**ما هو الانترنت؟**

لعلك تستعمل الانترنت كل يوم ولكن بدون أن تعلم تاريخه أو كيف يشتغل، و لذلك سنعود بالتاريخ إلى الوراء و بالضبط إلى بداية الستينات، في خضم الحرب الباردة بين الاتحاد السوفييتي و الولايات المتحدة الامريكية، فقد احتاجت الولايات المتحدة لبناء شبكة لتبادل المعلومات بينها و بين حلفائها تضمن أنها لن تنقطع تحت أي من الظروف، و من هنا كانت الانترنت التي خرجت من مختبرات الوكالة الأمريكية للمشاريع الدفاعية المتقدمة، داربا. إذن فلا غرابة أن الاسم الأصلي للانترنت كان اربانات (ARPANET).

ولكن، في مثل هذا الوقت، لم يكن الويب موجوداً بعد. يخلط الكثير من الناس اليوم بين الانترنت و الويب، و الحقيقة ان الويب يعد جزءاً واحدا من الاجزاء العديدة لشبكة الانترنت، ولذلك سنتعرف على الويب في العنصر التالي.

**إذا لم يكن الويب هو الانترنت، فما هو إذا !؟**

حتى نفهم ماهية الويب، دعنا نكمل القصة التي سردناها في العنصر السابق.

لقد استمر الانترنت بالتطور و التوسع مما خلق حاجة لتوحيد المعايير التي يتم على اساسها بناء هذه الشبكات و تشغيلها و ذلك ادى إلى ظهور بروتوكولات جديدةللتحكم في كيفية تبادل المعلومات على هذه الشبكة مثل بروتوكول الـتحكم بالنقل (TCP) و بروتوكول الانترنت (IP)، وهذا التوحيد في المعايير أتاح المجال لظهور الويب حيث قدم عالم في معهد ماساتشوستس للتكنولوجيا اسمه **تيم بيرنرز لي** سنة 1989 نظاما جديداً يشتغل كجزء من الانترنت و يعتمد على لغة تسمى HTML اضافة إلى الروابط التشعبية و الواجهات سهلة الاستخدام اضافة إلى برامج خاصة تسمى متصفحات الانترنت. هذا النظام هو الويب.

**ما هي اللغة الأساسية للويب؟**

يستعمل الويب لغة تسمى بـ "لغة ترميز النص المتشعب" و يرمز لها اختصارا بـ **HTML** و لا تعتبر هذه الأخيرة لغة برمجة بل هي لغة ترميز تستعمل لوصف هيكلة محتوى الموقع.

تتألف لغة الـ  **HTML** في اساسها من عناصر تسمى الوسوم (tags). دعونا نأخذ النص التالي كمثال :



يمكن إدراج مثل هذا النص في صفحتنا عن طريق الوسم الآتي:

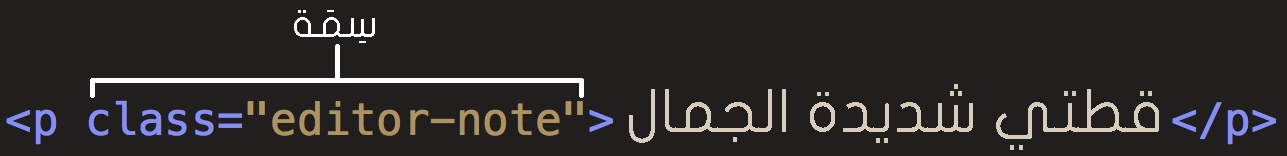
دعنا نكتشف هذا العنصر أكثر:



الأجزاء الرئيسية من عنصرنا هي:

