

<?php (the_AMAZON_#1_bestseller); ?>

ALL SOURCE CODE INCLUDED FREE

THE JOY of PHP

A Beginner's Guide to Programming Interactive
Web Applications with PHP and mySQL



<?php (Alan_Forges); ?>

جدول المحتويات

| | |
|-------|--------------------------|
| _____ | مقدمة |
| _____ | مقدمة |
| _____ | ما هي لغة PHP؟ |
| _____ | تقديم PHP |
| _____ | مثال |
| _____ | قليلا من التاريخ |
| _____ | تثبيت وتكوين PHP |
| _____ | مقدمة |
| _____ | مستخدمو Windows |
| _____ | باستخدام Wampserver |
| _____ | مستخدمو Mac |
| _____ | كيف أعرف أنها تعمل؟ |
| _____ | Oracle VM Virtual Box |
| _____ | الثبيت لموقع ويب |
| _____ | يمارس |
| _____ | مقدمة إلى HTML |
| _____ | مقدمة |
| _____ | HTML الأساسية |
| _____ | العناصر الأساسية لـ HTML |
| _____ | العلامات |
| _____ | العلامات المتداخلة |
| _____ | العلامات المطلوبة |
| _____ | نوع الوثيقة |
| _____ | رأس |
| _____ | العلامات الاختيارية |
| _____ | العلامات الفوقيّة |
| _____ | علامات مفيدة |
| _____ | الرؤوس |
| _____ | DIV |
| _____ | الصور / الصورة |

الروابط

القواعد

يمارس

جداول HTML

رؤوس جدول HTML

نماذج HTML

علامة الإدخال

أساليب وإجراءات نموذج HTML

مقدمة - بناء جملة PHP الأساسية أول صفحتك التي تم تمكن

يها - أهلاً بالعالم!

بعض المرح على الفور

عداد العد التنازلي

يمارس

المحررين والبقاء منظمين

المحررين

Microsoft Expression Web

PhpDesigner

صاروخ

المحررين الحرة

يشمل

مثال أساسى

مثال HTML

مثال رمز

أفضل الممارسات

المتغيرات والأرقام والتواريخ والسلسل

المتغيرات

اصطلاحات التسمية المتغيرة وأفضل الممارسات

أعداد

الحساب الأساسي

اختصارات حسابية شائعة

وظائف عددية مفيدة

سلسل

وظائف سلسلة مفيدة

مصادر التوثيق

يمارس

كيفية تفسير وثائق PHP.NET

قيمة الإرجاع

اسم وظيفة

حدود

بلغ

مثال: حساب العمر بالسنوات

نطاق متغير

جمل التحكم

مقدمة

لو

إذا كان غير ذلك

تمرين: ساعات مفتوحة

مفتاح PHP

حلقات PHP

بينما PHP

J PHP

كيفية استخدام قاعدة بيانات ، مثل mySQL

مقدمة

ما هي قواعد البيانات؟

الشروع في العمل مع phpMyAdmin ما هو؟

استخدام phpMyAdmin لإنشاء قاعدة بيانات

مقدمة في SQL

استخدام phpMyAdmin لإنشاء جدول في قاعدة بيانات

تحديد جدولنا الأول

ما الذي يميز السيارة؟

أنواع البيانات

أعداد

الشخصيات

بلغ

تمرين: قم بإنشاء جدول

العمل مع جمل SQL

أدخل البيانات

حدد البيانات

بيانات أين

عوامل المقارنة

ترتيب حسب

تحديث البيانات

حذف البيانات

استخدام MySQL و PHP معاً

مقدمة

شفرة!

قائمة الكود: createdb.php Code

HTML مهلاً، أين؟ createdb.php

إنشاء نماذج لعرض البيانات وإضافتها وتحريرها وحذفها

مقدمة

النماذج التي تضيف بيانات إلى قاعدة بيانات

نموذج أساسى

كود HTML

نموذج العمل

كود PHP

SQL A Brief Time Out... تشمل الملفات وإدخال

تضمين الملفات

حقن SQL

النماذج التي تعرض بيانات الملخص

شفرة

شرح الكود

تمرين: التغيير والتبديل في SELECT

تحسين مظهر الجدول باستخدام CSS

شرح CSS

تعديل النموذج للارتباط بصفحة التفاصيل

النماذج التي تعرض بيانات تفصيلية

شفرة

شرح الكود

النماذج التي تقوم بتحرير البيانات

النماذج التي تحذف البيانات

رمز لحذف البيانات

شرح الكود

يمارس

متغيرات الجلسة

مقدمة

الجلسات

بدء جلسة PHP

استخدام متغيرات الجلسة

تخزين متغير

استرجاع متغير

التحقق من وجود متغير

إتلاف الجلسة

العمل مع الصور

مقدمة

تمرين: عرض الصور

سحب عدد غير معروف من الصور من قاعدة بيانات

تمرين: إنشاء جدول قاعدة بيانات لتخزين الصور. تمرين: قم بتعديل صفحة viewcar.php لإظهار

عدة صور

أوضح الكود

تحميلات ملف PHP

مقدمة

قم بإنشاء نموذج تحميل ملف

قم بإنشاء برنامج نصي لمعالجة الملف الذي تم تحميله

ViewCarsAddImage.php

شرح الكود

مراجعات PHP ونصائح

مقدمة

[اقتباسات مفردة مقابل اقتباسات مزدوجة](#)

[علامة المساواة](#)

[عوامل المقارنة](#)

[اعتبارات أمنية](#)

[مقدمة](#)

[موازنة الأمان وقابلية الاستخدام](#)

[حقن SQL](#)

[مصادر إضافية](#)

[الملحق أ: تثبيت PHP على موقع ويب](#)

[كيفية التثبيت على خادم ويندوز](#)

[كيفية التثبيت على خادم Linux](#)

[مفكرة](#)

[ملاحظة من أحد مؤيدي كيك ستارتر.](#)

[باشا كاجان ، مطور قريباً PHP-](#)

1

مقدمة

مقدمة

هذا الكتاب مخصص للمطور الذي صادف للتو PHP ويتساءل ما هي الصفة الكبيرة ، وأيضاً لغير المبرمج الذي بدأ للتو - ولا يعرف من أين يبدأ.

كما قال كونفوشيوس ذات مرة بحكمة ، "أسمع وأنس. أرى وأنذكر. أفعل وأفهم. "هذا الكتاب سوف يجعلك تفعل يتم تقديم الكتاب كدراسة حالة لـ "Sam's Used Cars" وستقوم بناء موقع ويب لأعمال أثناء تقدمنا. خذ وقتك وقم بالتمارين. كافح قليلا قبل أن تبحث عن الإجابات.

عند الحديث عن الإجابات ، فإن موقع الويب المصاحب لهذا الكتاب هو <http://www.joyofphp.com> حيث يمكنك العثور على جميع مقتطفات التعلمات البرمجية (إجابات التمارين) من هذا الكتاب - بالإضافة إلى بعض الأشياء الجيدة الإضافية. إذا أعجبك الكتاب ، فيرجح تحديد موقعه على Amazon.com ومنه مراجعة إيجابية. إذا لم تعجبك ، أو وجدت شيئاً تعتقد أنه بحاجة إلى الإصلاح ، أو لديك فكرة عن "المزيد من متعة" ، يرجى مراسلتي عبر البريد الإلكتروني على AlanForbes@Outlook.com

ما هي لغة PHP؟ يقدم PHP في سياقاتها العديدة والمتنوعة. يشرح الفرق بين خادم PHP وملف PHP ولغة PHP. يصف أيضا نقطة PHP وهي إنشاء صفحات ويب ديناميكية.

يصف تثبيت وتكوين PHP كيفية تثبيت PHP على جهاز الكمبيوتر الخاص بك. بالطبع ، لا يمكنك القيام بالكثير من برمجة PHP إذا لم يكن لديك ، لذلك هذا شرط أساسى مهم لبقية الكتاب.

تضع مقدمة إلى HTML أساساً من خلال مناقشة HTML. PHP هي لغة تعدل وتنتج ، لذلك عليك أن تعرف HTML كأساس لاستخدام PHP لتعديل HTML. نحن نغطي العلامات المطلوبة والاختيارية ، بالإضافة إلى الإضافات الكافية لمساعدتك على المضي قدماً.

بناء جملة الأساسية يقدم لغة PHP. نوضح هنا كيفية تداخل PHP و HTML وفي نفس الملف ، وماذا تتوقع عند القيام بذلك.

يمتحنك بعض المرح على الفور فرصة لتجربة بعض PHP قبل أن نذهب إلى أبعد من ذلك بكثير ، لنمنحك إحساساً مبكراً بالبهجة التي ستختبرها عندما إتقانها.

يتحدثEditors and Staying Organized عن كيفية تحرير ملف PHP بالفعل ، ويوجه القراء نحو بعض الأدوات المتاحة لتسهيل التحرير. أيضا ، البقاء منظم

هي عادة جيدة لتببدأ بها.

بعد ذلك نناقش المتغيرات **والأرقام والتاريخ والسلسل**. يتناول هذا الفصل كيفية إنشاء واستخدام متغير في **PHP** وكذلك كيفية إجراء وظائف حسابية ورقمية مفيدة. وهو يغطي السلسل ووظائف السلسل المفيدة ، كما يغطي التواريخ ودلالات التاريخ. يصف أيضًا كيفية قراءة وثائق **PHP** عند الحاجة

أكتر.

يغطي الفصل الخاص بهياكل التحكم **كيفية إضافة منطق شرطي إلى تطبيقك ، وكيفية أداء المهام القابلة للتكرار بطريقة آلية**.

يعتبر الفصل **كيفية استخدام قاعدة بيانات ، مثل MySQL** محتوى ثقيلاً. هنا تعرف على **ماهية قاعدة البيانات** ، وكيف تعمل الجداول ، وكيفية التعامل مع عبارات **SQL**. أقدم أيضًا أداة ، **PHPMyAdmin** وهي أداة رائعة لمساعدتك على بدء استخدام **MySQL**. يقدم لك **Sam** ، الذي يريد موقعاً على شبكة الإنترنت يسمح للزائرين برؤية السيارات التي لديه للبيع ، دون الحاجة إلى تعديل **HTML** الم موقعه باستمرار. سوف تكون **Sam's Used Cars** موقعاً على شبكة الإنترنت يعتمد على قاعدة البيانات.

في الفصل **استخدام MySQL و PHP معاً** ، بدأنا في **ربط الموضوعين معاً واستخدام MySQL و PHP في وقت واحد لإنشاء موقع ويب ديناميكيّة حقاً**.

مع تقدمنا في دراسة الحالة الخاصة بنا ، سنتنقل من النظرية إلى الممارسة. يبدأ الفصل "إنشاء نماذج لعرض البيانات وإضافتها وتحريرها وحذفها" في وضع بعض **نطريتنا موضع التنفيذ من خلال إنشاء أمثلة محددة لصفحات الويب التي تنفذ إجراءات على قاعدة البيانات الخاصة** بنا.

تسمح لك **متغيرات الجلسة بإنشاء متغير لتخزين قيمة يمكنك استخدامها في أي مكان في موقع الويب الخاص بك**، على سبيل المثال ، عند تسجيل الدخول إلى **Amazon.com** ستلاحظ أن كل صفحة تقول "مرحباً ، يا لأن" أو شيء مشابه -باستثناء ، بالطبع ، باسمك بدلاً من اسمي. تقدم متغيرات الجلسة طريقة واحدة لتحقيق هذا التأثير في **PHP**.

لن يكون لموقع الويب الخاص بالسيارات المستعملة قيمة كبيرة إذا لم تتمكن من رؤية صور السيارات ، لذلك في فصل العمل مع الصور ، نغطي التقنيات القوية لاستخدام قاعدة البيانات الخاصة بك لربط **سيارات معينة بصور محددة**.

تحميلات ملف PHP . بناءً على الفصل السابق الخاص بالصور ، قمنا بتوسيع وظائف موقعنا على الويب للسماح للمستخدمين بـ **تحميل صور السيارات مباشرة من المستعرض** ، بدلاً من الاضطرار إلى نسخ الصور إلى القرص الصلب.

كل اللغات لها مراوغات خاصة بها ، وتقدم **بعض ميزات PHP** **PHP Quirks and Tips** التي قد تبدو غريبة بالنسبة للبعض.

أخيراً ، نناقش اعتبارات **الأمان**. لا ينبغي أن يكون الأمان فكرة متأخرة عند إنشاء تطبيق ويب ، ولكن جاء أخيراً في الكتاب فقط لأنك لا تستطيع تأمين شيء ما إلا إذا فهمت أولاً كيفية عمله. لا تتحطى هذا الفصل!

2

ما هي لغة PHP؟

PHP هي لغة برمجة يمكنك استخدامها لإنشاء تطبيقات الويب. إنه مجاني وقوى وسهل الإعداد والتعلم نسبياً ، ولديه امتدادات وأطر عمل متاحة تقريرياً للقيام بأي شيء يمكن أن تخيله. يمكنك البدء بسرعة ، ولن تتخلص منها لاحقاً عندما تصبح جيداً حقاً. في رأيي المتواضع ، PHP هي لغة رائعة تستحق الوقت والجهد الذي تبذله في تعلمها.

بصراحة ، إنها مجرد متعة بسيطة أيضاً.

هيا بنا نبدأ. المفهوم الأساسي الذي تحتاج إلى فهمه هو أن صفحة الويب هي مجرد مجموعة من النصوص ، منظمة بطريقة معينة ، والتي يتم عرضها بواسطة المتصفح. فقط عدد قليل من الشركات يصنع المتصفحات ، لكن ملايين الأشخاص ينشئون صفحات ويب - ويمكنك أنت أيضاً.

تحتاج معظم برامج الكمبيوتر إلى طريقة ما لمعرفة ما إذا كان الملف مخصصاً لها أم لا. في عالم الكمبيوتر الشخصي ، يتم تحقيق ذلك عن طريق امتدادات الملفات. (تحمل معي ، هذا الجزء التالي مناسب). على سبيل المثال ، يرتبط ملف باسم "book.docx" بـ Microsoft Word لأن امتداده (النص الذي يلي النقطة) هو ".docx". وبالمثل ، قد ينتهي ملف PowerPoint بـ "my.ppt" أو البرامج الأخرى لها أيضاً امتدادات فريدة خاصة بها.

تحتوي صفحة الويب عادةً ، ولكن ليس حصرياً ، على امتداد مثل ".mth" أو ".mth.html" ، مما يشير إلى أنها ملف HTML على جهاز الكمبيوتر الخاص بك ، أو على جهاز كمبيوتر مختلف في مكان ما على الإنترنت. المتصفح لا يهتم. فيما يلي مثال بسيط لملف HTML موجود على الإنترنت:

<http://www.tsowell.com/columns.htm>

كانت هذه حول أبسط صفحة ويب يمكن أن أجدها ... رابطان فقط لصفحات أخرى. لاحظ أن الأحرف الأربع الأخيرة هي ".mth" ، مما يشير إلى أنه ملف HTML مخصص للعرض في المستعرض. إذا كان هذا الملف موجوداً على جهاز الكمبيوتر المحلي الخاص بك ، على سبيل المثال في مجلد "المستندات" ، فكل ما عليك فعله لرؤيته في المتصفح هو النقر فوقه نقرًا مزدوجًا.

سيعرف متصفحك كيفية القيام بالباقي وسترى شيئاً كهذا:

- [Current and Archived Columns](#)
- [Books of Collected Columns](#)

هذا الملف المحدد ليس على جهاز الكمبيوتر المحلي الخاص بك ، ومع ذلك. إنه على خادم خارج على الإنترنت. إذن كيف يصل الملف إلى متصفحك عند النقر فوقه؟ في خطوة

في تبسيطه ، لا داعي للقلق بشأن هذا الجزء كثيراً. يحتوي الكمبيوتر الآخر ، المعروف باسم "خادم الويب" ، على الملف ويعرف كيفية توصيله إليك.

في حالة ملف HTML مثل هذا ، يرسل الخادم الملف كما هو دون فعل أي شيء به. بمعنى آخر ، الملف الذي يحصل عليه المتصفح هو نفسه تماماً الملف الموجود على نظام الملفات ، بغض النظر عما إذا كان موجوداً في الأصل على جهاز الكمبيوتر الخاص بك أو على الخادم.

نسميهـا صـفـحة وـيـب ثـابـتـة .

تقديم PHP

الآن دعنا نضيف PHP إلى الصورة. لغة PHP لها عدة معانٍ اعتماداً على السياق الذي تستخدـمـ فـيـهـ ، لـذـاـ سـأـحـاـوـلـ شـرـحـهـ جـمـيـعـاـ. يوجد "خـادـمـ" ، وهو خـادـمـ PHPـ وـيـبـ يـقـومـ بـتـشـغـيلـ بـرـنـامـجـ خـادـمـ PHPـ عـلـيـهـ. دـعـنـاـ نـقـارـنـ خـادـمـ "عـادـيـ" ، خـادـمـ لاـ يـقـومـ بـتـشـغـيلـ يـتـلـقـيـ خـادـمـ الوـيـبـ "عـادـيـ" طـلـبـاـ مـنـ الـمـتـصـفـحـ ، وـيـحدـدـ مـوـقـعـ الـمـلـفـ الـمـنـاسـبـ ، وـيـرـسـلـهـ إـلـىـ الـمـتـصـفـحـ كـمـاـ هـوـ ، دـوـنـ أـيـ تـلـاعـبـ. بـمـعـنـيـ آخرـ ، إـنـهـ يـخـدـمـ صـفـحـاتـ الوـيـبـ الثـابـتـةـ فـقـطـ .

بـمـجـدـ إـضـافـةـ PHPـ إـلـىـ خـادـمـ وـيـبـ ، سـتـحـصـلـ عـلـىـ وـظـائـفـ إـضـافـيـةـ - دـوـنـ الـاستـغـنـاءـ عـنـ أـيـ وـظـائـفـ مـوـجـودـةـ. لـيـزـالـ بـإـمـكـانـ الـخـادـمـ الـاسـتـمـرـارـ فـيـ إـرـسـالـ مـلـفـاتـ HTMLـ إـلـىـ الـمـتـصـفـحـ ، وـلـكـ يـمـكـنـهـ أـيـضاـ مـعـالـجـةـ الـمـلـفـاتـ قـبـلـ إـرـسـالـهـ إـلـىـ الـمـتـصـفـحـ.

يـُـشـارـ إـلـىـ الـمـلـفـ الـذـيـ تـمـ التـلـاعـبـ بـهـ قـبـلـ إـرـسـالـهـ إـلـىـ الـمـتـصـفـحـ بـصـفـحةـ الوـيـبـ الـدـيـنـامـيـكـيـةـ .

A static web page never changes, unless a person specifically edits the page.

A dynamic web page can be different every time it is viewed by a browser, because the server edits the page prior to sending it to the browser, according to what instructions the programmer has coded into that specific page.

مثال

هـنـاـ مـثـالـ. لـنـفـتـرـضـ أـنـ لـدـيـكـ صـفـحةـ وـيـبـ تـرـيدـ أـنـ يـظـهـرـ التـارـيخـ الـحـالـيـ عـلـيـهـ. باـسـتـخـدـامـ صـفـحةـ وـيـبـ ثـابـتـةـ ، سـيـعـيـنـ عـلـيـكـ الدـخـولـ إـلـىـ الـصـفـحةـ وـتـحرـيرـهـاـ كـلـ يـوـمـ لـتـحـدـيـثـ التـارـيخـ. هـذـاـ مـنـ شـائـعـهـ أـنـ يـصـبـحـ مـعـنـيـاـ سـرـيـعـاـ جـداـ!

سيـبـدوـ كـوـدـ HTMLـ بـهـذـاـ الشـكـلـ:

```
<html>
  <body>
    Hello world! Today's date is the 7th of October 2012
  </body>
</html>
```

باـسـتـخـدـامـ ، يـمـكـنـكـ السـمـاحـ لـلـخـادـمـ بـإـجـرـاءـ التـغـيـرـاتـ نـيـابـةـ عـنـكـ. بـمـعـنـيـ آـخـرـ ، يـمـكـنـ لـ PHPـ

أضف التاريخ الصحيح ديناميكياً إلى الصفحة في كل مرة يتم فيها عرض الصفحة إذا أدخلت قليلاً من كود PHP مثل هذا:

```
<html>
<body>
    Hello world! Today's date is the <?php echo date('jS \o\f F Y'); ?>
</body>
</html>
```

لاحظ أنه تم استبدال النص ثابت بتاريخ 7 أكتوبر 2012 بـ <?php echo date('jS \o\f F Y'); ?>. ما فعلناه هو استبدال النص الثابت برمز سيتم تحويله إلى نص ثابت بواسطة الخادم. (معينة سريعة - يظهر كود PHP بالتوالي مع كود HTML العادي ويتم تحديده من خلال الظهور ضمن علامتي <?.?.>).

في ٢٠١٢٧٥٤٦٩٣ لـ[أكتوبر](#) لهذا الملفين باستخدام متصفحك في ، فستكون الصفحتان متماثلتين تماماً.

لـ[أكتوبر](#) من أكتوبر ، يأخذ الخادم الكود أعلاه ويجعله إلى هذا:

```
<html>
<body>
    Hello world! Today's date is the 8th of October 2012
</body>
</html>
```

والشيء الرائع هو أنه يعمل كل يوم ، دون أي مزيد من التلاعب. هل بدأت ترى الفرج؟

كيف يعرف الخادم ما إذا كانت الصفحة يجب أن تكون ديناميكية أو ثابتة؟ الخادم العادي يعرف فقط الصفحات الثابتة. يعرف خادم PHP أنه يجب معالجة الملف (يكون ديناميكياً) إذا كان "ملف PHP وأنه لا ينبغي معالجته (يكون ثابتاً) إذا كان ملف HTML.

ما الفرق بين ملف HTML وملف PHP؟ هو في الأساس مجرد ملف HTML به بعض التعليمات البرمجية بداخله تخبر الخادم بتبديل جزء الكود وإدراج نص (أو HTML في مكانه. ملف PHP هو ملف HTML فقط" تم حفظه بامتداد مختلف ".php". - هذا مثال:

<http://php.net/manual/en/tutorial.firstpage.php>

A **PHP file** is just an HTML file saved using a **.php** extension instead of an **.html** or **.htm** extension, which tells the server to look in the page for code.

ما هو "الرمز الإضافي" الذي يتم إدخاله داخل ملف PHP الذي يأمر الصفحة بأن تكون

التلعب بها؟ هذه هي لغة ، PHP التي تخبر الخادم كيف وأين يجب معالجة الصفحة قبل إرسالها إلى المتصفح. بمعنى آخر ، هي لغة برمجة تُستخدم لإنشاء صفحات ويب ديناميكية.

PHP is a language that can be used to create dynamic web pages. In fact, that is the **point of PHP.**

كيف يعرف الخادم أي أجزاء من الصفحة يجب أن تكون ثابتة وأي أجزاء يجب أن تكون ديناميكية؟ بشكل عام ، يترك الخادم الصفحة وشأنها. ومع ذلك ، إذا رأى النص ، `<?php` فسيتم التعامل مع كل النص التالي مثل الكود ، حتى يصل إلى `?>` الذي يشير إلى الخادم للعودة إلى إرسال الصفحة كما هي.

Text on a PHP page is normally static. The PHP server will dynamically convert the text that appears in between `<?php` and `?>` tags into static text, after evaluating what the code means.

قليلًا من التاريخ

تم إنشاء PHP في الأصل بواسطة Rasmus Lerdorf في عام 1995. يتم الآن إنتاج التطبيق الرئيسي له بواسطة مجموعة PHP ويعمل كمرجع رسمي لغة PHP. PHP هو برنامج مجاني تم إصداره بموجب ترخيص PHP وهو غير متوافق مع رخصة GNU العامة (GPL) بسبب القيود المفروضة على استخدام مصطلح PHP.

بينما كانت PHP تعني في الأصل الصفحة الرئيسية الشخصية ، يقال الآن أنها تمثل ، PHP: Hypertext Preprocessor اختصار تكراري.

3

تثبيت وتكوين PHP

مقدمة

قبل أن تتمكن من البدء في استخدام ، يجب أن يكون لديك نسخة منها. لأغراض هذا الكتاب ، سنقوم بتثبيت PHP على جهاز الكمبيوتر المحلي الخاص بك حتى تتمكن من اللعب بها دون الكثير من الجلبة.

لاحظ أن التطبيقات التي تكتبها على جهاز الكمبيوتر المحلي الخاص بك ، أو بواسطة أجهزة الكمبيوتر الأخرى الموجودة على شبكتك المحلية.

بالنسبة لبقية هذا الكتاب ، ستحتاج إلى Apache و MySQL و PHP. يشار إلى هذه المجموعة معاً باسم LAMP. فإذا كان Windows ، Linux فإنه يسمى ، و عندما تقوم بتشغيله على ، يطلق بعض الأشخاص على هذه المجموعة اسم "LAMP Stack". وقد يشيرون إلى المجموعة باسم "Mkds" ، وقد يشارون إلى المجموعة باسم.

لحسن الحظ ، هذه تركيبة شائعة جدًا ولا يتعدى عليك معرفة كل شيء بنفسك.

مستخدمو Windows

بصفتي أحد مستخدمي Windows يمكنني شخصياً أن أضمن سهولة التثبيت والاستخدام ، والذي يمكن العثور عليه على <http://www.wampserver.com/en/>

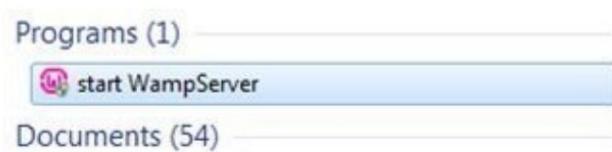
"WampServer" هي بيئه تطوير ويب يسمح لك بإنشاء تطبيقات ويب باستخدام Apache2 و PHP و قاعدة بيانات MySQL. يتضمن WampServer أيضًا برنامجًا يسمى PhpMyAdmin والذي يسمح لك بإدارة قواعد البيانات الخاصة بك بسهولة.

جزء من التثبيت ، سيتم إنشاء دليل "www" تلقائياً (عادةً على c:\wamp\www).

أنشئ دليلاً فرعياً في "www" وضع ملفات PHP داخل هذا المجلد.

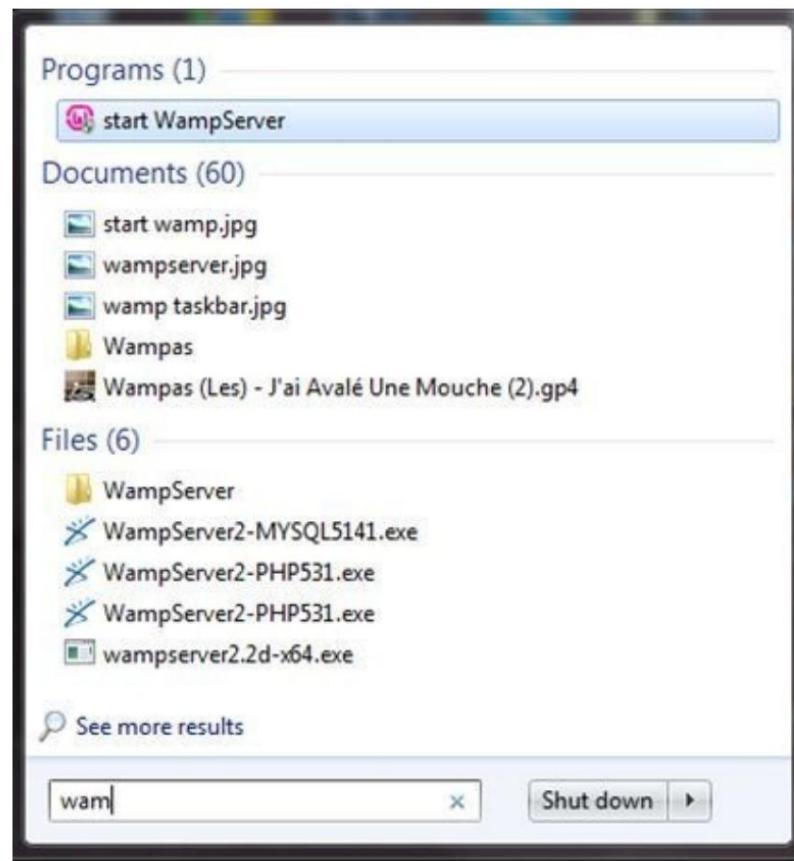
باستخدام Wampserver

بمجرد تثبيت ، يمكنك بدء تشغيله بتحديد بدء WampServer من ملف قائمة ابدأ في ، كما هو موضح أدناه:



إذا لم يكن موجوداً في قائمة "ابداً" ، فابحث عن البرامج التي تبدأ بـ "WAM" كما هو موضح

أقل:



يؤدي هذا إلى إضافة رمز سطح المكتب على الجانب الأيمن من شريط المهام ، محاط بدائرة باللون الأحمر:



ستكون الأيقونة حمراء إذا كان WampServer لا يعمل ، والأخضر إذا كان قيد التشغيل. سيؤدي النقر فوق الرمز إلى إظهار قائمة منبثقة ، تشبه قائمة ابدأ في Windows.

انقر على رابط "localhost" في قائمة WampServer أو افتح متصفح الإنترنت وانتقل إلى URL: http://localhost

مستخدمو Mac

XAMPP هو الحل الأبسط والأكثر عملية وكاملة لخادم الويب لنظام التشغيل Mac OS X. يتضمن التوزيع خادم Apache 2 مدمجاً مع أحدث إصدارات MySQL و PHP و Perl. يأتي كحزمة تثبيت لنظام التشغيل Mac OS X تحتوي على جميع الملفات الضرورية ولا تتطلب أي تبعيات.

إذا كنت مطور ويب متعمداً أو متخصصاً لنظام التشغيل Mac وتحتاج إلى تشغيل خادم ويب أو إنشاء صفحات ويب ديناميكية أو استخدام قواعد بيانات ، فهذا هو يومك المحظوظ!

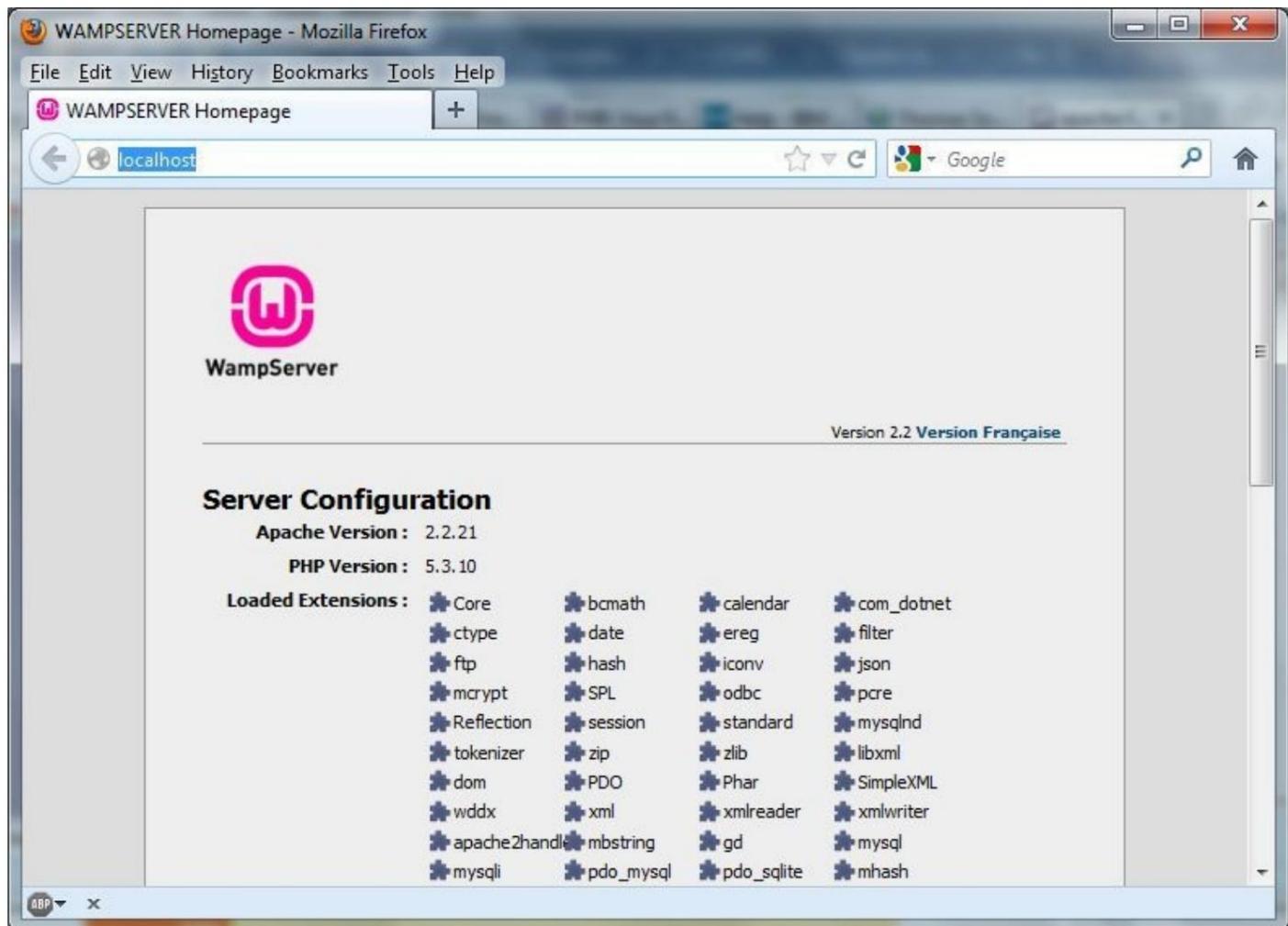
هذا الإصدار مخصص لنظام التشغيل Mac OS X 10.4 (Intel & PPC) والإصدارات الأحدث.

