سكريبت إرسال بطاقات بسيط يحتوي على ملفين الملف الأول فيه البطاقات وعنوان البريد الإلكتروني والملف الثاني هو الذي يقوم بعملية الإرسال

الملف الأول هو `chcard.php` وكوده كالتالي:

```
<html dir="rtl">
<form action =card.php method = "post">
اختر البطاقة التي تريد إرسالها
  <br>
  <br>
  <INPUT TYPE = "radio" NAME = "card" value= "
http://www.khalaad.f2s.com/MADINA9_small.JPG" checked>
البطاقة الأولى
  <br>
  <br>
  
  <br>
  <br>
  <INPUT TYPE = "radio" NAME = "card" value= "
http://www.khalaad.f2s.com/Haram3.jpg" >
البطاقة الثانية
  <br>
  
  <br>
اسمك
  <br>
  <input type="text" name = "myname">
  <br>
بريدك الإلكتروني
  <br>
  <input type="text" name = "you">
  <br>
بريد صديقك
  <br>
  <input type="text" name = "to">
```

```

<br>
موضوع التهنئة
<input type="text" name = "sub">
<br>
الرسالة
<br><br>
<textarea rows=10 cols=20 name = "msg" >
</textarea>
<br>
" >إرسال البطاقة<INPUT TYPE = submit value= "
</form>
</html>

```

الملف الثاني يقوم بعملية إرسال البطاقة وقم بكتابة الكود التالي وحفظه في ملف باسم card.php

```

<?
$message = " $myname بإرسال بطاقة إليك " . "\n" . " وهو يقول في نص رسالته
لك: ". "\n" . "$card" . "\n"; وتجدها على الرابط التالي:
"From:$you"); ، "$message" ، "$sub" ، mail("$to"
<center> "echo مبارك، ، لقد تم إرسال الرسالة بنجاح;
?>

```

ملاحظة:

الدالة \n تقوم فقط ببدء سطر جديد لأننا لا نستطيع استخدام
 في نص الرسالة.

obbeikandi.com