1. **وسم الفتح**: يتألف من اسم العنصر (في حالتنا p)، محاطاً بقوس مفتوح وآخر مغلق. يحدد هذا المكوِّن بداية العنصر، أو بداية التأثير الذي يُضفيه (في حالتنا بداية الفقرة).
2. **وسم الإغلاق**: مشابه لسابقه، ما عدا أنَّه يتضمن مائلة أماميّة (/) قبل اسم العنصر. يحدد هذا المكوِّن نهاية العنصر (في حالتنا نهاية الفقرة). يُعد نسيان وضع وسم الإغلاق أحد الأخطاء الشائعة التي يقع بها المبتدئون والتي تؤدي إلى مشاكل مزعجة.
3. **المحتوى**: يُمثِّل هذا المكوِّن محتوى العنصر (في حالتنا نص الفقرة).
4. **العنصر**: هو الكيان ككل المؤلَّف من وسم الفتح، ووسم الإغلاق، ويتوسطه محتواه.

قد يحتوي عنصر ما على سِمات إضافية ، وهي تحمل معلومات إضافية حول العنصر لا تظهر في المحتوى الفعلي. في المثال التالي، قمنا بإضافة سمة تدعى (class) إلى العنصر السابق:



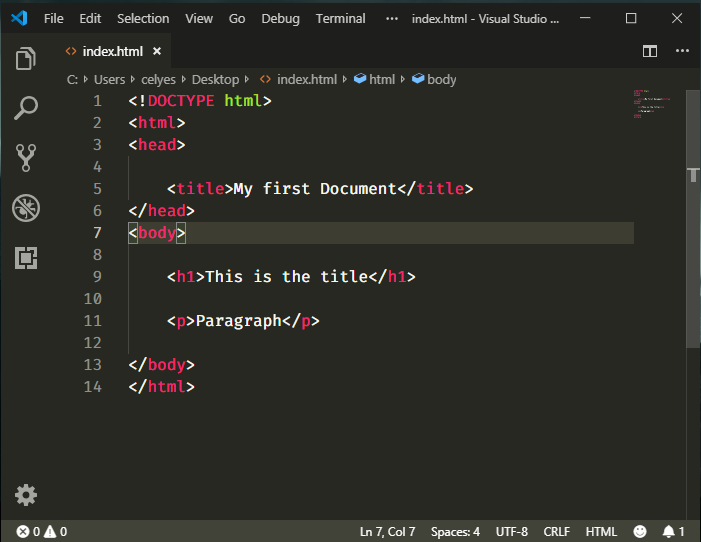
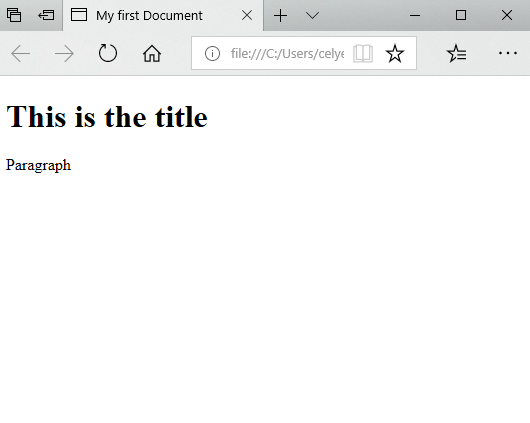
هناك بعض العناصر لا يوجد فيها محتوىً. تُدعى تلك العناصر **بالعناصر الفارغة**. خذ العنصر [<img>](https://developer.mozilla.org/ar/docs/Web/HTML/Element/img) الذي يستعمل لادراج صورة في صفحتنا كما يلي:



هناك عدة وسوم في لغة الـ HTML و لكل منها مهمته الخاصة. خلاصة القول هي الصورة التالية التي هي مثال لابسط صفحة انترنت يمكن انشاؤها بواسطة هذه اللغة، و ما ينتج عن هذه الصفحة عند تشغيلها على المتصفح.