يمكن العثور على XAMPP على <http://www.apachefriends.org/en/xampp-macosx.html>

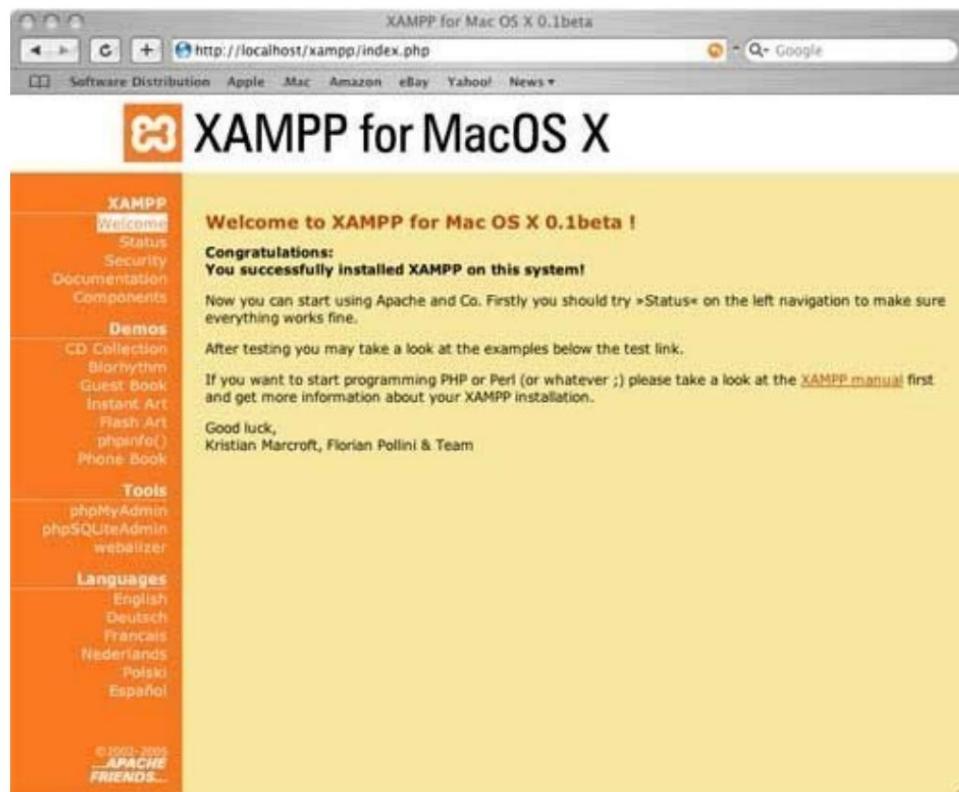


كيف أعرف أنها تعمل؟

حسناً ، كان ذلك سهلاً ، لكن كيف يمكنك التحقق من أن كل شيء يعمل حقاً؟ ما عليك سوى كتابة عنوان URL التالي في متصفح الويب المفضل لديك: <http://localhost> شيئاً مثل هذا:



سيرى مستخدمو Mac شيئاً مثل هذا:



Oracle VM Virtual Box

إذا كنت بارعاً في التعامل مع أجهزة الكمبيوتر ، فقد ترغب في تجربة جهاز ظاهري مُعد مسبقاً. Oracle VirtualBox مجاني للاستخدام ، ويمكن مشاركة المجلدات بين المضيف وجهاز الضيف ، مما يتيح لك ببساطة حفظ الملف الذي تعمل عليه وتحديث المستعرض الخاص بك ، ولا داعي للتحميل عبر FTP / SFTP لاختبار تغييراتك. إذا كان هذا يبدو كخيار جيد بالنسبة لك ، فإن إليك رابط البرنامج التعليمي على إعداده. يعد استخدام Oracle Virtual Box خارج نطاق هذا الكتاب.

الثبيت لموقع ويب

إذا أردت جعل تطبيقك متاحاً للجميع على الإنترن特 ، فستحتاج إلى تثبيت PHP وتطبيقاتك على خادم يمكن الوصول إليه بشكل عام. يتم تناول هذا الموضوع في الملحق أ: تثبيت PHP على موقع ويب.

يمارس

استخدم المحرر الخاص بك (على سبيل المثال ، Programmers 'Notepad) لإنشاء ملف يحتوي على السطر التالي:

```
<?php echo phpversion(); ?>
```

احفظ الملف كملف `phpinfo.php` في المكان الصحيح على محرك الأقراص الثابتة (على سبيل المثال ، `\www`).
`C:\wamp`

أخيراً ، افتح الملف باستخدام متصفح عن طريق كتابة `http://localhost/phpinfo.php`

4

مقدمة إلى HTML

مقدمة

كما وصفناها ، PHP هي لغة تستخدم لإنشاء صفحات ويب ديناميكية. تتم كتابة صفحات الويب بلغة HTML ويتم استخدام بحيث يتغير HTML في صفحة معينة اعتماداً على مواقف معينة تحددها أنت.

نظرًا لاستخدام PHP لإنشاء HTML على الصفحة ، فمن المنطقي أنك بحاجة إلى فهم HTML الأساسي قبل أن تتمكن من المضي قدماً.

أحياناً يُشار إلى HTML هي تقنية ذات صلة تستخدم لتحديد شكل وأسلوب صفحة . Cascading Style Sheets (CSS) أكثر بساطة على أنها ورقة أنماط.

إذا كنت تفهم بالفعل HTML و CSS فيمكنك الانتقال إلى الفصل التالي.

HTML الأساسي

HTML هو البناء الأساسية للويب ، لذلك من الضروري أن يكون لديك فهم أساسى ل מהية HTML وكيف تعمل. HTML هي لغة ترميز تستخدمها المتصفحات حتى يعرفوا كيفية عرض المستند كصفحة ويب.

بغض النظر عما إذا كان المستند يبدأ بتنسيق HTML مكتوب يدوياً أو يتم إنشاؤه باستخدام ASP أو JSP أو PHP يتم تحويل المستند في النهاية إلى HTML وإرساله إلى المتصفح لعرضه على شخص.

HTML هي لغة ترميز تحدد بنية المستند ومخططه وتقدم محتوى منظماً. لا يقصد بالترميز تحديد شكل وأسلوب المحتوى على الصفحة بما يتجاوز المفاهيم الأولية مثل الرؤوس والفقرات والقوائم.

تم إهمال جميع سمات العرض التقديمي لـ HTML وهي كلمة رائعة لـ "من فضلك لا تستخدمها بعد الان ، على الرغم من أنها لا تزال تعمل". تنص أفضل الممارسات الحالية في تصميم صفحة HTML على أنه يجب تصميم معظم الأنماط في أوراق الأنماط ، وهي مجموعة من القواعد التي تصف الشكل الذي يجب أن تبدو عليه الصفحة. تعدد أوراق الأنماط موضوعاً في حد ذاتها ، وليس مهم جدًا في هذه المرحلة من تعلمك. ومع ذلك ، سترغب في وضع أوراق الأنماط في قائمة القراءة المستقبلية.

تعد كتابة HTML وعرضها أمرًا سهلاً للغاية (وممتعاً) ، وهو بالطبع عامل كبير في جعله ذائع الصيت. إذا كنت تقرأ هذا المستند على جهاز كمبيوتر ، فلديك بالفعل كل ما تحتاجه لتجربته الان. كل ما تحتاجه لإنشاء صفحة ويب (صفحة HTML) هو جهاز كمبيوتر ومحرر نصوص (يكي شيء بسيط مثل Notepad) ومتصفح للعمل مع ، لا تحتاج إلى خادم أو أي برنامج خاص على الإطلاق. ما عليك سوى إنشاء الملف وحفظه بامتداد .mth أو .lmt . وفتحه مباشرة في متصفحك.

Windows users will find [Programmer's Notepad](#) to be a great editor for creating HTML files.

العناصر الأساسية لـ HTML

تحتوي جميع المستندات المستندة إلى HTML على نفس العناصر الأساسية. وهي تتكون من علامات تحدد الأجزاء المختلفة من المستند - من حيث يبدأ وينتهي ، وكل شيء بينهما. يستخدم HTML العناصر ("العلامات") لترميز أقسام النص. يمكن أن تشمل هذه العناوين والعنوانين الفرعية والقوائم والنص الغامق أو المسطر وبالطبع الروابط.

تُقرأ مستندات HTML من اليسار إلى اليمين ومن أعلى إلى أسفل.

العلامات [لتمييز](#) العلامات عن النص العادي ، تظهر العلامات داخل أقواس: < .> تحتوي معظم العلامات على علامة فتح وإغلاق ، تُعرف أيضًا باسم علامة البداية والنهاية. تبدأ علامة الفتح بـ </> وتبدأ علامة النهاية بـ .>. على سبيل المثال ، يشير إلى البدء بالخط العريض و يشير إلى الإيقاف (النهاية) بالخط العريض.

على سبيل المثال هنا عنصر فقرة:

```
<p>This is the first paragraph.</p>
```

في هذا المثال ، تكون الوسم <p> و </p> يتم استخدامهما لتحديد النص الموجود بداخله كفقرة. هناك شيء يستحق الإشارة إليه هنا وهو أنك لست مضطراً لوضع كل شيء في سطر واحد. يعمل الكود أعلاه تماماً مثل هذا أدناه:

```
<p>
  This is the first paragraph
</p>
```

في الواقع ، المسافة الباردة ليست ضرورية أيضاً ، على الرغم من أنها تعمل بالتأكيد على تحسين إمكانية القراءة. ضع في اعتبارك أن شخصاً ما (ربما أنت) قد يضطر إلى تعديل HTML في المستقبل ، لذا فإن جعله قابلاً للقراءة يعد فكرة جيدة.

جميع تنسيقات العلامات هي نفسها. تبدأ بعلامة أقل من: < .> و تنتهي بعلامة أكبر من: < .> دائمًا ما يتم إدخاله داخل < > هو اسم العلامة. جزء كبير من تعلم HTML هو تعلم العلامات المحددة وما يفعلونه.

العلامات المتداخلة

بشكل عام ، يمكن أن تتدخل معظم العلامات داخل علامات أخرى ، ولكن بالطبع قد تكون هناك استثناءات لهذه القاعدة.

هنا ترى علامة غامقة متداخلة داخل علامة فقرة:

```
<p>
  This is the first paragraph, with some <b>bold</b> text in it.
</p>
```

لا تحتوي كل العناصر على قطعة فتح وختام. على سبيل المثال ،
 لا

لها `
` مقابل ، ولا `<hr>` أيضاً.

The `<hr>` tag inserts a horizontal rule (line) and the `
` tag inserts a new line.

العلامات المطلوبة

تبدأ صفحة HTML بعلامة `<html>` وتنتهي بـ `</html>`. يتم وضع نص الصفحة داخل علامات النص الأساسية.

```
<!DOCTYPE html>
<html>
<body>
<p>This is my first paragraph.</p>
</body>

</html>
```

إذا كانت صفحة الويب تفتقد إلى علامة `<DOCTYPE>` أو بها نوع من علامة "الانتقالية" ، فسيتم عرض الصفحة في ما يسمى بوضع "المراوغات". وضع المراوغات لا يمكن التنبؤ به إلى حد ما ، ولا تحصل دائمًا على ما تتوقعه.

لذلك ، من المهم أن يكون لديك علامة على شكل مستند إذا كنت تريده عرض صفحة الويب الخاصة بك في وضع المعايير ، كما هو متوقع.

رأس

عنوان المستند هو المكان الذي ستذهب إليه معلومات العنوان والميتا . بشكل عام ، يمكنك أيضًا وضع أي أنماط CSS وعلامات نصية وعلامات ارتباط إلى الملفات الخارجية في الرأس أيضًا ، إذا كان لديك أي منها.

```
<!DOCTYPE html>
<html>
<head>
<title>
    This is the title of the page.
</title>
</head>
<body>
    The body text goes here.
</body>
</html>
```

العلامات الاختيارية

العلامات الفوقيّة

تعد علامة `Meta` جنباً إلى جنب مع علامة الارتباط ، فريدة من نوعها من حيث أنها علامات HTML الوحيدة التي لا تتطلب علامة إغلاق ولا إغلاق / في نهاية العلامة ، ولا تزال تعتبر صحيحة من الناحية النحوية.

الشيء الآخر بخصوص العلامات الوصفية هو أنها العلامة الوحيدة التي ، بشكل عام ، ليس لها أي تأثير على تخطيط الصفحة أو معالجتها ؛ يتم استخدامها لتقديم معلومات حول الصفحة و / أو الموقع الذي يتم عرضه. العلامة الوصفية هي في الأساس زوج مفتاح / قيمة ، ويمكن أن تحتوي كل علامة على زوج واحد فقط من القيم. تستخدم العلامات الوصفية بشكل أساسي بواسطة محركات البحث.

علامات مفيدة

الرؤوس

تُستخدم الرؤوس لتنظيم المعلومات في مجموعات هرمية.

1 </h1> العنوان

2 </h2> العنوان

3 </h3> العنوان

العنوان 4 </h4> العنوان

5 </h5>

6 </h6> العنوان

علامات الرأس هي عناصر على مستوى الكتلة ، مما يعني أنها تشغل سطراً كاملاً بمفردها ، ولا يُسمح بأي ترميز آخر داخل علامات العنوان.

DIV

علامة DIV هي علامة يمكنك استخدامها لإنشاء تقسيم منطقي داخل المستند الخاص بك. تعمل DIVs مع CSS وتسمح لك بكتابة قواعد CSS التي تحدد كيفية تنسيق النص داخل DIV.

الصور / الصورة بالإضافة صورة إلى المستند الخاص بك ، يمكنك استخدام علامة "صورة". لإدراج صورة في مستند html الخاص بك ، استخدم الصيغة التالية:

```

```

يمكن للقيمة التي تضعها في السمة 'src' إما تحديد رسم موجود في نظام الملفات المحلي ، أو يمكنك تحديد عنوان URL كامل ، والذي يسترد الصورة من مكان آخر على الإنترنت.

الروابط

الرابط يأخذ المستخدم إلى مكان آخر عندما ينقر عليه. يمكن أن يكون الارتباط إلى جزء معين من المستند المفتوح أو إلى صفحة جديدة بالكامل.

يأخذ المستخدم إلى صفحة جديدة.

```
<a href="http://www.joyofphp.com">Visit Joy of PHP!</a>
```

ينقل المستخدم إلى مكان مختلف (كما هو مشار إليه في العلامة ``) في الصفحة الحالية:

```
<a href="#top">Go to top</a>
```

القوائم

هناك نوعان من القوائم - مرتبة وغير مرتبة. يتم ترتيب القائمة المرتبة ، مثل ، 3 ، 2 ، 1 بينما القائمة غير المرتبة هي قائمة بعناصر تعداد نقطي. هناك علامات لبدء وإيقاف القائمة ، وعلامات لكل عنصر في القائمة.

تبدأ القائمة المرتبة بعلامة `.` تبدأ القائمة غير المرتبة بعلامة `.` يبدأ كل عنصر قائمة ، بغض النظر عن نوع القائمة ، بـ `` العلامة وينتهي بـ ``.

```
<ul>
<li>Coffee</li>
<li>Milk</li>
</ul>
```

| | |
|---|---|
| <p>An ordered list:</p> <ol style="list-style-type: none"> 1. The first list item 2. The second list item 3. The third list item | <p>An unordered list:</p> <ul style="list-style-type: none"> • Coffee • Tea • Chocolate Milk |
|---|---|

يمارس

قم بإنشاء ملفين أساسيين بتنسيق HTML ووضعهما في المكان الصحيح على جهاز الكمبيوتر الخاص بك حتى تتمكن من فتحهما في المستعرض. قم بتضمين القوائم والقرارات والنص الغامق والمائل. لم أخبرك كيف تجعل المهمة مائلة. بالنظر إلى أن علامة الخط الغامق هي `,` ، `فماذا تفترض أن علامة الخط المائل؟`

جداول

الجداول رائعة. إنها تحل الكثير من المشكلات ، لكن لا ينبغي استخدامها لتخطيط الصفحة بشكل عام . يجب استخدام جداول HTML فقط لعرض البيانات التي تنتهي إلى شبكة أو بمعنى آخر حيث تصف البيانات عددًا من الكائنات التي لها نفس الخصائص. على سبيل المثال ، إذا كان من المنطقي عرض البيانات في Microsoft Excel ، فاستخدم جدولًا.

يتم تعريف الجداول بعلامة `<table>` . ينقسم الجدول إلى صفوف (علامة `<tr>`) وينقسم كل صف إلى خلايا بيانات (علامة `<td>`) . يرمز `td` إلى "بيانات الجدول" ، ويحمل محتوى خلية البيانات. يمكن أن تحتوي العلامة `<td>` على نصوص ، روابط ، صور ، قوائم ، نماذج ، جداول أخرى ، إلخ.

مثال على الجدول

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

هذه هي الطريقة التي ستبدو بها شفرة HTML أعلاه بمجرد ترجمتها بواسطة المستعرض. سيرسم المتصفح خطوطاً حول الخلايا لأنني قمت بتضمين '1' = 'border' في علامة الفتح.

| | |
|---------------|---------------|
| row 1, cell 1 | row 1, cell 2 |
| row 2, cell 1 | row 2, cell 2 |

جداؤل HTML وسمة الحدود

إذا لم تحدد سمة حد ، فسيتم عرض الجدول بدون حدود.
قد يكون هذا مفيداً في بعض الأحيان ، ولكن في معظم الأحيان ، نريد أن تظهر الحدود.

لعرض جدول بلا حدود ، ما عليك سوى إسقاط سمة الحدود:

```
<table>
<tr>
<td>Row 1, cell 1</td>
<td>Row 1, cell 2</td>
</tr>
</table>
```

رؤوس جدول HTML

يتم تحديد الرؤوس في الجدول (الصف العلوي الذي يصف البيانات بدلاً من كونها بيانات) بعلامة `<th>`.

تعرض جميع المتصفحات الرئيسية النص الموجود في عنصر `<th>` بخط عريض ومتوسط.

```
<table border="1">
<tr>
<th>Header 1</th>
<th>Header 2</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

كيف يبدو رمز HTML أعلاه في متصفحك:

| Header 1 | Header 2 |
|---------------|---------------|
| row 1, cell 1 | row 1, cell 2 |
| row 2, cell 1 | row 2, cell 2 |

يمكن أن تخلق الجداول مشاكل في إمكانية الوصول. نظراً لأنه من المفترض أن تُقرأ الجداول من اليسار إلى اليمين ، فإن استخدامها للتخطيط يمكن أن يتسبب في أن تقرأ ببرامج قراءة الشاشة المحتوى خارج الترتيب وتسبب إرباكاً للمستخدمين الذين يعتمدون على برامج قراءة الشاشة.

نماذج HTML

نماذج HTML هي نوع خاص من صفحات HTML يمكن استخدامها لتمرير البيانات إلى الخادم. بمجرد أن يحصل الخادم على البيانات ، قد يقوم بمعالجتها وإرسال بعضها مرة أخرى ، أو قد يخزنها في قاعدة بيانات لاستخدامها لاحقاً.

سيحتوي نموذج HTML على عناصر إدخال مثل التسميات وحقول النص وخانات الاختيار وأزرار تحديد الاختيار وأزرار الإرسال والمزيد. يمكن أن يقدم النموذج أيضاً قوائم ، يمكن للمستخدم من خلالها التحديد ، أو منطقة نص حيث يُسمح بالكتابة متعددة الأسطر.

الهيكل الأساسي للنموذج هو كما يلي:

```
<form>
  .
  input elements
  .
</form>
```

يتم وضع علامات النموذج داخل علامة `<body>`. يتم إرسال البيانات الموجودة في النموذج إلى الصفحة المحددة في سمة الإجراء الخاصة بالنموذج. الملف المحدد في سمة الإجراء عادة

ي فعل شيئاً مع المدخلات المستلمة:

```
<form name="input" action="PostSignup.php" method="get">
```

سنفطي إجراءات النموذج لاحقاً.

علامة الإدخال

أكثر عناصر النموذج شيوغاً هو عنصر `<input>` ، والذي يستخدم لجمع المعلومات من المستخدم. يحتوي عنصر `<input>` على العديد من الاختلافات ، والتي تعتمد على سمة النوع . يحتوي عنصر `<input>` أيضاً على عنصر اسم ، لذا يمكنك الرجوع إليه لاحقاً. بشكل عام ، بناء الجملة هو:

```
<input type="type" name="name"/>
```

يمكن أن يكون عنصر `<input>` من نوع النص ومربع الاختيار وكلمة المرور ورر الاختيار ورر الإرسال والمزيد. يتم وصف أنواع المدخلات الشائعة.

حقول النص: يحدد حقل إدخال من سطر واحد يمكن للمستخدم إدخال نص فيه:

```
<form>
First name: <input type="text" name="firstname"/><br>
Last name: <input type="text" name="lastname"/>
</form>
```

هكذا ستبدو شفرة HTML أعلاه في المتصفح:

| | |
|-------------|---------------------------------------|
| First name: | <input name="firstname" type="text"/> |
| Last name: | <input name="lastname" type="text"/> |

حقل كلمة المرور: يحدد حقل كلمة المرور يشبه حقل النص تماماً ، باستثناء عدم عرض النص المكتوب في ملف شاشة.

```
Password: <input type="password" name="pwd"/>
```

حقل كلمة المرور لا يؤمن البيانات ، إنه يخفيها فقط عن البشر.

أزرار الراديو: `(نوع الإدخال = "راديو")` يحدد زر اختيار. تتيح أزرار الاختيار للمستخدم تحديد خيار واحد (واحد فقط) من عدد محدود من الخيارات المعروضة:

```
<body>
<form>
Pick your favorite color: <br/>
<input type="radio" name="color" value="red"/>Red<br/>
<input type="radio" name="color" value="blue"/>Blue<br/>
<input type="radio" name="color" value="green"/>Green
</form>
</body>
```

هكذا تبدو شفرة HTML أعلاه في المتصفح:

Pick your favorite color:
 Red
 Blue
 Green

مربعات الاختيار: يحدد مربع اختيار. تتيح مربعات الاختيار للمستخدم تحديد خيارات صفرية أو أكثر من عدد محدود من الخيارات.

```
<form>
<input type="checkbox" name="vehicle" value="Bike">I have a bike<br/>
<input type="checkbox" name="vehicle" value="Car">I have a car
</form>
```

زر الإرسال: يحدد زر الإرسال.

يتم استخدام زر الإرسال عندما يملأ المستخدم النموذج ويكون جاهزاً لإرسال ("إرسال") البيانات التي قام بإدخالها إلى الخادم. يتم إرسال البيانات إلى الصفحة المحددة في سمة الإجراء الخاصة بالنماذج ، والتي سيتم تناولها في القسم التالي.

أساليب وإجراءات نموذج HTML

عند تحديد نموذج ، هناك سمتان مطلوبتان: الإجراء والطريقة. تشير سمة الإجراء (`= إلى اسم الملف الذي سيتم إرسال النموذج إليه`). تحدد سمة الطريقة (`= كيفية إرسال النموذج`).

عادةً ما يقوم الملف المحدد في سمة الإجراء بعمل شيء مع المدخلات المستلمة ، مثل وضعه في قاعدة بيانات أو إعادة بعض القيم إلى المستخدم، فيما يلي مثال على نموذج بسيط يحتوي على سمات الإجراء والطريقة.

```
<form name="input" action="form_action.php"
method="get">
    Your name: <input type="text" name="name"/>
    <input type="submit" value="Submit">
</form>
```

لأغراض هذا الكتاب ، سنفترض أن سمة الإجراء تحدد اسم ملف PHP كاما ستري ، ملف PHP المحدد في سمة الإجراء

الوصول إلى جميع القيم في النموذج الذي تم إرساله.

سنفطي إجراءات النموذج بمزيد من التفصيل في قسم إنشاء نماذج لعرض البيانات وإضافتها وتحريرها وحذفها.

5

قواعد PHP الأساسية

مقدمة - بناء جملة PHP الأساسية

يبدأ نص PHP دائمًا بـ `<?php` وينتهي بـ `?>`. يمكن وضع نص PHP في أي مكان في المستند.

```
<?php  
 //Imagine some code here ...  
?>
```

In PHP, we use `//` to make a one-line comment or `/*` and `*/` to make a comment block

يجب أن يكون ملف PHP ممتداد `.php`. يحتوي ملف PHP عادةً على علامات HTML وبعض أكواد PHP النصية. قبل أن نذهب إلى ذلك ، من المهم ملاحظة أن PHP حساسة لحالة الأحرف. تأكد من اتباع نفس الغلاف الذي تراه في الأمثلة.

أول صفحتك الممكّنة بلغة PHP - أهلاً بالعالم!

قم بإنشاء ملف باسم `hello.php` ووضعه في الدليل الجذر لخادم الويب الخاص بك (`C:\wamp\www`)؟
بالمحظى التالي:

```
<html>  
  <head>  
    <title>PHP Test</title>  
  </head>  
  <body>  
    <?php echo '<p>Hello World!</p>'; ?>  
  </body>  
</html>
```

استخدم متصفحك للوصول إلى الملف بعنوان URL لخادم الويب الخاص بك ، منتهيًا بمرجع الملف / `hello.php`.
محليًا ، سيكون عنوان URL لهذا مثل `http://localhost/hello.php` أو `http://127.0.0.1/hello.php` ولكن هذا يعتمد على
تكوين جهاز الكمبيوتر الخاص بك.

إذا تم تكوين كل شيء بشكل صحيح ، فسيتم تحليل هذا الملف بواسطة PHP وبشكل سحري

تحولت إلى .إذا سارت الأمور على ما يرام ، فسيتم إرسال HTML التالي إلى متصفحك:

```
<html>
<head>
<title>PHP Test</title>
</head>
<body>
<p>Hello World!</p>
</body>
</html>
```

هذا البرنامج بسيط للغاية ، ولا تحتاج حقاً إلى استخدام PHP لإنشاء صفحة كهذه. كل ما يفعله هو عرض "Hello World" في المتصفح. ومع ذلك ، تعتبر هذه الطريقة الكلاسيكية لتقديم لغة برمجة - توضح للمستخدمين كيفية قول "مرحباً بالعالم".

The PHP echo statement sends whatever text follows the statement to the browser.

لاحظ أنه لا يوجد شيء مميز بشكل خاص حول هذا الملف. يعرف الخادم أن هذا الملف يحتاج إلى تفسير بواسطة PHP لأنك استخدمت الامتداد ، ".php" والذي تم تكوين الخادم لتمريره إلى PHP. فكر في هذا على أنه ملف HTML عادي يحتوي على مجموعة من العلامات الخاصة المتاحة لك والتي تقوم بالكثير من الأشياء المثيرة للاهتمام.

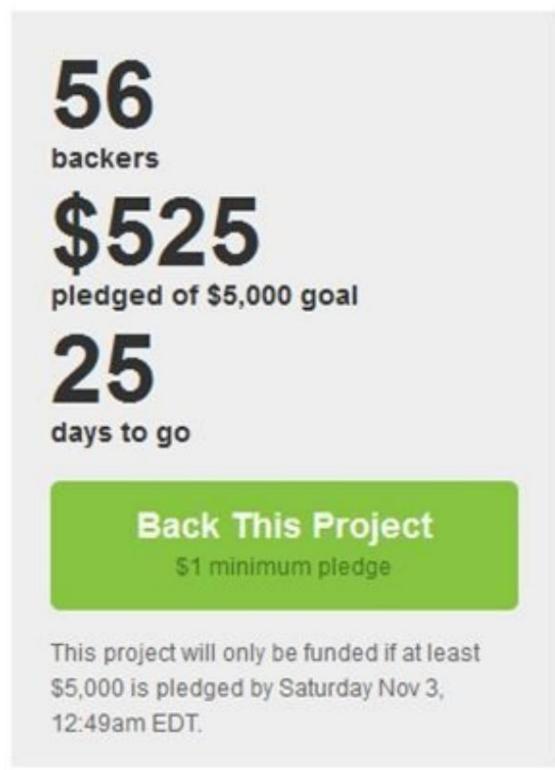
الهدف من المثال هو إظهار تنسيق علامة PHP الخاص. في هذا المثال استخدمنا `?<?=` للإشارة إلى بداية علامة PHP. ثم نضع تعليمة PHP ونضع الأيسر بإضافة علامة الإغلاق ، ؟. يمكنك الدخول والخروج من وضع PHP في ملف HTML مثل هذا في أي مكان تريده.

6

بعض المرح على الفور

عداد العد التنازلي

نظرًا لأن معظمكم الذين اشتروا هذا الكتاب قاموا بذلك من خلال رعاية المشروع على Kickstarter فأنت على دراية بفكرة العد التنازلي لتاريخ ووقت محددين. الفكرة هي خلق الإثارة والشعور بالإلحاح.



كيف يمكنك إنشاء مثل هذا الشيء في PHP؟ نحتاج أولاً إلى تحديد التاريخ المستهدف - الوقت الذي نقوم بالعد التنازلي إليه. في حالة مشروع Kickstarter الذي أطلق هذا الكتاب ، كان الوقت المستهدف هو 30 سبتمبر 2012. يمكنك إنشاء متغير في PHP للاحتفاظ بالوقت المستهدف باستخدام `mktime()` (أجعل الوقت) تعمل على النحو التالي:

```
$target = mktime(0, 0, 0, 9, 30, 2012);
```

mktime

The `mktime()` function is used to get the timestamp for a specified date. It is phrased as `mktime(hour, minute, second, month, day, year, is_dst)`

Hour - Number of the hour

Minute - Number of the minute

Second - Number of seconds past the minute

Month - Number of the month

Day - Number of the day

Year - Two or four digit representation of the year.

If you use 2 digit 00-69 will be prefixed with 20

(2000-2069) and 70-99 will be prefixed with 19

(1970-1999). If you use 100 it will be interpreted as 2000.

`Is_dst` - Represents daylight savings time 1=yes,
0=no, and -1= default/unknown.

بعد ذلك ، نحتاج إلى الحصول على التاريخ الحالي. يمكننا فعل ذلك بهذا الخط:

```
$today = time();
```

أعتقد أنه يمكنك معرفة ما تفعله وظيفة `time()` بنفسك. :

بعد ذلك ، علينا الآن إيجاد الفرق بين الوقت الحالي والوقت المستهدف.

للقیام بذلك ، نحتاج ببساطة إلى طرح:

```
$difference =($target-$today);
```

نظرًا لأنه يتم قياس الطابع الزمني بالثواني ، نحتاج إلى تحويله إلى أي وحدة نريدها. إذا أردنا أن نقسم ساعات على ، 3600 ولكن في مثالنا سنستخدم الأيام ، لذلك نحتاج إلى القسمة على 86400(عدد الثواني في اليوم). نريد أيضًا التأكد من أن الرقم لدينا هو عدد صحيح ، لذلك نحن سيسخدم الدالة `int`.