لحد الآن قمنا بتغطية بعض العناصر الفردية الأساسية في هذه اللغة، ولكن تلك العناصر ليست بتلك الفائدة لوحدها. الآن سنأخذ نظرة حول كيف يمكن للعناصر الفردية أن تنمزج معاً لتشكل صفحة ويب متكاملة. دعنا نعاود رؤية الشفرة البرمجية التي يمكن من شأنها أن تكون أبسط صفحة HTML كاملة:





**ما لا تستطيع الـ HTML فعله:**

إن الـ HTML هي لغة رائعة للتعبير عن صفحات الانترنت الثابتة، لكن ماذا لو أردنا من صفحتنا أن تفعل شيئا آخر عدا عن عرض معلومات ثابتة؟ ماذا لو أردنا ان نجعل المستخدم يتفاعل اكثر؟ على سبيل المثال يدخل معلومات أو يحصل على معلومات متغيرة من قاعدة بيانات معينة؟ لا يمكن للـ HTML ان تفعل هذا. تذكر انها صممت لعرض معلومات ثابتة لا تتغير.

لماذا هي ثابتة ؟ تذكر ان الـ HTML ليست لغة برمجة، مهمة الخادم هنا هي انه يسلمك الصفحة فقط. انت تطلب الصفحة عبر كتابة الرابط الخاص بها في متصفحك و الخادم يبحث عنها لديه و عندما يجدها يسلمك إياها و انتهت القصة. لذلك حتى نجعل صفحتنا تفاعلية، يمكننا ان نجعل الخادم الخاص بنا يؤدي دورا جديد بفضل أداة جديدة. هذه الأداة هي في الاساس لغة برمجة تسمى PHP (في حالتنا هنا).

تسمح الـ PHP للخادم بأن يتلاعب بمحتوى الصفحة مباشرة قبل إرسالها إلى المستخدم كما يلي: يتم تشغيل ملف PHP على الخادم حيث يمكن لهذا الملف ان يتلاعب بمحتوى الصفحة و يعدله أو يولد محتوى جديد حسب طلب المستخدم، حيث يكون ذلك المحتوى على شكل HTML ثم يتم إرساله للمستخدم و يعرض في متصفحه.

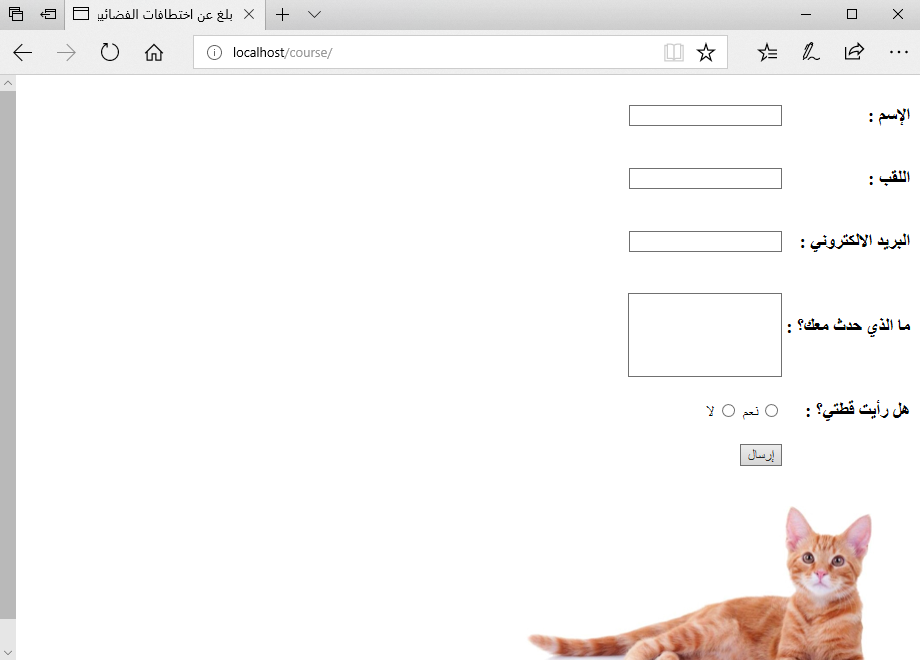
**مشكلة سليم:**

لنتعرف على سليم. لدى سليم قطة ظريفة و لكن لسوء حظه، اختفت هذه القطة. المشكلة هنا هي ان البحث في الجوار عنها ليس كافيا، لانها ببساطة اختطفت من طرف كائنات فضائية وبالتالي، عليه أن يبحث في المجرة كلها! فكر سليم في حل وهو أن ينشئ موقعا يمكن الناس من مشاركة تجاربهم المشابهة لتجربته مع الفضائيين.