```
$days =(int) ($difference/86400);
```

عندما نجمعها معاً ، نحصل على الكود النهائي:

```
<?php  
$target = mktime(0, 0, 0, 2, 10, 2007) ;  
$today = time () ;  
$difference =($target-$today) ;  
$days =(int) ($difference/86400) ;  
print "Our event will occur in $days days";  
?>
```

يمارس

أنشئ عدداً للعد التنازلي لحدث مهم في حياتك.

7

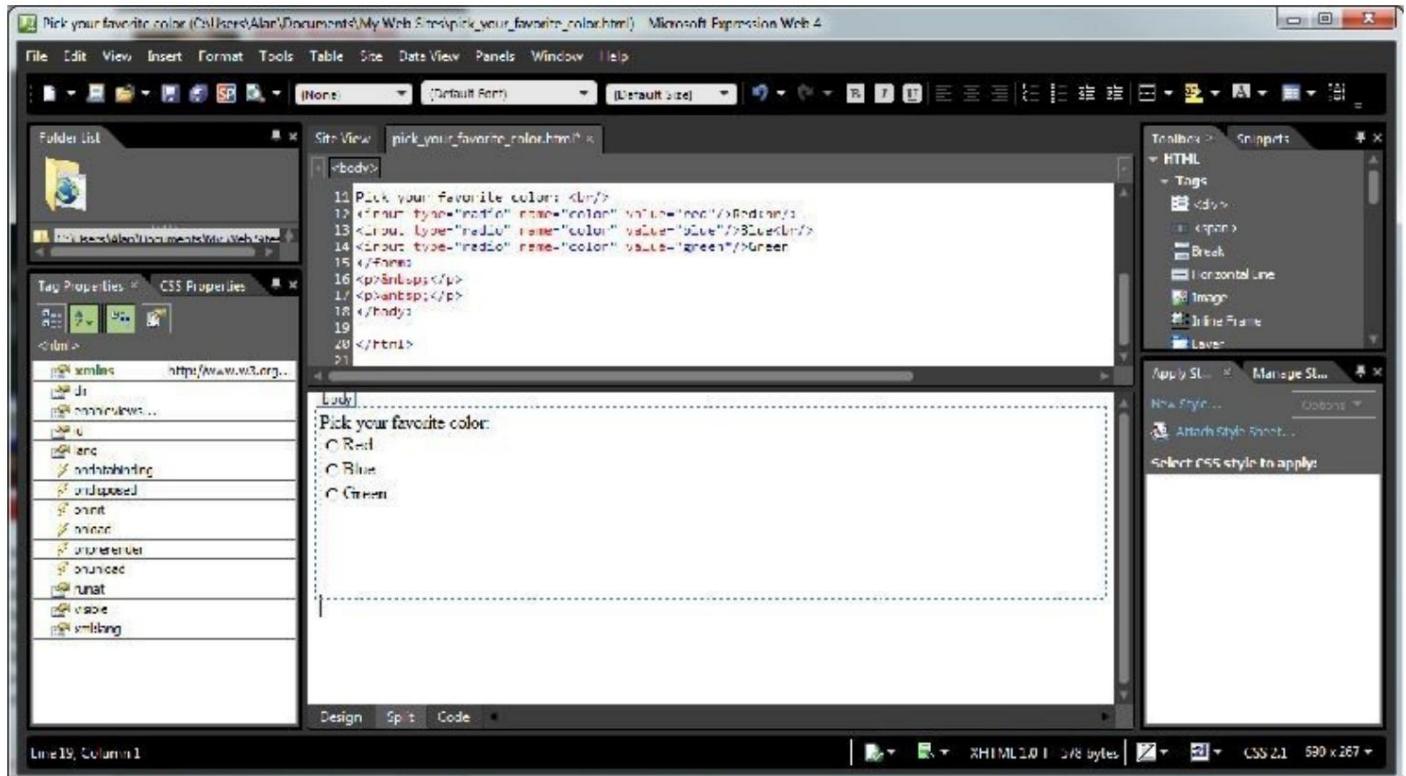
المحررين والبقاء منظمين

المحررين

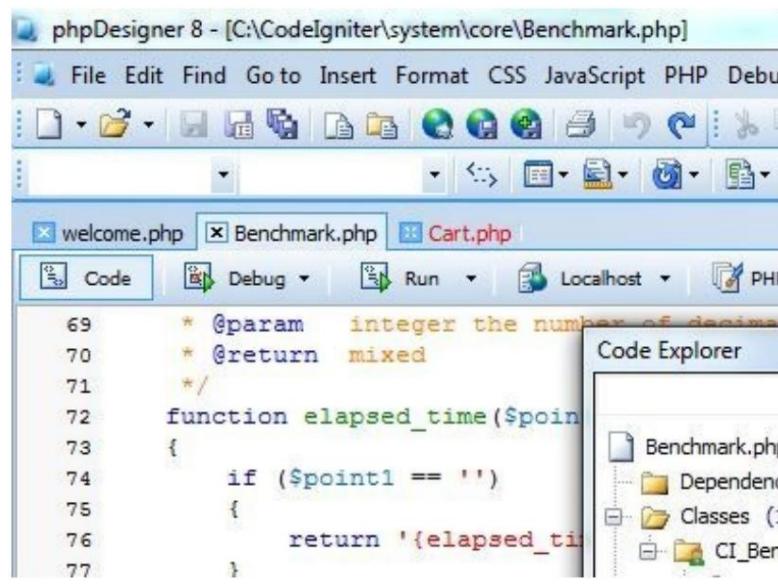
المحرر هو البرنامج الذي تستخدمه لكتابة شفرة HTML الخاصة بك والتعليمات البرمجية. على سبيل المثال ، Microsoft Word هو المحرر الذي تستخدمه لكتابة المستندات. إذا كنت ستصبح جاداً في تعلم PHP وكتابة الكثير من الأكواد ، فإن المحرر المصمم خصيصاً لـ PHP سيكون مفيداً للغاية.

أستخدم محررين مختلفين ، اعتماداً على ما أركز عليه. عندما أكتب ، فإنني في الغالب أستخدم Expression Web 4. عندما أكتب كود PHP ، أستخدم برنامجاً يسمى Microsoft phpDesigner.

ما هو لطيف في Microsoft Expression Web هو أنه يوفر شاشة مقسمة - كود HTML في الأعلى والرمز كما سيتم عرضه في الأسفل. ويمكنك إجراء تعديلات في أي من الجزأين ، ويقوم تلقائياً بتحديث الجزء الآخر.

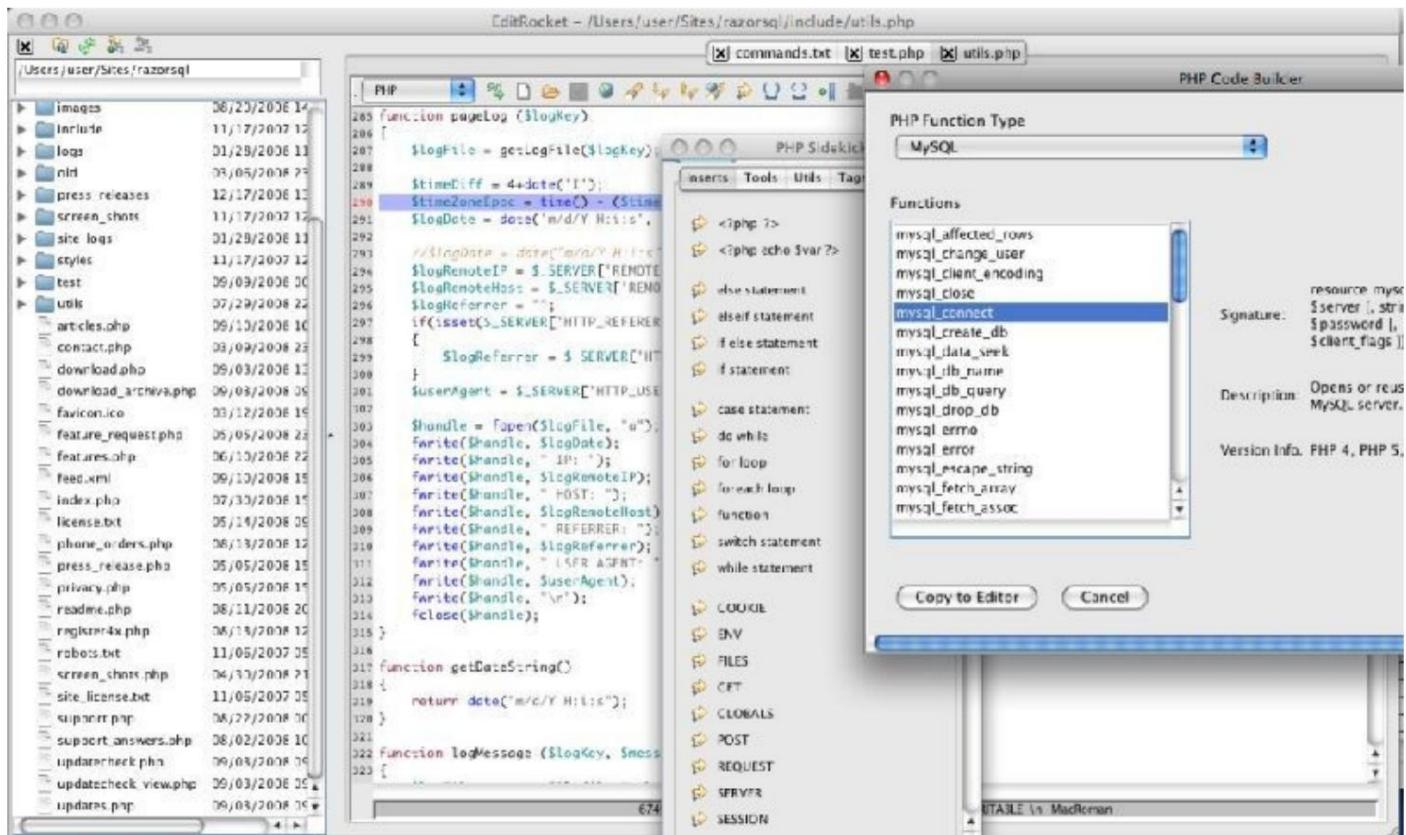


عندما أعمل باستخدام كود PHP الذي يظهر أدناه. الشيء الجميل في ذلك PhpDesigner هو أنه يرمز إلى نصوص PHP و HTML وهذا يجعل من السهل جدًا رؤية ما تفعل عليه.



صاروخ

[أولئك الذين يعملون منكم على أجهزة Mac سيحبون على الأرجح EditRocket](#)



المحررين الحرة

عندما تبدأ للتو ، من المنطقي أن تبدأ بالمحررين المجانيين. بعد ذلك ، إذا كنت تحب العمل باستخدام ، PHP فمن الأسهل كثيراً تبرير إتفاق الأموال على محرر لأنك تعلم أنك ستستخدمه ، وستكون لديك فكرة أفضل عما تبحث عنه.

[المفكرة المبرمج هو محرر جيد لجميع الأغراض. برنامج Notepad ++ هو واحد آخر.](#)

[Dev-PHP](#) هو محرر خاص بلغة PHP

[تيكت رانجل هو محرر جيد لنظام التشغيل Mac.](#)

يشمل

واحدة من أعظم ميزات PHP هي عبارة include (وعبارة "تطلب" ذات الصلة).

ما تمكّنه هذه الميزة هو أنه يمكنكأخذ أجزاء الصفحة التي يمكن تكرارها مراراً وتكراراً في صفحات متعددة ووضع تلك الأجزاء في ملف منفصل ، والذي يمكنك إدراجه في أي مكان تريده فقط باستخدام تعليمات التضمين.

يمكن أن يؤدي استخدام ملفات التضمين إلى توفير الكثير من العمل. على سبيل المثال ، إذا كان موقع الويب الخاص بك يحتوي على رأس أو تذييل قياسي ، أو إذا كان يحتوي على قائمة للتنقل ، فمن المحتمل أن تظهر هذه العناصر في كل صفحة تقريباً. مع HTML العادي ، هذا يعني الكثير من التعليمات البرمجية المكررة. لا بأس في تكرار الشفقة ، حتى تحتاج إلى تغييرها - عندها تصبح مشكلة كبيرة.

باستخدام PHP وملف التضمين ، ما عليك سوى تغيير الملف نفسه ، وسيتم تحديث كل صفحة تشير إليه تلقائياً.

include and **require** are virtually identical, except if the specified file cannot be found. **include** will continue with only a warning, but **require** will stop the script dead.

Which to use? It depends if you would *like* to include it, or if you will *require* the file to be included.

مثال أساسى

افترض أن لديك ملفاً يسمى "top.php" يحتوي على html الذي ترغب في ظهوره أعلى عدة صفحات. يمكنك تضمينه بسهولة في صفحة مثل هذه:

```
<html>
<body>
    <?php include 'top.php'; ?>
    <h1>Welcome to Al's Car Wash!</h1>
    <p>Some text.</p>
</body>
</html>
```

مثال HTML

لنفترض أن لديك قائمة قياسية يجب أن تظهر في عدة صفحات. هذا هو الكود الذي يتألف منه القائمة:

```
<a href="/default.php">Home</a>
<a href="/products.php">Products</a>
<a href="/services.php">Services</a>
<a href="/about.php">About Us</a>
<a href="/contact.php">Contact Us</a>
```

لكل صفحة تريد القائمة فيها ، ما عليك سوى تضمين النص المميز باللون الأحمر:

```
<html>
<body>
  <div class="leftmenu">
    <?php include 'menu.php'; ?>
  </div>

  <h1>Welcome to Sam's Used Cars.</h1>
  <p>We are glad that you came to visit us today.</p>

</body>
</html>
```

مثال رمز

افترض أن لدينا ملف تضمين مع تعريف بعض متغيرات ، وأن هذا الملف يسمى ("variables.php"):

```
<?php
$color='black';
$pet='dog';
?>
```

ثم يمكن استخدام المتغيرات في ملف الاستدعاء:

```
<html>
<body>

  <h1>Welcome to Sam's Used Cars.</h1>
  <?php include 'variables.php';
echo "I have a $color $pet"; // I have a black dog
?>

</body>
</html>
```

[أفضل الممارسات](#)

استخدام ملفات التصميم هو أفضل ممارسة. كلما قل الرمز الذي يجب عليك تكراره ، كان ذلك أفضل. سترى نموذج الشفرة الذي يتماشى مع هذا الكتاب يستخدم على نطاق واسع للتصميم صياغات.

8

المتغيرات والأرقام والتاريخ والسلسل

المتغيرات

المتغير هو مكان حيث يمكنك تخزين الأشياء ، مثل رقم أو تاريخ أو بعض النصوص. يمكنك وضع نص أو أرقام في متغيرات حتى تتمكن من استعادتها لاحقاً ، أو حتى تتمكن من معالجتها. تسمى المتغيرات المتغيرات لأن القيمة التي تحتوي عليها يمكن أن تختلف. ومن ثم ، فإن المتغيرات متغيرة. على سبيل المثال ، لتخزين القيمة 3.89 في متغير لتتبع سعر البنزين ، يتم تطبيق الصيغة التالية:

```
$gas_price = 3.89;
```

تخزين سعر الغاز هو استخدام جيد لمتغير لأننا نعلم أن سعر البنزين ليس ثابتاً! في يوم مختلف ، قد يحتوي المتغير على قيمة مختلفة:

```
$gas_price = 3.79;
```

على الرغم من أنه من الناحية النظرية يمكنك تسمية المتغير الخاص بك حول أي شيء تريده ، فمن الأفضل إعطاء المتغيرات الخاصة بك اسمًا يسهل على البشر فهمه.

على سبيل المثال ، في PHP سيكون من الجيد تماماً تسمية المتغيرات الخاصة بك \$a و \$b و \$c لتخزين معلومات حول لون السيارة وطرازها وسنتها ، ولن ينتج عن ذلك رمز سهل القراءة. انظر /variables-bad-form/ <http://www.joyofphp.com/>

```

<html>
  <body>
    <h1>Variables</h1>
    <?php
      echo "<br>";
      $a="Ford";
      $b="Explorer";
      $c = 2003;

      echo "This $a $b is a $c <br>";

      $gas_price = 3.89;
      echo "the price of gas is $gas_price";
    ?>
  </body>
</html>

```

على الرغم من أن الكود أعلاه قد يكون صحيحاً من الناحية التركيبية ، فمن الأفضل كثيراً أن تأخذ لحظة أو اثنين وتفكر في الغرض من المتغيرات الخاصة بك وتسميتها لغرضها.

قارن ما ورد أعلاه مع الكود التالي ، الذي له نفس إخراج الكود السابق:

```

<html>
  <body>
    <h1>Variables</h1>
    <?php
      echo "<br>";
      $brand="Ford";
      $model="Explorer";
      $year = 2003;

      echo "This $brand $model is a $year <br>";
      $gas_price = 3.89;
      echo "the price of gas is $gas_price";
    ?>
  </body>
</html>

```

انظر [الرابط](http://www.joyofphp.com/variables-good-form/) للأولاً توافق على أن النمط الثاني من البرمجة أسهل في القراءة

والمتابعة؟ في مثال صغير مثل هذا ، لا يهم كثيراً ، ولكن عندما تبدأ في كتابة رمز أطول وأكثر تعقيداً ، فإن اصطلاحات التسمية ستحدث فرقاً كبيراً.

ليس عليك التصريح عن متغير في PHP قبل استخدامه. سيوجد المتغير بمجرد تعيين قيمة له.

```
$text = "Hello World";
$num = 123;
```

لاحظ أنه عند تعيين نص لمتغير ، فإنك تضع النص بين علامتي اقتباس.

اصطلاحات التسمية المتغيرة وأفضل الممارسات

يتم تمثيل المتغيرات في PHP بعلامة الدولار متبوعة باسم المتغير. اسم المتغير حساس لحالة الأحرف ، مما يعني أن PHP ستتعامل مع السعر بالدولار والسعر \$ كمتغيرين مختلفين. لا يمكنني المبالغة في التأكيد على هذا كمصدر محتمل للارتكاب - أسماء المتغيرات حساسة لحالة الأحرف ، لذا انتبه.

تتبع أسماء المتغيرات نفس القواعد مثل التسميات الأخرى في PHP. يبدأ اسم المتغير الصالح بحرف أو شرطة سفلية ، متبوعاً بأي عدد من الأحرف أو الأرقام أو الشروط السفلية.

Note: **\$this** is a special variable that can't be assigned.

عندما يتعلق الأمر بأفضل الممارسات لتسمية المتغيرات الخاصة بك ، فهناك عدة طرق للقيام بذلك. تقترح إحدى المدارس الفكرية أن كل متغير وأول كلمة في كل متغير تبدأ بحرف كبير ، مثل \$GasPrice. يفضل البعض الآخر سعر gas_price دولاًر. لا يهم حقاً الطريقة التي تختارها ، ولكن المهم هو أنك متسق.

أعداد

أعتقد أننا جميعاً نعرف ما هي الأرقام. في PHP لا يتغير تصريح مسبقاً ما إذا كان المتغير سيحتوي على رقم أو سلسلة أو تاريخ. عندما تعلن عن متغير وتحرص له رقمًا ، تعرف PHP أنه رقم. هناك العديد من الوظائف التي يمكنك استخدامها لمعالجة الأرقام.

الحساب الأساسي

+عامل الإضافة

-عامل الطرح. يمكن أيضًا استخدامها للنفي مثل هذا 9-

*عامل الضرب

/عامل القسمة ، عامل المعامل ، يعيد الباقي بعد القسمة. على سبيل المثال ، 3 25% ستعطينا 1.

يمكنك إسناد نتيجة تعبير إلى متغير ، ويمكنك استخدام المتغيرات في التعبيرات. إذا أردت معرفة تكلفة ملء خزان بنزين ، يمكنك كتابة كود PHP على النحو التالي:

```
$price = 3.79;
$gallons = 10;
$total = $price * $gallons;
echo "The bill for this trip to the gas station is $total";
```

اختصارات حسابية شائعة

= $+ \$a$ يضيف قيمة إلى المتغير الحالي. على سبيل المثال ، يضيف $1 + \$a$ واحداً إلى المتغير $\$a$.

= $- \$a$ يطرح واحداً من المتغير $\$a$. على سبيل المثال ، $1 - \$a$ = تطرح قيمة من المتغير الحالي.

وظائف رقمية مفيدة هناك العديد من الوظائف المتعلقة بالأرقام. فيما يلي بعض من أكثرها شيوعاً وفائدة:

- تُرجع (abs) القيمة المطلقة لرقم تُرجع $(round)$ رقمًا إلى أقرب عدد صحيح تُرجع $(sqrt)$ الجذر التربيعي لرقم
- أعتقد أنك حصلت على الفكرة. إذا كنت بحاجة إلى دالة رقمية ، فإن الاحتمالات عالية جدًا أن PHP لديه ذلك.
-
-

السلسلة هي سلسلة من الأحرف ليست أرقاماً. في تفسير أبسط ، إنه نص. يمكن اعتبار أي مجموعة من الأحرف والمسافات سلسلة. هذه الجملة عبارة عن سلسلة.

في ، من المهم أن تقوم بإنشاء سلسلة محاطة بعلامات اقتباس مفردة أو علامات اقتباس مزدوجة. إذا قمت بتضمين سلسلة نصية داخل علامتي اقتباس مفردتين ، فستعيد PHP هذه السلسلة بالضبط. عندما تقوم بإحاطة سلسلة بين علامتي اقتباس ، فإن أي متغيرات داخل السلسلة سيتم استبدالها بقيمها.

```
<?php
echo 'this is a string';
echo "this is a string";
//both output the same thing
$icecream ="chocolate";
echo 'I like $icecream icecream'; // prints I like $icecream ice cream
echo "I like $icecream icecream"; // prints I like chocolate ice cream
?>
```

وظائف السلسلة المفيدة هناك مجموعة كبيرة من الوظائف في PHP يمكن استخدامها لمعالجة متغيرات السلسلة.

وهنا عدد قليل:

- يحول سلسلة نصية إلى مكافئتها في $htmlentities()$

- يحول كود HTML مرة أخرى إلى سلسلة ()
str_pad()
- يكرر سلسلة لعدد محدد من المرات ()
str_repeat()
- يستبدل بعض الأحرف في سلسلة (حساسة لحالة الأحرف) ()
str_replace()
-
-

مرة أخرى ، أعتقد أنك فهمت الفكرة. إذا كنت بحاجة إلى وظيفة سلسلة ، الاحتمالات عالية جدًا أن PHP لديه ذلك.

مصادر التوثيق

يوجد هنا مصدراً رائعاً للعثور على وظائف محددة:

عندما تزور ، يوجد مربع بحث في الزاوية اليمنى العليا من كل <http://www.w3schools.com> <http://www.php.net> صفحة. ما عليك سوى كتابة اسم الوظيفة هنا ، أو أي شيء تبحث عنه ، وسيقوم <http://www.php.net> بإرجاع قائمة [بالصفحات ذات الصلة](#).

يمارس

حاول البحث في "substr" عن [w3schools.com](http://www.w3schools.com) و [PHP.Net](http://www.php.net)

كيفية تفسير وثائق [PHP.NET](#) عند عرض صفة مرجعية وظيفية ، تحتاج إلى فهم كيفية تمثيل بناء الجملة في الوصف ، والذي سيبدو ، في حالة وظيفة السلسلة الفرعية ، كما يلي:

substr

(PHP 4, PHP 5)

substr — Return part of a string

Description

```
string substr ( string $string , int $start [, int $length ] )
```

Returns the portion of *string* specified by the *start* and *length* parameters.

قد يبدو هذا مثل هراء في البداية - لقد فعل ذلك بالتأكيد بالنسبة لي. ولكن بمجرد أن تتعلم فك تشفيرها ، ستري أنه كل ما تحتاجه لفهم كيفية استخدام هذه الوظيفة. هذه نسخة مشرورة:

substr — the function name
 (PHP 4, PHP 5) — which versions of PHP have this function
 substr — Return part of a string — What the function does

Description

The function syntax

```
string substr ( string $string , int $start [, int $length ] )
```

Returns the portion of *string* specified by the *start* and *length* parameters.

لا يكون وصف الوظيفة واضحًا دائمًا في البداية ، ولكن عندما تقرأ الأمثلة التالية ، يصبح من السهل فهمها.

يمكن التفكير بشكل أفضل في وظيفة السلسلة الفرعية على أنها "سلسلة فرعية" أو "مجموعة فرعية من سلسلة . " بشكل عام ، أسماء الوظائف هي نسخ مختصرة لما تفعله بالفعل. يتيح لك Substr استخراج جزء من سلسلة.

دعنا نتعرف على كيفية تفسير بناء جملة الوظيفة ، كما هو موضح أدناه مرة أخرى باللون الأزرق.

```
string substr ( string $string , int $start [, int $length ] )
```

من اليسار إلى اليمين:

قيمة الإرجاع

على يسار اسم الوظيفة (substr) هي القيمة المرجعة (سلسلة).

القيمة المعاادة هي ما تحصل عليه عند تشغيل الوظيفة. في هذه الحالة ، ستُرجع هذه الدالة قيمة سلسلة. قد تُرجع الدوال الأخرى أعداداً صحيحة ، ومصفوفات ، وكائنات ، وما إلى ذلك. في بعض الحالات ، ستُرى دالة تُرجع شيئاً يسمى ، booleان وهو اختصار لـ ، وهذا يعني أن الوظيفة ستعيد إما True أو False.

إذا رأيت دالة ترجع فارغاً ، فهذا يعني أنه لم يتم إرجاع أي شيء. "فارغ" يعني مختلط أن الوظيفة يمكنها إرجاع مزيج من أنواع الإرجاع مثل عدد صحيح ، سلسلة ، مصفوفة ، إلخ.

اسم وظيفة

الجزء التالي هو اسم الوظيفة نفسها. في المربع الأزرق أعلاه ، وعلى موقع ويب ، يكون اسم الوظيفة بخط غامق.

حدود

الجزء التالي هو المعلمات التي تم تمريرها إلى الوظيفة مفصولة بفواصل. في كل حالة ، هناك مثال على نوع القيمة التي يجب تمريرها. في حالة دالة ، فإن المعلمتين الأوليين هما \$string و \$start. هذا يعني أنك تمرر دالة السلسلة الفرعية سلسلة وعدد صحيح. السلسلة \$ هي السلسلة التي تريد مجموعة فرعية منها ، و \$start هي الموضع الذي تريده بدء البحث فيه.

في بعض الأحيان ستُرى قيمة محددة داخل بناء الجملة. هذا يعني أن هذه هي القيمة الافتراضية. في حالات أخرى ، تكون المعلمات بين أقواس مربعة. يشير هذا إلى أن ملف

المعلومة اختيارية ، ولا تحتاج إلى تعبيئها ، ولكن يمكنك استخدامها إذا كنت تريده ذلك.

أثناء التمرير لأسفل في الوثائق ، سترى استخدامات أكثر غموضاً لوظيفة معينة. على سبيل المثال ، في حالة ، يمكنك تمرير الأرقام السالبة إلى أي من معلمات الأرقام ، وسيتم حسابها من نهاية السلسلة ، بدلاً من البداية. مفيد؟ بالتأكيد. شائع؟ ليس كثيرا.

فيما يلي مثال بسيط لاستخدام دالة substr.

```
$myString = "Hello World";
$myNewString = substr( $myString , 0, 5 );
echo $myNewString;
```

سيؤدي هذا إلى طباعة "Hello" لأن الأحرف الخمسة الأولى من "Hello World" هي "Hello".

للح

هناك مجموعة متنوعة من وظائف التاريخ في PHP، وأكثرها وضوحاً هو التاريخ().

string date (string \$format, [int \$timestamp = time()])

Returns a string formatted according to the given format string using the given integer timestamp or the current time if no timestamp is given. In other words, timestamp is optional and defaults to the value of time().

تتيح لك وظيفة التاريخ تنسيق التاريخ بأي طريقة قد تخيلتها تقريباً. المعلومة الأولى هي سلسلة التنسيق ، والتي يمكنك إنشاؤها باستخدام أي مجموعة من الأحرف التالية:

لتنسيق الأيام

- d ترجع يوم الشهر كرمين بأصفار بادئة ، مثل 01 إلى 31.

- I إرجاع تمثيل نصي ليوم على هيئة ثلاثة أحرف ، مثل Sun إلى Mon - j. إرجاع يوم الشهر بدون أصفار بادئة ، مثل 1 إلى 31. (حرف صغير - ("L") إرجاع تمثيل نصي كامل ليوم الأسبوع ، مثل الأحد أو السبت.

- N إرجاع تمثيل رقمي ISO-8601 ليوم الأسبوع (مضاف في PHP 5.1.0) مثل 1 (ليوم الاثنين) حتى 7 (ليوم الأحد).

- S ترجع لاحقة ترتيبية إنجليزية ل يوم من الشهر على هيئة حرفين مثل st أو nd أو rd. يمكنك استخدام هذا جنباً إلى جنب مع (أعلاه) إذا كنت تريده إنشاء شيء مثل الأول من يناير.

- W إرجاع تمثيل رقمي ل يوم الأسبوع ، مثل 0 (ليوم الأحد)

حتى 6(لليوم السبت). - إرجاع يوم من أيام السنة (بدءاً من 00 مثل 0 إلى 365

لتنسيق الأسابيع

- Wترجع رقم أسبوع ISO-8601 من السنة ، الأسابيع التي تبدأ يوم الاثنين (تمت إضافتها في PHP 4.1.0 مثل 32 للأسبوع الثاني والثلاثين في السنة).

لتنسيق الشهور

- Fإرجاع تمثيل نصي كامل لشهر ، مثل يناير أو مارس.

- mإرجاع تمثيل رقمي لشهر بأصفار بادئة مثل 01 أو 12.

- Mترجع تمثيلاً نصياً قصيراً لشهر على هيئة ثلاثة أحرف ، مثل Jan أو Dec. n -

- tإرجاع عدد الأيام في الشهر المحدد ، مثل 28 أو 29 أو 30 أو 31.

لتنسيق السنوات

- Lترجع ما إذا كانت سنة كبيسة مثل 1 إذا كانت سنة كبيسة ، 0 بخلاف ذلك. - 0إرجاع رقم سنة ISO-8601 مكون من 4أرقام. هذا له نفس قيمة ، 7 باستثناء

أنه إذا كان رقم أسبوع (W) ISO يتبع إلى العام السابق أو التالي ، فسيتم استخدام تلك السنة بدلاً من ذلك. تمت إضافة هذا الخيار في PHP 5.1.0.

- Zإرجاع تمثيل رقمي كامل لسنة ، مرة أخرى باستخدام 4أرقام.

- utعرض تمثيلاً مكوناً من رقمين للسنة.

لتنسيق الوقت

أ - إرجاع الأحرف الصغيرة am أو pm.

- Aإرجاع الأحرف الكبيرة AM أو PM.

- و إرجاع تنسيق 12 ساعة للساعة بدون أصفار بادئة.

- Gإرجاع تنسيق 24 ساعة للساعة بدون أصفار بادئة ، من 0 إلى 23. h - إرجاع تنسيق 12 ساعة للساعة بأصفار بادئة ، مثل 01 إلى 12.

- Hإرجاع تنسيق 24 ساعة للساعة بأصفار بادئة ، مثل 00 إلى 23.

- إرجاع الدقائق بأصفار بادئة ، مثل 00 إلى 59.

- sإرجاع الثواني ، بأصفار بادئة ، مثل 00 إلى 59.

لتنسيق المناطق الزمنية

- eيعرض معرف المنطقة الزمنية (تمت إضافته في PHP 5.1.0) مثل C و UTC و GMT و Azores. و Atlantic /

ـ I(كبير) - (ترجع ما إذا كان التاريخ في التوقيت الصيفي أم لا مثل 1 إذا كان التوقيت الصيفي ، و 0 بخلاف ذلك.

- Oترجع الفرق إلى توقيت غرينتش (GMT) بالساعات ، مثل +0200.

- تُرجع الفرق إلى توقيت غرينتش (GMT) بنقطتين بين الساعات والدقائق (تمت إضافتها في ، مثل .00: 02: PHP 5.1.3).

- يعرض اختصار المنطقة الزمنية ، مثل ، MDT ، CET ، EST ، الخ.

لتنسيق التاريخ / الوقت الكامل

- إرجاع تاريخ ISO 8601 مضاد في (5 مضاد مثل PHP 2013-02-12T15: 19: 21 + 00: 00. r - مثل RFC 2822).

الخميس ، 21 ديسمبر 2000 16:01:07 +0200

- تُرجع عدد الثواني التي انقضت منذ وقت ، Unix Epoch وهو 1 يناير 1970 00:00:00 بتوقيت جرينتش.

ستتم طباعة الأحرف غير المعروفة في سلسلة التنسيق كما هي. سيعود تنسيق Z دائمًا 0 عند استخدام () .

مثال: حساب العمر بالسنوات بافتراض أنه كجزء من إجراء التسجيل الذي طبعت عيد ميلاد ، ستعيد هذه الوظيفة أعمارهم بالسنوات.

```
<?php

//calculate years of age (input string: YYYY-MM-DD)
function age($age){
    list($year,$month,$day) = explode("-",$age);
    $year_diff = date("Y") - $year;
    $month_diff = date("m") - $month;
    $day_diff = date("d") - $day;
    if ($day_diff < 0 || $month_diff < 0)
        $year_diff--;
    return $year_diff;
}

?>
```

انظر المثال النصي AgeInYears.php

نطاق متغير

يحدد نطاق المتغير مكان الوصول إلى القيمة. إذا تم الإعلان عن متغير في السطر الخاص به على الصفحة ، فسيكون متاحًا في أي مكان على تلك الصفحة. إذا تم التصريح عن متغير داخل دالة ، فسيكون متاحًا فقط داخل هذه الوظيفة.

إذا كنت تريد أن يكون متغيرًا معيناً متاحًا في كل مكان ، فقم بالإعلان عنه باستخدام الكلمة الأساسية العامة ، مثل ؟! global \$ a = 'Hello'!

هناك نوع خاص من المتغيرات التي يمكن الوصول إليها في كل صفحة تشكل تطبيق الويب الخاص بك. تمت تفطية هذا الموضوع في متغيرات الجلسة.

9

جمل التحكم

مقدمة

بيت القصيد من PHP هو جعل صفحة الويب ديناميكية. ديناميكي يعني أنك لا تحصل بالضرورة على نفس النتيجة عند عرض نفس الصفحة من وقت آخر. تعرض صفحة الويب في بعض الأحيان مجموعة نصية واحدة ، وفي أحيان أخرى تعرض مجموعة أخرى. تسمح لك هيكل التحكم بالتحكم في الشروط التي تحدد القواعد التي تحدد متى وكيف يحدث ذلك.

لـ

تعد عبارة `if` واحدة من أهم ميزات العديد من اللغات ، بما في ذلك PHP. يسمح بالتنفيذ المشروط لأجزاء الكود. في ، الصيغة البسيطة لتعليمية `if` هي كما يلي

```
if (expression)
statement
```

يتم تقييم التعبير إلى قيمته المنطقية. إذا تم تقييم التعبير إلى ، ، فإنها تُعرف بالعبارة ، وإذا تم تقييمها إلى - .
تجاهلهما `TRUE` `FALSE` .

A **Boolean** value is one that can be reduced to one of two values: True or False

إذا كنت ترغب في تنفيذ عبارات متعددة إذا كان الشرط صحيحًا ، فقم بتحجيم جميع العبارات داخل كتلة التعليمات البرمجية باستخدام الأقواس المترجة ، كما هو موضح

```
if (expression)
{
    statement 1;
    statement 2;
}
```

يعرض المثال التالي "إذا كان `a` أكبر من `b`"
أكبر من `a`:

```
<?php
if ($a > $b)
    echo "a is bigger than b";
?>
```

[إذا كان غير ذلك](#)

غالباً ما ت يريد تنفيذ عبارة إذا تم استيفاء شرط معين ، وبيان مختلف إذا لم يتم استيفاء الشرط . هذا هو الغرض من عبارة - else لتحديد الإجراء إذا لم يتم استيفاء الشرط.

بشكل أكثر رسمية ، تقوم else بتمديد عبارة if لتنفيذ عبارة في حالة تقييم التعبير في عبارة if إلى FALSE.

على سبيل المثال ، إذا كان اليوم هو الاثنين ، فسيتم إغلاق المتجر. إلا فهو مفتوح من الساعة 10 صباحاً حتى 9 مساءً. الدالة date_default_timezone_set يحدد المنطقة الزمنية التي سيتم استخدامها.

```
<html>
    <body>
        <h1>Open Hours</h1>
        <?php
            date_default_timezone_set ( "EST" );
            if ( date("l")=='Sunday' ) {
                echo "Sorry, we are closed today.";
            } else {
                echo "We are open today from 10 AM to 9 PM";
            }
        ?>
    </body>
</html>
```

تمرين: ساعات مفتوحة

لنفترض أنك تدير متجرًا من نوع ما (صالة ألعاب رياضية ، ومتجر دونات ، ومنصة مزرعة ، و موقف سيارات مستعملة ، وما إلى ذلك) ، وتختلف ساعات فتحه كل يوم. أيام السبت والأحد هي أكثر الأيام ازدحامًا ، لذا فهي مفتوحة من 9 صباحاً إلى 9 مساءً. يوم الإثنين هو يوم إجازتك ، وبقية أيام الأسبوع ، تكون الساعات من 10 صباحاً إلى 6 مساءً ، باستثناء الصيف (يوليو وأغسطس) عندما تظل مفتوحة حتى الساعة 7 مساءً.

ترغب في أن تعرض الصفحة الرئيسية لموقع الويب الخاص بك بشكل بارز ساعات اليوم ، على غرار لقطة الشاشة أدناه:



PHP مفتاح

بيان التبديل يشبه سلسلة من عبارات IF على نفس التعبير. في العديد من المناسبات ، قد ترغب في مقارنة نفس المتغير (أو التعبير) بالعديد من القيم المختلفة ، وتنفيذ جزء مختلف من التعليمات البرمجية بناءً على القيمة التي تساويها. هذا هو الغرض من بيان التبديل.

```
<?php
switch ($make) {
    case "Ford":
        echo "Your car is a Ford";
        break;
    case "Chevrolet":
        echo "Your car is a Chevrolet";
        break;
    case "Toyota":
        echo "Your car is a Toyota";
        break;
}
?>
```

حلقات PHP

بينما

حلقة while هي واحدة من أبسط أنواع الحلقات في PHP في الأساس ، تستمرة في فعل شيء ما طالما أن الشرط صحيح.

الصيغة الأساسية لحلقة while هي كما يلي:

```
while (expression){
    do something;
    do something else;
}
```

هذا مثال:

```
<?php
$i = 1;
while ($i <= 10):
    echo $i;
    $i++;
endwhile;
?>
```

سيطبع هذا الرمز الأرقام من 1 إلى 10. الاستمرار الشائع لحلقة while هو الاستمرار في طباعة شيء ما طالما كان لديك سجلات قاعدة بيانات للمعالجة.

حلقات for أكثر تعقيداً بعض الشيء. صيغة حلقة for هي كما يلي:

```
for (expression1; expression2; expression3)
{
    statement;
    statement;
}
```

يتم تنفيذ التعبير الأول (التعبير 1 مرة واحدة ، مهما حدث ، في بداية الحلقة).

في بداية كل تكرار خلال الحلقة ، يتم تقييم التعبير 2. إذا تم تقييم التعبير 2 إلى TRUE ، تستمرة الحلقة و يتم تنفيذ العبارة (العبارات) المتداخلة. إذا تم التقييم إلى FALSE ، يتنهى تنفيذ الحلقة.

في نهاية كل تكرار ، أي بعد تنفيذ جميع العبارات ، يتم تنفيذ التعبير 3.

ضع في اعتبارك المثال التالي ، الذي يعرض أيضاً الأرقام من 1 إلى 10:

```
for ($i = 1; $i <= 10; $i++) {
    echo $i.'  
';
```

وإليك كيف يعمل. يعين \$i قيمة Expression1. يحدث هذا بغض النظر عن أي شيء. تختبر \$i قيمة Expression2 لمعرفة ما إذا كان \$i أقل من 10. نظراً لأن \$i أقل من 10 ، تنفذ العبارات التالية. إذا لم يكن \$i أقل من 10 ، فلن يتم تنفيذ أي من العبارات.

في هذه الحالة يوجد بيان واحد فقط echo \$i. '
' . وهو بعد تنفيذ العبارة (العبارات) ، يتم تنفيذ التعبير 3 ، الذي يزيد \$i بواحد. الآن \$i يساوي 2 وهو أقل من 10 لذا يتعدد صدى ، 2 وهكذا.

10

كيفية استخدام قاعدة بيانات ، مثل mySQL

مقدمة

من المؤكد أن فكرة صفحات الويب الديناميكية رائعة ، ولكن يمكنك فقط الذهاب بعيداً مع ما هو مضمن في ، مثل تغيير PHP الصفحة بناءً على يوم الأسبوع. ما تريده حقاً هو جعل صفحة الويب فريدة لكل زائر ، وهنا يأتي دور قواعد البيانات.

سنبدأ هذا الفصل بافتراض أن القارئ ليس لديه أي معرفة على الإطلاق بـ MySQL أو قواعد البيانات. أولاً ، سنشرح قواعد البيانات ، ثم سننشئ واحدة بالطريقة السهلة - باستخدام phpMyAdmin. ثالثاً ، سنغطي كيفية إنشاء قواعد البيانات والجداول باستخدام SQL وفي الفصل التالي سنبين كيف يمكن القيام بكل هذا باستخدام PHP.

ما هي قواعد البيانات؟

لنبدأ برنامجنا التعليمي بمقدمة لموضوع الاختبار ، سام ، الذي يدير وكالة لبيع السيارات المستعملة. عندما بدأ عمله لأول مرة ، لم يكن لديه سوى عدد قليل من السيارات - لذا كان تتبعها أمرًا سهلاً للغاية. ولكن بعد فترة ، بدأت تجارته في النمو. سرعان ما كان لديه 10 سيارات في أرضه ، وبعد عام كان لديه 25 سيارة.

كل سيارة لديها عدد من السمات الفريدة لتبينها ، مثل الطراز ، والطراز ، واللون ، والسنة ، ورقم التعريف الشخصي ، وعدد الركاب ، وشكل الجسم ، و MPG ، وتكلفة الاستحواذ ، وسعر الطلب ، وما إلى ذلك ، كما يمكنك أن تخيل ، في مرحلة ما يكون الإنسان فقط لا يمكنه تتبع كل هذه المعلومات ، وحتى إذا تمكنت من الاحتفاظ بها في نصاتها الصحيح ، فإنه يحتاج أيضًا إلى نقل هذه المعلومات إلى مندوبي المبيعات ، الذين لا يتمتعون دائمًا بنفس الشغف تجاه أعمال Sam التي يتمتع بها. يحتاج سام إلى القدرة على طباعة "ورقة الغش" بسرعة لكل سيارة يمكن أن يشير إليها موظف المبيعات عندما يتوجه أحد العملاء في ساحة انتظار السيارات.

وبالطبع ، يود سام حقاً أن يكون لديه موقع ويب يسمح للأشخاص بالبحث عن نوع السيارة التي يريدون معرفة ما إذا كان لديه أي مرشح ، وتصفح جميع السيارات المتاحة لديه - وهو بالطبع تغيير دائمًا.

نظرًا لأن مخزون السيارات يتغير دائمًا ، فلن يكون موقع الويب الثابت هو الحل. الحل هو موقع ويب قائم على قاعدة بيانات.

تساعد قواعد البيانات في تنظيم وتعقب الأشياء. تسمح لك قواعد البيانات باستخدام الإبداع لتجميع الأشياء معًا بطرق ذات مغزى ، وتقديم نفس مجموعة المعلومات بطرق مختلفة إلى جماهير مختلفة.

"Databases" are simply an organized collection of data stored in a computer.

ت تكون قواعد البيانات من "جدول" واحد أو أكثر. تتكون الجداول من أجزاء تسمى "صفوف" و "أعمدة" مماثلة لما قد تراه في جدول البيانات. يعلن قسم الأعمدة في كل جدول عن خصائص كل جدول بينما يحتوي كل صف على بيانات فريدة لكل عنصر في الجدول.

قد يبدو الأمر معقداً ولكنه في الواقع بسيط للغاية. خذ المثال أدناه ، وهي إحدى الطرق التي يمكن أن يبدأ بها Sam في تنظيم مجموعة سياراته. (لاحظ أنه للإيجاز ، لا يتم عرض جميع سمات السيارة الممكنة).

الجدول: سيارات

| ID | VIN | Make | Model | Style | Year | Price |
|----|------------|--------|----------|-----------|------|-------|
| 1 | 1328237824 | Ford | Explorer | SUV | 2005 | 5995 |
| 2 | 4797834923 | Dodge | RAM | Pickup | 2008 | 7200 |
| 3 | 2394923724 | Mazda | 6 | Passenger | 2010 | 9995 |
| 4 | 2342323634 | Subaru | Outback | Passenger | 2007 | 4500 |

يمكننا أن نرى بوضوح أن العناصر الموجودة في هذا الجدول تحتوي على سبعة أعمدة مُعرفة على أنها ID و VIN و Make و Model و Style و Year و Price. يحتوي الجدول على أربعة صنفون تصف أربع سيارات مختلفة Ford Explorer و Dodge RAM و Mazda 6 و Dodge Outback. Subaru

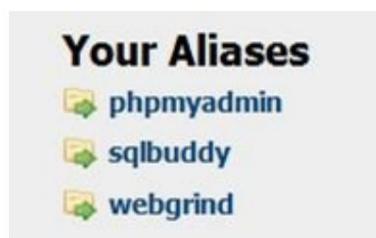
فيما يلي مراجعة سريعة لما تعلمناه.

- الجداول هي مجرد مجموعة من الأشياء التي تريد تتبعها.
- تكون الجداول من صنفون وأعمدة.
- تحتوي الأعمدة على سمات مختلفة لكل عنصر في هذا الجدول. تحتوي الصنفون في الجدول على مثيلات مختلفة محددة بشكل فريد بواسطة أعمدة الجدول.
- قواعد البيانات عبارة عن مجموعة من الجداول.

الشروع في العمل مع phpMyAdmin

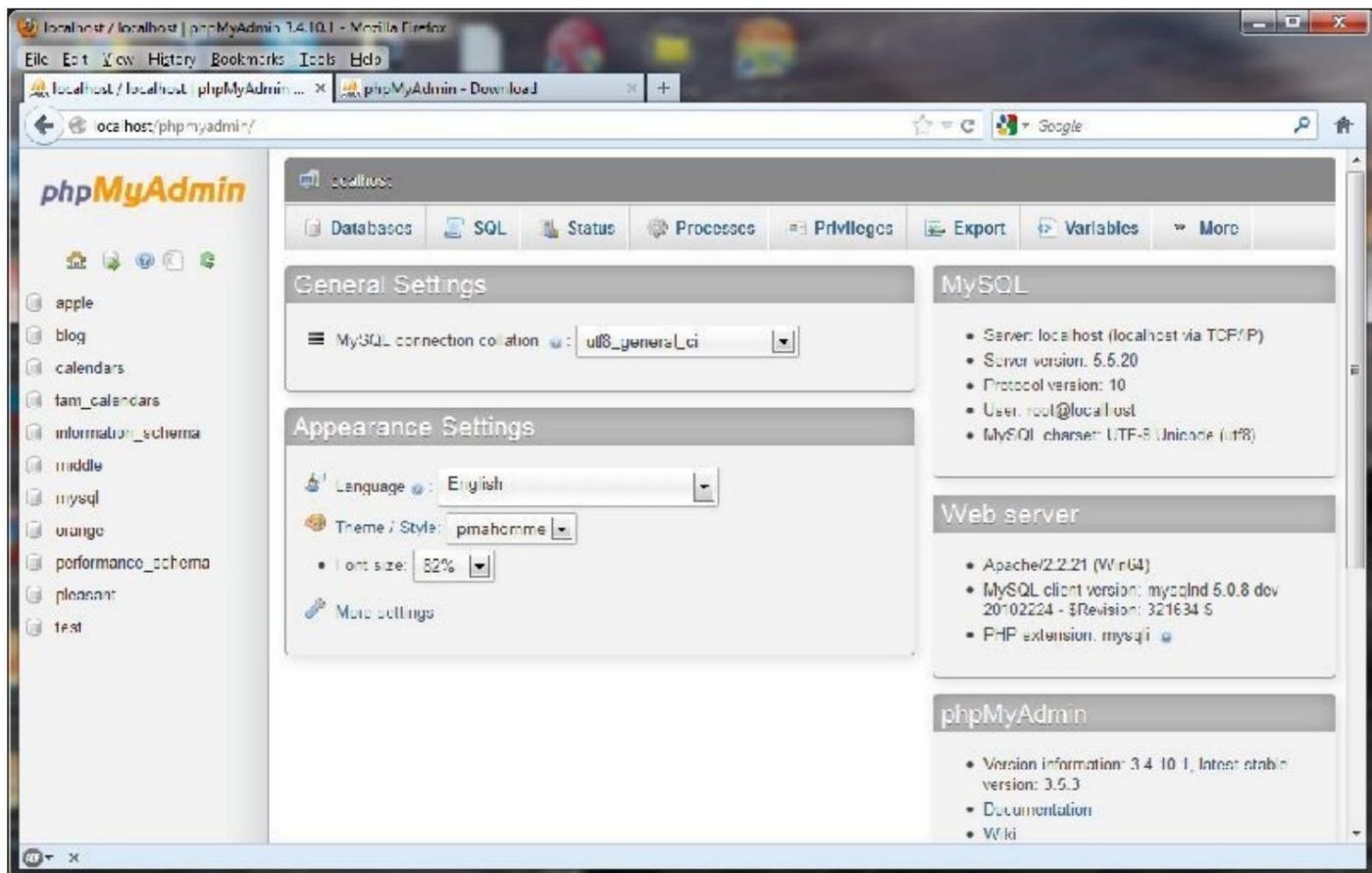
تذكر من القسم كيف أعرف أنه يعمل في الصفحة 23 أنه إذا انتقلت إلى http://localhost:23 أنه إذا انتقلت إلى localhost صفة تم إنشاؤها بواسطة [الخادم المحلي](#) WAMPSERVER بك. في حالتي هو.

في الجزء السفلي من الصفحة ، ستري رابطاً للأسماء المستعارة الخاصة بك



لست متأكداً مما إذا كانت أجهزة Mac ترى هذا ، ولكن إذا كنت لا ترى الرابط ، فقط أدخل هذا العنوان في متصفحك أو انقر فوق هذا الرابط: http://localhost / phpmyadmin /

سيؤدي النقر فوق هذا الرابط إلى نقلك إلى صفحة مشابهة لهذه:



ما هو phpMyAdmin؟

إن phpMyAdmin هي أداة برمجية مجانية - وقد تمت كتابتها في PHP - وهي مصممة للتعامل مع العديد من المهام الإدارية الشائعة في MySQL باستخدام متصفح. يدعم phpMyAdmin مجموعة واسعة من العمليات باستخدام MySQL. يتم دعم العمليات الأكثر استخداماً من خلال واجهة المستخدم (إدارة قواعد البيانات والجداول والحقول وال العلاقات والفهارس والمستخدمين والأذونات وما إلى ذلك) ، ولا يزال لديك القدرة على تنفيذ جملة SQL مباشرة إذا كنت تفضل ذلك. يأتي phpMyAdmin مزوداً بمazon جيد من الوثائق ، والمستخدمون مدعاوون لتحديث صفحات الوiki لمشاركة الأفكار والتعليقات. سيحاول فريق phpMyAdmin مساعدتك إذا واجهت أي مشكلة ، لكنني لم أواجه أي مشكلة معها شخصياً.

ما يفعله ، إنه يعمل بشكل جيد.

استخدام phpMyAdmin لإنشاء قاعدة بيانات

Databases

Create new database

Cars

Collation

Create

انتقل أولاً بحيث يكون لديك phpMyAdmin على الشاشة. انقر فوق الارتباط الذي يشير إلى قواعد البيانات: في المربع الذي يشير إلى إنشاء قاعدة بيانات جديدة ، اكتب كلمة "سيارات" ، ثم انقر فوق الزر "إنشاء". إذا كان يعمل بشكل صحيح ، فسترى مربع تأكيد أصفر يظهر على الشاشة لفترة وجيزة ، على النحو التالي:

Databases



مقدمة في SQL

هذا يعادل إصدار أمر SQL

`CREATE DATABASE Cars;`

وفي الواقع ، نفذ SQL أمر phpMyAdmin بالضبط في الخلفية من أجلك عندما تنظر على الزر. بمعنى آخر ، يمكنك التفكير في كأداة تقوم ببناء أوامر SQL نيابة عنك.

SQL (pronounced "sequel") or Structured Query Language) is a language all its own. SQL is a special-purpose programming language designed for managing data in relational database management systems, such as mySQL. SQL can be used to create databases, create tables, and insert, update, and delete data into tables.

استخدام phpMyAdmin لإنشاء جدول في قاعدة بيانات الآن بعد أن تم إنشاء قاعدة البيانات ، نود استخدامها. ابحث عن قاعدة بيانات السيارات في قائمة قواعد البيانات ، ثم اضغط على اسم قاعدة البيانات.

Databases

Create new database Collation

| <input type="checkbox"/> Database | |
|--|---|
| <input type="checkbox"/> apple | <input type="button" value="Check Privileges"/> |
| <input type="checkbox"/> blog | <input type="button" value="Check Privileges"/> |
| <input type="checkbox"/> calendars | <input type="button" value="Check Privileges"/> |
| <input checked="" type="checkbox"/> cars | <input type="button" value="Check Privileges"/> |
| <input type="checkbox"/> fam | <input type="button" value="Jump to database"/> <input type="button" value="Check Privileges"/> |

سيوفر phpMyAdmin صفحة مشابهة لـ:

localhost/phpmyadmin/

phpMyAdmin

localhost > cars

No tables found in database

Create table on database cars

Name: Number of columns:

هذا يعادل أمر SQL:

USE cars;

هذا يخبر قاعدة بيانات MySQL أنك ستعمل في قاعدة البيانات *سيارات* حتى تقول غير ذلك.

لقد أنشأت للتو قاعدة البيانات الخاصة بمجموعة السيارات المستعملة الخيالية. سنقوم بتطوير قاعدة البيانات هذه أكثر كلما تقدمنا.

تحديد جدولنا الأول

لقد قمت حتى الآن بإنشاء قاعدة البيانات الخاصة بك ، واكتشفت البنية العامة لـ IPHPMyAdmin. ستحتاج إلى وضع جدول داخل قاعدة البيانات التي قمت بإنشائها. في حالة قاعدة بيانات السيارات الخاصة بنا ، سنحتاج إلى تحديد الجدول لوصف السيارات والشاحنات التي يعرضها سام للبيع في ساحة سيارته المستعملة.

هذه لقطة شاشة أخذتها من [التي تصف السيارة www.cars.com](http://www.cars.com)

Features:

| | |
|--|--|
| Price: 8995 | Make: Ford |
| Model: Explorer Eddie Bauer | Year: 2004 |
| Mileage: 90832 | Condition: Used |
| Exterior Color: Red Fire Metallic | Interior Color: Med Parchment |
| Body Style: Sport Utility | Transmission: 5-SPEED AUTOMATIC TRANSMISSION W/OD |
| Stock Number: cb232 | Vin Number: 1FMZU74K94UB86311 |

قبل إنشاء الجدول الخاص بك ، فكر فيما ستضعه في الجدول وما هي السمات المختلفة التي قد تميز صفاً واحداً (سيارة) عن آخر.

ما الذي يميز السيارة؟

يمكنني التفكير في عدد من الخصائص أو السمات التي تميز سيارة عن أخرى في ساحة انتظار سيارات مستعملة.

- رقم هوية المركبة (VIN)
- سنة
- يصنع
- نموذج
- تقليل
- اللون الخارجي
- اللون الداخلي
- **السؤال عن السعر**
- سعر الشراء
- عدد الأميال
- الانتقال
- تاريخ الشراء
- تاريخ البيع
- سعر البيع

يجب أن يكون ذلك كافياً على الأقل لنبدأ. الآن علينا معرفة نوع البيانات التي سنضعها في هذه الفئات.

أنواع البيانات

لأغراض التعلم ، هناك ثلاثة أنواع فقط من البيانات التي ستحتاج إلى استخدامها.

هم:

1.أرقام.

2.الشخصيات.

3.التمر.

أعداد

الأرقام ، كما قد يعطيها الاسم ، هي أي نوع من المعلومات الرقمية. سوف

تحتاج إلى استخدام أي نوع من الكسور العشرية للبيانات التي ستقوم بتخزينها؟ في هذه الحالة ، ستحتاج إلى استخدام نوع البيانات العشري أو عدد عشري. إذا لم يكن الأمر كذلك ، يمكنك استخدام نوع البيانات int (اختصار لعدد صحيح) أو bigint (عدد صحيح كبير - الذي يشغل مساحة أكبر ، لكن يمكنه التعامل مع أعداد أكبر).

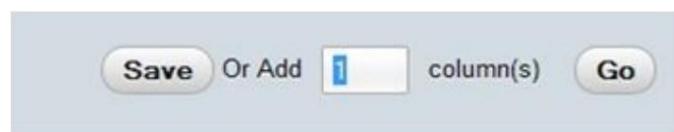
الشخصيات

نوع الحرف في MySQL هو نوع البيانات الذي تستخدمه لتخزين السلسل النصية. تُستخدم الأحرف لتخزين تمثيل حرف أو كلمة أو سلسلة من الكلمات. على سبيل المثال ، سيكون كل من الحرف A وعبارة "Hello World" من نوع الحرف. تسمى MySQL هذا بـ VARCHAR اختصاراً للأحرف المتغيرة، إنه متغير لأنك تقوم فقط بتعيين الحد الأقصى لعدد الأحرف التي يمكن أن يحتفظ بها الحقل ، وإذا أدخلت قيمة تحتوي على عدد أقل من الأحرف ، فسيتم تخزين القيمة الأقصر. تقدم قواعد البيانات الأخرى ، مثل Microsoft SQL Server ، نوع CHAR ، وهو الذي يسمى بأي حرف غير مستخدمة بمسافات. لماذا يريد أي شخص أن لا تخيل ، لذلك من أجل البساطة سنلتزم بـ VARCHAR في الوقت الحالي.

استخدم نوع البيانات varchar(n) لتحديد العمود الذي ترغب في تمثيله بحرف. استبدل n بأكبر عدد من الأحرف يمكن أن يحتوي عليه عمود في الجدول (حتى 255). يتم حساب المسافات وعلامات الجدولة والأسطر الجديدة في فقرة كأحرف.

بلح

التاريخ هي طريقة لتخزين التواريخ في قاعدة البيانات. هل تريد فقط تخزين التاريخ وليس الوقت؟ استخدم تاريخ نوع البيانات. هل تريد تخزين الوقت وليس التاريخ؟ استخدم وقت نوع البيانات. تريد تخزين التاريخ والوقت؟ استخدم نوع البيانات والوقت.



دعونا نلقي نظرة على خصائص السيارات لدينا لتحديد نوع البيانات التي ينبغي أن تكون.

- رقم معرف السيارة - (VIN) تحتوي جميع المركبات التي تسير على الطريق على رقم تعريف شخصي مكون من 17 حرفاً ، والذي لا يتضمن الأحرف (i) أو (o) أو (Q) لتجنب الالتباس مع الأرقام 1 و 0 . فارتشار (17)
- السنة - تكون من أرقام بدون فاصلة عشرية. - Int Make يتكون من نص. فارتشار (25)
- النموذج - يتكون من النص والعدد العربي. فارتشار (25)
- Trim - يتكون من نص. فارتشار (25)
- اللون الخارجي - يتكون من نص. فارتشار (25)
- اللون الداخلي - يتكون من نص. فارتشار (25)
- السعر المطلوب - يتكون من أرقام ذات فاصلة عشرية. سعر الشراء العشري - يتكون من أرقام بعلامة عشرية. عدد الأميال العشري - يتكون من أرقام بدون فاصلة عشرية. - Int Transmission يتكون من نص. فارتشار (25)
- تاريخ (اقتناء) الشراء - تاريخ البيع - التاريخ

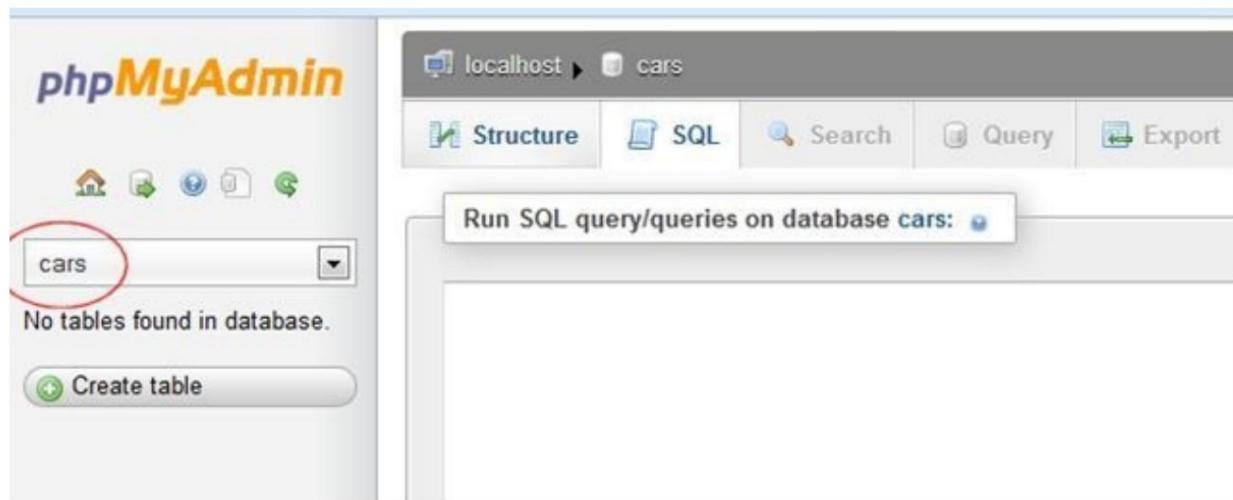
■ سعر البيع - يتكون من أرقام بدون فاصلة عشرية. كثافة العمليات

هذا يلخص الجدول الذي نحتاج إلى إنشائه لتبني سياراتنا. نظراً لأن VIN هو العنصر الوحيد الفريد حقاً في القائمة ، فسنجعله "المفتاح الأساسي".

يعني تعريف العمود كمفتاح أساسي أن العمود سيكون قادراً فقط على الحصول على قيمة فريدة (أي لا يمكن لأي شيء أن يكرر نفسه). في حالة هذا الجدول المحدد ، فهذا يعني أنه لا يمكنك إدخال سيارتين بنفس رقم تعيين المركبة في قاعدة البيانات ، لأننا أخبرنا للتو mySQL أن هذا غير مسموح به. بعض الأمثلة على ذلك في الحياة اليومية هي أرقام لوحات السيارات وأرقام بطاقات الائتمان وأرقام الضمان الاجتماعي. من المفترض أن تكون كل هذه الأرقام فريدة لكل شخص. نفس المفهوم ينطبق على الجداول في قواعد البيانات.

كلما كان ذلك ممكناً ، من الجيد التأكد من أن الجدول الذي تقوم بإنشائه يحتوي على شكل من أشكال المفتاح الأساسي لإعطاء شيء ما لتعريف الصفة بشكل فريد.

كيف يمكنني عمل جدول بهذه المعلومات؟ سؤال رائع. على الرغم من أنها أنشأنا قاعدة البيانات باستخدام معالج ، فمن الآن فصاعداً سنستخدم SQL.



في نافذتك باستخدام phpMyAdmin ، تأكد من تحديد جدول السيارات (شاهد حفاظ بدائرة باللون الأحمر أدناه) ، ثم انقر فوق علامة تبويب SQL لإظهار مربع الأوامر. تأكد من رؤية المضيف المحلي <-السيارات فوق الصندوق. إذا لم تقم بذلك ، فما عليك سوى النقر فوق ارتباط السيارات على الجانب الأيمن ثم علامة التبويب SQL للوصول إلى هناك.

اكتب الأمر التالي في المربع وانقر فوق انتقال.

```
CREATE TABLE INVENTORY ( VIN varchar(17) PRIMARY KEY, YEAR INT, Make varchar(50), Model varchar(100), TRIM varchar(50), EXT_COLOR varchar (50), INT_COLOR varchar (50), ASKING_PRICE DECIMAL (10,2), SALE_PRICE DECIMAL (10,2), PURCHASE_PRICE DECIMAL (10,2), MILEAGE int, TRANSMISSION varchar (50), PURCHASE_DATE DATE, SALE_DATE DATE)
```

تهانينا! لقد قمت بإنشاء جدول المخزون.

Your SQL query has been executed successfully (Query took 0.0099 sec)

```
CREATE TABLE INVENTORY(
VIN VARCHAR(17) PRIMARY KEY,
YEAR INT,
Make VARCHAR(50),
Model VARCHAR(100),
TRIM VARCHAR(50),
EXT_COLOR VARCHAR(50),
INT_COLOR VARCHAR(50),
ASKING_PRICE DECIMAL(10,2),
SALE_PRICE DECIMAL(10,2),
PURCHASE_PRICE DECIMAL(10,2),
MILEAGE int,
TRANSMISSION varchar(50),
PURCHASE_DATE DATE,
SALE_DATE DATE);
```

[Edit] [Create PHP Code]

Run SQL query/queries on database cars:

إليك نصيحة مفيدة بشكل لا يصدق: انقر على الرابط "إنشاء كود PHP" الموجود على الجانب الأيمن من الشاشة وما ستعود إليه هو:

```
$sql = " CREATE TABLE INVENTORY ( VIN varchar(17)
PRIMARY KEY, YEAR INT, Make varchar(50), Model
varchar(100), TRIM varchar(50), EXT_COLOR varchar
(50), INT_COLOR varchar (50), ASKING_PRICE DECIMAL
(10,2), SALE_PRICE DECIMAL (10,2), PURCHASE_PRICE
DECIMAL (10,2), MILEAGE int, TRANSMISSION varchar
(50), PURCHASE_DATE DATE, SALE_DATE DATE)";
```

السبب في وجود مثل هذا الارتباط هو أن أي شيء يمكنك القيام به في MySQL يمكنك إخبار PHP بالقيام به من أجلك في التعليمات البرمجية. يمثل هذا سطراً صالحًا من كود PHP حيث يتم تعين قيمة سلسلة للمتغير \$sql للاحتفاظ بعبارة SQL. بالطبع ، هناك المزيد الذي يجب القيام به خارج هذا السطر الفردي من التعليمات البرمجية ، ولكن لا تقلق - سنغطي هذا الأمر قريباً.

Showing as PHP code

```
$sql = " CREATE TABLE INVENTORY (ID int PRIMARY KEY, VIN varchar(17), YEAR INT, Make varchar(50), Model
varchar(100), TRIM varchar(50), EXT_COLOR varchar (50), INT_COLOR varchar (50), ASKING_PRICE DECIMAL
(10,2), SALE_PRICE DECIMAL (10,2), PURCHASE_PRICE DECIMAL (10,2), MILEAGE int, TRANSMISSION
varchar (50), PURCHASE_DATE DATE, SALE_DATE DATE)";
```

[Inline] [Edit] [Without PHP Code] [Submit Query]

Run SQL query/queries on database cars:

```
CREATE TABLE INVENTORY (ID int PRIMARY KEY, VIN varchar(17), YEAR INT, Make varchar(50), Model
varchar(100), TRIM varchar(50), EXT_COLOR varchar (50), INT_COLOR varchar (50), ASKING_PRICE DECIMAL
(10,2), SALE_PRICE DECIMAL (10,2), PURCHASE_PRICE DECIMAL (10,2), MILEAGE int, TRANSMISSION
varchar (50), PURCHASE_DATE DATE, SALE_DATE DATE)
```

تمرين: قم بإنشاء جدول

قم بإنشاء جدول باستخدام جملة `CREATE TABLE` ، ثم احذف الجدول وقم بإنشائه مرة أخرى باستخدام `phpMyAdmin`. أيهما أسهل؟

العمل مع جمل SQL

أدخل البيانات

الآن بعد أن تم إنشاء جدول ، فإن الخطوة المنطقية التالية هي وضع بعض البيانات في جدولنا. في عالم SQL يتم تحقيق ذلك باستخدام الأمر `INSERT`.

The syntax for inserting into a table is as follows:

`INSERT INTO {Name of table} (Column Names) VALUES (Column Values);`

انقر فوق علامة التبويب SQL مرة أخرى ، واتبع الأمر التالي (إذا استطعت) ، واضغط على `Enter`.