ولكن حتى يحصل على تلك المعلومات من الآخرين، فهو يحتاج إلى أن شيء يستقبل تلك المعلومات منهم... مهلا لحظة! نحن نعرف كيف نصنع شيئا مشابها بواسطة الـ HTML... سنستعين بالحقول (forms and inputs).

**الحقول تساعد سليم في فهم القصة:**

يهدف موقع سليم هذا إلى وصله بالاشخاص الذين عاشوا تجارب مشابهة، لعله يجد دليلا او يتوصل إلى حقائق تساعده في رحلة البحث عن قطته. يعلم سليم تماما بأنه يحتاج حقول الـ HTML حتى يمكن للناس أن يدخلوا المعلومات اللازمة في حال ما إذا صادفوا قطته في أحد رحلاتهم الفضائية، ولكنه يحتاج مساعدتك لجعل ذلك ممكناً. إليك شكل الموقع الذي يفكر فيه سليم:



إليك الشفرة البرمجية التي استخدمناها للحصول على التيجة اعلاه:



**هناك خطب ما !**

قام سليم برفع النموذج على موقعه، و لكنه لم يتلقَ أي استجابة... هل يمكن ان يكون اختفاء قطته حدثا معزولا لهذه الدرجة؟ ام ان هناك خللا في الموقع؟

إليك ما قاله بعض من زوار الموقع بشأن هذا الخلل:

**الشخص الأول:** عندما اقوم بالضغط على إرسال لا يحدث أي شيء على الاطلاق... انه ميت تماما.

**الشخص الثاني:** قمت بملء جميع الحقول بدقة و لكن لم يحدث شيء اطلاقا... كل ما حدث هو ان عنوان الصفحة تغير و زاد بما يلي من نص لا افهمه :



يا له من أمر مربك حقا !

بعد ما قام سليم بتحقيقاته بشأن الخلل توصل إلى نتيجة مفادها أن الصفحة تعمل بشكل سلس إلى غاية اللحظة التي يضغط فيها المستخدم على "إرسال" أين تتوقف عن العمل و لا يحدث أي شيء أو تغير يدل على أن المعلومات التي ادخلها قد تم حفظها.

إذن، ما يحتاجه سليم هنا هو آلية يمكن بواسطتها أن يجمع تلك المعلومات التي ادخلها الزوار في الحقول و أن يخزنها في مكان ما. بشكل أدق، انه يحتاج إلى الـ PHP.

**أين تشتغل الـ PHP و كيف؟**

تعمل الـ PHP على مستوى الخادم، وتنتهي الملفات الخاصة بها بامتداد .php عادة ما يتشابه ملف الـ PHPمع ملفات الـ HTML من الداخل و في الشيفرة البرمجة الخاصة بهما و ذلك راجع لان ملف الـ php قادر على ان يحتوي على كود html داخله ايضا. في الواقع، عندما يعالج الخادم ملف php فإنه يحول في النهاية إلى html و يتم ارساله إلى المستخدم.

**سنستعمل الـ PHP للوصول إلى البيانات المدخلة في الحقول:**

إذن، يحتاج سليم إلى ملف PHP مهمته استقبال تلك البيانات و تخزينها في مكان ما لاستعمالها لاحقا. ليس ذلك فحسب بل يجب ان تكون طريقة الوصول إلى تلك البيانات فعالة مهما كانت قيمتها.