```
INSERT INTO `cars`.`inventory` (`VIN`, `YEAR`, `Make`,
`Model`, `TRIM`, `EXT_COLOR`, `INT_COLOR`,
`ASKING_PRICE`, `SALE_PRICE`, `PURCHASE_PRICE`,
`MILEAGE`, `TRANSMISSION`, `PURCHASE_DATE`,
`SALE_DATE`) VALUES ('5FNYF4H91CB054036', '2012',
'Honda', 'Pilot', 'Touring', 'White Diamond Pearl',
'Leather', '37807', NULL, '34250', '7076', 'Automatic',
'2012-11-08', NULL);
```

من الواضح أن كتابة SQL ليست صعبة من الناحية المفاهيمية ... لكنها مملة وعرضة للخطأ ، خاصة وأن العبارة تطول. هذا البيان:

```
INSERT INTO `cars`.`inventory` ( `YEAR`, `Make`,
`Model`, `ASKING_PRICE`) VALUES ( '2012', 'Honda',
'Pilot', '37807');
```

من السهل جداً متابعتها ، ولكن الخطوة التالية أصعب قليلاً:

```
INSERT INTO `cars`.`inventory` (`VIN`, `YEAR`, `Make`,
`Model`, `TRIM`, `EXT_COLOR`, `INT_COLOR`,
`ASKING_PRICE`, `SALE_PRICE`, `PURCHASE_PRICE`,
`MILEAGE`, `TRANSMISSION`, `PURCHASE_DATE`,
`SALE_DATE`) VALUES ('5FNYF4H91CB054036', '2012',
'Honda', 'Pilot', 'Touring', 'White Diamond Pearl',
'Leather', '37807', NULL, '34250', '7076', 'Automatic',
'2012-11-08', NULL);
```

الاختلاف الوحيد هو عدد الحقول. بناء الجملة هو نفسه ، ولكن التحدي يصبح التأكيد من وجود علاقة رأس برأس لكل اسم وقيمة عمود ، وأنهما بالترتيب الصحيح - أسماء الأعمدة والقيم الخاصة بها ، أي.

كما ترى ، من السهل كتابة عبارة INSERT. كلنا نفعل ذلك. لحسن الحظ ، يجعل phpMyAdmin من السهل إنشاء عبارات SQL مماثلة. ما عليك سوى النقر فوق الجدول ، ثم النقر فوق الزر "إدراج" وإدخال القيم في المربعات ، كما هو موضح:

| Column | Type | Function | Null | Value |
|--------------|---------------|----------|-------------------------------------|-------|
| VIN | varchar(17) | | <input checked="" type="checkbox"/> | |
| YEAR | int(11) | | <input checked="" type="checkbox"/> | |
| Make | varchar(50) | | <input checked="" type="checkbox"/> | |
| Model | varchar(100) | | <input checked="" type="checkbox"/> | |
| TRIM | varchar(50) | | <input checked="" type="checkbox"/> | |
| EXT_COLOR | varchar(50) | | <input checked="" type="checkbox"/> | |
| INT_COLOR | varchar(50) | | <input checked="" type="checkbox"/> | |
| ASKING_PRICE | decimal(10,2) | | <input checked="" type="checkbox"/> | |
| SALE PRICE | decimal(10,2) | | <input checked="" type="checkbox"/> | |

بمجرد النقر فوق الزر ، سيقوم phpMyAdmin بإنشاء عبارة SQL نيابة عنك وإدراج السجل ، وحتى عرض تحويله إلى سطر من كود PHP نيابة عنك.

إليك خدعة يستخدمها المحترفون: بمجرد أن يكون لديك سطر واحد من SQL يعمل ، فمن السهل جدًا نسخه ولصقه وتعديل القيم للسيارة التالية. استمر وأدخل بعض القيم الإضافية حتى تحصل على 5 أو 6 سيارات في طاولتك. هنا آخر واحد:

```
INSERT INTO `cars`.`inventory` (`VIN`, `YEAR`, `Make`, `Model`, `TRIM`, `EXT_COLOR`, `INT_COLOR`, `ASKING_PRICE`, `SALE_PRICE`, `PURCHASE_PRICE`, `MILEAGE`, `TRANSMISSION`, `PURCHASE_DATE`, `SALE_DATE`) VALUES ('4T4BF3EK5AR031954', '2010', 'Toyota', 'Camry', 'LE', 'Magnetic Gray Metallic', 'Tan cloth', '16977', NULL, '14250', '32673', '6 Speed Automatic', '2012-11-08', NULL);
```

لا تقلق إذا أخطأت. سوف تحذر MySQL ، وتمنعك من تشغيل الأوامر غير الصحيحة. لست بحاجة إلى إدخال 10 أو 20 سيارة ؛ يشتمل نموذج التعليمات البرمجية على ملف

البرنامج النصي الذي يفعل ذلك من أجلك. فقط افعليها مرات كافية لتحصل عليها

[حدد البيانات](#)

إن بناء جملة SQL مباشرة إلى الأمام ، على الأقل من الناحية النحوية. لقد استخدمناها حتى الآن لإنشاء قاعدة بيانات وإنشاء جدول داخل قاعدة البيانات هذه وإدراج البيانات في الجدول.

لم يتبق لنا سوى عدد قليل من المعاملات الأساسية لإتقانها: قراءة البيانات وتحديث البيانات وحذف البيانات. يشير بعض الأشخاص إلى هذا باختصار مبهج ، CRUD من أجل الإنشاء والقراءة والتحديث والحذف.

— — —

تم قراءة البيانات باستخدام عبارة SELECT تحدد عبارة SELECT قيمة أو مجموعة قيم من جدول وترجع هذه القيمة (القيم) إلى المستخدم.

إليك طريقة سهلة للتذكرها: تتيح لك عبارة SELECT أن تكون انتقائياً. ذكي ، إيه؟

The syntax for selecting data from a table is as follows:

SELECT (Column Names) FROM (Table Name);

لنبدأ بعبارة SELECT البسيطة. في ، انقر فوق أيقونة السيارات على الجانب الأيسر ثم انقر فوق علامة التبويب SQL في أعلى الصفحة. اكتب الأمر التالي واضغط على Go.

حدد * من المخزون :

بشكل عام ، يُطلق على حرف العلامة النجمية (*) في لغة الكمبيوتر حرف بدل ويعني أساساً "كل شيء" ، لذلك يجب أن تُرجع نتيجة الأمر أعلاه جميع الصفوف والأعمدة في جدول المخزون ، وتبدو مشابهة لما يلي:

localhost > cars > inventory

Show query box

Showing rows 0 - 2 (3 total, Query took 0.0007 sec)

```
SELECT *
FROM `inventory`
LIMIT 0 , 30
```

Show : 30 row(s) starting from row # 0 in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

| | VIN | YEAR | Make | Model | TRIM | EXT_COLOR | INT_COLOR | ASKING_PRICE |
|--------------------------|------------------|------|--------|----------|-----------|------------------------|-----------|--------------|
| <input type="checkbox"/> | 4T4BF3EK5AR031 | 2010 | Toyota | Camry | LE | Magnetic Gray Metallic | Tan cloth | 16977.00 |
| <input type="checkbox"/> | 5FNYF4H91CB05403 | 2012 | Honda | Pilot | Touring | White Diamond Pearl | Leather | 37807.00 |
| <input type="checkbox"/> | 1FMZU73EX5UB068 | 2012 | Ford | Explorer | XLT Sport | Black | Tan cloth | 37807.00 |

Check All / Uncheck All With selected: Change Delete Export

Show : 30 row(s) starting from row # 0 in horizontal mode and repeat headers after 100 cells

Query results operations

Print view Print view (with full texts) Export Display chart Create view

إذا كتبت هذا البيان بشكل صحيح ، يجب أن ترى المحتويات الكاملة لجدولك "مخزون". لتحديد أعمدة معينة فقط في الجدول ، اكتب جميع الأعمدة التي تريدها في هذا الجدول مفصولة بفواصل. اكتب الأمر التالي واضغط على Go.

SELECT `Make`, `Model`, `ASKING_PRICE` FROM `inventory`

يجب أن نرى شيئاً من هذا القبيل:

localhost > cars > inventory

Show query box

Showing rows 0 - 2 (3 total, Query took 0.0007 sec)

```
SELECT `Make`, `Model`, `ASKING_PRICE`
FROM `inventory`
LIMIT 0 , 30
```

Show : 30 row(s) starting from row # 0 in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

| | Make | Model | ASKING_PRICE |
|--------------------------|--------|----------|--------------|
| <input type="checkbox"/> | Toyota | Camry | 16977.00 |
| <input type="checkbox"/> | Honda | Pilot | 37807.00 |
| <input type="checkbox"/> | Ford | Explorer | 11999.00 |

Check All / Uncheck All With selected: Change Delete Export

Show : 30 row(s) starting from row # 0 in horizontal mode and repeat headers after 100 cells

لاحظ أنني أضفت الدائرة الحمراء والخط لتظهر لك المكان الذي تبحث فيه. قامت قاعدة بيانات MySQL بإرجاع الأعمدة التي حددتها فقط باستخدام عبارة SELECT.

[بيانات أين](#)

لقد تعلمت حتى الآن كيفية الحصول على جميع الصفوف والأعمدة من جدول ، وكيفية الحصول على أعمدة محددة من جدول ، ولكن ماذا عن الصفوف المحددة؟

هذا هو المكان الذي يأتي دور بيان WHERE يعطي بيان WHERE مجموعة محددة من المعايير لقاعدة بيانات MySQL بحيث تكون النتائج أكثر تحكماً وذات صلة بما تريده. على سبيل المثال ، لنفترض أنك تريد تحديد جميع Ford Explorer الموجودة في المخزون ، أو جميع Toyotas التي تقل قيمتها عن 15000 دولار. بند WHERE يجعل هذا ممكنا.

```
SELECT `Model`, `TRIM`, `EXT_COLOR` FROM inventory
WHERE Make = 'Ford'
```

يجب أن تكون النتائج هي كل سيارة صنعتها شركة Ford في قاعدة البيانات. إذا كنت تريد فقط Ford Explorer فستحتاج إلى WHERE Make = 'Ford' AND Model = 'Explorer'.

بالطبع ، إذا كنت تبحث عن شراء سيارة ، فستكون مهتماً فقط بتلك السيارات التي لم يتم بيعها بالفعل ، لذلك قد يكون الاستعلام التالي مناسباً بشكل أفضل:

```
SELECT `Make`, `Mode`, `EXT_COLOR` FROM inventory
WHERE YEAR > = 2010 AND `SALE_DATE` IS Null
```

NULL هي كلمة خاصة تعني أن الحقل لا يحتوي على قيمة ، ولسبب ما لا يمكنك قول ، عليك أن تقول IS NULL. أنا متأكد من أن هناك سبباً لذلك ، لكن هذا لا يهم حقاً. انه ما هو عليه.

[عوامل المقارنة](#)

هناك العديد من عوامل المقارنة المختلفة بالإضافة إلى = و IS.

| Operation | Symbol |
|--------------------------|--------|
| Equal To | = |
| Not Equal To | != |
| Greater Than | > |
| Less Than | < |
| Greater Than Or Equal To | >= |
| Less Than Or Equal To | <= |

تذكر أن تحيط سلسلة مع اقتباسات أو أقواس في كل مرة ترغب في استخدامها في عبارات SQLلن يعملوا بطريقة أخرى. MySQL ، يذهب الأمر SELECT دائماً بعد عبارة WHERE في.

للعثور على جميع السيارات ذات العام 2010 أو الأحدث ، من الواضح إلى حد ما أنها بحاجة إلى استخدام عامل أكبر من أو يساوي إلى المحدد أعلاه. اكتب الأمر التالي في المترجم واضغط على Go.

```
SELECT `Make`, `Mode`, `EXT_COLOR` FROM inventory
WHERE YEAR > = 2010 AND `SALE_DATE` IS Null
```

ترتيب حسب

من المحتمل أن تكون جملة ORDER BY واحدة من أسهل الأوامر وأكثرها فعالية في SQL يمكنك إرفاقه في نهاية أي عبارة SELECT لوضع النتائج بترتيب العمود الذي تحدده.

```
SELECT * FROM inventory ORDER BY YEAR DESC
```

يجب أن يعرض البيان أعلى السيارات بالترتيب حسب العمود "السنة" مع ظهور أحدث السيارات في الأعلى. هذا بسبب وضع المعدل ، DESC أو التنازلي ، في نهاية الأمر.

```
SELECT * FROM inventory WHERE YEAR > = 2006
ORDER BY YEAR ASC
```

يجب أن يعرض البيان أعلى السيارات بالترتيب حسب العمود "السنة" مع ظهور أقدم السيارات في الأعلى. وذلك لأن المعدل ، ASC أو التصاعدي ، يتم وضعه في نهاية الأمر.

يمكن أيضًا استخدام معدل ORDER BY مع عبارة WHERE مثل:

```
SELECT * FROM inventory WHERE YEAR > = 2006
ORDER BY YEAR ASC
```

فقط تذكر أن الأمر WHERE يذهب دائمًا قبل الأمر ORDER BY . إذا قمت بخلطها ، فسوف تحصل على خطأ.

للحد من عدد النتائج التي تتلقاها في عبارة ORDER BY ، يمكنك استخدام جملة التحديد بعد كتابة "تصاعدي" أو "تنازلي" ، مثل تحديد * من المخزون ؛ ORDER BY YEAR DESC limit 10 الرقم بعد الحد يحدد عدد النتائج التي يتم إرجاعها.

تحديث البيانات

لتحديث السجلات الموجودة في قاعدة بيانات ، يمكنك استخدام عبارة UPDATE . قد يكون هذا مفيدًا ، على سبيل المثال ، عندما يتم بيع سيارة في المخزون بسعر طلب أقل.

صيغة بيان التحديث هي

```
UPDATE table_name
SET column1=value, column2=value2, ...
WHERE some_column=some_value
```

لتغيير سعر الطلب لسيارة في قاعدة البيانات الخاصة بنا ، يمكنك استخدام عبارة مثل:

```
UPDATE `cars`.`inventory` SET `ASKING_PRICE` =
'36999' WHERE `inventory`.`VIN` =
'5FNYF4H91CB054036';
```

حذف البيانات

لحذف السجلات من قاعدة بيانات ، يمكنك استخدام عبارة DELETE ، مع تحديد اسم الجدول وعبارة WHERE التي تحدد السجلات المراد حذفها.

```
DELETE FROM table_name
WHERE some_column=some_value
```

على سبيل المثال ، لحذف سيارات Caravan من المخزون ، يمكنك استخدام أمر مشابه لـ

```
DELETE from inventory WHERE `Model` = 'Caravan'
```

إذا كنت تريده حذف كل شيء من جدول قاعدة البيانات ، فيمكنك تخطي عبارة WHERE واستخدام حرف البدل لصديقنا مع عبارة مثل

```
DELETE * from inventory
```

11

استخدام MySQL و PHP معًا

مقدمة

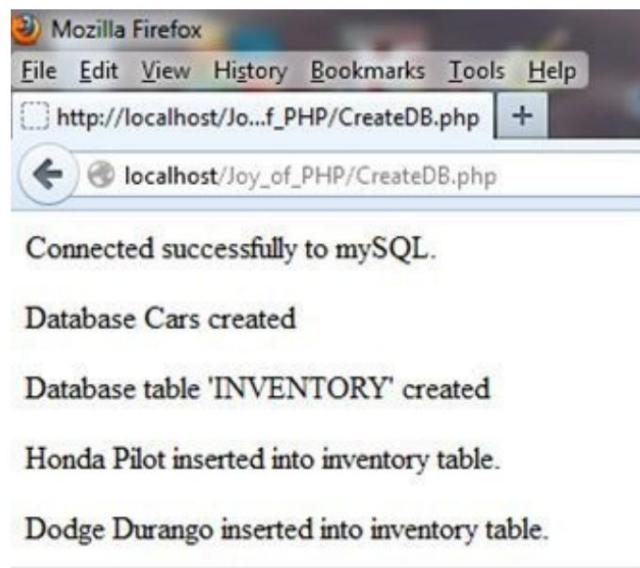
في الفصل السابق ، تعلمنا جميع أساسيات استخدام قاعدة البيانات ، في حالتنا MySQL من المحتمل أن تعمل جميع عبارات SQL التي تعلمناها حتى الآن مع أنظمة قواعد البيانات الأخرى ، مثل Microsoft SQL Server. مثل SQL ، تعمل أوامر الشائعة عبر جميع قواعد البيانات. ستجد أحياناً اختلافات طفيفة ، لكن أنظمة قواعد البيانات الأخرى خارج نطاق هذا الكتاب.

حتى الآن قمنا بإنشاء عبارات SQL الخاصة بنا إما يدوياً ، أو بمساعدة phpMyAdmin. في هذا الفصل ، سنستخدم PHP و MySQL معاً. هذا هو المكان الذي تبدأ فيه بالفعل في التحسن.

سفرة!

ستعمل قائمة كود PHP التالية على أتمتة جميع الخطوات التي غطيناها في الفصل السابق لإنشاء قاعدة بيانات وإنشاء جدول وإدراج سجلات في الجدول.

إذا كان كل شيء يعمل على النحو المنشود ، يجب أن ترى شاشة مثل هذه:



The screenshot shows a Mozilla Firefox window. The address bar displays "http://localhost/Joy_of_PHP/CreateDB.php". The page content area shows the following text:

```

Connected successfully to mySQL.

Database Cars created

Database table 'INVENTORY' created

Honda Pilot inserted into inventory table.

Dodge Durango inserted into inventory table.

```

تم ترقيم الكود التالي لتسهيل شرحه ، ولكن تذكر أنه يمكنك الحصول على نموذج الكود من موقع الويب <http://www.joyofphp.com>. يسمى هذا البرنامج النصي ["createdb.php"](#).

لا تحتاج إلى دراسة كل سطر كما هو معرض هنا. فقط قم بإلقاء نظرة سريعة عليها. في القسم التالي ، سأاستعراض كامل النص سطراً بسطر وأشرح كل واحد.

قائمة الكود: createdb.php

```

1. <?php
2. / **
3. * متنعة كود عينة
4. * يوضح كيفية إنشاء قاعدة بيانات وإنشاء جدول وإدراج السجلات.
5. */
6.
7. $ mysqli = new mysqli ('localhost', 'root', 'mypassword');
8.
9. if (! $ mysqli) {
"    . mysqli_error ($ mysqli)) : 10. يموت ("تعذر الاتصال:
11. }
12. صدى "تم الاتصال بنجاح ب '.
13. mySQL. <BR>
14. *
15. * إنشاء جدول لا يرجع مجموعة نتائج /
16. if ($ mysqli-> query ("CREATE DATABASE Cars") === TRUE) {17. echo "<p>
</P>" إنشاء قاعدة بيانات للسيارات :
18. }
19. آخر
20. {
21. صدى "خطأ في إنشاء قاعدة بيانات سيارات: ." mysqli_error ($ mysqli). "<br>"; 22. }
22. حدد قاعدة بيانات للعمل معها
23. //
24. mysqli $-> select_db
25. Echo ("Selected the Cars
26.
" Model varchar (100), 29. TRIM varchar (50), EXTCOLOR varchar (50), INT_COLOR varchar (50) , ASKING_PRICE DECIMAL (10.2) ,
27. $ query = 28. (VIN varchar (17) PRIMARY KEY, YEAR INT, Make varchar (50),
DECIMAL (10.2), MILEAGE int, TRANSMISSION varchar (50), PURCHASE_DATE DATE, SALE_DATE DATE ) ";
30. SALE_PRICE DECIMAL (10.2), PURCHASE_PRICE
$ query: 33. // echo "<p> ***** </p>": 34.
31. // echo "<p> ***** </p>": 32. // echo
(mysqli $-> query ($ query) === TRUE) 35. {
36. صدى "إنشاء جدول قاعدة البيانات المخزون ؟ </P>
37. }
38. آخر
39. {

```

:40. خطأ: <p> صدى // خطأ: الخلية خطأ() .

41. {

42. // يتم تخزين التواريف في MySQL بتنسيق SQL .

43. \$ query = "INSERT INTO cars ." . " TRIM ." . " EXT_COLOR ." . " INT_COLOR ." . " ASKING_PRICE ." . " SALE_PRICE
" . " YYYY-MM-DD" .

45- القيمة:

'2012', 'Honda', 'Pilot', 'Touring', 'White Diamond Pearl', 'Leather', '37807', NULL, '34250', '7076', 'Automatic', '2012-11-08', NULL) : ";

46. ('5FNYF4H91CB054036',

47.

48.

49. إذا كان تم إدراج Pilot (\$ mysqli-> query (\$ query) == TRUE) {

50. echo "<p>" .

51. } .

52. آخر .

53. {

54. صدى خطأ في إدخال هوندا باليوت: <p> . mysqli_error (\$ mysqli) : .

55. صدى خطأ في إدخال دودج دورانجو: <p> . mysqli_error (\$ mysqli) : .

56. echo \$ query: .

57. صدى خطأ في إدخال دودج دورانجو: <p> . mysqli_error (\$ mysqli) : .

58. } .

59.

60. // أدخل دودج دورانجو .

61.

62. \$ query = "INSERT INTO cars ." . " PURCHASE_PRICE ." . " MILEAGE ." . " TRANSMISSION ." . " PURCHASE_DATE ." . " SALE_DATE ") .

63- القيمة:

'2009', 'Dodge', 'Durango', 'SLT', 'Silver', 'Black', '2700', NULL, '2000', '144000', '4WD Automatic', '2012-12-05', NULL) : ";

64. ('LAKSDFJ234LASKRF2',

65.

66. إذا كان تم إدراج دورانجو (\$ mysqli-> query (\$ query) == TRUE) {

67. echo "<p>" .

68. } .

69. آخر .

70. {

71. صدى خطأ في إدراج المراوغة: <p> . mysqli_error (\$ mysqli) : .

72. echo \$ query: .

73. <p> . mysqli_error (\$ mysqli) : .

74. صدى خطأ في إدراج المراوغة: <p> . mysqli_error (\$ mysqli) : .

75. } .

76.

77.

```
78. mysqli $-> close ();
79.<
```

شرح الكود: createdb.php

بعد ذلك ، سوف أطلعك على الكود ، سطراً بسطراً. من فضلك خذ الوقت الكافي لمتابعة معي ، فهذه هي الطريقة الوحيدة للحصول عليها حقاً. نعم ، كل سطر مهم .

```
1. <?php
```

السطر 1 هو علامة البداية لـ ، ويخبر مترجم PHP أن ما يلي هو رمز وليس HTML.

```
2. / **
```

متعة نموذج كود * PHP 3.

* يوضح كيفية إنشاء قاعدة بيانات وإنشاء جدول وإدراج السجلات.

```
5. */
```

```
6.
```

السطور 5-2 هي تعليقات. التعليقات جيدة ، لذا ضع الكثير من التعليقات في التعليمات البرمجية الخاصة بك.

```
7. $ mysqli = new mysqli ('localhost', 'root', 'mypassword');
```

ينشئ السطر 7 متغيراً يسمى \$ con (للاتصال) وبقيمة مساوياً للدالة المضمونة للاتصال بـ mySQL. تحتاج إلى توفير اسم المضيف واسم المستخدم وكلمة المرور لخادم mySQL الخاص بك. إذا لم يكن لديك اسم المستخدم وكلمة المرور الصحيحة ، فسترى هذا:



```
9. if (! $ mysqli {
```

السطر 9 هو بداية عبارة ، if والتي تقول أساساً "إذا لم تكون متصلة". علامة التعجب ليست عامل التشغيل. الهدف من هذا الخط هو الاختبار لمعرفة ما إذا كان السطر 7 قد نجح.

```
10. " . mysqli_error ($ mysqli)) :
```

السطر 10 هو ما يجب فعله إذا فشل الاتصال. 'die' هو أمر يوقف تنفيذ المزيد من التعليمات البرمجية ويطبع النص الذي يليه. لو كنت أنا من اختر ، لربما سميت هذا الأمر "stop" بدلاً من ، "die" لكنه يوضح هذه النقطة.

11.}

mySQL.
 12. صدى "تم الاتصال بنجاح بـ ' .

يطبع السطر 12 "تم الاتصال بنجاح بـ . هذا هو السطر الأول الذي تراه في المتصفح.

* / * نتائج مجموعة مرجع لا جدول إنشاء 15.

تم 16. if (\$ mysqli-> query ("CREATE DATABASE Cars") === TRUE) {17. echo "<p>

إنشاء قاعدة بيانات للسيارات :</p>"

18.}

آخر 19.

20. {

21. صدى "خطأ في إنشاء قاعدة بيانات السيارات: ."
 .

22.}

السطر 15 هو تعليق يشرح وظيفة السطر التالي.

يتم طباعة السطر 17 إلى المستعرض إذا تم تشغيل عبارة SQL في السطر 15 بدون أخطاء.

يقوم السطر 21 بطباعة معلومات الخطأ إلى المستعرض إذا لم يتم تشغيل عبارة SQL في السطر 15 بنجاح.

// 23. حدد قاعدة بيانات للعمل معها

السطر 23 تعليق. التعليقات جيدة.

: 24. mysqli \$-> select_db ("سيارات")

ينشئ السطر 24 متغيراً يسمى selected \$ والذي يستخدم وظيفة مضمونة لتحديد قاعدة بيانات ، mySQL باستخدام الاتصال الذي تم إنشاؤه في السطر 7.

25. Echo ("Selected the Cars database") :

يطبع السطر "Selected the Cars database" 25 إلى المتصفح.

"

إنشاء مخزن الجدول

27. \$ query =

28. (VIN varchar (17) PRIMARY KEY, YEAR INT, Make varchar (50), Model varchar (100),

29. TRIM varchar (50) , EXT_COLOR varchar (50) , INT_COLOR varchar (50) , ASKING_PRICE DECIMAL (10,2) ,

(10,2), PURCHASE_PRICE DECIMAL (10,2), MILEAGE int, TRANSMISSION varchar (50), PURCHASE_DATE DATE, SALE_DATE DATE) :

30. SALE_PRICE DECIMAL

تنشئ الأسطر 30 - 32 متغيراً يسمى query \$ والذي يحتوي على جملة SQL. تذكر أن phpMyAdmin أنشأ لنا هذا السطر من التعليمات البرمجية. شيء جيد أيضاً ، لأنه من السهل القيام به.

31. // echo "<p> ***** </p>" ;

```
32. // echo $query;
33. // echo "<p> ***** </p>" ;
```

الأسطر 33 - 31 هي تعليقات الآن ، لكنها كانت في السابق جزءاً من البرنامج النصي الذي طبع قيمة المتغير \$query . كان لدي هذا هناك لمساعدتي في معرفة سبب عدم نجاحها في البداية ، وتركتها هناك كمثال لما يجب فعله عندما لا يفعل البرنامج النصي ما كنت تعتقد أنه سيفعله تماماً. قمت بعد ذلك بنسخ إخراج السطر 32 إلى الحافظة ولصقه في phpMyAdmin للحصول على نصائح حول بناء الجملة.

```
34. if (mysqli $-> query ($query) === TRUE)
```

السطر 34 ينفذ عبارة SQL "الاستعلام" (\$query) ثم يختبر نتيجة جملة SQL الموجودة في المتغير \$mysqli .

```
35. {
```

36. صدى "إنشاء جدول قاعدة البيانات" المخزون ؛ "</ P>"

```
37. }
```

يقوم السطر 36 بطباعة الرسالة "تم إنشاء جدول قاعدة البيانات" المخزون "إذا كان السطر 34 هو أ

نجاح.

آخر38.

```
39. {
```

40. صدى </ p>. mysqli_error (\$mysqli) : 41.{<p> خطأ: .

يطبع السطر 40 الرسالة "Error: MySQL" خطأ إذا فشل السطر 34. نأمل أن تخبرك القيمة التي تم إرجاعها بواسطة () mysql_error بشيء مفيد حول سبب فشلها . في بعض الأحيان يحدث ذلك بالفعل.

42. يتم تخزين التواريخ في MySQL بتنسيق // YYYY-MM-DD

السطر 42 هو تعليق لتذكيرني (وأنت) بتنسيق التواريخ بالطريقة التي تتوقعها MySQL

```
cars ". "SALE_PRICE", "PURCHASE_PRICE", "MILEAGE", "TRANSMISSION", "PURCHASE_DATE", "SALE_DATE")  
43. $query = "INSERT INTO"
```

45-القيم

```
'Pilot', 'Touring', 'White Diamond Pearl', 'Leather', '37807', NULL, '34250', '7076', 'Automatic', '2012-11-08', NULL); :  
46. ('5FNYF4H91CB054036', '2012', 'Honda',
```

46-43 تغيير قيمة الاستعلام \$ إلى جملة SQL جديدة ، وهذه المرة INSERT .

```
49. if (mysqli-> query ($query) === TRUE) {
```

49 من اختبارات تنفيذ جملة SQL الموجودة في الاستعلام \$ المتغير

50. في Honda Pilot إدراج **< p >** جدول الجرد. "تم إدراج الرسالة 50 السطر يطبع خطأ."
 جدول المخزون **< p />** إذا كان السطر 49 ناجحاً. تضع العلامات **< p >** الرسالة في سطراها الخاص.

51.}

آخر 52.

53. {

54. خطأ في إدخال هوندا بایلوت: **< p />** الخلية خطأ(): صدى.55. صدى **< p >** : 56. echo \$ query;***** **< p >** ***** صدى :**"< p >**

58.}

49. الأسطر 47-54 طباعة رسالة إذا فشل السطر.

60. // Insert a Dodge Durango

61.

```
62. $query = "INSERT INTO `cars`.`inventory` (`VIN`,  

`YEAR`, `Make`, `Model`, `TRIM`, `EXT_COLOR`,  

`INT_COLOR`, `ASKING_PRICE`, `SALE_PRICE`,  

`PURCHASE_PRICE`, `MILEAGE`, `TRANSMISSION`,  

`PURCHASE_DATE`, `SALE_DATE`)
```

63. VALUES

```
64.('LAKSDFJ234LASKRF2', '2009', 'Dodge', 'Durango',  

'SLT', 'Silver', 'Black', '2700', NULL, '2000', '144000',  

'4WD Automatic', '2012-12-05', NULL);";
```

65.

60. // دودج دورانجو أدخل.

61.

```
62. $ query = "INSERT INTO" cars ". ." PURCHASE_PRICE ". ." MILEAGE ". ." TRANSMISSION ". ." PURCHASE_DATE ". ." SALE_DATE ")
```

63- القيم

```
'2009', 'Dodge', 'Durango', 'SLT', 'Silver', 'Black', '2700', NULL, '2000', '144000', '4WD Automatic', "2012-12-05" . NULL) : "  

64. ('LAKSDFJ234LASKRF2',
```

65.

66. إذا كان {\$query == TRUE} (

67. صدى **< p >** دودج دورانجو تضاف إلى جدول الجرد. : **< p />**

68.}

آخر 69.

70- {

71. خطأ في إدراج المراوغة: **< p >** صدى : **< p />** الخلية خطأ(): صدى.**"< p >**

73. echo \$ query;

```
"<p> ***** </p>" صدى : 74.  
75.}  
76.
```

الخطو⁷⁶ تفعل نفس الشيء مثل ، 58 - 43 باستثناء سيارة مختلفة.