لفعل ذلك، اول ما نحتاج فعله هو إضافة ميزتين في صفحتنا حتى نقوم بتهيئة الـ php لاستقبال تلك المعلومات، و هما كما يلي:



لاحظ السطر الاول، هناك ميزتين اضفناهما و هما:

action="process.php" تحدد هذه الميزة مسار الملف الذي سيقوم بمعالجة المعلومات التي سيدخلها الزوار في الحقل. في حالتنا هذه هو ملف اسمه process.php و هو يتواجد في نفس المسار مع ملفنا الحالي.

method="post" تحدد هذه الميزة نوع الطلب الخاص بالـ http . نميز نوعين من الطلبات:

post نستعمله عندما نريد ان نرسل معلومات إلى الخادم.

get و نستعمله عندما نريد ان نجلب معلومات من الخادم.

فلنلق نظرة على ملف process.php . لا تقلق إذا بدا لك محتوى الملف غامضا. سنفهمه معا سطرا بسطر.



ما الذي تمثله الأسطر السابقة ؟

هذا السطر الأول :



ابتداءً من هذا السطر فصاعدا، نحن نتعامل مع الـ PHP إلى غاية العلامة التي تدل على اغلاق الجزء الخاص بهذه اللغة وهي :



تقوم الاسطر الموالية له بتخزين القيم التي ادخلها المستخدم في **متغيرات**. ما هي المتغيرات ؟ هي عبارة عن حاويات نستخدمها لنخزن قيمة معينة حتى نتمكن من إجراء عمليات عليها لاحقا أو نستخدمها فيما بعد.



يحتوي كل سطر على متغير، و كل واحد من هذه المتغيرات يخزن قيمة لحقل واحد. يمكنك تسمية المتغيرات بأي اسم تشاء و لكن كن حريصا على اختيار اسم يعبر عن المحتوى الحقيقي لذلك المتغير.

الآن نحن نتحدث! الان بعدما قمنا بتخزين قيمة الحقول في المتغيرات، يمكننا ان نصل إليهم كما نشاء. فلنقم بطباعتهم الآن باستعمال الشيفرة الآتية:



تقوم الكلمة المفتاحية **echo** بطباعة ما بين الفارزتين في المتصفح. لاحظ بأننا قمنا بواسطتها بطباعة ما تحتويه المتغيرات مع القليل من النص مثل "ما حدث معي هو:" و "إسمي هو: " هذا مثال واضح على كيفية التلاعب بمحتوى الصفحة من خلال الـ .PHP

**بعض القواعد لكتابة شيفرة PHP بشكل صحيح:**

* تبدأ الشيفرة البرمجية الخاصة بالـ PHP دائما بالرمز **<?php** و تنتهي بالرمز **?>**.
* يجب على جميع التعبيرات الخاصة بالـ PHP ان تنتهي بفاصلة منقوطة **;**
* إذا كان هناك صفحة تحتوي على شيفرة php، فمن الافضل لها ان تنتهي بامتداد .php
* تبدأ المتغيرات بعلامة الدولار (**$**) .
* اذا اردت تسمية المتغيرات بكلمتين، فافصل بينهما باستعمال العلامة السفلية (**\_**)

**المتغيرات بشكل أعمق:**

المتغيرات هي عبارة عن حاويات لقيم معينة. ابسط مثال يمكن ان يقرب لذهنك الفكرة هو كأس من المياه، الكأس في حد ذاته هو المتغير و الماء هو قيمة ذلك المتغير. انظر إلى الصورة الآتية:



يمثل هذا الجزء القيمة الفعلية للمتغير. يمكن لهذا الجزء ان يكون رقما او نصا أو نوعا آخر من البيانات.

يعد هذا الجزء اسم المتغير، يمكن تسميته بأي اسم نريده شريطة ألا يبدأ بأرقام أو برموز خاصة و شريطة أن يكون معبرا و ذو معنى.

يمكن للمتغيرات ان تحمل انواعاً مختلفة من البيانات مثل الارقام و النصوص و الارقام العشرية.

**موقع سليم يعمل جيدا... بشكل أكثر من اللازم.**

إلى هنا، يعمل كل شيء بسلاسة و لكن هناك خطبا ما... إذا أراد لاحقا أن يرى ما قام الناس بإدخاله من معلومات؟ هل سيطلب منهم إعادة إدخالها مجددا؟ هذا غير منطقي أبداً.