```
78. mysqli $-> close ();  
79. <
```

يغلق السطر 78 الاتصال بـ . السطر 79 هو علامة النهاية لـ ، وسيتم التعامل مع أي نص يتبعه على أنه ، HTML بدلاً من رمز.

مرحباً ، أين ؟ HTML

ربما لاحظ القارئ الذكي أن هذا البرنامج النصي لم يظهر داخل النمط المعتاد لـ .
<HTML> <Body> <html code

ومع ذلك فقد نجحت. كيف ذلك؟ لقد اكتشفت هذا بالصدفة تماماً. إنها ليست إحدى وظائف PHP ولكن يبدو أن بعض المتصفحات ستتماً إطار عمل HTML إذا "نسيت" القيام بذلك ، وهو ما فعلته مرة واحدة. جربها بنفسك. إنها تعمل. هل هذه أفضل ممارسة؟ لا ، لا أستطيع أن أتخيل ذلك. لكن بينما تتعلم فإنها تتيح لك التركيز على كود PHP.

12

إنشاء نماذج لعرض البيانات وإضافتها وتحريرها وحذفها

مقدمة

لقد تعلمنا حتى الآن كيفية استخدام SQL لإنشاء قواعد البيانات وإضافة السجلات وتحرير السجلات وحذف السجلات واختيار السجلات. ثم تعلمنا كيفية استخدام PHP لإجراء نفس العمليات.

بعد ذلك سنصبح أكثر روعة. سنتعلم كيفية استخدام نماذج SQL التي تؤدي العملية مع HTML لإنشاء عبارات PHP التي تؤدي العملية.

النماذج التي تضيف بيانات إلى قاعدة بيانات

[نموذج أساسى](#)

لنبدأ بمثال بسيط يسهل اتباعه. إليك نموذج بسيط من أربعة مجالات:

Sam's Used Cars

VIN:

Make:

Model:

Price:

من الواضح أنها لا تحتوي على جميع سمات السيارة التي حددناها سابقاً ، وليس من الجميل النظر إليها ، لكنها بسيطة ، وستوضح النقطة دون أي خردة إضافية تعرّض طريقك. فهم المفهوم.

[HTML كود](#)

يتبع رمز إنتاج مثل هذا النموذج

<HTML>

<head>

PHP </title> بهجة<title>
</head>

```

<الجسم>
    <سيارات سام المستعملة>
        <h1>VIN: <input name = "VIN" type = "text" /> <br />
            <br />
            <input name = "Make" type = "text" /> <br />
            <br />
            <input name = "Model" type = "text" /> <br />
            <br />
            السعر: <input name = "Asking_Price" type = "text" /> <br />
            <br />
            = "Submit1" type = "submit" value = "submit" /> <br /> &nbsp; </form>
            <input name
</body>

</html>

```

حتى الآن ما لدينا هو HTML فقط ، وفي الواقع لن يفعل النموذج أي شيء إذا ضغطت على زر الإرسال ... حتى الآن.

نموذج العمل

لجعل النموذج يفعل شيئاً ما بالفعل ، نحتاج إلى تعديل علامة .
 <form><input name = "post" type = "submit" value = "SubmitCar.php" method = "Post"> يقول هذا يخبر المتصفح أنه عند إرسال النموذج بالضغط على زر الإرسال ، يجب أن يمر هذا النموذج إلى نص "SubmitCar.php" بعنوان "Post" للقيام بذلك.

يمكن إرسال النماذج إما باستخدام الطريقة "get" أو "post".
 ليس هناك حفاظاً سبب وجيه لاستخدام "get" عند إرسال نموذج ، وذلك لتبسيط الأمور ، سنستخدم "post" فقط كلما أرسلنا نموذجاً.

سنستخدم لاحقاً في الكتاب لغرض مختلف.

كود PHP

هذا ما سنتحققه. نريد البرنامج النصي المشار إليه في النموذج للحصول على القيم من النموذج ، وإنتاج جملة SQL INSERT باستخدام هذه القيم ، وكتابة عبارة SQL إلى المتصفح حتى نتمكن من رؤيتها ، وتنفيذ جملة SQL التي أنشأناها للتو ، وأخيراً ، واسمحوا لنا أن نعرف ما إذا كان يعمل.

إذا سارت الأمور على ما يرام ، يجب أن ينتج النص شيئاً مشابهاً لهذا:

إدراج في المخزن ، VIN(الشركة المصنعة ، الطراز ، ASKING_PRICE)

VALUES ('9T4BF3EKXBR153775', 'Ford', 'Fiesta', 800)

تم الاتصال بنجاح بـ mySQL
اختيار قاعدة بيانات السيارات.

لقد نجحت في إدخال Ford Fiesta في قاعدة البيانات.

إليك رمز ملف SubmitCar.php والمتوفر أيضاً في نموذج التعليمات البرمجية.
مرة أخرى ، ليس عليك دراستها هنا لأنني سأطلعك عليها بعد ذلك. في الوقت الحالي ، ما عليك سوى إلقاء نظرة سريعة عليها.

```

1. <html>
2. <head>
3.   <title>Car Saved</title>
4. </head>
5. <body bgcolor="#FFFFFF" text="#000000" >
6.
7. <?php
8. // Capture the values posted to this php program from the text
fields in the form
9.
10.$VIN =  $_POST['VIN'] ;
11.$Make =  $_POST['Make'] ;
12.$Model =  $_POST['Model'] ;
13.$Price =  $_POST['Asking_Price'] ;
14.
15.//Build a SQL Query using the values from above
16.
17.$query = "INSERT INTO Inventory
18. (VIN, Make, Model, ASKING_PRICE)
19. VALUES (
20. '$VIN',
21. '$Make',
22. '$Model',
23. $Price
24. )";
25.
26.// Print the query to the browser so you can see it
27.echo ($query. "<br>");
28.

```

```

29.$mysqli = new mysqli('localhost', 'root', 'password', 'cars' );
30./* check connection */
31.if (mysqli_connect_errno()) {
32.    printf("Connect failed: %s\n", mysqli_connect_error());
33.    exit();
34.}
35.
36. echo 'Connected successfully to mySQL. <BR>';
37.
38./select a database to work with
39.$mysqli->select_db("Cars");
40. Echo ("Selected the Cars database. <br>");
41.

42./ Try to insert the new car into the database */
43.if ($result = $mysqli->query($query)) {
44.    echo "<p>You have successfully entered $Make $Model into
the database.</P>";
45.}
46.else
47.{

48.    echo "Error entering $VIN into database: " .
mysql_error()."<br>";
49.}
50.$mysqli->close();
51.?>
52.</body>
53.</html>

```

1. <html>

السطر 1 هو الافتتاح <html>(وهو مغلق في السطر 53).

2. <head>

3. <title>Car Saved</title>
4. </head>

تشكل السطور من 2 إلى 4 علامة الرأس ، بينما يحدد السطر 3 عنوان الصفحة.

5.<body bgcolor="#FFFFFF" text="#000000" >

يفتح السطر 5 علامة الجسم (التي يتم إغلاقها في السطر 52) لاحظ أننا استخدمنا المعلمة الاختيارية لتعيين ألوان الخلفية والنص.

7. <?php

السطر 7 هو علامة الفتح ، <?php للإشارة إلى أن النص التالي هو رمز وليس HTML.

8. // Capture the values posted to this php program from the text fields in the form

السطر 8 تعليق. لا يمكنك أبداً أن يكون لديك الكثير من التعليقات في التعليمات البرمجية الخاصة بك. اعتاد مبكراً على الإفراط في التعليق على التعليمات البرمجية الخاصة بك. لم أسمع أبداً أي شخص يشتكى من أن الكود الذي كانوا يحاولون اكتشافه به تعليقات "كثيرة جدًا" تشتت الانتباه عن أناقة المنطق.

```
10. $VIN = $_POST['VIN'];
11. $Make = $_POST['Make'];
12. $Model = $_POST['Model'];
13. $Price = $_POST['Asking_Price'];
```

تحصل الأسطر 10-13 على القيم التي كانت في النموذج وقم بتعيينها للمتغيرات في PHP. \$ REQUEST هو متغير خاص يستخدم لجمع البيانات بعد إرسال نماذج HTML. أنت تتبعه باسم الحقل في نموذج الذي تريد استرداده.

علق عدد من قراء الإصدار الأول من هذا الكتاب على أنه لا يجب أن تثق أبداً في المعلومات التي يقدمها المستخدمون لك ، حتى في تطبيق الشركة مثل هذا ، حيث يكون المستخدمون موثوقون بشكل عام. لذا فإن الطريقة الأكثر أماناً لتحقيق ما فعلناه أعلاه هي استخدام وظيفة mysql_real_escape_string() المسمى في PHP. على سبيل المثال ، \$ Make = mysql_real_escape_string (\$ _ POST ['Make']) :

The mysql_real_escape_string() function escapes special characters in a string for use in an SQL statement

15. //Build a SQL Query using the values from above

السطر 15 تعليق. التعليقات جيدة.

```
17.$query = "INSERT INTO Inventory
18. (VIN, Make, Model, ASKING_PRICE)
19. VALUES (
20. '$VIN',
21. '$Make',
22. '$Model',
23. $Price
24. )";
```

الأسطر 24 - 17 بناء أمر SQL INSERT كان من الممكن أن يكون الكل في سطر واحد ، لكن من الأسهل قراءته بهذه الطريقة. لاحظ أن المتغيرات \$VIN و \$Model و \$Price موضوعة في الصيغة كما هي. لاحقا ، عندما يتم تنفيذ الكود فعليا ، ستستبدل PHP أسماء المتغيرات بقيمها الفعلية.

```
// Print the query to the browser so you can see it
```

السطر 26 تعليق. التعليقات جيدة.

```
27.echo ($query. "<br>");
```

يكتب السطر 27 بيان SQL إلى المستعرض ، على السطر الخاص به. يضيف ". "بعد الاستعلام
 \$ في نهاية السطر. هذا ما يضعها على خطها الخاص. لم يكن السطر 27 مطلوباً لكي تعمل الوظيفة. إنه موجود حتى تتمكن من رؤية كيفية ترجمة PHP للمتغيرات إلى قيمها عند إنتاج جملة ، SQL والتي يتم تخزينها بدورها في المتغير الاستعلام . \$.

To add two strings together in PHP use the . character. In line 27 we have added two strings together-- \$query and “
”, then used the echo command to write it to the browser.

Authors Note: If I was designing PHP, I would use the & character to join two strings. That would make it consistent with other languages, but it is what it is.

```
29. $mysqli = new mysqli('localhost', 'root', 'password',
'cars' );
```

يقوم السطر 29 بإجراء اتصال بقاعدة بيانات MySQL عن طريق تمرير اسم الخادم ("المضيف المحلي") ، واسم المستخدم ("الجذر") ، وكلمة المرور ("كلمة المرور") ، وقاعدة البيانات الأولية ("السيارات"). لاحظ أن كلمة مرورك ستكون مختلفة على الأرجح.

```
30. /* check connection */
```

السطر 30 هو تعليق. يستخدم الصيغة البديلة للدلالة على تعليق.

```

31. if (mysqli_connect_errno()) {
32.     printf("Connect failed: %s\n",
33.         mysqli_connect_error());
34. }

```

اختبار الخطوط 34-31 لمعرفة ما إذا كان الاتصال الذي تم إجراؤه مع الخط 29 يعمل أم لا. إذا لم يكن كذلك ، فإنه يطبع رسالة خطأ ثم يتوقف عن تنفيذ المزيد من التعليمات البرمجية (السطر - 33 خروج). مخرج() هو بديل لأمر يموت.

**36. echo 'Connected successfully to mySQL.
';**

يطبع السطر 36 على المتصفح رسالة "تم الاتصال بنجاح بـ mySQL". لن يتم تنفيذ هذا الخط إذا تم استدعاء السطر 33. منذ أن وصلنا إلى هذا الحد ، يمكننا أن نستنتج أنها في الواقع قمنا بالاتصال.

38. //select a database to work with

السطر 38 هو تعليق يشرح الغرض من السطر التالي.

```

39. $mysqli->select_db("Cars");
40. Echo ("Selected the Cars database. <br>");

```

يحدد الخط 39 قاعدة بيانات "السيارات" ، ويطبع السطر 40 هذه الحقيقة.

42./* Try to insert the new car into the database */

السطر 42 تعليق. ترى موضوعاً هنا ، أليس كذلك؟ كلما أضفت المزيد من التعليقات ، كان من الأسهل اكتشاف الكود الخاص بك عندما تعود لاحقاً لألقاء نظرة عليه.

```

42. /* Try to insert the new car into the database */
43. if ($result = $mysqli->query($query)) {
44.     echo "<p>You have successfully entered $Make
$Model into the database.</P>";
45. }
46. else
47. {
48.     echo "Error entering $VIN into database: " .
mysql_error()."<br>";
49. }

```

الخط 43 هو النهاية الكبرى. هنا نقوم بالفعل بتنفيذ جملة SQL على قاعدة بيانات السيارات. السطر 43 هو بداية جملة if ويطبع السطر 44 رسالة نجاح بينما يطبع السطر 48 رسالة فشل.

ملاحظة: يجب أن يقرأ السطر 48 حفلاً استخدام "mysql_error ()" وليس "mysqli_error ()". تم تصحيح هذا في نموذج التعليمات البرمجية. كما وأشار قارئ ذكي من الإصدار الأول ، لا يمكنك خلط mysql و mysqli في نفس النص - فهما ليسا متماثلين. على أي حال ، فإن ملف

تم إهمال امتداد mysql لصالح امتداد mysqli.

50. \$mysqli->close();

يغلق الخط 50 الاتصال بقاعدة بيانات MySQL . هذا ليس مطلوباً تماماً ، حيث ستظل الصفحة تعمل إذا لم تقم بذلك ، ولكن من الواضح أنها فكرة جيدة لأنه إذا لم تقم بذلك ، فسيقوم الخادم في النهاية بتطوير مشكلات وسيطلب في النهاية إعادة التشغيل.

51. ?>

يغلق السطر 51 علامة PHP التي تم فتحها في السطر 7 ، مما يشير إلى أن الأسطر التالية هي HTML وليس كود PHP.

52.</body> 53.</html>

يغلق السطرين 52 و 53 علامة النص الأساسي وعلامات HTML على التوالي.
واو ، لقد جعلناها من خلال النص بأكمله. إذا كنت لا تزال معـي ، فلديك مستقبل جيد في تطوير ! PHP أبق معـها!

A Brief Time Out... تشمل الملفات وإدخال SQL

تضمين الملفات

قد تذكر من القسم السابق حول يتضمن فكرة إعادة استخدام الكود عن طريق تضمين محتويات ملف واحد في ملف آخر. هذا هو الوقت المناسب لإعادة النظر في هذا الموضوع المهم.

لقد قمنا حتى الآن بإنشاء ملفي PHP مختلفين - الأول لإنشاء قاعدة بيانات وجدول ، والثاني في القسم أعلاه لإدراج البيانات في قاعدة البيانات باستخدام نموذج ويب. كما يمكنك التخمين من عناوين الأقسام التي ستظهر لاحقاً في هذا الفصل ، فنحن على وشك إنشاء المزيد من البرامج النصية التي ستسمح لنا أيضاً بتحرير البيانات وحذفها.

سيكون لكل من هذه النصوص شيئاً مشتركاً - رمز يتصل بقاعدة بيانات ، وفي كل حالة ستكون هذه الشفرة متطابقة تماماً. حتى الآن ، قمنا بالتطوير على جهاز الكمبيوتر الخاص بـنا ، لذلك كان اسم المضيف هو "localhost" . في وقت ما في المستقبل القريب ، بعد أن كتبت عشرات البرامج النصية أو أكثر في المستقبل ، وفجأة قررت نقل تطبيقك إلى كمبيوتر آخر - يمكن الوصول إليه من الإنترنت. لن يكون اسم المضيف هو نفسه. ولن يكون اسم المستخدم وكلمة المرور متماثلين على الأرجح. ماذا لو تم إخراج كلمة المرور الخاصة بك واحتاجت إلى تغييرها؟

بدون تدخيـلي المفـيد هنا ، ستواجهـه احتمـال تغيـير العـشرات من مـلفـات .php - الـبحث عن السـطـر الـذـي يـقرـأ شـيـئـاً مـثـل ...

```
$mysqli = new mysqli('localhost', 'root', 'password', 'cars');
```

... وتغييرـه ليـعكس اـسـمـ المـضـيفـ الجـديـدـ أوـ اـسـمـ المـسـتـخـدـمـ أوـ كـلـمـةـ المـرـوـرـ. أـوـكـ لنـ يـكـونـ هـنـاكـ مـتـعـةـ فـيـ هـذـهـ المـهـمـةـ عـلـىـ الإـطـلاـقـ.

من الآن فصاعداً ، ستنقل جزء الكود الذي يتصل بقاعدة البيانات إلى ملف منفصل ، وستشير جميع البرامج النصية الجديدة من هذه النقطة فصاعداً إلى هذا الرمز باستخدام عبارة `include` . إذا تغيرت أي من القيم ، فسيتعين علينا فقط تغييرها في مكان واحد ... الملف الذي يشير إليه جميع القيم الأخرى.

فقط تخيل متعة تغيير سطر واحد من التعليمات البرمجية ورؤيه هذا التغيير ينتشر عبر عشرات الصفحات. هذا ما أتحدث عنه. تعد ميزة التضمين واحدة من أكثر الميزات المفيدة والأكثر أهمية في `PHP` في رأيي المتواضع.

... سنستخدم الخط

```
include 'db.php';
```

لإخبار `PHP` بإدخال محتويات ملف `db.php` في البرنامج النصي الحالي. تأكد من استخدام تضمين الملفات كلما استطعت ، لأن ... الدقائق القليلة الإضافية التي تستغرقها لنقل بعض التعليمات البرمجية إلى ملف منفصل تكون أكثر من مدفوعة عندما يتغير هذا الرمز.

حقن `SQL` بشكل عام ، ليس من الجيد أن تأخذ كل ما يدخله المستخدم في نموذج وتمريره مباشرةً إلى برنامج نصي `SQL` كما فعلنا في المثال أعلاه. إذا كان المستخدم ضاراً (وماهراً) ، فقد يقوم بإدخال كود `SQL` في أحد النماذج الخاصة بك وقد يكون لهذا تأثير كبير على ما يفعله البرنامج النصي بالفعل. على سبيل المثال ، تخيل نموذجاً أساسياً لاسم المستخدم / كلمة المرور ويدخل المستخدم "أو - 1 = 1" في حقل كلمة المرور ، كما هو موضح:

| | |
|---------------------------------------|-----------|
| Username: | Brian |
| Password: | 'or 1=1-- |
| <input type="button" value="Submit"/> | |

الآن العبارة التي يتم تنفيذها في قاعدة البيانات هي التالية:

حدد * من المستخدمين حيث اسم المستخدم = "Brian" وكلمة المرور = "أو - 1 = 1--"

نظرًا لأن `1 = 1` يكون دائئراً صحيحاً ، فسيعيد هذا الاستعلام جميع المستخدمين. (لاحظ أنه تم التعليق على آخر اقتباس). لذلك ، في البرنامج النصي أعلاه ، يكون `sqlsrv_has_rows` صحيحًا ، وسيتم إرجاع جميع صفوف كلمة مرور اسم المستخدم.

إدخال `SQL` ممكن هنا لأن إدخال المستخدم متسلسل مع كود `SQL` المنفذ. تمثل إحدى طرق منع حدوث ذلك في إزالة أي شرطات مائلة أو علامات اقتباس من مدخلات المستخدم. يوضح مقتطف الشفرة التالي هذا:

```
// للحماية من حقن :
myusername): $ model = stripslashes ($ mypassword): $ make = mysql_real_escape_string ($ myusername)
SQL $ make = stripslashes ($
```

`$ = mysql_real_escape_string ($ mypassword)` نموذج :

مع المخاطرة بذكر ما هو واضح ، يزيل الأمر `stripslashes` أي مائلة للمستخدمين ويقوم الأمر `mysql_real_escape` بإزالة أحرف الاقتباس.

هناك طريقة أفضل لتقليل فرصة حقن SQL وهي استخدام عبارات معدة ، ولكن هذا موضوع خارج نطاق كتاب للمبتدئين هذا.
إذا كنت تريد معرفة المزيد ، فإن إليك مكاناً جيداً للبدء / topic/54239-introduction-to-mysqli-and-prepared-statement /
-> <http://www.dreamincode.net/forums/>

النماذج التي تعرض بيانات الملخص

من أول الأشياء التي نرغب في القيام بها مع `Sam's Used Cars` هو عرض قائمة بجميع السيارات التي تلبي المعايير المحددة. في البداية ، ستكون معاييرنا هي اختيار جميع السيارات ، ولكن فيما بعد يمكنك تعديل الاستعلام لإرجاع سيارات معينة فقط ببساطة عن طريق تعديل عبارة `SELECT` في الكود.

الطريقة التي سيعمل بها هذا هي أننا سنقوم بتنفيذ عبارة `Select` لاسترداد السيارات التي تطابق المعايير ، ثم نقوم بالتكرار عبر جميع الصفوف. سنضع كل صف من البيانات في جدول منسق بشكل جيد.

Sam's Used Cars

Complete Inventory

| Make | Model | Asking Price |
|--------|---------|--------------|
| Nissan | Maxima | 15000.00 |
| Mazda | 626 | |
| Mazda | 626 | 1200.00 |
| Toyota | Camry | 12000.00 |
| Honda | Pilot | 36999.00 |
| Toyota | Camry | 18000.00 |
| Toyota | Corolla | 5000.00 |
| Isuzu | Trooper | 7700.00 |

يمكن العثور على شفرة المصدر باسم `viewcars.php`. إذا سارت الأمور على ما يرام ، يجب أن تبدو الصفحة كما يلي:

بالطبع ، إخراج نص بسيط ليس جذاباً بشكل خاص للنظر إليه ، ولكن مع إضافة القليل من CSS يمكننا جعله يبدو كما يلي:

Sam's Used Cars

Complete Inventory

| Make | Model | Asking Price |
|--------|---------|--------------|
| Nissan | Maxima | 15000.00 |
| Mazda | 626 | |
| Mazda | 626 | 1200.00 |
| Toyota | Camry | 12000.00 |
| Honda | Pilot | 36999.00 |
| Toyota | Camry | 18000.00 |
| Toyota | Celica | 5000.00 |
| Izusu | Trooper | 7700.00 |
| Ford | Focus | 4400.00 |
| Ford | Fiesta | 800.00 |
| Mazda | 626 | |
| Dodge | Durango | 2700.00 |
| Nissan | Rogue | 30000.00 |

لكن دعونا لا نقدم كثيرا على أنفسنا. أولاً ، هذا هو الكود الذي ينتج النسخة الأساسية من الجدول. يكون إخراج هذا البرنامج النصي أكثر إثارة للاهتمام إذا كان لديك الكثير من السيارات في قاعدة البيانات الخاصة بك ، لذلك إذا لم تكن قد قمت بذلك بالفعل ، فاستخدم البرنامج النصي "createdb.php" المضمن في نموذج الكود لملء جدول المخزون الخاص بك بالكثير السيارات.

شفرة

```

1. <html>
2. <head>
3.   <meta charset="utf-8">
4.   <title>Sam's Used Cars</title>
5. </head>
6.
7. <body>
8. <h1>Sam's Used Cars</h1>
9. <h3>Complete Inventory</h3>
10. <?php
11. include 'db.php';
12. $query = "SELECT * FROM INVENTORY";

```

```
13.* Try to query the database */  
14.if ($result = $mysqli->query($query)) {  
15. // Don't do anything if successful.  
16}  
17.else  
18.{  
19. echo "Error getting cars from the database: " .  
mysql_error()."<br>";  
20}  
21.  
22// Create the table headers  
23.echo "<table id='Grid' style='width: 80%'><tr>";  
24.echo "<th style='width: 50px'>Make</th>";  
25.echo "<th style='width: 50px'>Model</th>";  
26.echo "<th style='width: 50px'>Asking Price</th>";  
27.echo "</tr>\n";  
28.  
29.$class = "odd"; // Keep track of whether a row was even  
or odd, so we can style it later  
30.  
31// Loop through all the rows returned by the query,  
creating a table row for each  
32.while ($result_ar = mysqli_fetch_assoc($result)) {  
33. echo "<tr class=\"$class\">";  
34. echo "<td>" . $result_ar['Make'] . "</td>";  
35. echo "<td>" . $result_ar['Model'] . "</td>";  
36. echo "<td>" . $result_ar['ASKING_PRICE'] . "</td>";  
37. echo "</td></tr>\n";  
38.  
39. // If the last row was even, make the next one odd and  
vice-versa  
40. if ($class=="odd"){  
41.     $class="even";  
42. }  
43. else  
44. {  
45.     $class="odd";
```

```

46.    }
47.}
48.echo "</table>";
49.$mysqli->close();
50.?>
51. </body>
52.
53.</html>

```

شرح الكود

لن أمشي معك عبر كل سطر بعد الآن ، لأنني لم أعد أعتقد أنك بحاجة إليه. من الآن فصاعداً ، سأشرح فقط الأشياء المهمة.

السطر 11 هو أول استخدام لنا لخيار التضمين الذي يشير إلى ملف خارجي يسمى db.php والذي سيتم تضمينه في هذا البرنامج النصي تماماً كما لو كان جزءاً من نفس الملف. لقد أبرزت السطر 11 أعلاه باللون الأزرق والرمز أدناه باللون الأزرق ، علىأمل أن تفهم بشكل أفضل كيفية عمله. يتم استبدال محتوى المربع الأزرق أدناه في الكود الخاص بالخط الأزرق (11)أعلاه ، بحيث يتم دمج كلا الملفين في نص واحد.

محتويات ملف "db.php" موضحة أدناه:

```

<?php
$mysqli = new mysqli('localhost', 'root', 'mypassword',
'cars' );
/* check connection */
if (mysqli_connect_errno()) {
    printf("Connect failed: %s\n", mysqli_connect_error());
    exit();
}
//select a database to work with
$mysqli->select_db("Cars");

?>

```

ال kod الموجود في ملف db.php مطابق للشفرة الموضحة في السطر 29 في القسم السابق ، لذلك لن أشرحها مرة أخرى هنا. هذه فائدة رئيسية أخرى لتضمين الملفات.
بمجرد أن يعمل الكود الموجود بداخله ، لن تضطر حقيقةً إلى التفكير فيه كثيراً بعد الآن.

12.\$query = "SELECT * FROM INVENTORY";

السطر 12 هو الاستعلام الذي ينتج قائمة السيارات المراد عرضها. في هذه الحالة البسيطة ، نختار جميع السيارات.

```

14. if ($result = $mysqli->query($query)) {
15. // Don't do anything if successful.
16. }
17. else
18. {
19. echo "Error getting cars from the database: " .
mysql_error()."<br>";
20. }

```

الأسطر 20 - 14 تقوم بتشغيل الاستعلام وتعرض رسالة خطأ في حالة فشل الاستعلام.

```
23. echo "<table id='Grid' style='width: 80%><tr>";
```

السطر 23 هو علامة افتتاحية لإنشاء جدول بمعرف "الشبكة". المعروف اختياري ولكنه يجعل من السهل تطبيق الأنماط على الجدول لاحقاً. النمط = العرض: 80% يمنع العمود من التمدد لملء الشاشة بأكملها ؛ بدلاً من ذلك يأخذ 80% من العرض. <tr> يبدأ صف الجدول بعلامة الفتح.

```

24. echo "<th style='width: 50px'>Make</th>";
25. echo "<th style='width: 50px'>Model</th>";
26. echo "<th style='width: 50px'>Asking Price</th>";
27. echo "</tr>\n";

```

الأسطر 24-27 تنشئ الصف الأول من الجدول ، الصف الذي يحتوي على عناوين الأعمدة الخاصة بالماركة والطراز والسعر. السطر 27 هو علامة صف جدول إغلاق ، متبعاً بسطر جديد.

```
29.$class ="odd"; // Keep track of whether a row was
even or odd, so we can style it later
```

يحدد السطر 29 قيمة متغير يسمى فئة \$ إلى "فردي" لأن صفات البيانات الأولى في جدولنا سيكون فردياً. أثناء قيامنا بتكرار كل صف من البيانات ، سنقوم بالتناوب بتعيين فئة \$ إلى قيمة إما "فردي" أو "زوجي". نقوم بذلك حتى نتمكن من تصميم الجدول لاحقاً بحيث تظهر الصفوف البديلة ألواناً مختلفة لتسهيل الأمر على العينين.

```

31. // Loop through all the rows returned by the query,
creating a table row for each
32. while ($result_ar = mysqli_fetch_assoc($result)) {
33. echo "<tr class=\"$class\">";
34. echo "<td>" . $result_ar['Make'] . "</td>";
35. echo "<td>" . $result_ar['Model'] . "</td>";
36. echo "<td>" . $result_ar['ASKING_PRICE'] . "
</td>";
37. echo "</td></tr>\n";

```

تنشئ السطور من 31 إلى 37 صفاً في جدول HTML ليتوافق مع كل صف في ملف

جدول قاعدة البيانات الذي استخرجناه باستخدام الاستعلام. تحتوي كل خلية جدول على بيانات من جدول mySQL على سبيل المثال ، السطر (:)34 يجب أن ينتج شيئاً مثل

</ td><td>

لأن result_ar ['Make'] تقول الحصول على قيمة Make (أحد الأعمدة في الجدول ، وفي هذه الحالة ("Ford") وضعه هنا بين علامتي . <td> خذ الوقت الكافي لفهم ما يفعله هذا الخط حقاً ، لأنه إذا كنت تستطيع فهم ذلك ، فيمكنك فعل أي شيء تقريباً! تذكر . حرف يعني الانضمام إلى هاتين السلسلتين.

39. // If the last row was even, make the next one odd
and vice-versa

```
40. if ($class=="odd"){
41.     $class="even";
42. }
43. else
44. {
45.     $class="odd";
46. }
```

السطور 46 - 39 فقط قم بتبديل قيمة فئة \$ من زوجي إلى فردي.

47. }

يغلق الخط 47 حلقة while.

48. echo "</table>";

يغلق السطر 48 الجدول بعلامة </table>.

49. \$mysqli->close();

يغلق السطر 49 قاعدة بيانات mySQL.

50. ?>

يشير السطر 50 إلى نهاية كود PHP.

51. </body>

السطر 51 هو نهاية النص في صفحة HTML.

53. </html>

أخيراً ، يشير السطر 53 إلى نهاية HTML.

تمرين: التغيير والتبديل في SELECT

ارجع وقم بتعديل هذا الرمز بحيث لا يختار جميع السيارات بل مجموعة فرعية

يمتعك.

تحسين مظهر الجدول باستخدام CSS الذي يحسن مظهر النموذج. تتم إضافة معلومات النمط هذه إلى قسم <head> من الصفحة ، ولكن غالبًا ما يضع الأشخاص أنماطًا في ورقة أنماط منفصلة أيضًا. راجع ملف viewcarswithstyle.php لمشاهدة النموذج أثناء العمل.

إن شرح كيفية عمل CSS هو خارج نطاق هذا الكتاب ، وموضوع في حد ذاته. لكن الشيء المهم هو أن نرى مدى سهولة تمكنا من تغيير مظهر جدول HTML باستخدام القليل من معلومات النمط. ألق نظرة على ورقة الأنماط الكاملة هنا ، وسأشرحها بعد ذلك.

```
1. <style>
2. /* The grid is used to format a table */
3. #Grid
4. {
5. font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
6. width:80%;
7. border-collapse:collapse;
8. margin-left: auto;
9. margin-right: auto;
10.}
11.#Grid td, #Grid th
12.{  
  
13.font-size:1em;
14.border:1px solid #61ADD7;
15.padding:3px 7px 2px 7px;
16.}
17.#Grid th
18.{  
19.font-size:1.1em;
20.text-align:left;
21.padding-top:5px;
22.padding-bottom:4px;
23.background-color:#C2D9FE;
24.color: lightslategray;
25.
```

```

26.}
27.#Grid tr.odd td
28.{ 
29.color:#000000;
30.background-color: #F2F5A9;
31.}
32.
33.#Grid tr.even
34.{ 
35.color:#000000;
36.background-color: white;
37.}
38.#Grid head
39.{ 
40.color:#000000;
41.background-color:teal;
42.}
43.
44. </style>

```

شرح CSS

1. **<style>**
2. **/* The grid is used to format a table */**

يفتح السطر الأول علامة ، لإخبار المتصفح أن ما يلي هو ورقة أنماط. السطر 2 تعليق.

```

3. #Grid
4. { 
5. font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
6. width:80%; 
7. border-collapse:collapse;
8. margin-left: auto;
9. margin-right: auto;
10. }

```

يقول السطر 3 لتحديد عنصر على الصفحة بمعرف الشبكة. رمز # هو المحدد لتحديد شيء محدد باستخدام معرف ، وما يليه هو اسم الشيء المحدد الذي تريد تحديده. انظر السطر 23 من نص PHP السابق ، والذي عين معرف الجدول الخاص بنا على "الشبكة" مع صدى السطر !>"<table id = 'Grid' style =' width: 80%'>" نظراً لأن لدينا جدولًا به ، id = 'Grid' أفسسنا تطبيق هذا النمط.

كل ما يلي بين رمزي {و} يحدد نمط هذا العنصر.
نختار الخط ، الحدود ، الهامش ، إلخ.