نحتاج هنا لوسيلة تخزن البيانات بشكل دائم. بصيغة أخرى، نحتاج إلى قاعدة بيانات، و بما أننا نستعمل الـ PHP فخيارنا الاول سيكون قاعدة بيانات من نوع MySQL.

**الـ MySQL رائعة لتخزين البيانات:**

قبل أن نستعمل الـ MySQL، دعونا نعرف كيف تنظم المعلومات داخل هذا النوع من قواعد البيانات.

يتم تسيير قاعدة البيانات من قبل خادم بيانات (database server) يعمل جنبا إلى جنب مع خادمنا الرئيسي.

توضع البيانات في قاعدة بيانات و هي بدورها مقسمة إلى جداول، كل جدول يحتوي على عدد من صفوف، و كل صف يحتوي على عدد من الحقول. في الواقع هناك عدة حالات نستعمل فيها العديد من الجداول.

**قاعدة بيانات سليم:**

إذن، فقد استقر سليم على رأي، سيستخدم قاعدة بيانات من نوع MySQL ليخزن المعلومات التي سيدخلها زوار موقعه، و ليحقق ذلك، فهو سيقوم بالتعديل على ملف process.php بحيث انه سيقوم بإدخال تلك البيانات في جداول ليقوم بتخزينها و عرضها لاحقا، بدل ان يقوم بعرضها مباشرة.

ولكن قبل ذلك، فلنتعرف على هيكلة قاعدة البيانات الخاصة بنا لنعرف كيف نخزن تلك المعلومات بكفاءة.

نحتاج لقاعدة بيانات واحدة تحتوي على جدول واحد، و في هذا الجدول، هناك مجموعة من الحقول بالتأكيد.

كل حقل في ذلك الجدول يمثل احد الحقول في نموذج الـ html الخاص بسليم الذي قمنا بتصميمه سابقا.

سنسمي جدولنا باسم aliens\_abduction ، اي بالعربية اختطاف الفضائيين و تجدر بنا الاشارة إلى انه يمكنك تسمية هذا الجدول بأي اسم تشاء و لكن احرص عن يكون الاسم معبرا عن المحتوى الذي ستضعه في الجدول. أبقِ كل شيء واضحا.

اول خطوة لانشاء الجدول هي فتح نافذة لسطر الأوامر في النظام (cmd) و كتابة الأمر الآتي:



لا بأس إن بدا لك هذا الأمر مربكا. لقد حدث نفس الأمر لي ايضا في السابق.

ما يفعله هذا الأمر ببساطة هو انه ينشئ جدولا تحت اسم aliens\_abduction

و يدرج داخل ذلك الجدول حقولا هي firstname و lastname و email و what\_happened و spotted\_my\_cat.

يمثل الجدول الآتي اسم الحقل و ما سنخزنه داخله:

|  |  |
| --- | --- |
| الجدول : aliens\_abduction | |
| اسم الحقل | **ما سيتم تخزينه فيه** |
| firstname | إسم الزائر |
| lastname | لقب الزائر |
| email | البريد الالكتروني للزائر |
| what\_happened | ما قاله الزائر بشأن ما حدث معه |
| spotted\_my\_cat | هل تم رؤية قطة سليم ام لا (نعم / لا) |

سنفهم باقي الكلمات التي في ذلك الأمر في وقتها.

**فلنحتفظ بما يدخله الزوار:**

هل تتذكر اننا قمنا بجلب تلك البيانات التي يدخلها المستخدم في الحقول ؟ هذا يعني انه لدينا وصول لما يدخله المستخدم و منه، سنقوم بتخزينها في الجدول الذي قمنا بإنشاءه.

ولكن قبل أن نقوم بإدخال هذه المعلومات، يجب أن نربط برنامجنا بقاعدة البيانات أولا، بصيغة أخرى أن ننشئ اتصالا بينهما.