```
11. #Grid td, #Grid th
12. {
13. font-size:1em;
14. border:1px solid #61ADD7;
15. padding:3px 7px 2px 7px;
16. }
```

يحدد السطر 11 أن الأسطر التالية تنطبق فقط على علامتي <td> و <th> إذا ظهرت داخل عنصر بمعرف "الشبكة".

يصبح كل سطر يلي أكثر تحديداً حول كيفية تنسيق العنصر. محدد معين يستبدل عام واحد. لذلك بدأنا في تحديد التنسيق الافتراضي للشبكة ، لكننا لاحقاً قمنا بتعديل عناصر معينة لعنصر الشبكة. الجزء التالي هو كيفية تلوين الصفوف البديلة بشكل مختلف:

```
27.#Grid tr.odd td
28.{  
29.color:#000000;  
30.background-color: #F2F5A9;  
31.}
```

يقول السطر 27 لتحديد علامة ، <td> داخل صف الجدول <tr> إذا كان عضواً في الفصل الفردي. انظر إلى الناتج HTML عن البرنامج النصي. ستري صفاً في الجدول محدداً على النحو التالي: <tr class = 'odd'><tr class = 'even'> أو .

يوجد محدد آخر لرؤوس الجدول. من المنطقي إذا نظرت إليها لفترة كافية. يعد الرمز # في CSS محدداً. حسناً ، هذا كل شيء الآن. ربما سأقوم يوماً ما بكتاب "Joy of CSS". اسمحوا لي أن أعرف ...

تعديل النموذج للارتباط بصفحة التفاصيل آخر شيء يحتاجه هذا النموذج هو طريقة الارتباط بسيارة معينة. عندما ينقر زائر الموقع على سيارة معينة في صف واحد ، يجب أن يأخذهم لمزيد من التفاصيل حول تلك السيارة المحددة. بمعنى آخر ، يجب أن يتم نقلهم إلى صفحة "تفاصيل السيارة". سنضطر إلى إنشاء تلك الصفحة ، بالطبع.

The `<a>` tag defines a hyperlink, which is used to link from one page to another. Amazingly, 'A' actually stands for Anchor. I don't associate anchors with jumping around, but I guess someone did, and the tag somehow stuck.

The most important attribute of the `<a>` element is the `href` attribute, which indicates the link's destination.

لاحظ أنه لكي يعمل هذا ، سنحتاج إلى إنشاء صفحة التفاصيل لأنه بخلاف ذلك سيفشل الارتباط بشكل طبيعي. لا شيء يحدث بشكل تلقائي. بافتراض وجود صفحة التفاصيل ، يمكننا تعديل الكود في السطر 34 الذي يقرأ على النحو التالي:

```
echo "<td>" . $result_ar['Make'] . "</td>";
```

لقراءة بدلاً من ذلك على النحو التالي:

```
echo "<td><a href='viewcar.php?VIN=".$result_ar['VIN']."'>" . $result_ar['Make'] . "</a></td>";
```

ما يفعله هذا هو إنشاء "مرساة" أو رابط يجعل العمود الأول من كل صف رابطاً قابلاً للنقر. يجب أن يُخرج HTML مشابهاً لـ:

```
<td><a href='viewcar.php?VIN=123234FE221'>Nissan</a></td>
```

يمكنك أن ترى أن عنوان URL الذي تم إنشاؤه سيكون مشابهاً لـ `/viewcar.php?VIN=123234FE221` كما هو موضح أعلاه. هذا يخبر المتصفح بفتح ملف `viewcar.php` وتمريره سلسلة الاستعلام `VIN = VIN`. يطلق عليه سلسلة استعلام لأن الغرض الأساسي من تمرير البيانات إلى نموذج بهذه الطريقة هو حتى يتمكن من استخدام البيانات في استعلام - `SQL` وهذا بالضبط ما سنفعله.

هل تذكر مرة أخرى عندما قلت استخدام "Post" بدلاً من "Get" عند إرسال نموذج؟ إذا كنت قد استخدمت ، فسيؤدي النقر فوق زر الإرسال إلى إرسال عنوان URL طويل جداً إلى المتصفح مع الحق جميع أسماء الحقول والقيم في نهايته كسلسلة استعلام بتنسيق مشابه لـ `?Make = Ford & Model = Explorer` إلخ. سنستفيد من هذه التقنية لإنشاء سلسلة استعلام خاصة بنا وتمريرها إلى برنامج نصي.

في الوقت الحالي ، سيؤدي النقر فوق الارتباط إلى حدوث خطأ فقط ، لأن ملف `viewcar.php` غير موجود بعد ، ولكن هذا ما سنقوم ببنائه بعد ذلك.

النماذج التي تعرض بيانات تفصيلية

بمجرد أن يتعرف زائر الموقع على السيارة التي يريد المزيد من المعلومات عنها ، سيرغب متسوق السيارة في النقر فوق سيارة معينة لمعرفة المزيد عنها. لذلك سنقوم بإنشاء صفحة PHP للتعامل مع هذا. سوف نسمى هذه صفحة تفاصيل السيارة ، وسيكون اسم الملف الخاص بها هو viewcar.php.

مرة أخرى ، سنبني المثال بسيطًا نسبيًا لغرض اتباع المنطق.
إذا سارت الأمور على ما يرام ، فسيؤدي النقر فوق سيارة من الشاشة السابقة إلى ظهور نموذج مشابه لـ:



Sam's Used Cars

2012 Honda Pilot

Asking Price: 36999.00

Exterior Color: White Diamond Pearl

Interior Color: Leather

[شفرة](#)

```

1. <html>
2. <head>
3. <title>Sam's Used Cars</title>
4. </head>
5.
6. <body>
7.
8. <h1>Sam's Used Cars</h1>
9. <?php include 'db.php';
10.$vin = $_GET['VIN'];
11.$query = "SELECT * FROM INVENTORY WHERE VIN='$vin'";
12./* Try to query the database */
13.if ($result = $mysqli->query($query)) {
14. // Don't do anything if successful.
15.
16.else
17.
18. echo "Sorry, a vehicle with VIN of $vin cannot be found " .
mysql_error()."<br>";
19.

```

```

20.// Loop through all the rows returned by the query, creating a
table row for each
21.while ($result_ar = mysqli_fetch_assoc($result)) {
22.    $year = $result_ar['YEAR'];
23.    $make = $result_ar['Make'];
24.    $model = $result_ar['Model'];
25.    $trim = $result_ar['TRIM'];
26.    $color = $result_ar['EXT_COLOR'];
27.    $interior = $result_ar['INT_COLOR'];
28.    $mileage = $result_ar['MILEAGE'];
29.    $transmission = $result_ar['TRANSMISSION'];
30.    $price = $result_ar['ASKING_PRICE'];
31.}

32.echo "$year $make $model </p>";
33.echo "<p>Asking Price: $price </p>";
34.echo "<p>Exterior Color: $color </p>";
35.echo "<p>Interior Color: $interior </p>";
36.
37.$mysqli->close();
38.?>
39.
40.</body>
41.</html>

```

شرح الكود

1. `<html>`

يفتح السطر الأول علامة HTML ويبدأ المستند.

2. `<head>`
3. `<title>Sam's Used Cars</title>`
4. `</head>`

السطور من 2 إلى 4 هي علامات الرأس ، وفي ما بينهما تحدد عنوان المستند ، "Sam's Used Cars".

8. `<h1>Sam's Used Cars</h1>`

السطر 8 هو HTML عادي ؛ يقوم بطباعة سيارات سام المستعملة في أعلى الصفحة بنمط عنوان رئيسي.

9. <?php include 'db.php';

يحدد السطر 9 أن البرنامج النصي الحالي يتضمن ملف db.php الذي يسجل الدخول إلى قاعدة بيانات MySQL.

10. \$vin = \$_GET['VIN'];

ينشئ السطر 10 متغيرًا يسمى \$vin ويخصص له القيمة التي تتبع VIN في سلسلة URL. لكي يعمل هذا النموذج ، يجب عليك تمرير رقم VIN مثل هذا: /viewcar.php?VIN=123234FE221. نستخدم الأمر \$_GET لأنه عندما ترسل نموذجًا باستخدام الحصول على القيم ، يتم إلهاقها بعنوان URL بطريقة مماثلة.

11. \$query = "SELECT * FROM INVENTORY WHERE VIN='\\$vin"';

يبني السطر 11 استعلامًا باستخدام القيمة التي تم تمريرها إلى النموذج في سلسلة الاستعلام ، وب الخاص بها إلى الاستعلام المتغير المسمى بذكاء. انظر لماذا نسميها "سلسلة الاستعلام"؟

```
12. /* Try to query the database */
13. if ($result = $mysqli->query($query)) {
14. // Don't do anything if successful.
15. }
16. else
17. {
18. echo "Sorry, a vehicle with VIN of $vin cannot be
found ". mysql_error()."<br>";
19. }
```

تقوم السطور 12-19 بتشغيل الاستعلام مقابل قاعدة بيانات MySQL وإنشاء شيء يسمى "مجموعة النتائج". مجموعة النتائج هي مجموعة البيانات التي تنتج من تشغيل الاستعلام. يتم تعريف مجموعة النتائج هذه إلى المتغير \$result.

**20. // Loop through all the rows returned by the query,
creating a table row for each**

```
21. while ($result_ar = mysqli_fetch_assoc($result)) {
22. $year = $result_ar['YEAR'];
23. $make = $result_ar['Make'];
24. $model = $result_ar['Model'];
25. $trim = $result_ar['TRIM'];
26. $color = $result_ar['EXT_COLOR'];
27. $interior = $result_ar['INT_COLOR'];
28. $mileage = $result_ar['MILEAGE'];
29. $transmission = $result_ar['TRANSMISSION'];
30. $price = $result_ar['ASKING_PRICE'];
31. }
```

خطوات 20-31 حلقة خالل "كل" الصنوف التي تم إرجاعها كنتيجة للاستعلام. في حالتنا ، نظرًا لأن VINs فريدة من نوعها ، فإننا نتوقع فقط استعادة صفات واحدة من البيانات مرة أخرى ، لكننا نستخدم بشكل أساسي نفس التقنية التي تعلمناها في القسم السابق - النماذج التي تعرض بيانات الملخص.

```

22. $year = $result_ar['YEAR'];
23. $make = $result_ar['Make'];
...

```

تحدد الأسطر من 22 إلى 30 سلسلة من المتغيرات بقيم أعمدة البيانات المحددة ، والتي تتطابق مع أسماء الأعمدة في جدول قاعدة البيانات "مخزون".

31. }

يغلق السطر 31 حلقة while.

[النماذج التي تقوم بتحرير البيانات](#)

إذا فهمت كيفية إنشاء نماذج تضييف بيانات إلى قاعدة بيانات ، وفهمت النماذج التي تعرض بيانات التفاصيل ، فليس من الامتداد (من الناحية المفاهيمية على أي حال) لإنشاء نموذج يقوم بتحرير البيانات. ما عليك سوى إنشاء نموذج مثل النموذج الذي قمت بإنشائه لإضافة البيانات ، ولكن قبل عرضه ، يمكنك استرداد البيانات من قاعدة البيانات وتعيينها مسبقاً بالقيم.

بدلاً من تنفيذ أمر SQL Insert عندما ينقر المستخدم فوق إرسال ، قم بتنفيذ تحديث بدلاً من ذلك.

[النماذج التي تحذف البيانات](#)

لتحذف سجل معين من قاعدة بيانات ، تحتاج إلى طريقة للمستخدم لتحديد البيانات التي يريد حذفها. لقد تعلمت بالفعل كيفية القيام بذلك في قسم النماذج التي تعرض بيانات الملخص. في قسم تعديل النموذج للارتباط بصفحة التفاصيل ، أنشأنا رابط [HREF](#) ينقل المستخدم إلى صفحة تفاصيل ، ويمكنك استخدام نفس الأسلوب لنقلهم إلى صفحة حذف ، مثل تلك الموضحة أدناه:

رمز لحذف البيانات

```

1. <html>
2. <head>
3. <title>Sam's Used Cars</title>
4. </head>
5. <body>
6. <h1>Sam's Used Cars</h1>
7. <?php
8. include 'db.php';
9. $vin = $_GET['VIN'];
10.$query = "DELETE FROM INVENTORY WHERE VIN='$vin'";
11.echo "$query <BR>";
12./* Try to execute the DELETE against the database

```

```

22.$mysqli->close();
23.
24.?>
25.
26.</body>
27.</html>

```

شرح الكود

```

1.<html>
2.<head>
3.<title>Sam's Used Cars</title>
4.</head>
5.<body>
6.<h1>Sam's Used Cars</h1>

```

تحدد الأسطر 6-1 أساسيات الصفحة. نفتح ، <html>ونفتح ونغلق علامات ، <head>ونبدأ الجسم بعنوان رئيسي يعلن "Sam's Used

```

7. <?php
8. include 'db.php';

```

الأسطر من 6 إلى 7 مألوفة لنا الآن. نفتح علامة php ونضيف سطر الإدراج لربطنا بقاعدة بيانات MySQL الخاصة بنا.

```

9. $vin = $_GET['VIN'];
10. $query = "DELETE FROM INVENTORY WHERE
VIN='$vin'";
11. echo "$query <BR>";

```

يحصل السطر 9 على رقم VIN من سلسلة الاستعلام. تذكر ، سيتم استدعاء هذه الصفحة مع؟ " .. = "23ABC VIN ملحق بالنهاية. يعني السطر 10 بيان حذف SQL باستخدام ، ، لذلك نحن نعرف السيارة التي يجب حذفها. يكتب السطر 11 ببساطة الاستعلام على الشاشة حتى نتمكن من رؤية الاستعلام الذي أنشأناه. ليس مطلوباً بشكل صارم أن تعمل الوظيفة.

```

14. if ($result = $mysqli->query($query)) {
15. Echo "The vehicle with VIN $vin has been deleted.";
16. }
17. else
18. {
19. echo "Sorry, a vehicle with VIN of $vin cannot be
found ". mysql_error()."<br>";
20. }

```

تقوم السطور من 14 إلى 20 بالعمل الفعلي. ينفذ السطر 14 الاستعلام ، ويعيد True إذا نجح الاستعلام. إذا كان الأمر كذلك ، فإن السطر 15 يطبع رسالة نجاح على الشاشة ، وإذا لم يكن الأمر كذلك ، فإن السطر 19 يطبع رسالة فشل على الشاشة.

```
22. $mysqli->close();  
23.  
24. ?>  
25.  
26. </body>  
27. </html>
```

تغلق بقية الصفحة اتصال قاعدة البيانات ، وتغلق علامة ، و `php` وتغلق علامة النص الأساسي ، وأخيراً تغلق علامة `html`.

يمارس

إضافة وظيفة التحرير والحذف ، ما عليك سوى إضافة عمودين جديدين إلى الجدول مع روابط للتحرير والحذف ، واستدعاء صفحة المناسبة. تم توفير `deletecar.php` بينما يجب أن تقوم بإدخال `Editcar.php` بنفسك. إذا لم تتمكن مطلقاً من تشغيل `Editcar.php` فقد قمت بتضمينه في نموذج التعليمات البرمجية. فقط اعمل لنفسك معروفاً وجرب أن تتحققه.

للإجابة على هذا التحدي ، انظر إلى نموذج عرض الصفحة ، `carswithstyle2.php` والذي تم تضمينه في نموذج التعليمات البرمجية.

13

متغيرات الجلسة

مقدمة

عادة ما يكون للمتغيرات في PHP نطاق محدد ومحدود - بشكل عام ، المتغير متاح فقط في الصفحة التي تم الإعلان عنها. الاستثناء الرئيسي لهذه القاعدة هو عندما تقوم بتعريف متغير داخل دالة ، فإنه يعمل فقط في هذه الوظيفة.

ولكن ماذا لو كنت تريدين الوصول إلى نفس المتغير عبر صفحات متعددة في تطبيقك؟ على سبيل المثال ، أنا متسوق منتظم على Amazon.com. إذا كنت كذلك ، فربما لاحظت أنه بمجرد تسجيل الدخول ، ستتجدد اسمك أعلى كل صفحة .



من المفترض أن هناك متغيراً في نص ما يسمى شيئاً مثل \$FirstName مثل "Alan" ، ربما يمكنك بسهولة كتابة مثل هذا السيناريو. هنا تلميح:

```
echo "$FirstName's Amazon.com";
```

ولكن كيف تنتقل هذه القيمة \$FirstName من صفحة إلى أخرى وأنا أتجول في الموقع؟ وكيف يتبع الموقع مئات متغيرات \$FirstName الفريدة لجميع العملاء الفريدين الذين يتواجدون على الموقع في نفس الوقت؟ الجواب هو متغيرات الجلسة.

الجلسات

متغير الجلسة هو نوع خاص من المتغيرات ، بمجرد تعبيئه ، يكون متاحاً لجميع الصفحات في التطبيق طالما كان المستخدم مفتوحاً لمتصفحه ، أو حتى يتم إنتهاء الجلسة صراحةً من قبل المطور (أنت).

إن الشيء العظيم في متغيرات الجلسة هو أن PHP سوف تتبع بطريقة سحرية متغير جلسة معين يذهب مع كل مستخدم معين. لذا ، في حين أن تجربتي في Amazon.com ستقول لك شيئاً مختلفاً (ما لم يكن اسمك هو Alan بالطبع). تعمل الجلسات من خلال إنشاء معرف فريد (UID) لكل زائر وتخزين المتغيرات على أساس هذا المعرف الفريد. عادةً ما يتم تخزين المعرف الفريد العمومي (UID) في ملف تعريف ارتباط.

A cookie, also known as an HTTP cookie, web cookie, or browser cookie, is a small piece of data sent from a website and stored in a user's web browser while a user is browsing a website.

لا يهم حقاً كيفية عملهم ، الشيء المهم هو أنهم يقومون بعملهم . وهم رائعون جداً إنها تفتح مجالاً كاملاً من الاحتمالات لتخفيض تطبيق الويب الخاص بك لعميل معين. على سبيل المثال ، في حالة Sam's Used Cars يمكن أن تطلب من العميل لون السيارة المفضل لديه ، والنوع / الطراز ، والميزات ، وما إلى ذلك. من هذه النقطة فصاعداً ، يمكنك تخصيص الصفحات لتعكس تفضيلات العملاء. على سبيل المثال ، انظر ، هذه السيارة بها فتحة سقف! (وهو أحمر أيضا!) إنه مجرد نموذج للتطبيق ، لذا لا يأس من كتابة ميزات مزعجة لتعلم شيء ذي قيمة.

بمجرد أن يغلق المستخدم متصفحه ، سيتم مسح ملف تعريف الارتباط وستنتهي الجلسة. لذا فإن الجلسات ليست مكاناً جيداً لتخزين البيانات التي تبوي الاحتفاظ بها لفترة طويلة. المكان المناسب لتخزين البيانات طويلة المدى هو قاعدة البيانات. بالطبع ، يمكن أن تعمل الجلسات وقواعد البيانات معًا. على سبيل المثال ، يمكنك تخزين تفضيلات المستخدم في قاعدة بيانات ، واستردادها من قاعدة البيانات عندما يقوم المستخدم "بتسجيل الدخول" أو كتابة عنوان بريده الإلكتروني أو القيام بكل ما قمت بترميزه له لتعريف نفسه. بمجرد استرداد البيانات ، قم بتعيين التفضيلات لمتغيرات الجلسة وستكون متاحة منذ ذلك الحين.

بدء جلسة PHP

قبل أن تتمكن من تخزين معلومات المستخدم في جلسة PHP الخاصة بك ، يجب عليك أولاً بدء الجلسة باستخدام وظيفة `session_start()`. قبل ظهر الدالة `session_start()` يجب أن تظهر علامة <html> وإلا فلن تعمل.

```
<?php session_start(); ?>

<html>
    <body>
        <p>Hello world</p>
    </body>
</html>
```

سيبدأ الكود أعلاه جلسة المستخدم مع الخادم ويسمح لك ببدء حفظ معلومات المستخدم في متغيرات الجلسة.

استخدام متغيرات الجلسة

الطريقة الصحيحة لتخزين واسترداد متغيرات الجلسة هي استخدام متغير `$_SESSION` :

تخزين متغير

```
<?php
session_start();
// store session data
$_SESSION['FirstName']='Alan';
?>

<html>
<body>
```

استرجع متغير

```
<?php
//retrieve session data
echo $_SESSION['FirstName']."'s Amazon";
?>
```

الإخراج: آلان أمازون

التحقق من وجود متغير يمكنك التحقق لمعرفة ما إذا كان متغير جلسة متاحًا أم لا باستخدام وظيفة .() iset

bool iset (mixed \$var [, mixed \$...])

Determines if a variable is set and is not NULL. You can unset a variable with **unset()**.

هذا مثال:

```
<?php
session_start();
if(isset($_SESSION['FirstName']))
echo $_SESSION['FirstName']."'s Amazon";
else
$_SESSION['views']=1;
echo "Welcome to Amazon";
?>
```

إتلاف الجلسة

إذا كنت ترغب في حذف بعض بيانات الجلسة ، يمكنك استخدام وظيفة .() unset() إذا كنت تريد حذفها كلها ، فاستخدم وظيفة .() session_destroy() لحذف متغير جلسة محدد:

```
<?php  
session_start();  
if(isset($_SESSION['FirstName']))  
    unset($_SESSION['FirstName']);  
?>
```

يمكنك أيضًا تدمير الجلسة بالكامل عن طريق استدعاء وظيفة () session_destroy()

```
<?php  
    session_destroy();  
?>
```

ملاحظة: ستعيد () session_destroy() جلستك وستفقد جميع بيانات الجلسة المخزنة. هذه طريقة سهلة لتنفيذ وظيفة تسجيل الخروج.

إذا كنت ترغب في معرفة المزيد عن متغيرات الجلسة ، فلدي كتاب كامل حول هذا الموضوع بعنوان "متعة PHP: الغوص العميق في الجلسات".

14

العمل مع الصور

مقدمة

لن يكون موقع الويب الخاص بالسيارات المستعملة ذا فائدة كبيرة لمتسوق السيارات العادي دون تقديم صور للسيارات ، لذلك سنغطي في هذا الفصل كيفية إضافة الصور إلى موقعنا. سيكون الأمر بسيطًا إلى حد ما إذا كان لكل سيارة صورة واحدة مرتبطة بها -في هذه الحالة ، يمكننا ببساطة إضافة عمود إضافي إلى جدول المخزون لدينا يسمى "صورة" (أو شيء وصفي مماثل ، مثل "صورة_أولية") والتي من شأنها تخزين اسم ملف الصورة المرتبط بالسيارة المعينة.

ثم نقوم ببناء PHP لاسترداد اسم الصورة وإدخاله في علامة صورة HTML في صفحة تفاصيل السيارة.

To display an image in your HTML page, use the syntax **** in the simplest form or add optional parameters to specify height and width, and text to display if the image isn't available (or for screen readers) such as ****

بالطبع ، PHP مناسبة تماماً لذلك. سنقرأ اسم الملف من قاعدة البيانات ونستخدم PHP لإنشاء علامة الصورة ديناميكياً.

على سبيل المثال ، يمكننا تعديل مثالنا السابق ، والذي يوضح التفاصيل الخاصة بسيارة معينة عن طريق إضافة الخطوط المميزة باللون الأحمر على النحو التالي:

```
1. <html>
2. <head>
3. <title>Sam's Used Cars</title>
4. </head>
5.
6. <body>
7.
8. <h1>Sam's Used Cars</h1>
9. <?php include 'db.php';
10.$vin = $_GET['VIN'];
11.$query = "SELECT * FROM INVENTORY WHERE VIN='".$vin."'";
12./* Try to query the database */
13.if ($result = $mysqli->query($query)) {
14. // Don't do anything if successful.
15.}
16.else
17.{ 
18. echo "Sorry, a vehicle with VIN of $vin cannot be found " .
mysql_error()."<br>";
19.}
20.// Loop through all the rows returned by the query, creating a
table row for each
21.while ($result_ar = mysqli_fetch_assoc($result)) {
22. $year = $result_ar['YEAR'];
23. $make = $result_ar['Make'];
24. $model = $result_ar['Model'];
25. $trim = $result_ar['TRIM'];
26. $color = $result_ar['EXT_COLOR'];
27. $interior = $result_ar['INT_COLOR'];
28. $mileage = $result_ar['MILEAGE'];
29. $transmission = $result_ar['TRANSMISSION'];
```

```

31. $image = $result_ar['Primary_Image'];
32.}
33.echo "$year $make $model </p>";
34.echo "<p> Asking Price: $price </p>";
35.echo "<p> Exterior Color: $color </p>";
36.echo "<p> Interior Color: $interior </p>";
37.echo "<IMG src='$image'>";
38.
39.$mysqli->close();
40.?>
41.
42.</body>
43.</html>

```

يفترض هذا المثال أن لدينا عموداً في قاعدة البيانات الخاصة بنا يسمى ، والذى يحتوي على اسم ملف ملف Primary_Image الصورة المخزن على خادمنا. تحتوى الصفحة الرئيسية لملفات العينة على برنامج نصي يقوم بإجراء هذا التعديل ، إذا كنت تميل إلى هذا الحد.

إذا كانت الصور في مجلد يسمى "الصور" ، فسيقرأ السطر:

```
echo "<IMG src='images\\$image'>";
```

تمرين: عرض الصور احصل على المثال أعلاه للعمل. قم بإنشاء مجلد صور أسفل المجلد الذي يقوم تشغيل تطبيق مجموعة السيارات ووضع بعض الصور فيه. قم بتعديل جدول المخزون لإضافة حقل Primary_Image وإدخال بعض القيم في هذا الحقل لربط سيارات معينة بصور معينة.

قم بعمل نسخة من البرنامج النصي viewcar-backup.php (أطلق عليه اسم viewcar.php إذا احتجت إليه لاحقاً) ، ثم قم بتعديل viewcar.php كما هو موضح باللون الأحمر أعلاه بحيث يقرأ موقع الصورة خارج قاعدة البيانات ويدرج صورة في الصفحة . باستخدام علامة

سحب عدد غير معروف من الصور من قاعدة بيانات

بافتراض أنك حصلت على التمرين أعلاه للعمل ، يجب أن تعرف أنه أمر رائع. تهانينا ، أنت رائع رسمياً. لكن يمكننا أن نفعل أكثر من ذلك بكثير. مجرد وجود صورة واحدة لسيارة لا يعكس حقيقة توقع الزائر لموقع السيارة. من المرجح أن يرغب زائر موقع Sam's Used Cars على الويب في رؤية العديد من الصور لسيارة يهتم بها ، وسيتعين على موقعنا استيعاب ذلك. قد تحتوي بعض السيارات على صورة واحدة فقط ، بينما قد تحتوي بعض السيارات على 10 أو أكثر. ستكون مختلفة لكل سيارة. فكيف نحقق هذا؟ من الواضح أن وجود عمود واحد يسمى Primary_Image ليس هو الحل الدائم. بمجرد أن تعرضه على سام ، سيقول بالتأكيد "لكن ماذا لو كان لدى صورتان للسيارة لأعرضهما؟" هذه هي طبيعة تطوير الويب في بعض الأحيان. فكرة جيدة واحدة تطلق شرارة أخرى. لا تحصل

محبّط عندما يحدث هذا ، لكن فكر في نفسك ، "واو ، لقد ألهمت فكرة أفضل!"

أسهل طريقة للتعامل مع عدد متغير من الصور هي إنشاء جدول قاعدة بيانات لتخزينها فيه.

دعنا نضيف جدولًا يسمى "الصور" إلى قاعدة بيانات السيارات الخاصة بنا. يجب أن يحتوي على معرف الأعمدة و VIN و ImageFile.

تمرين: إنشاء جدول قاعدة بيانات لتخزين الصور. استخدم phpMyAdmin لإنشاء هذا الجدول ، على هذا النحو.

The screenshot shows the phpMyAdmin interface for creating a new table named 'cars'. The table structure is defined with three columns:

| # | Column | Type | Collation | Attributes | Null | Default | Extra | Action |
|---|-----------|--------------|-------------------|------------|------|---------|----------------|---|
| 1 | ID | int(11) | | | No | None | AUTO_INCREMENT | Change Drop More |
| 2 | VIN | varchar(17) | latin1_swedish_ci | | No | None | | Change Drop More |
| 3 | ImageFile | varchar(200) | latin1_swedish_ci | | No | None | | Change Drop More |

Below the table structure, there are buttons for 'Add column(s)' and 'Go'.

أنت الآن بحاجة إلى ملء الجدول ببعض البيانات النموذجية. هذا ما فعلته. انتقل إلى <http://www.cars.com> وأبحث عن بعض السيارات. انسخ رقم VIN إلى الحافظة ، واحفظ بعض صور السيارة على محرك الأقراص الثابتة. أدخل صُفًّا في جدول الصور لكل صورة تحفظها ، وأدخل رقم تعريف السيارة لكل صورة. يجب أن يكون هناك أيضًا إدخال مقابل في جدول الجرد لتلك السيارة ، بنفس رقم تعريف السيارة بالضبط. من السهل القيام بذلك في phpMyAdmin. لا تقلق بشأن محاولة أتمتة هذا الجزء منه حتى الآن.

تمرين: قم بتعديل صفحة viewcar.php لعرض صور متعددة بمجرد أن يكون لديك بعض البيانات النموذجية التي تطابق أرقام VIN محددة مع صور محددة ، فمن السهل جدًا عرض تلك الصور على الصفحة مع وصف السيارة. إليك مقتطف الشفرة الذي يمكنك إلحاقه بنموذج viewcars.php لاستخراج أسماء الصور للسيارة المحددة.

افتراض هذا البرنامج النصي هو أن لديك رقم VIN للسيارة في المتغير ، \$ vin وأنك قمت بتضمين "db.php" لإنشاء اتصال قاعدة البيانات.

```

1. <?php
2. $query = "SELECT * FROM images WHERE VIN='$vin'";
3. /* Try to query the database */
4. if ($result = $mysqli->query($query)) {
5.     // Got some results
6.     // Loop through all the rows returned by the query, creating a
7.     // table row for each
8.     while ($result_ar = mysqli_fetch_assoc($result)) {
9.         $image = $result_ar['ImageFile'];
10.        echo "<img src='uploads/$image' width= '250'> ";
11.    }
12.$mysqli->close();

```

وأوضح الكود

**2. \$query = "SELECT * FROM images WHERE
VIN='\$vin'" ;**

يقوم السطر 2 بإعداد الاستعلام حيث نختار جميع الحقول في جدول الصور للسيارة المحددة. (WHERE VIN =).

4. if (\$result = \$mysqli->query(\$query)) {

يقوم السطر 4 بتشغيل الاستعلام والتحقق لمعرفة ما إذا تم إرجاع أي نتائج من قاعدة البيانات.

**7. while (\$result_ar = mysqli_fetch_assoc(\$result)) {
8. \$image = \$result_ar['ImageFile'];
9. echo " "
10. }**

الأسطر 10 - 7 حلقات خلال مجموعة النتائج عدة مرات مثل الصنفوف. بمعنى آخر ، إذا كانت هناك خمس صور لسيارة معينة ، فسيتم إرجاع خمسة صنفوف من البيانات وستدور حلقة while حوالي خمس مرات.

**11. }
12. \$mysqli->close();**

يغلق السطر 11 عبارة if ويغلق السطر 12 الاتصال بقاعدة بيانات MySQL.

15

تحميلات ملف PHP

مقدمة

في القسم السابق ، قمنا بالتقاط صور لسياراتنا ثم حفظناها يدوياً على القرص الصلب. هذا رائع ، لكنه ممل. ما سيكون رائعاً حقاً هو اختيار سيارة في مخزوننا والنقر فوق زر يسمى "إضافة صورة" ، والسماح للنص البرمجي بمعالجة الباقي - وضع الملف في المكان المناسب وإنشاء الصف الصحيح في جدول الصور باستخدام رقم VIN للسيارة المختارة.

هذا ما سنفعله بعد ذلك.

قم بإنشاء نموذج تحميل ملف

في أبسط أشكاله ، إليك نموذج HTML يمكنك استخدامه لتحميل ملف.

```
<html>
<body>

<form action="upload_file.php" method="post"
enctype="multipart/form-data">
    <label for="file">Filename:</label>
    <input type="file" name="file" id="file"><br>
    <input type="submit" name="submit"
value="Submit">
</form>

</body>
</html>
```

هناك بعض الأشياء التي تستحق الإشارة إليها.

أولاً ، لاحظ سمات النموذج: الإجراء 'upload_file.php' يعني أنه عند النقر فوق زر الإرسال ، سيتم تمرير نتيجة منشور النموذج إلى البرنامج النصي upload_file.php لمزيد من المعالجة. بعد ذلك ، يعد "multipart / form-data" بالنسبة لـ enctype = "multipart / form-data".

نحن هنا نحدد نوع الترميز الذي سيستخدمه النموذج. يجب عليك تحديد أنه متعدد الأجزاء / بيانات النموذج إذا كنت تقوم بتضمين عنصر تحكم تحميل ملف في نموذج ، لذلك يعرف المتصفح أن يمرر الملف كملف ، وليس مجرد كتلة نصية كبيرة أخرى.

لدينا أيضًا نوع جديد من مربعات الإدخال. في الماضي ، كنا نستخدم مربعات الإدخال في الغالب للسماح للمستخدمين بكتابة نص. عندما تحدد أن نوع الإدخال = "ملف" ، يتعامل المستعرض معه بشكل مختلف. سيوضع زر تصفح بجوار حقل الإدخال ،

السماح للمستخدم بتحديد ملف من جهاز الكمبيوتر الخاص به.

قم بإنشاء برنامج نصي لمعالجة الملف الذي تم تحميله

حدد النموذج أعلاه أن المنشور تتم معالجته بواسطة "upload_file.php". يستخدم هذا البرنامج النصي لعمل شيء مع الملف بمجرد تحميله. البرنامج النصي الذي يليه يردد ببساطة إلى المتصفح بعض سمات الملف الذي تم تحميله للتو. هناك ، بالطبع ، سمات ملفات أخرى لن نغطيها ، لأنك ربما لن تحتاج إلى استخدامها على الإطلاق.

```
<?php
if ($_FILES["file"]["error"] > 0)
{
    echo "Error: " . $_FILES["file"]["error"] . "<br>";
}
else
{
    echo "Upload: " . $_FILES["file"]["name"] . "<br>";
    echo "Type: " . $_FILES["file"]["type"] . "<br>";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . "
kB<br>";
    echo "Stored in: " . $_FILES["file"]["tmp_name"];
}
?>
```

لقد أبرزت الأجزاء التي يجب أن تتطابق باللون الأصفر. بمعنى آخر ، إذا كان اسم عنصر تحكم الإدخال في نموذج التحميل يشير إلى الملف ك ، مثل ، </input> فسيتعين عليك أيضًا الإشارة إليه على أنه foo في البرنامج النصي التالي ، مثل .
\$_FILES ["foo"] ["name"]. الاسم الفعلي لا يهم ، ولكن ما يهم هو الاتساق.

عندما تقوم بتحميل ملف باستخدام ، يتم تخزين الملف في مجلد مؤقت. ما لم تفعل شيئاً محدداً مع الملف ، فسيختفي قريباً.

بالنسبة إلى ، Sam's Used Cars فإن الشيء المثالي الذي يجب فعله هو تحميل الملف ، ونسخ الملف في مجلد معين ، ثم إنشاء سجل في جدول الصور يُدرج VIN للمركبة المناسبة واسم ملف الصورة التي حملناها للتو. في نموذج البيانات ، راجع البرنامج النصي ViewCarsAddImage.php لرؤية هذا المفهوم الدقيق قيد التنفيذ.

ViewCarsAddImage.php الكود:

```
1. <?php
2. include 'db.php';
3. $vin = trim($_POST['VIN']);
4. if ($_FILES["file"]["error"] > 0)
5. {
6. echo "Error: " . $_FILES["file"]["error"] . "<br>";
7. }
8. else
9. {
10. echo "Upload: " . $_FILES["file"]["name"] . "<br>". "\n";
11. echo "Type: " . $_FILES["file"]["type"] . "<br>". "\n";
12. echo "Size: " . ($_FILES["file"]["size"] / 1024) . " kB<br>". "\n";
13. echo "VIN: ".$vin."<br>";
14. echo "Stored temporarily as: " .
$_FILES["file"]["tmp_name"]."<br><BR>". "\n";
15. $currentfolder = getcwd();
16. echo "This script is running in: " . $currentfolder."<br>". "\n";
17. $target_path = getcwd() ."/uploads/";
18. echo "The uploaded file will be stored in the folder:
".$target_path."<br>". "\n";
19.
20. $target_path = $target_path . basename( $_FILES['file']['name']);
21. $imagename = "uploads/". basename( $_FILES['file']['name']);
22. echo "The full file name of the uploaded file is "".
$target_path."<br>". "\n";
23.
```

```
24. echo "The relative name of the file for use in the IMG tag is " .  
$imagename ."  
  
25.  
26.if(move_uploaded_file($_FILES['file']['tmp_name'], $target_path)) {  
27. echo "The file ". basename( $_FILES['file']['name']). " has been  
uploaded  
28.  
29. // Create a database entry for this image  
30. if (mysqli_connect_errno()) {  
31.     printf("Connect failed: %s\n", mysqli_connect_error());  
32.     exit();  
33. }  
34.  
35. echo 'Connected successfully to mySQL. <BR>';  
36. $file_name = $_FILES["file"]["name"];  
37. $query = "INSERT INTO images (VIN, ImageFile) VALUES ('$vin',  
'$file_name')";  
38. echo $query."  
\n";  
39. echo "<a href='AddImage.php?VIN=";  
40. echo $vin;  
41. echo ">Add another image for this car </a></p>\n";  
42./* Try to insert the new car into the database */  
43.if ($result = $mysqli->query($query)) {  
44.     echo "<p>You have successfully entered $target_path into  
the database.</P>\n";  
45.  
46. }  
47. else  
48. {  
49.     echo "Error entering $VIN into database: " .  
mysql_error()."<br>";  
50. }
```

```

51. $mysqli->close();
52. echo "<img src='$imagename' width='150'><br>";
53.
54. } else{
55. echo "There was an error uploading the file, please try again!";
56. }
57. }
58.
59. include 'footer.php'
60.?>

```

شرح الكود

1.<?php
2. include 'db.php';

يفتح السطر الأول علامة ، ويضيف السطر 2 ملف التضمين الضروري للاتصال بقاعدة البيانات الخاصة بنا.

3. \$vin = trim(\$_POST['VIN']);

ينشئ السطر 3 متغيراً يسمى \$vin ويخصص له القيمة التي تم تمريرها إليه باستخدام النموذج الذي تم ترحيله. مرة أخرى ، راجع هذا عملياً مع نماذج البرامج النصية المضمنة في هذا الكتاب. ليست هذه هي الطريقة الوحيدة التي كان بإمكاننا القيام بها. يمكننا أيضاً تمرير VIN في سلسلة استعلام ، وهي التقنية التي استخدمناها في viewcar.php

```

4. if ($_FILES["file"]["error"] > 0)
5. {
6. echo "Error: " . $_FILES["file"]["error"] . "<br>";
7. }

```

تحتبر السطور من 4 إلى 7 لمعرفة ما إذا كان الملف قد تم تحميله بالفعل. إذا لم يكن الأمر كذلك ، فسيتم طباعة خطأ باستخدام السطر 6.

8. else
9. {

بدءاً من السطر ، 9 يبدأ البرنامج النصي في معالجة الملف الذي تم تحميله.

```

10. echo "Upload: " . $_FILES["file"]["name"] . "<br>".
"\n";
11. echo "Type: " . $_FILES["file"]["type"] . "<br>".
"\n";
12. echo "Size: " . ($_FILES["file"]["size"] / 1024) . "
kB<br>" . "\n";
13. echo "VIN: ".$vin."<br>";

```

طبع الأسطر 10-12 معلومات حول الملف ، بينما يطبع السطر 13 رقم ، بينما يطبع السطر 13 VIN فقط للتأكد من أننا حصلنا عليه دون أي مشاكل.

```
14. echo "Stored temporarily as: " . $_FILES["file"]
["tmp_name"]."<br><BR>". "\n";
```

يخبرنا السطر 14 بالاسم الذي استخدمته لتخزين الملف الذي تم تحميله مؤقتاً.

للمزيد: لاحظ أنه في نهاية السطر ، أكتب أيضًا ، "\n" مما يعني إضافة سطر جديد في نهاية هذا السطر. لا يؤثر هذا على البرنامج النصي على الإطلاق ، ولكنه يضع سطراً جديداً على HTML الذي تم إنشاؤه بواسطة البرنامج النصي. يؤدي وضع \n في نهاية السطر في البرامج النصية إلى تسهيل قراءة تعليمات HTML البرمجية عندما تنظر إلى صفحة ما وتحدد "عرض المصدر" - وهو أمر يتبع على كل مطور PHP القيام به من وقت آخر.

```
15. $currentfolder = getcwd();
16. echo "This script is running in: " . $currentfolder."
<br>". "\n";
```

يستخدم السطر 15 الأمر getcwd() لمعرفة اسم المجلد الذي يعمل فيه البرنامج النصي الحالي. لماذا أردت ذلك؟ لأنني أرغب في وضع الملف الذي تم تحميله في مجلد موجود ضمن المجلد الحالي ، وللقيام بذلك تحتاج إلى معرفة مكانك. يخرج السطر 16 ما تعلمته للتو.

getcwd() The getcwd() function returns the current directory. This function returns the current directory on success and FALSE on failure.

```
17. $target_path = getcwd() . "/uploads/";
```

في السطر ، 17 أنشأنا متغيراً يسمى \$target_path وقمنا بتعيين قيمة له عن طريق إضافة سلسلتين معاً باستخدام . شخصية. الملفات التي تم تحميلها هي للطابع /uploads/. نحن نقوم بإنشاء المسار الهدف لتحديد المكان الذي نريد وضع

```
18. echo "The uploaded file will be stored in the folder:
".$target_path."<br>". "\n";
```

يخرج السطر 18 نتيجة الحساب لتعيين المسار الهدف.

```
20. $target_path = $target_path . basename(
$_FILES['file']['name']);
```

في السطر ، 20 قمنا بتعديل المسار الهدف مرة أخرى ، هذه المرة بـ إلأحاق اسم الملف الأصلي للملف الذي تم تحميله به.

```
21. $imagename = "uploads/". basename(
$_FILES['file']['name']);
```

يحسب السطر 21 اسم ملف الصورة فقط بدون مسار الملف بالكامل. هذا لأنه عندما تعمل مع علامات ، لا يتبع عليك تحديد المسار الكامل للصورة ؛ ما عليك سوى تحديد مكانها بالنسبة إلى مكان وجودك.

```

22. echo "The full file name of the uploaded file is "".
$target_path."<br>". "\n";
23.
24. echo "The relative name of the file for use in the
IMG tag is " . $imagename ."<br><br>". "\n";

```

يُخرج السطران 22 و 24 قيم هذه الحسابات حتى تتمكن من معرفة النتيجة. بالطبع ، إذا كان هذا موقع ويب " حقيقياً " لسيارة مستعملة ، فلن ترغب في نقل كل هذه المعلومات الإضافية إلى المتصفح.

The reason for all these echo statements is so you can follow along with the code as it executes.

```

26. if(move_uploaded_file($_FILES['file']['tmp_name'],
$target_path)) {
27. echo "The file ". basename( $_FILES['file']
['name']). " has been uploaded<br>". "\n";

```

ينقل السطر 26 الملف الذي تم تحميله من الموقع المؤقت المعين بواسطة PHP إلى المسار الهدف الذي حسبته في السطر 20. يخبرك السطر 27 بهذه الحقيقة.

```

29. // Create a database entry for this image
30. if (mysqli_connect_errno()) {
31.     printf("Connect failed: %s\n",
mysqli_connect_error());
32.     exit();
33. }
34.
35. echo 'Connected successfully to mySQL. <BR>';

```

بعد ذلك ، نريد إنشاء سجل في جدول الصور يشير إلى ملف الصورة الجديد هذا. تمهد السطور من 29 إلى 35 الطريق لحدوث ذلك.

```

36. $file_name = $_FILES["file"]["name"];
37. $query = "INSERT INTO images (VIN, ImageFile)
VALUES ('$vin', '$file_name')";
38. echo $query."<br>\n";

```

في السطر 36 نحصل فقط على اسم الملف الذي تم تحميله ، دون أي معلومات عن المسار على الإطلاق. هذا لأننا نريد فقط إدراج اسم الملف في قاعدة البيانات. عند الإشارة إلى الملف لاحقاً بعلامة `` يمكننا دائمًا تحديد مسار إذا لزم الأمر.

يبني السطر 37 الاستعلام لإدراج السجل في قاعدة البيانات ، ويكتب السطر 38 ما هو الاستعلام. كان السطر 38 مفيداً للغاية عندما كنت أكتب هذا السيناريو في الأصل ، لأنه بالطبع لم ينجح في المرة الأولى التي جربته فيها. تعتبر رؤية الاستعلام الفعلي هي الخطوة الأولى لمعرفة سبب عدم نجاح استعلام معين.

```

39. echo "<a href='AddImage.php?VIN=';
40. echo $vin;
41. echo ">Add another image for this car </a>
</p>\n";

```

تنشئ الخطوط 39 إلى 41 رابطاً يتبع لنا إضافة صورة أخرى لهذه السيارة بسهولة إذا كان لدينا واحدة.

```

42. /* Try to insert the new car into the database */
43. if ($result = $mysqli->query($query)) {
44.     echo "<p>You have successfully entered
$target_path into the database.</P>\n";
45.
46. }
47. else
48. {
49.     echo "Error entering $VIN into database: " .
mysql_error()."<br>";
50. }
51. $mysqli->close();

```

تقوم الأسطر 42 إلى 51 بتنفيذ الاستعلام وطباعة رسالة نجاح أو فشل. يغلق السطر 52 الاتصال بـ MySQL.

```

52. echo "<img src='$imagename' width='150'>
<br>";

```

ينشئ السطر 52 علامة صورة للملف الذي حملناه للتو حتى تتمكن من رؤية شكله. عندما أنشأت هذا لأول مرة ، كانت الصور كبيرة جدًا لدرجة أنها استحوذت على الشاشة بأكملها ، لذلك أضفت عرض السمة '150' لإبقاء الصور بحجم معقول. هذا يخبر المتصفح بتغيير حجم الصورة.

16

مراجعات PHP ونصائح مقدمة

كل لغة لها مراجعاتها. عندما أواجه جواب PHP التي ليست بدبيهية على الفور ، أو إذا وجدت نصيحة رائعة يمكن أن يجعل حياتك أسهل ، فسوف تنتقل إلى هذا القسم.

اقتباسات مفردة مقابل اقتباسات مزدوجة

عند العمل مع الجمل ، من المهم فهم الاختلاف في كيفية تعامل PHP مع علامات الاقتباس المنفردة (صدى (: "Hello \$ name") مقارنة بعلامات الاقتباس المزدوجة (صدى (: "Hello \$ name")

ستعرض السلاسل المنفردة المقتبسة الأشياء "كما هي" تماماً. لن يتم استبدال المتغيرات بقيمها. المثال الأول أعلاه (':!\$ name') سيطبع Hello \$ name. المثال الثاني أعلاه ("Hello \$ name") سيطبع Hello Alan إذا احتوى المتغير \$ name على "Alan".

هذا شيء سهل الخلط ، لذا اقرأه مرة أخرى. (:

علامة المساواة

غالباً ما تكون علامة التساوي مصدراً للارتباك. يتم استخدام علامة المساواة الفردية لتعيين قيمة لمتغير ، على سبيل المثال : 'Alan' = \$ FirstName

يمكن أيضاً استخدام علامة التساوي للمقارنة بالقيم ، إذا جمعت اثنين منهم معاً وأدرجتهما في عبارة .f على سبيل المثال ، 'Alan' == \$ FirstName سيعود صحيحاً بالنسبة لي ، كما يوضح الكود التالي

```
<?php
$FirstName = 'Alan';
if ($FirstName == "Alan")
{
    echo "Hello Mr. Awesome";
}
else
{
    echo "Hello $FirstName";
}
?>
```

راجع [مقارنات عينة التعليمات](#) البرمجية الشيء الغريب في اختبار التساوي المزدوج هو أن PHP ستحاول تحويل المتغيرين اللذين تم مقارنتهما إلى أنواع مختلفة لمعرفة ما إذا كانت ستحصل على تطابق. على سبيل المثال ، إذا كان `$a = 1` و `$b = "1"` فقد تعتقد أنهما غير متساوين لأنهما نوعان مختلفان. (أحدهما رقم والآخر عبارة عن سلسلة.)

ومع ذلك ، فإن مقارنة `$a == $b` باستخدام المقارنة `==`سيعود إلى القيمة `true` ، لأنك إذا قمت بتحويل `$b` من سلسلة النوع إلى رقم النوع ، فسيكون المتغيرين متساوين.

When you convert a variable from one type to another type, that's called 'Casting'. I could also have written that PHP will cast the string into a number, then find the two variables equal, but you have to admit that does sound awful geeky. But I mention it because you may well come across the term.

إذا كنت ترغب في اختبار ما إذا كانت القيمتان لهما نفس القيمة ونفس النوع ، فيمكنك مقارنتهما باستخدام ثلاثة علامات متساوية. بهذه الطريقة ، سيعود `$a === $b` إلى القيمة `false`.

```
<html>
  <body>
    <h1>Comparison Operators</h1>
    <?php
      $FirstName = 'Alan';
      if ($FirstName == "Alan")
      {
        echo "Hello Master";
      }
      else
      {
        echo "Hello $FirstName";
      }
      echo "<br>";
      $a = 1;
      $b = "1";
      if ($a == $b)
      {
        echo '$a is equal to $b';
      }
      else
      {
        echo '$a is not equal to $b';
      }

      echo "<br><BR>";
      if ($a === $b)
      {
        echo '$a is equal to $b';
      }
      else
      {
        echo '$a is not equal to $b';
      }
    ?>
  </body>
</html>
```

عوامل المقارنة

| Example | Name | Result |
|-------------------------------|--------------------------|---|
| <code>\$a == \$b</code> | Equal | TRUE if <code>\$a</code> is equal to <code>\$b</code> after type juggling. |
| <code>\$a === \$b</code> | Identical | TRUE if <code>\$a</code> is equal to <code>\$b</code> , and they are of the same type. |
| <code>\$a != \$b</code> | Not equal | TRUE if <code>\$a</code> is not equal to <code>\$b</code> after type juggling. |
| <code>\$a <> \$b</code> | Not equal | TRUE if <code>\$a</code> is not equal to <code>\$b</code> after type juggling. |
| <code>\$a !== \$b</code> | Not identical | TRUE if <code>\$a</code> is not equal to <code>\$b</code> , or they are not of the same type. |
| <code>\$a < \$b</code> | Less than | TRUE if <code>\$a</code> is strictly less than <code>\$b</code> . |
| <code>\$a > \$b</code> | Greater than | TRUE if <code>\$a</code> is strictly greater than <code>\$b</code> . |
| <code>\$a <= \$b</code> | Less than or equal to | TRUE if <code>\$a</code> is less than or equal to <code>\$b</code> . |
| <code>\$a >= \$b</code> | Greater than or equal to | TRUE if <code>\$a</code> is greater than or equal to <code>\$b</code> . |

إذا قارنت رقمًا بسلسلة أو إذا تضمنت المقارنة سلاسل رقمية ، فسيتم تحويل كل سلسلة إلى رقم و يتم إجراء المقارنة رقميًا.

تنطبق هذه القواعد أيضًا على بيان التبديل. لا يحدث تحويل النوع عندما تكون المقارنة `==` أو `!=` لأن هذا يتضمن مقارنة النوع بالإضافة إلى القيمة.

17

اعتبارات أمنية

مقدمة

كما رأينا ، تعد PHP لغة سهلة التعلم للغاية ، وسيتعلّمها الكثيرون من الأشخاص الذين ليس لديهم أي خلفية رسمية في البرمجة بطريقة إضافة نشاط داخلي إلى موقع الويب الخاصة بهم.

لسوء الحظ ، هذا يعني غالباً أن مبرمجي PHP خاصةً المبتدئين في تطوير الويب ، غير مدركين للمخاطر الأمنية المحتملة التي يمكن أن تحتويها تطبيقات الويب الخاصة بهم.

غالباً ما يتم التغاضي عن الأمان عند تصميم مشروع ويب ، لأنه لا يوجد "متعة" في التفكير في قيام شخص ما بالتسلاّل إلى تطبيقك الجديد الدامع.

يعد الأمان أمراً يصعب قياسه ، ومن المستحيل تحديد ما إذا كان التطبيق آمناً حقاً أم لا - لا توجد سوى درجات من الأمان. بطبيعة الحال ، كلما بذلت المزيد من الجهد لجعل تطبيق ما آمناً ، كان أكثر أماناً. تكمّن الحيلة بالطبع في إيجاد التوازن الصحيح في الوقت والجهد - والنفقات.

من السهل إلى حد ما وغير المكلف نسبياً توفير مستوى كافٍ من الأمان لمعظم التطبيقات. ومع ذلك ، إذا كانت احتياجاتك الأمنية متطلبة للغاية - لأن المعلومات المخزنة في تطبيقك ذات قيمة كبيرة (أو حساسة للغاية ، مثل رموز الإطلاق النووية) - فيجب عليك حيئنـ ضمان مستوى أعلى من الأمان على الرغم من التكاليف المتزايدة المرتبطة بها . تذكر أن الخرق الأمني يمكن أن يكون مكلفاً للغاية.

موازنة الأمان وقابلية الاستخدام

للأسف ، فإن العديد من الخطوات المتخذة لزيادة أمان تطبيق الويب تقلل أيضاً من قابليته للاستخدام. كلمات المرور ، وانقضاض المهلات ، ومستويات التحكم في الوصول والأدوار كلها تخلق عقبات للمستخدمين الشرعيين. في حين أن هذه الخطوات ستزيد من أمان التطبيق ، فلا يمكنك جعله آمناً بحيث لا يمكن لأي شخص استخدامه.

أبرمت عقداً لمدة عام زائد كمطور في وكالة حكومية لم تذكر اسمها ادعت أنها واعية جدًا بالأمن. لقد طلّبوا فحصاً شاملًا للخلفية قبل التوظيف ، وكان على الجميع ارتداء شارات عالية التقنية للتنقل في المبني.

حتى أنه كان لدينا حراس عند مدخل المبني. كان الأمر "آمناً للغاية" لدرجة أنها اضطررنا إلى تغيير كلمات المرور الخاصة بنا كل 30 يوماً إلى كلمة مرور لم نستخدمها من قبل ، ويجب أن تكون كلمة المرور هذه من 10 أحرف على الأقل وتحتوي على أرقام وحروف وحالة مختلطة وعلامات ترقيم - ولا يمكن العثور عليها في القاموس.

باختصار ، لقد طلبو كلمات مرور لا يمكن لأي إنسان أن يتذكرها بالفعل ، ولم يكن النظام قابلاً للاستخدام. إذا كان جهاز الكمبيوتر الخاص بك خاماً لمدة 15 دقيقة أو أكثر ، فسيطلب منك كتابة كلمة المرور مرة أخرى. كل شخص عملت معه في هذا المشروع تم تدوين كلمة المرور الخاصة به على قطعة من الورق بجوار جهاز الكمبيوتر الخاص بهم. من الواضح أن "الصلاحيات الموجودة" في قسم الأمان قد اختارت الأمان بدلاً من قابلية الاستخدام لدرجة أن الأمن الذي كانوا يبحثون عنه تم اختراقه تماماً.

حقن SQL

واحدة من أعظم نقاط القوة في PHP هي سهولة التواصل مع قواعد البيانات ، مثل MySQL. مثال Sam's Used Car Lot. من هذا الكتاب وآلاف من مواقع الويب البارزة الأخرى ، مثل ، الاعتماد على قواعد البيانات لتعمل. http://Facebook.com

مع هذه القوة تأتي المخاطر أيضاً. أكثر مخاطر الأمان شيوعاً التي تواجهها عند التفاعل مع قاعدة بيانات هو شيء يسمى حقن SQL - عندما يستخدم المستخدم عن عمد جزءاً من تطبيقك لتشغيل استعلامات SQL غير المصرح بها وغير المقصودة في قاعدة البيانات الخاصة بك.

دعنا نستخدم مثلاً شائعاً. على الرغم من أننا لم نقم بتغطيتها في هذا الكتاب ، إلا أن العديد من الأنظمة التي تتطلب من المستخدم تسجيل الدخول تتميز بسطر من كود PHP يشبه إلى حد كبير هذا:

```
$authorized = mysql_query("SELECT Username,
Password, UserLevel FROM Users WHERE Username =
'".$_POST['username']."' and Password =
'".$_POST['password']."'");
```

يأخذ البرنامج النصي اسم المستخدم وكلمة المرور اللذين تم إدخالهما في النموذج ويبني استعلاماً باستخدام النص الذي أدخله المستخدم.

هل تبدو مألوفة؟ ستري العديد من الاختلافات في هذا مع استمرار رحلتك إلى PHP. فإذا ما هي المشكلة؟ لا يبدو أن مثل هذا الرمز يمكن أن يسبب ضرراً كبيراً. لكن دعنا نقول للحظة أنني أدخلت ما يلي في مربع الإدخال "اسم المستخدم" في النموذج وأرسلته:

" 1 = 1 # أو

يخبر رمز التجزئة MySQL (#) أن كل ما يليه هو تعليق ويتجاهله. سيبدو الاستعلام الذي سيتم تنفيذه بواسطة MySQL كما يلي:

```
SELECT Username, Password FROM Users
WHERE Username = " OR 1=1 #' and
Password = "
```

يخبر الرمز MySQL أن يتجاهل أي نص يتبع ، تاركاً عبارة WHERE Username = the SQL من WHERE Username = ' . 'سيتطابق مع كل صفاتي - وهو الجزع للبيطرا لشيعييللابويتيعلم اكتبييع ةيسبلعالمهستيور . ' . 'وكلمات المرور من قاعدة البيانات. ما قد يحدث بعد ذلك هو أنه إذا كان أول اسم مستخدم وكلمة مرور هو المستخدم المسؤول ، فإن الشخص الذي أدخل ببساطة بعض الرموز في مربع الرموز في المستخدم يتم تسجيل دخوله الآن كمسؤول موقع الويب الخاص بك ، كما لو كان يعرف بالفعل اسم المستخدم وكلمة المرور للمؤمن ، التي

ربما لا ، ولا ينبغي أن أعرف.

مع القليل من الإبداع الذي يتجاوز نطاق هذا الكتاب ، يمكن استخدام SQL Injection لإنجاز بعض الحيل السيئة التي ربما لم تفكر بها أبداً عند تصميم التطبيق الخاص بك.

لحسن الحظ ، من السهل جدًا وضع حواجز على الطرق تساعد في منع هذا النوع من الضعف. من خلال التحقق من الفوائل العليا في العناصر التي ندخلها في قاعدة البيانات ، وإزالتها أو استبدالها ، يمكننا منع أي شخص من تشغيل كود SQL الخاص به في قاعدة البيانات الخاصة بنا.

الوظيفة أدناه ستفعل الحيلة:

```
function make_safe($variable) {
    $variable =
    mysql_real_escape_string(trim($variable));
    return $variable;
}
```

بعد ذلك سنحتاج إلى تعديل استعلامنا. بدلاً من استخدام متغيرات POST مباشرةً ، سنمرر جميع البيانات التي يوفرها المستخدم من خلال وظيفة مثل make_safe .

مثل:

```
$username = make_safe($_POST['username']);
$password = make_safe($_POST['password']);
$authorized = mysql_query("SELECT Username,
Password, UserLevel FROM Users WHERE Username =
'".$username."' and Password = '".$password."'"');
```

الآن ، إذا أدخل المستخدم البيانات الضارة أعلاه ، سيبدو الاستعلام كما يلي ، وهو غير ضار تماماً. سيحدد الاستعلام التالي من قاعدة بيانات حيث يكون اسم المستخدم مساوياً لها

"" أو ". # 1 = 1 ."

```
SELECT Username, Password, UserLevel
FROM Users
WHERE Username = '\' OR 1=1 #' and
Password = "
```

الآن ، ما لم يكن لديك مستخدم باسم مستخدم غير معتمد وكلمة مرور فارغة ، فلن يتمكن المهاجم من إلحاق أي ضرر.

من المهم التتحقق من جميع البيانات التي تم تمريرها إلى قاعدة البيانات الخاصة بك مثل هذه ، مهما كنت تعتقد أنها آمنة.

مصادر إضافية

<http://www.addedbytes.com/articles/writing-secure-php/writing-secure-php-1/> <http://phpsec.org/projects/guide/> راجع أيضاً

الملحق أ: تثبيت PHP على موقع ويب

كيفية التثبيت على خادم ويندوز

قام Microsoft بأتمنة العملية بأكملها بشكل ملائم لخوادم Microsoft. كل ما عليك فعله هو زيارة <http://www.microsoft.com/web/gallery/>

أنظمة التشغيل المدعومة هي Windows 8 وWindows 7 وWindows Vista SP2 وWindows XP SP3 + وWindows Server 2012 وWindows Server 2008 R2 وWindows Server 2008 وWindows Server 2003

يجب أن يكون لديك امتيازات المسؤول على جهاز الكمبيوتر الخاص بك لتشغيل Web Platform Installer.

إليك مستند يصف كيفية تثبيت PHP على Windows Server 2008 / <http://www.howtogeek.com/50432/how-to-install-php-on-iis-7-for-windows-server-2008/>

كيفية التثبيت على خادم Linux

جميع توزيعات Linux تأتي مع PHP. ومع ذلك ، يوصى بتنزيل أحدث كود مصدر لـ PHP وترجمته وتثبيته على Linux. سيؤدي ذلك إلى تسهيل ترقية PHP بشكل مستمر فور توفر تصحيح أو إصدار جديد للتنزيل من PHP.

يوفر Geek Stuff جولة رائعة خلال هذه العملية. / <http://www.thegeekstuff.com/2008/07/instruction-guide-to-install-php5-from-source-on-linux/>

مقدمة

مرحبا ، وشكرا لك على قراءة كتابي. بصفتي مؤلفا جديدا ، فأنا مهتم بشكل خاص بالحصول على ملاحظاتك ، ومعرفة ما أعجبك وما لم يعجبك ، والنمو المهني.

كما قلت في بداية الكتاب ، لا تتردد في مراسلتي عبر البريد الإلكتروني على AlanForbes@outlook.com.

إذا لم يعجبك الكتاب ، فيرجى مراسلتي عبر البريد الإلكتروني وإخباري كيف يمكنني تحسينه. أجيبي على كل من يأخذ الوقت لمساعدتي ، والكتاب بالفعل أفضل بكثير من الإصدار الأول بفضل القراء أمثالك.

إذا أعجبك الكتاب ، فسأكون ممتنًا إذا كنت ستأخذ لحظة وترك لي تعليقاً على أمازون بالنقر فوق هذا الرابط
asin = B00BALXN70
-> https://www.amazon.com/review/create_review?ie=UTF8&

أيضا ، إذا كنت تحب أسلوب الكتابة الخاص بي وترغب في التعرف على البرمجة النصية من جانب العميل ، فيرجى التحقق من [أفضل الكتب مبيعا](#) ، [The Joy of jQuery: دليل المبتدئين إلى](#) [مكتبة جافا سكريبت الأكثر شهرة في العالم](#)

شكرا مرة أخرى ، وترميز سعيدا!

ملاحظة من أحد مؤيدي كيك ستارتر.

أنا شخصياً كنت أعاني من مشروع تعلم PHP لبعض الوقت.
هناك العديد من الدروس على الإنترنت ؛ بعضها أفضل ، في حين أن معظمها فظيع. سوف أشرح ...

لقد بدأ صبرنا ينفد بشكل متزايد ، حيث أن الإنترنت تجعلنا معتادين على مفهوم "هنا والآن". يجب علينا جميعاً أن نعترف بأن نطاق انتباهنا محدود نوعاً ما.
لذلك ، أعتقد أننا جميعاً يمكن أن نعترف بأن أصعب جزء في تعلم مهارة جديدة في الوقت الحاضر ليس فهم المجال ، بل الجلوس والاستمتاع بالانصباط اللازم لتعلم مهارة جديدة. التعلم ليس مسلياً ، لا سيما في مجال البرمجة الذي يعتمد على التقنية في الغالب.

هذا هو السبب في أن عنوان "متعة برمجة" لفت انتباهي أثناء التنقل في Kickstarter "الفرح" و "البرمجة" في نفس الجملة ضمن مفهوم "التعلم محترم للغاية وغير مألف تماماً لدرجة أنني قررت اتخاذ قفزة من الإيمان والاستثمار في المشروع. سألت نفسي ، "ما هو أسوأ ما يمكن أن يحدث ، برنامج تعليمي PHP لطيف آخر؟" لكنني كنت آمل سراً أن ينشأ شيء خاص من الإيمان الكبير الذي أبداه 583 من المؤيدين الآخرين.

هدأني قرار عرض الكتاب في مرحلة مبكرة. حتى في هذه المرحلة المبكرة ، بدا النهج برمته مختلفاً تماماً عما رأيته في أماكن أخرى.

النصوص سهلة الوصول وسهلة الفهم - وهي مثالية لأشخاص مثلـي.
يجب أن أقول إن الأمر يتطلب قدراً كبيراً من القوة لمنع نفسي من بدء الدورة في الوضع الحالي للكتاب ، وانتظار المشروع النهائي والمكرر.
العذر الوحيد الذي أقدمه لنفسي هو أنني متأكد من أن آلان سيتمكن من جعله أفضل من "بيتا". أثناء كتابة ما سبق ، أدركت أنني أتطلع حقاً إلى الحصول على الكتاب والمشاركة في الدورة التدريبية ... !

أود أنأشكر آلان على توليه هذا المشروع الضخم ، وإظهار روح رياضية جيدة طوال المhana بأكملها ، والحفاظ على مستوى عالي من التواصل مع جمهوره المستهدف عبر التعليقات والتعليقات ، ومراعاة العديد من الاقتراحات المقدمة إليه. يسعدني حقاً أن أرى مؤلفاً تقنياً يعطي الأولوية لجمهوره أولاً ، بينما لا يزال قادرًا على تقديم رؤيته وطريقته. كما أشكر كل من المؤيدين الآخرين على مساعدة آلان ودعم فكرته. أنا متأكد من أننا جميعاً سوف نستفيد أكثر من مجرد تبرعنا.

استمتع بالقراءة!

-باشا كاجان ، مطور PHP قريباً-