يكون ذلك كما يلي:



لدينا اربعة متغيرات هنا، اولهم يحمل اسم الخادم الخاص بنا. بما اننا في خادمنا المحلي فإن الإسم هو localhost.

ثانيهم هو اسم المستخدم الخاص بالخادم و ثالثهم هي كلمة المرور الخاص بذلك المستخدم، و أخيرا اسم قاعدة البيانات التي نعمل عليها.

هل هذا كافٍ؟ لا بالتأكيد. يجب أن نضيف الية تمكننا من الاتصال فعليا، سنضيف السطر الآتي:



لقد اتصلنا بقاعدة البيانات و تم الأمر بنجاح. بقي علينا أن ندرج تلك البيانات في القاعدة.

**فلنحتفظ بما يدخله الزوار - الجزء الثاني:**

كما ذكرنا فيما سبق، لقد قمنا بربط الاتصال بقاعدة البيانات و بقي علينا أن نحتفظ بما يدخله الزوار بشأن قطة سليم.

توفر الـ MySQL العديد من الأوامر للتفاعل مع قاعدة البيانات، حيث يمكن لنا أن نضيف معلومات جديدة أو نحذفها، أو نقوم بتغييرها أو جلبها.

بما اننا نريد إدخال بيانات للاحتفاظ بها، فسنستخدم أحد اكثر الأوامر شيوعا في الـ MySQL هو الأمر INSERT. حيث يتيح لنا هذا الأمر إدخال المعلومات في قاعدة البيانات الخاصة بنا.

تكون صيغة الأمر INSERT كما يلي:



شرح الأمر:

نبدأ الأمر بكتابة الكلمتين"**INSERT INTO**".

ثاني خطوة ستكون كتابة إسم الجدول الخاص بنا. في هذه الحالة سيكون **aliens\_abduction** .

الخطوة الثالثة هي فتح القوسين، ثم كتابة أسماء الحقول التي نريد إدراج القيم فيها بالترتيب، مع مراعاة الفصل بينها باستعمال الفاصلة، قم إغلاق القوس بعد آخر حقل

الخطوة الرابعة هي كتابة الكلمة "**VALUES**".

الخطوة الخامسة و الأخيرة هي فتح قوسين و كتابة القيم التي نريد ادخالها و كما في الخطوة الثالثة نفصل بين كل قيمة و قيمة بالفاصلة.

**إنتبه!**

يجب أن يكون ترتيب العناصر التي الخطوة الخامسة كما في الخطوة الثالثة، في المثال أعلاه، سيتم تخزين القيمة value1 في الحقل column1 و القيمة value2 في الحقل column2 وهكذا دواليك.

في النهاية، ستكون الشفرة البرمجية في صفحة process.php كما يلي:





**عرض البيانات:**

إن ادخال البيانات يعد أمر سهلا، لقد حللنا نصف مشكلة سليم، ولكن يبقى هناك شعور بأن لا فائدة من عملنا ما لم نستطع عرض تلك البيانات و الاستفادة منها بشكل واقعي، و لهذا الغرض، سننشئ صفحة أخرى تجلب تلك البيانات و تعرضها في جدول منسق و جميل.

هل تتذكر أننا قلنا أن الـ MySQL بإمكانها جلب البيانات من القاعدة الخاصة بنا؟ هذا ما سنفعله الآن! سنستعين بأحد أكثر أوامر الـ MySQL شهرة وهو الأمر SELECT. و تكون صيغته كما يلي:



نبدأ الأمر بكتابة كلمة **SELECT** ثم نقوم بتحديد الحقول التي نريد جلبها من الجدول، و إذا كنا نريد جلب اكثر من حقل، فسيتم الفصل بين اسمائها بواسطة الفاصلة، ثم نكتب الكلمة **FROM** ثم إسم الجدول وانتهينا!

دعنا نطبق الأمر على جدولنا، حيث سنستبدل الكلمات اللازمة حتى نحقق الغرض الذي نريده و بالتالي، سيكون الأمر كما يلي:

