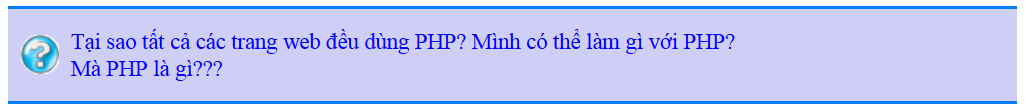


# Một trang web động với PHP



Ya ya ya! Đừng hốt hoảng các bạn của tôi, bài viết này sẽ giải thích với các bạn tất cả!  
Tôi giả định rằng các bạn đọc bài viết này vì các bạn chưa từng biết gì về PHP, nghĩa là các bạn là những beginners, còn những ai đã biết PHP, tôi xin vui mừng nhận các ý kiến đóng góp của các bạn! Các bạn nên đọc bài viết này theo thứ tự các chương, đừng nhảy vội, rồi các bạn sẽ bất ngờ vì những gì các bạn có thể làm sau đó cho mà xem!

Bài viết bao gồm các phần sau:

* [Cơ bản về PHP](http://thanhtra.nguyen.free.fr/web/tuto-php/modau.php#coban)
* [Cơ Sở Dữ Liệu](http://thanhtra.nguyen.free.fr/web/tuto-php/modau.php#csdl)

## Phần I: Cơ bản về PHP

[1) Giới thiệu về PHP](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongI.php)

|  |  |
| --- | --- |
| 1-1 | [PHP là gì ?](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongI.php#whatisphp)  [Khác nhau giữa PHP và HTML](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongI.php#different) |

[2) Chương trình sử dụng : WAMP](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongII.php)

|  |  |
| --- | --- |
| 1-2 | [Tại sao phải dùng WAMP ?](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongII.php#whyWAMP)  [Cài đặt WAMP](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongII.php#install)  [Chỉnh thông số cho WAMP](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongII.php#configurer) |

[3) Bước đầu với PHP](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongIII.php)

|  |  |
| --- | --- |
| 1-3 | [Tag trong PHP](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongIII.php#tag)  [Hiển thị text](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongIII.php#text)  [Chú thích](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongIII.php#chuthich) |

[4) Biến số](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong4.php)

|  |  |
| --- | --- |
| 1-4 | [Biến là gì?](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong4.php#bien)  [Gán và hiển thị biến](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong4.php#gan)  [Những tính toán đơn giản](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong4.php#calcul)  [Chuyển biến](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong4.php#transmettre) |

[5) Hàm](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong5.php)

|  |  |
| --- | --- |
| 1-5 | [Tự tạo hàm](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong5.php#taoham)  [Biến PHP thành cái đồng hồ biết nói](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong5.php#dongho) |

[6) Điều kiện](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong6.php)

|  |  |
| --- | --- |
| 1-6 | [Cấu trúc cơ bản: if … else](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong6.php#coban)  [Luân chuyển Switch](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong6.php#luanchuyen) |

[7) Thực hành: trang bảo vệ bằng mật khẩu](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong7.php)

|  |  |
| --- | --- |
| 1-7 | [Tạo một trang web được bảo vệ bởi mật khẩu](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong7.php#baove) |

[8) Vòng lặp](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong8.php)

|  |  |
| --- | --- |
| 1-8 | [Một vòng lặp đơn giản: while](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong8.php#while)  [Vòng lặp phức tạp hơn: for](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong8.php#for) |

[9) Bảng (mảng)](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong9.php)

|  |  |
| --- | --- |
| 1-9 | [Bảng đánh số](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong9.php#numerote)  [Bảng kết hợp](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong9.php#associatif) |

## Phần II: Cơ sở dữ liệu

[1) Giới thiệu MySQL](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql1.php)

|  |  |
| --- | --- |
| 2-1 | [Ơ, mà MySQL là gì ta?](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql1.php#dinhnghia)  [Cấu trúc của một cơ sở dữ liệu](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql1.php#structure)  [Ê, tui có câu hỏi !](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql1.php#question) |

[2) PhpMyAdmin](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql2.php)

|  |  |
| --- | --- |
| 2-2 | [Tạo một bảng](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql2.php#taobang)  [Sửa đổi bảng](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql2.php#modifier)  [Các thao tác khác](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql2.php#autres) |

[3) Đọc dữ liệu](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql3.php)

|  |  |
| --- | --- |
| 2-3 | [Kết nối vào CSDL](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql3.php#connect)  [Lấy dữ liệu](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql3.php#recuperer)  [Tiêu chuẩn lựa chọn](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql3.php#critere)  [Đếm số hàng trong bảng dữ liệu](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql3.php#compter) |

[4) Viết dữ liệu](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql4.php)

|  |  |
| --- | --- |
| 2-4 | [Thêm dữ liệu](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql4.php#add)  [Sửa đổi DL](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql4.php#modifier)  [Xóa DL](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql4.php#delete) |

[5) Thực hành : mini-chat](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql5.php)

|  |  |
| --- | --- |
| 2-5 | [Tạo một mini\_chat](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql5.php#thuchanh) |

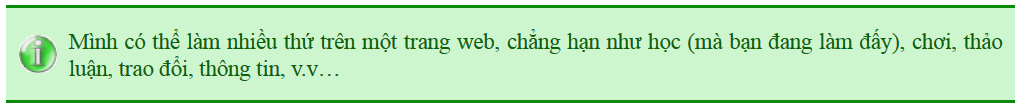
**Chương 1: Giới thiệu về PHP**

Trong chương này:

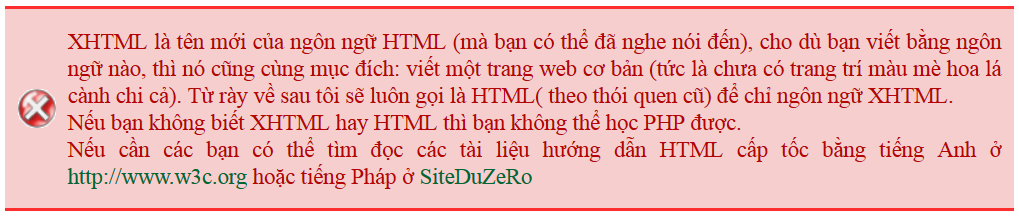
|  |  |
| --- | --- |
| 1-1 | [PHP là gì ?](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongI.php#whatisphp)  [Khác nhau giữa PHP và HTML](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongI.php#different) |

**PHP là gì?**

Hẳn bạn đã biết cái gì là một trang web rồi chứ gì? Nếu bạn không biết, thì ví dụ như trang này chẳng hạn, bạn đang xem một trang web đấy! Để xem một trang web, mình gõ địa chỉ của nó vào thanh địa chỉ, ví dụ <http://www.google.com.vn>. Gõ xong, enter, trình duyệt của bạn (Firefox, internet explorer, …) sẽ tải trang web về hiển thị lên màn hình cho bạn xem!



Bây giờ, chuyện thứ 2: bạn có nghe nói về (x)HTML chưa? Nếu chưa, bắt buộc bạn phải biết mới có thể tiếp tục được!



Như đã nói: XHTML là một ngôn ngữ cho phép bạn viết một trang web, mình có thể cho chữ in đậm, nghiêng, chèn hình ảnh, âm thanh,… bằng cách gõ các kí hiệu đặc biệt (tiếng anh gọi là các tags, tiếng pháp gọi balises). Trong bài viết này tôi gọi HTML cho gọn! Cái XHTML nó giống giống thế này này:



PHP là một ngôn ngữ được lồng vào giữa những đoạn mã HTML, đây là một ví dụ nhỏ về PHP:



Có gì mới ở đây ta ? Hãy xem hàng : <?php echo("You are the visitor n°" . $nb\_visitors); ?>   
Lúc nào cũng thấy ngôn ngữ HTML, giữa lòng nó mình lại bắt gặp câu trúc của PHP ! Những gì tôi sẽ hướng dẫn là thao tác trên những dòng kiểu này đấy ! À, mấy kí tự lạ lùng này ($ ;?>) có thể sẽ làm các bạn sợ, nhưng tôi bảo đảm rồi các bạn sẽ quen thôi !  
Một trang HTML thì có phần mở rộng là .html, còn một trang php thì phần mở rộng của nó là .php ! PHP có nhiều phiên bản (version) lắm, version hiện nay là v5 rồi!



Ừm, thiệt tình là không ! Dù sao đi nữa mình cũng cần phải có HTML để viết một trang web mà, mình không thể tránh khỏi HTML được đâu !  
Tóm lại, HTML rất tiện lợi,nhưng còn hạn chế. Với sự giúp đỡ của PHP bạn có thể làm được nhiều thứ lợi hại hơn cho trang web của bạn. Ví dụ hả ?

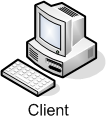
* Một diễn đàn, nơi mọi người thảo luận, trao đổi, thông tin cho nhau, giúp đỡ nhau khi ai đó gặp vấn đề,…
* Chat chit
* Một sổ lưu niệm hay lưu bút, nơi mà khách viếng thăm có thể ghi lại vào đó cảm nghĩ của họ về trang web của bạn, mà những người khác (ngoài bạn ra) cũng có thể đọc được !
* Newsletter : bạn có tin gì mới ? Chỉ cần click một cái là bạn có thể gửi mail tới tất cả những thành viên trong diễn đàn của bạn! Hoàn toàn tự động nhé, không phải mất công ngồi viết đâu!
* Một bộ đếm visitors
* Làm blog, mọi người có thể comment vào đó…

Và còn nhiều hơn nữa mà ... mình chưa biết! Hì hì ...  
Tóm lại, những gì cần nhớ là PHP giúp bạn tạo một trang web động, nó có thể tự động cập nhật trang web của bạn ngay cả khi bạn đi Ấn Độ du lịch, trang web của bạn vẫn cập nhật liên tục số người viếng thăm, những gì khách comment vào, những tin tức mới,…

**Khác nhau giữa HTML và PHP**

Cái mà tôi nói đây hơi rắc rối một tí, nhiều người dùng PHP thường không cần biết cái này! Nhưng tin tôi đi, Nếu bạn cố gắng hiểu được cơ chế của nó (nó chỉ lấy của bạn khoảng 10 phút thôi), không những bạn sẽ tiết kiệm được nhiều thời gian sau này mà còn sẽ hiểu được bạn *sắp làm gì* nữa kìa! Vậy cũng đáng để đọc chứ nhỉ?  
  
  Tôi định nói gì nhỉ? À, định giải thích rằng: khi ai đó vào thăm trang web của bạn, sau khi họ gõ địa chỉ, enter, rồi trang web được tải về hiển thị lên màn hình! Đồng ý, nhưng giữa khoảng thời gian sau khi bạn **enter** và khi trang web hiển thị lên, thì có chuyện gì đã diễn ra âm thầm trong đó không nhỉ?!  
Đương nhiên là không có thì tôi không nói làm gì! Điều này rất quan trọng, vì cơ chế hoạt động của HTML và PHP hoàn toàn khác nhau! Để tôi giải thích từ từ nhé:  
  Trước hết, bạn cần phải biết một khái niệm: **quan hệ client-server**! Quỷ quái gì nữa đây ? Không không, chả có gì quái quỷ bên trong đâu !

* **Client** ( khách ) nghĩa là bạn đấy mà ! Chính bạn là người đang âm thầm rón rén ngồi trước máy tính xin được xem trang web ! Tất cả những người xem web được gọi là client hết !   
  Tôi diễn tả máy client của bạn là cái này nhé :



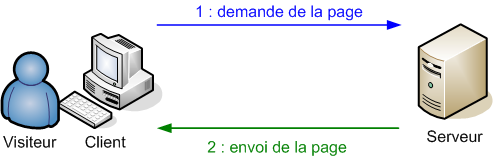
* **Server** ( chủ ) : chỉ có một server thôi ! Đó là một loại máy khổng lồ nối mạng suốt 24/24 (với một tốc độ kết nối cực nhanh), nó được đặt đâu đó trên thế giới, chạy liên tục 24/24 một mình, không ai chạm vào, không ai chơi gì trên nó cả ! Nó chỉ làm mỗi nhiệm vụ là phân phối các trang web cho client xem !   
  Cơ chế của nó là nó chứa trang web của bạn trên ổ cứng của nó, mỗi khi có ai hỏi ý xin xem, nó sẽ gửi cho người đó xem ! Phải chi ai hỏi tiền mà nó cũng cho thế này thì đã quá ! Xin là cho hà !  
  Để biểu diễn server, tôi dùng cái máy dưới đây nhé (chú ý rằng gã server này không có màn hình, vì chẳng dùng để làm gì cả, có ai thèm coi mặt nó đâu, nó vốn một mình, cô đơn, làm việc âm thầm lặng lẽ…)



   Bạn thấy sao, cũng không rối rắm lắm nhỉ ! Với những ai chưa hiểu, tôi xin ví dụ như sau :  
Tưởng tượng một cái nhà hàng nhé (KFC đi), bạn vô đó ăn, thì bạn là **CLIENT** ! Bạn gọi 2 phần gà chiên và khoai tây cộng một chai cocacola (ặc, đói quá !), thằng phục vụ (tức **server**) liền chạy đi lấy đưa cho bạn liền ! Rồi nếu có khách (client) khác tới, lại gọi 2 phần gà chiên khoai tây và cocacola như bạn, thằng phục vụ cũng chạy đi lấy liền ! Nó cứ làm như vậy hoài, liên tục, không ngừng nghỉ (nó mà nghỉ là ông chủ cho thôi việc liền) !  
   Internet cũng vậy, thằng phục vụ bây giờ là cái máy chủ to tướng khổng lồ không màn hình mà tôi nói hồi nãy, nó cũng làm việc liên tục không ngừng mỗi khi có ai đó xin được xem trang web nào đó, nó liền gửi cho người ta xem ! Nó làm như thể sợ người ta cho nó về hưu sớm vậy ! Không dám nghỉ !   
Rồi, hiểu **client-server** rồi nhé ! Bây giờ tôi sẽ giải thích PHP và HTML làm việc khác nhau thế nào !

**Trước hết, về HTML :**

Nhắc lại rằng một trang HTML có phần mở rộng là .html, ví dụ như exemple.html  
Tôi không đi vào chi tiết, chỉ giải thích hoạt động của nó bằng sơ đồ sau :

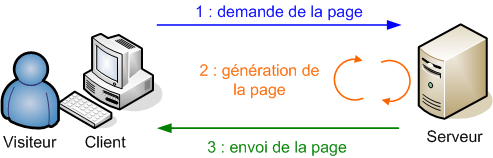
  
1: demande de la page: xin xem trang web   
2: envoi de la page: gửi cho xem trang web

Có hai bước :

1. Client (tức bạn, khách viếng thăm) muốn xem một trang web. Bạn xin server « anh server ơi, làm ơn gửi cho em xem trang nghihe.html với »
2. Server trả lời bằng cách quăng trang web vô màn hình của bạn « nè ! Đây là cái trang web mà mầy vừa đòi xem đó ku »

**Bây giờ tới PHP :**

Nó chèn thêm một bước nữa giữa hai bước nói trên, trang PHP được phát sinh bởi server trước khi gửi đến client. Sơ đồ hóa như vầy :

  
1: demande de la page: xin xem trang web   
2: génération de la page: phát sinh trang web  
3: envoi de la page: gửi cho xem trang web

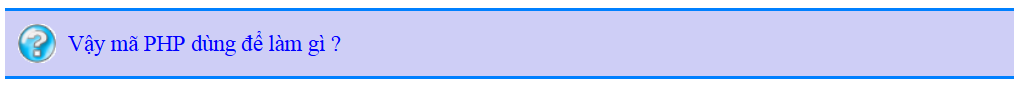
Các bước hoạt động như sau :

1. Client xin xỏ « anh server ơi, làm ơn gửi cho em xem trang nghihe.php đi »
2. Ở đây, một bước rất quan trọng, server không gửi trang web tới client ngay lập tức, mà nó phát sinh ra trang web trước ! Như vậy client không thể nào đọc được trang php có gì trong đó (chỉ có server đọc được thôi), client chỉ đọc được những trang HTML thôi ! Việc của server rất đơn giản : nó biến đổi trang php thành trang html để client có thể đọc được !
3. Cuối cùng, sau khi đã tạo xong trang HTML, giờ này nó chỉ có mã HTML bên trong thôi, rồi gửi tới client « nè ! Đây là trang web mà mầy đòi xem đó ! »

Vấn đề quan trọng của bạn bây giờ là hiểu cho rõ bước thứ 2 trên đây !



Lấy lại ví dụ từ đầu chương , tôi sẽ cho bạn xem :  
<?php echo("You are the visitor n°" . $nb\_visitors); ?>   
Máy tính của bạn không hiểu dòng này, chỉ có server hiểu và biến đổi nó thành HTML.



Mã PHP bao gồm những cấu trúc lệnh ! Giống như những dòng lệnh mà mình viết trong lập trình vậy (chẳng hạn như lập trình C, C++, Java, Python,...). Nó yêu cầu server cho biết giờ, số người khách vào thăm,… Nói ngắn gọn : PHP ra lệnh cho server ! Điều mà HTML không làm được.  
Đừng quên rằng trang PHP cũng có chứa mã HTML nữa, nhưng mà server không đụng đến, nó chỉ khoái chỗ nào có PHP thôi ! Rồi nhận lệnh mà PHP sai bảo, rồi biến tất cả thành HTML !  
Một điều đặc biệt ở đây là : trang web phát sinh chỉ được gửi đến một client duy nhất ! Nghĩa là khi có ai xin xem nữa, server sẽ lại bắt đầu phát sinh một trang khác ! (Thằng này kĩ dữ, không lấy đồ xài rồi hay second-hand cho người ta! Tốt!)  
Lưu ý rằng việc phát sinh này cũng chiếm một khoảng thời gian (vài phần nghìn giây, tùy dung lượng của trang), có nghĩa là server PHP phải mạnh hơn nhiều lần so với một server HTML thông thường ! Nếu trang web của bạn nổi tiếng, cùng lúc có thể có 30 người truy cập vào, cho nên server càng phải mạnh !

**Chương 2 : Chương trình sử dụng : WAMP**

    Chương đầu tiên đã giới thiệu sơ lược về thế giới PHP, nhưng vẫn chưa có gì cụ thể cả ! Tôi biết rằng các bạn cũng đang sốt ruột không muốn đọc nhiều, mà muốn bắt tay vào « vọc » liền ! Đừng nôn nóng, tôi sẽ không để các bạn đợi lâu đâu ! Ở chương này chúng ta sẽ bắt đầu thực hiện những thao tác trên PHP !  
Trong bài viết này tôi xin giới thiệu với các bạn một phần mềm tiếng Pháp rất hữu ích cho công việc của chúng ta sau này (bạn cũng có thể chọn ngôn ngữ là tiếng Anh nếu bạn không biết tiếng Pháp, tạm thời tôi chưa viết cho bản tiếng Anh, nhưng cũng giống nhau thôi, bạn chịu khó nhé !), chương trình tên là WAMPServer ! Cái này chỉ dùng trong Windows, bạn nào xài Mac xin dùng MAMP <http://www.mamp.info/en/download.html> , hoặc XAMPP cho Linux <http://sourceforge.net/project/showfiles.php?group_id=61776&package_id=60248>

* Trong chương này:

|  |  |
| --- | --- |
| 1-2 | [Tại sao phải dùng WAMP ?](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongII.php#whyWAMP)  [Cài đặt WAMP](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongII.php#install)  [Chỉnh thông số cho WAMP](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongII.php#configurer) |

**Tại sao tôi phải cần đến chương trình này ?**

    Như tôi đã trình bày ở chương trước, client (tức là BẠN) không hiểu được PHP (mà chỉ hiểu HTML thôi) ! Chỉ có server mới đọc được và hiểu được PHP. Vấn đề là bạn muốn viết một trang PHP, nhưng làm sao chạy thử coi nó có bị lỗi gì hay không, hay nó sẽ ra như thế nào,… Máy bạn không đọc được PHP cơ mà ! Vì vậy mình phải tìm cách nào đó « dạy » cho máy tính của mình hiểu được PHP !  
    WAMPServer chính là giải pháp ! Tôi gọi WAMP cho gọn! Nó là một package (gói) gồm nhiều chương trình cần thiết để xử lý PHP. Bạn không cần làm chi cả, chỉ cần cài WAMP, rồi những chương trình cần thiết ấy sẽ được cài đặt tự động!  
Trang web của WAMP đây: <http://www.WAMPserver.com/>   
Sau đây là những chương trình tự động cài đặt khi bạn cài WAMP:

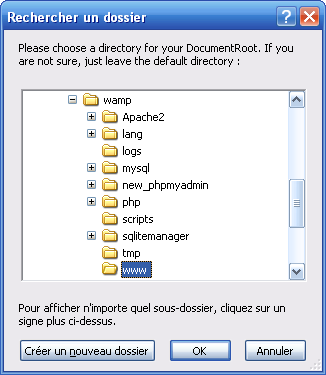
* Apache: chương trình sử dụng máy chủ, cho phép máy chủ phân phát các trang web, nhưng nó chỉ hiểu HTML thôi!
* PHP: coi như một “plugin” của Apache, nó cần có Apache để hoạt động. Nhờ PHP mà Apache biết làm việc với những trang Php, nói cách khác Apache+PHP = một server PHP.
* MySQL: một chương trình rất hữu ích sau này, tạm thời tôi chưa nói đến làm gì! Chỉ biết rằng nó cho phép mình sử dụng cơ sở dữ liệu (CSDL)! CSDL là gì hả? Tôi sẽ giải thích khi đến thời điểm cần phải giải thích, chỉ sợ các bạn bị tẩu hỏa nhập ma thôi!
* PhpMyAdmin: thằng này thì dùng để quản lý CSDL.



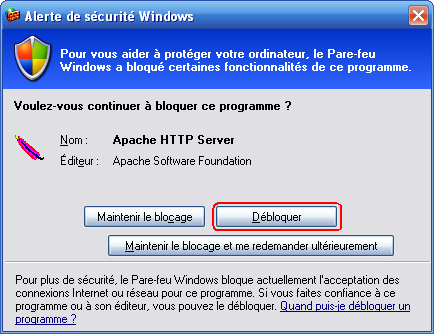
Bạn tạm thời không cần phải biết WAMP hoạt động ra sao, mà cái cần làm bây giờ là tải WAMP về cài cái đã! Nó là một chương trình khá kín đáo, khi chạy, nó trốn vào taskbar, không thèm hiện mặt lên, mà mình cũng không cần mò cái mặt nó làm gì! Bây giờ hãy xem làm sao cài đặt và cấu hình cho nó nhé!

## Cài đặt WAMP

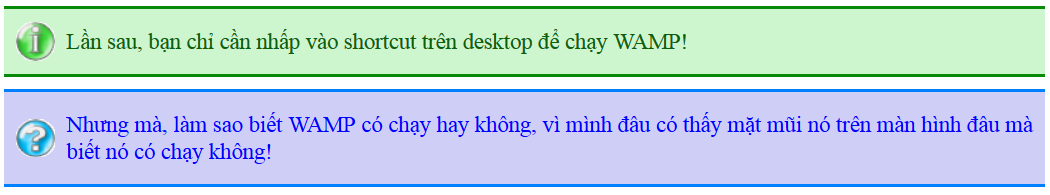
Tải WAMP về tại đây: <http://www.WAMPserver.com/dl.php> (khoảng 21Mo)  
Việc cài đặt cũng giống như cài bất cứ chương trình nào khác thôi, không gì đặc biệt. Khoảng gần cuối quá trình cài đặt, nó yêu cầu bạn chọn một thư mục mà sau này bạn sẽ đặt tất cả các file PHP của bạn vào đó để chạy thử! Bạn nên để nó mặc định (là www), nhấn OK!



Cuối phần cài đặt, servrer Apache sẽ chạy trước! Nếu tường lửa của máy bạn ngăn chặn chương trình, bạn chỉ việc nhấp vào Unlock (Débloquer) để cho phép nó chạy là được



Cuối cùng nó hỏi bạn có muốn chạy WAMP hay không, bạn hãy cho nó chạy!



Hãy nhìn xuống taskbar, ở chỗ gần cái đồng hồ ấy, có biểu tượng của WAMP đó!

install

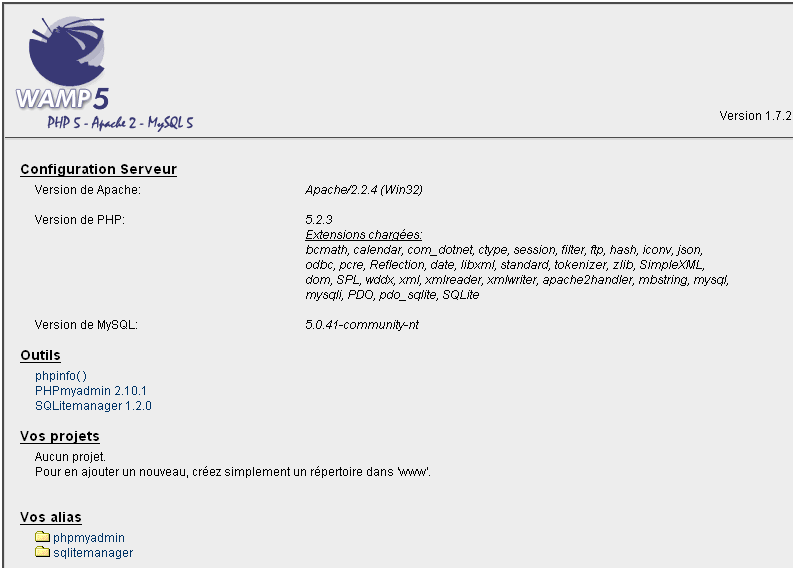
Nếu bạn đưa chuột vào nó, nó hiện lên “All services running”! Vậy là nó đã chạy rồi đấy! Chúc mừng! Bạn đã cài thành công WAMP!

**Cấu hình cho WAMP**

    Lưu ý rằng tôi đang dùng bản tiếng Pháp, nếu bạn dùng bản tiếng Anh thì cũng tương tự thôi! Nếu cần thiết tôi sẽ viết hướng dẫn bản tiếng Anh sau, nhưng tôi nghĩ chắc không cần đâu nhỉ? Chỉ cần nhìn vào “nó click chỗ nào thì mình click vào chỗ đó là OK thôi!" He he...  
Bây giờ, nhấp chuột trái vào biểu tượng của WAMP trên taskbar, nó trông giống thế này này:



* Hàng đầu tiên: Localhost! Nó sẽ đưa bạn đến trang chủ của WAMP, ở đấy bạn có thể quản lý tất cả các project của bạn (trong mục Vos projets (Your projects))



Cho đến lúc này, tôi chưa tạo một projet nào cả, việc tạo một projet cũng đơn giản thôi: bạn chỉ việc vào thư mục www rồi tạo một thư mục hay một file (tương ứng sẽ là một projet của bạn) Một projet tương ứng với một trang web, tạm thời bạn chỉ muốn làm một projet thôi, nhưng tương lai biết đâu bạn sẽ làm cùng lúc một đống projet thì sao!

* Hàng thứ hai trên menu WAMP là PhpMyAdmin, nó sẽ mở ra chương trình PhpMyAdmin! Tôi sẽ nói vụ này sau, giờ chưa phải lúc cần biết nó là gì đâu!
* Cuối cùng, hãy nhìn vào hàng thứ tư của menu WAMP: répertoire www, nhấp vào đây nó sẽ mở ra thư mục www trong windows, nơi mà bạn sẽ cất giữ các projet của mình trong đấy! Tôi khuyên bạn nên tạo một thư mục tên là tests để thử nghiệm các trang php mình sẽ tạo sau này! Trong trường hợp của tôi nó nằm ở C:\WAMP\www\tests

Sau khi đã tạo ra thư mục tests, bạn trở lại trang chủ của WAMP (Localhost), bạn phải thấy thư mục tests xuất hiện trong phần Vos projets (Your projects)

install

Nếu không thấy thì bạn hãy refresh (Ctrl+R) lại trang này nhé!  
Muốn chạy thử trang php nào bạn chỉ cần vào thư mục đó, nhấp vào trang php tương ứng là được! Nhưng việc này làm trên trang chủ của WAMP (Localhost) đấy nhé! Không phải vào windows nhấp đôi vào trang web để mở nó lên đâu! Nếu bạn làm như vậy, windows sẽ xuất hiện một hộp thoại hỏi bạn là "dùng chương trình gì để mở file này", và đương nhiên là không có chương trình nào trong máy bạn mở được cả! Cho nên tôi nhắc lại là việc test chương trình chỉ thực hiện trên trang chủ của WAMP mà thôi (ở Localhost á)

**Chương 3: Bước đầu với PHP**

    Cái tựa chắc cũng quá rõ rồi nhỉ? Mình sẽ làm quen bước đầu với PHP nào!  
Bạn sắp khám phá những cấu trúc đầu tiên của PHP và niềm khoái trá (xin lỗi, cái này nhiễm thầy Lân dạy toán ở Bách Khoa) của bạn khi thấy mấy dòng scripts làm treo máy! Hô hô hô !!!   
Yên tâm, tôi sẽ không làm cho cuộc đời thêm rắc rối đâu! Tất cả chỉ là những bước cơ bản về lập trình PHP thôi!  
Bạn sẵn sàng chưa? Lên đường!!!

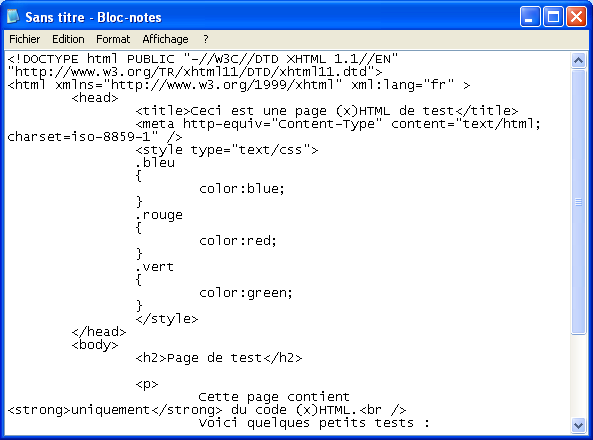
* Trong chương này:

|  |  |
| --- | --- |
| 1-3 | [Tag trong PHP](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongIII.php#tag)  [Hiển thị text](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongIII.php#text)  [Chú thích](http://thanhtra.nguyen.free.fr/web/tuto-php/chuongIII.php#chuthich) |

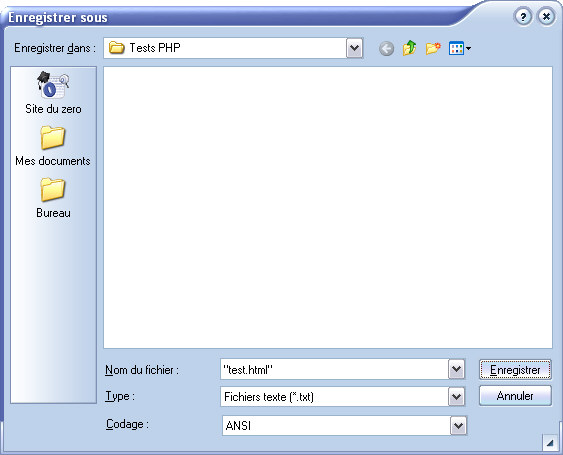
**Những tags trong PHP (tiếng Pháp gọi là balises)**  
Từ giờ trỡ đi chúng ta sẽ làm việc thường xuyên với mã nguồn của một trang web, vì vậy bắt buộc bạn phải có kiến thức tối thiểu về (x)HTML!  
Để sửa mã nguồn của một trang web bạn cần một Text Editor, bạn có nhiều giải pháp:  
    - Dùng Notepad của windows (Start--> All Programs--> Accessories--> Notepad) Thường thì phần mềm này cũng đủ để bạn soạn PHP rồi, nhưng …  
    - Tốt hơn hết là dùng Notepad++, nó có chức năng tô màu cho text để mình dễ phân biệt các đoạn mã, nó cũng có đánh số dòng nữa, cái này tiện lắm!   
Notepad++ là một phần mềm miễn phí, dễ sử dụng, nhẹ, tải tại đây: [NOTEPAD++](http://sourceforge.net/project/showfiles.php?group_id=95717&package_id=102072)  
Nếu bạn dùng hệ điều hành Mac, bạn có thể dùng Smultron <http://smultron.sourceforge.net/>, nếu là Linux thì bạn đã có sẵn Vim hay Emacs rồi!  
Cho dù bạn xài hệ điều hành nào đi nữa, các thao tác trên PHP đều như nhau cả thôi, không có gì thay đổi cả!  
Chúng ta hãy bắt đầu bằng việc tạo một trang web bằng HTML, như tôi đã nói PHP không thể sống thiếu HTML! Đoạn mã dưới đây chỉ chứa HTML thôi, bạn hãy copy nó vào Text Editor mà bạn dùng (Notepad hoặc notepad++) :



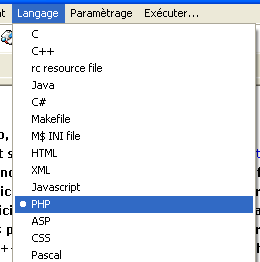
Tôi sẽ chỉ cho các bạn các thủ tục cần làm tùy theo bạn dùng Notepad hay Notepad++:  
**Notepad**  
Nếu bạn đã copy paste vào notepad, nó phải trông giống thế này:



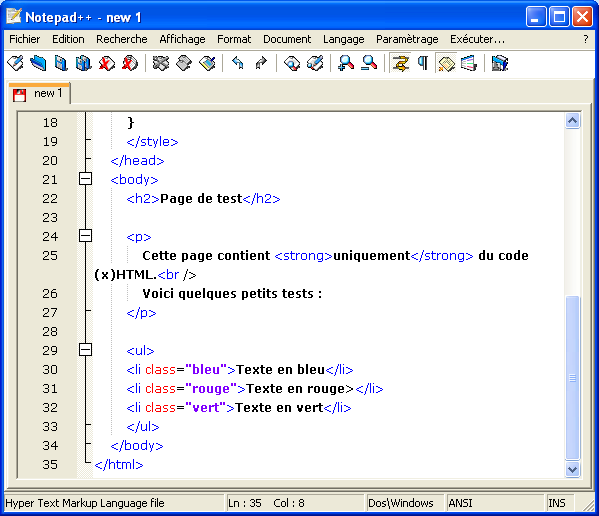
Để save lại dưới dạng một trang html, bạn vào File--> Save as, một hộp thoại mở ra, gõ tên trang web vào, nhớ có “.html” nữa nhé! Ví dụ “test.html”. Trông như thế này này:



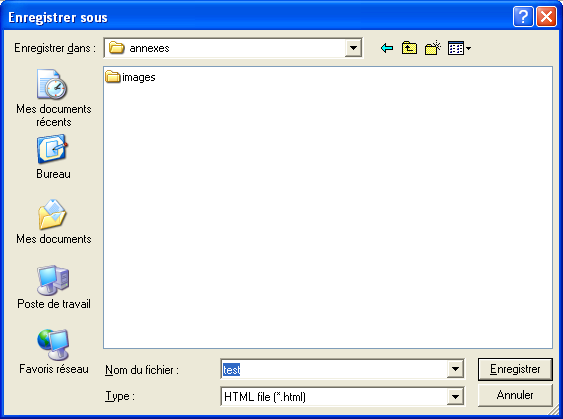
**Notepad++**  
Đầu tiên bạn vào menu chọn Langage (Language), chọn PHP



Bạn sẽ thấy code của bạn sẽ bị đổi màu, trông giống thế này:



Để save lại dưới dạng HTML, bạn làm như trên, ở đây chỉ cần gõ “test” là đủ, nhưng ở thanh Type, bạn chọn “HTML file”:



Hãy vào thư mục mà bạn đã save trang web để mở thử xem cái trang HTML mà bạn vừa save đó sẽ hiện ra như thế nào nhé! (Với những trang PHP bạn phải mở trong trang chủ của WAMP như tôi đã nhắc ở trên, nhưng với HTML thì mở trong windows được rồi).  
OK, cho tới đây thì chắc bạn cũng không lấy gì làm ngạc nhiên, mấy cái này chắc bạn đã làm như cơm bữa rồi phải không?  
Như bạn đã biết trang HTML trên được tạo thành bởi những tags, ví dụ <ul> là một tags! Không phải ngẫu nhiên mà tôi nhắc vụ này, vì để viết PHP chúng ta phải đặt vào một tag mới, hơi đặc biệt một chút. Nó bắt đầu bởi <?php và kết thúc bởi ?> . Và chúng ta sẽ đặt code PHP vào bên trong cái tag này!



Một điều quan trọng : một tag php có thể chứa nhiều hàng, chúng ta có thể nới rộng cái tag ra nhiều hàng không vấn đề gì cả, ví dụ :



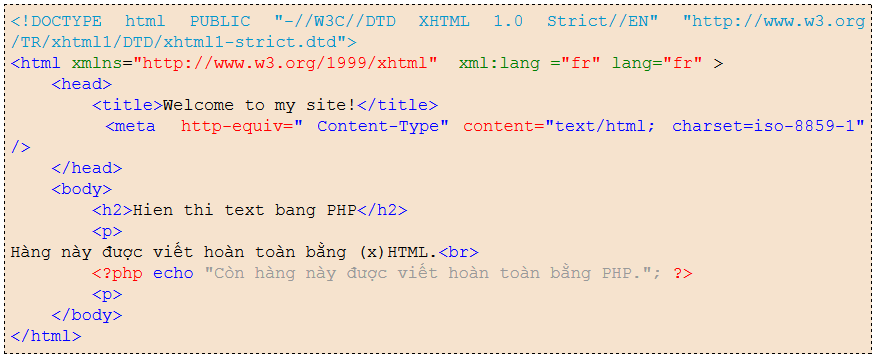
Tất cả những gì cần nhớ để viết code php là đặt vào cái tag <?php và ?>   
Lưu ý : Ngoài ra còn có những cách viết khác để chèn vào một tag php, ví dụ < ? ?>, <% %>, … cho nên đừng ngạc nhiên khi bạn nhìn thấy nó ở đâu đó ! Tuy nhiên <?php và ?> là cách viết chính xác nhất, cái mà tôi sẽ dùng trong suốt bài viết này !^^  
Mình hãy đặt code PHP vào một chỗ dễ coi nhất trong trang HTML (để dễ phân biệt ý mà), ví dụ như vầy :



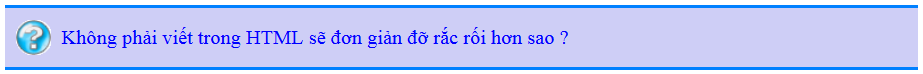
Chú ý rằng trang này không có gì để chạy cả, vì mình chưa đặt gì vào giữa hai tag PHP cả mà !

## Hiển thị text

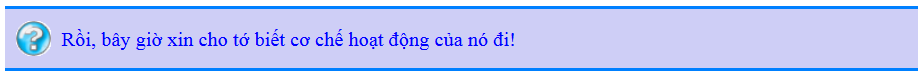
OK, nhìn như vậy cũng đẹp rồi nhỉ ? Nhưng mình còn phải đặt code PHP vào nữa chứ, cái này mới là món chính mà ! Cái hàm mà tôi sử dụng sau đây cho phép hiển thị text ! Giống như hàm printf trong C hay print trong Python vậy đó. Tôi sẽ viết code trước rồi sẽ giải thích chi tiết sau nhé !  
Mở notepad hoặc notepad++ ra, copy cái này vào :



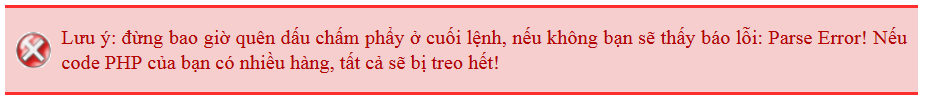
Save trang này với phần mở rộng là .php, ví dụ hienthitext.php trong thư mục tests mà tôi đã đề nghị các bạn tạo lúc cài WAMP ý !  
Để chạy thử trang php này, bạn hãy chạy WAMP (nếu bạn chưa chạy nó), rồi vào Localhots, trang chủ của WAMP mở ra, nếu bạn đã tạo thư mục tests, bạn phải thấy nó hiện ra trong phần Vos projets (Your projects) !  
Nhấp vào thư mục tests, một trang web mới xuất hiện cho thấy tất cả các trang php bạn đã tạo. Bạn phải thấy trang hienthitext.php trong đó, nhấp vô nó đi ! Coi cái gì hiện ra trước mắt bạn vậy ? PHP in ra một hàng, đơn giản !



Đúng vậy, nhưng bạn sẽ sớm thấy được điều thú vị của nó ! Tạm thời bạn hãy nhớ : echo cho ra text !



Lấy lại cái dòng mà mình muốn tìm hiểu:  
<?php echo "Còn hàng này được viết hoàn toàn bằng PHP."; ?>  
Như bạn đã thấy, code PHP được đặt giữa tag <?php ?>, cái mà mình quan tâm bây giờ là code nằm trong đó: echo "Còn hàng này được viết hoàn toàn bằng PHP."; Echo là một cấu trúc lệnh, nó ra lệnh cho máy: hãy hiển thị text ra cho ta!  
Text mà mình muốn hiển thị phải được đặt toàn bộ trong dấu ngoặc kép, kết thúc lệnh bằng dấu chấm phẩy! Dấu này giúp cho máy hiểu được “à, nó ra lệnh xong rồi đấy! Thi hành thôi”



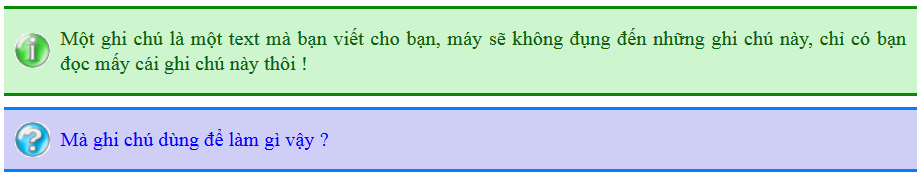
Mình cũng có quyền bắt nó hiển thị tag nữa! Ví dụ:  
<?php echo "Còn hàng này được viết <strong>hoàn toàn</strong> bằng PHP.";?>  
Chữ hoàn toàn sẽ được in đậm nhờ tag <strong> </strong>



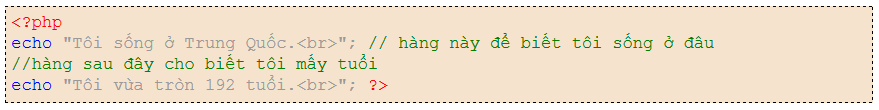
Ừ nhỉ, nếu mình đặt dấu ngoặc kép vào, máy sẽ hiểu là text cần hiển thị chỉ có bấy nhiêu thôi, nó sẽ dừng lại ngay chỗ bạn đặt dấu ngoặc kép thứ 2! Phần còn lại sẽ bị treo lủng lẳng!  
Giải pháp là như vầy: thêm dấu chéo ngược (\) vào trước dấu ngoặc kép cần hiển thị  
<?php echo "Còn hàng này được viết \"hoàn toàn\"bằng PHP.";?>  
Tôi đã nói gần như hết về chức năng của hàm echo, bây giờ tới lượt bạn! Thử cho hiển thị bất cứ câu gì bạn thích! Thử đặt thêm vài hàng php nữa xem (mỗi câu một hàng thôi nhé!) và đừng quên <br> để xuống dòng!  
Ví dụ, bạn có thể viết: <?php echo "Còn hàng này được viết \"hoàn toàn\"bằng PHP. <br>";?>  
Lưu ý rằng bạn có quyền đặt các tag HTML bên trong cấu trúc lệnh echo.

## Các ghi chú, chú thích, commentaires

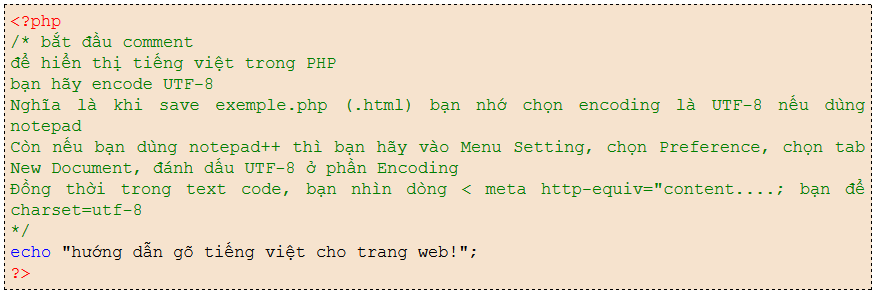
Trước khi kết thúc chương này, tôi muốn nói một điều mà tôi cho là rất quan trọng trong lập trình, cho dù với bất kì ngôn ngữ nào khác : những ghi chú (commentaires) !



Để cho bạn biết bạn đã viết gì, ý tưởng của bạn lúc đó là gì. Giả sử bạn ngưng công việc viết web khoảng một tuần, sau đó bạn lại tiếp tục thì bạn có nguy cơ không hiểu mình đã viết những gì, lúc đó mình nghĩ gì,…  
Để bắt đầu ghi chú, bạn dùng hai dấu chéo (slash) //  
Hoặc đặt ghi chú của bạn giữa hai dấu /\* và \*/  
Sau hai dấu slash, bạn bắt đầu viết vô tư ghi chú của bạn ! Ví dụ :



Nhưng với hai dấu slash, mình bị hạn chế ở chỗ chỉ ghi chú được một dòng mà thôi, với dấu /\* và \*/ mình có thể ghi chú được nhiều dòng, ví dụ:



# Chương 4 : Biến



Biến là một yếu tố không thể thếu trong các ngôn ngữ lập trình, và trong PHP cũng vậy, mình không thể tránh khỏi ! Biến không phải là cái gì ngoằn ngoèo lắc léo mà trái lại nó làm cho cuộc đời trỡ nên đơn giản ! Thiếu nó, bạn không thể tiến xa hơn được !  
Chương này khá dài, vì vậy bạn có thể đọc một nửa hôm nay, ngày mai một nửa. Bạn không nên bộp chộp vội vàng vì bạn sắp phải học những điều vô cùng quan trọng của PHP. Cuối chương bạn sẽ bắt đầu hiểu vì sao người ta thích PHP như vậy.

* Trong chương này:

|  |  |
| --- | --- |
| 1-4 | [Biến là gì?](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong4.php#bien)  [Gán và hiển thị biến](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong4.php#gan)  [Những tính toán đơn giản](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong4.php#calcul)  [Chuyển biến](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong4.php#transmettre) |

**Biến là gì ?**

Ngay cái tên thôi mình cũng nghĩ đến cái gì đó biến đổi thường xuyên, một điều riêng duy nhất của biến là « có thể thay đổi được » ! Nhưng cụ thể biến là gì ?  
Một biến, là một thông tin được chứa tạm thời trong bộ nhớ, nó không tồn tại lâu. Biến (hay thông tin) trong PHP chỉ tồn tại trong thời gian server phát sinh trang web (xem chương 1). Sau khi đã phát sinh xong trang web, tất cả các biến đều bị xóa đi vì chúng không còn dùng để làm gì nữa. Biến không phải như một file được chứa trong ổ cứng mà chỉ là một thông tin nhỏ tạm thời.  
Chính bạn là người tạo ra biến khi cần dùng đến.  
Một biến gồm 2 thành phần:

* **Tên biến**: để nhận ra nó là gì, dùng để làm gì, bạn nên cho nó một cái tên dễ nhận biết một chút, ví dụ như “tuoi\_cua\_toi”
* **Giá trị của biến**: là thông tin mà biến chứa, giá trị của biến có thể thay đổi, ví dụ “21 tuoi”.

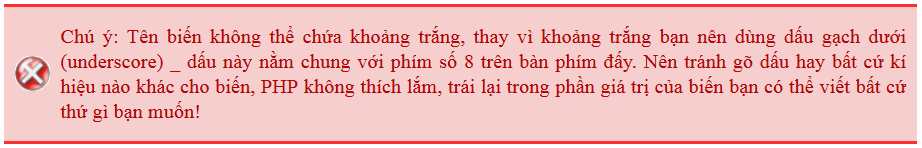
Ở đây tôi đã cho một ví dụ về một biến có tên là “tuoi\_cua\_toi”, biến này có giá trị là “21 tuoi”.  
Mình có thể sửa đổi khi cần đến giá trị của biến, hay thực hiện vài thao tác trên biến,… Khi đó mình sẽ gọi (tên) biến ra, và biến sẽ lịch sự cho mình biết giá trị của nó.   
Ví dụ một thời điểm nào đó, bạn gọi:  
    -Ê biến tuoi\_cua\_toi, chú chứa cái gì đó?  
    -21 tuoi  
    -Cảm ơn nghen!  
Bạn sẽ thấy mấy thứ nhăng nhít này (ngay khi nó còn mơ hồ, chưa rõ nghĩa lắm) sẽ rất cần thiết cho trang web của bạn!  
Ví dụ bạn muốn lấy tên hoặc nickname của một thành viên nào đó, bạn sẽ dùng một biến ten\_thanh\_vien chẳng hạn, rồi bạn chứa tên một thành viên nào đó trong biến này (khoai lang chẳng hạn). Rồi khi thành viên này đăng nhập vào trang web hay diễn đàn của bạn, bạn sẽ cho in ra hàng này: Xin chào Khoai lang!  
Hẳn bạn đã biết làm sao để in ra text rồi nhỉ? Lệnh echo mà tôi đã nói ở chương trước sẽ giúp bạn chuyện này dễ dàng.  
Bây giờ chúng ta hãy xem làm sao để sử dụng biến trong PHP nhé!

**Gán và hiển thị**

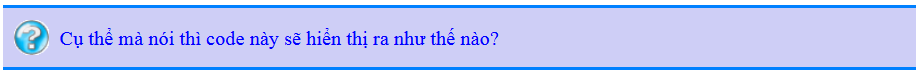
Bây giờ mình sẽ thử gán một giá trị cho biến, rồi hiển thị (in) ra giá trị đó. Bạn sẽ hiểu rõ hơn cách sử dụng biến.  
**Gán một giá trị cho một biến:**  
Ví dụ, nếu mình gõ vào như sau:



Nếu viết như vậy, bạn đã tạo ra một biến tên là “ten\_thanh\_vien” và biến này chứa giá trị là “Khoai Lang”!



Có vài kí hiệu mới nhỉ? Trước hết là dấu đô la ($), nó phải luôn luôn được đặt ở đầu tên biến. Đó là dấu hiệu để PHP nhận ra “à, thằng này là một biến đây!”. Không những PHP mà bạn cũng nhận ra nó là một biến khi bạn đọc code ở đâu đó khi bạn thấy dấu $ .   
Kế đến là dấu bằng (=) dùng để chỉ phép gán, giống như trong Pascal mình hay dùng dấu mũi tên <-- để gán giá trị cho một biến ấy mà.Ví dụ viết **$a<-- 15** thì có nghĩa là mình gán giá trị 15 cho biến a, trong php thì viết là **$a = 15;**. Nói nôm na gán cũng như áp đặt cho biến một giá trị nào đó. (mà mình có thể thay đổi nếu cần)   
Tiếp theo là một giá trị của biến được gán vào, chữ Khoai lang được đặt trong dấu ngoặc kép vì nó là giá trị kiểu text. Và cuối cùng xin đừng quên dấu chấm phẩy (;) để kết thúc một câu lệnh.



Chả ra gì cả! Bởi bạn chưa dùng lệnh echo mà, cho tới giờ PHP chỉ tạo ra một biến và gán cho nó một giá trị rồi giữ tạm thời trong bộ nhớ thôi, chứ chưa làm gì khác!  
Biến không nhất thiết phải chứa text, mà mình có thể gán cho nó một giá trị số hay giá trị logic (đúng sai) cũng được!  
Mình hãy cho 3 ví dụ về 3 kiểu “dữ liệu” khác nhau mà biến có thể chứa nhé!

* text: Cái này tôi vừa cho ví dụ, để cho biến một giá trị text, mình đặt nó trong dấu ngoặc kép như vầy:

C:\Users\Chu Quang Vien\Desktop\4_5.PNG

* số: chỉ khác với text ở chỗ là mình không cần phải đặt vào giữa hai dấu ngoặc kép thôi:

C:\Users\Chu Quang Vien\Desktop\4_6.PNG

* giá trị logic: chỉ có hai giá trị là ĐÚNG (true) hoặc SAI (false). Để phân biệt với giá trị text, mình không gõ dấu ngoặc kép cho true và false, ví dụ:



Bạn sẽ hiểu rõ hơn về giá trị logic này trong chương sử dụng điều kiện.  
**Hiển thị (in ra) giá trị của một biến :**  
Đơn giản chỉ dùng lệnh echo mà bạn đã biết, ví dụ :



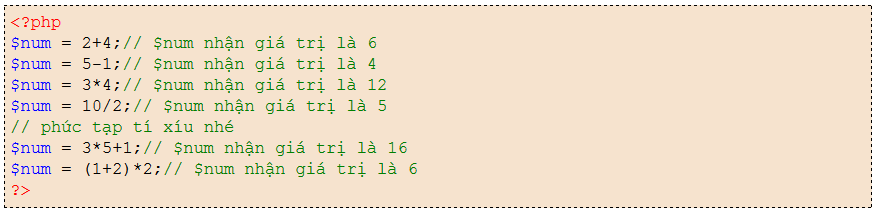
Nó in ra cái gì vậy? Tuyệt, tuyệt, nó in ra “khoai lang” kìa!   
Bạn hãy thử cho vài giá trị khác nhau bắt nó in ra thử xem!  
Với cấu trúc lệnh echo, bạn không nhất thiết chỉ in ra giá trị của biến, ví dụ:



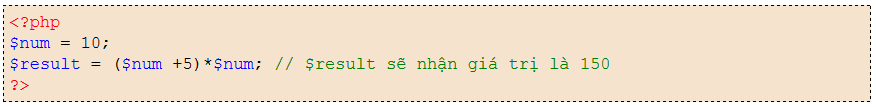
Nó sẽ in ra : xin chao khoai lang!  
Bạn hãy thử cho thêm vài biến bên trong echo xem sao!

## Vài tính toán đơn giản

Mình khoan tính toán cái gì ngoằn ngoèo rắc rối vội, chỉ đơn giản cộng trừ nhân chia thôi nhé!  
Mình sẽ chỉ làm việc với những biến chứa giá trị số, đây là 4 kí hiệu cần biết : +, -, \*, / (chắc bạn không lạ gì chúng phải không?)



Đây là những tính toán với nhiều biến:



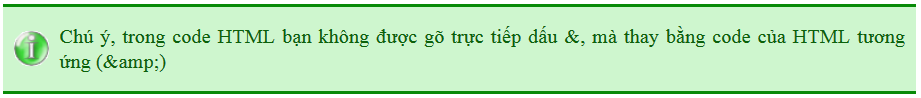
Bạn thử kiểm chứng lại xem (dùng lệnh echo in chúng ra như đã làm ở trên).

## Chuyển biến

Một thú vị của PHP: mình có thể chuyển một biến từ trang này sang trang khác!  
Bạn sẽ thấy điều này vô cùng tiện lợi, ví dụ chuyển biến $ten\_thanh\_vien từ trang này sang trang khác! Như tôi đã nói, biến sẽ bị hủy đi sau khi trang web được phát sinh xong. Vậy làm sao lấy biến $ten\_thanh\_vien vào trang khác được???  
  
**Vận chuyển bằng cách thay đổi address**  
  
Bạn có bao giờ thấy một địa chỉ (address) dài ngoằn như thế này chưa:  
http://www.mysite.com/info.php?ngay=27&thang=08&nam=2007&title=informations  
Mấy biến của bạn nằm trong đó đấy! Đây là một cách chuyển biến từ trang này sang trang khác!



Như bạn đã biết, để tạo một link trong HTML, mình dùng tag <a>, chẳng hạn:  
<a href="http://www.mysite.com/info.php"> click here to access my site </a>  
Sau chữ info.php, bạn đánh một dấu chấm hỏi (?), kế tiếp là tên biến, rồi dấu bằng, rồi giá trị của biến  
http://www.mysite.com/info.php?ngay=27  
Cái này sẽ tạo tự động một biến đặt biệt tên là $\_GET[“ngay”] có giá trị là 27  
Nếu bạn muốn tạo thêm biến khác, bạn chỉ cần đặt vào dấu & rồi tên biến mới, dấu bằng, giá trị của nó!

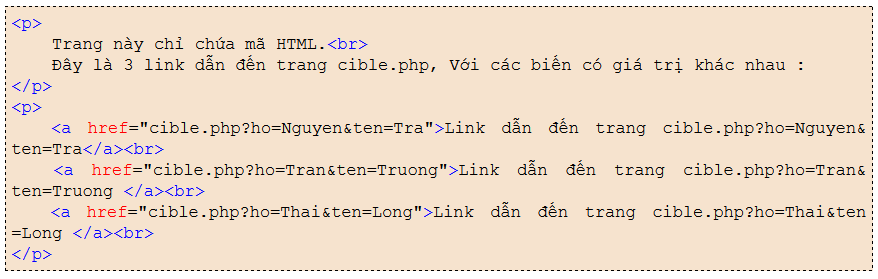


Ví dụ:  
http://www.mysite.com/info.php?ngay=27&amp;thang=08&amp;nam=2007&amp;title=informations   
Tất cả kí hiệu &amp; sẽ được trình duyệt của bạn chuyển thành &   
Ở đây, 4 biến đặc biệt đã được tạo :

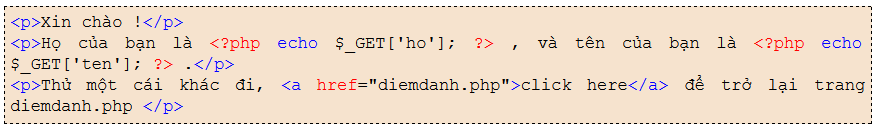
* $\_GET[‘ngay’] có giá trị là 27
* $\_GET[‘thang’] có giá trị là 08
* $\_GET[‘nam’] có giá trị là 2007
* $\_GET[‘title’] có giá trị là « informations »

Tôi nhận thấy rằng mấy biến này có hình dạng kì cục quá ! Nhưng đừng vì vậy mà bạn nản lòng dừng lại nhé !  
Mình hãy làm một ví dụ nhỏ để bạn thấy nó sẽ cho ra kết quả cụ thể như thế nào !  
Để thực hiện, mình cần 2 trang :  
Một trang chứa cái tag <a href="http://www.mysite.com/info.php"> click here to access my site </a>  
Một trang để lấy biến và giá trị của nó được chuyển đến!

Code HTML

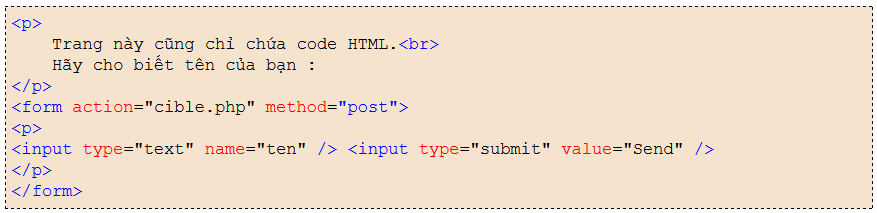


Code PHP

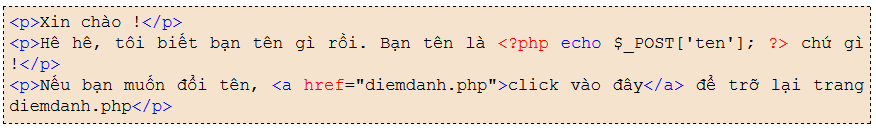


Phào, bạn thấy sao ? Dễ chịu chứ nhỉ ?   
Bạn bắt đầu để ý một điều thú vị của php rằng : source code của trang cible.php có chút xíu, mà trang web lại hiện ra mỗi lần mỗi khác !   
Tạm dừng ở đây để xem bạn tự test vài ví dụ nữa để làm quen với cách chuyển đổi biến kiểu này nhé !  
  
**Chuyển biến bằng trình đơn (formulaire ; form)**  
  
Một cách khác để chuyển biến từ trang này sang trang khác là dùng trình đơn (form), với những vùng gõ text, những ô vuông để chọn, những nút options,…  
Tôi dành hẳn một chương để nói về mấy vụ form này (phần III), coi vậy chứ cũng rắc rối lắm. Còn bây giờ mình hãy làm những cái đơn giản nhưng thường gặp thôi nhé !  
Cái đơn giản nhất là vùng text.  
Bạn đã gõ text vào vùng text trên một trang web lần nào chưa ? Tôi dám cá là nhiều lắm rồi, bạn gõ vô tội vạ bất cứ thứ gì có thể gõ được khi comment trong blog của tui đó mà ! Ha ha, đùa tí thôi !  
Như vậy là bạn đã biết vùng text là cái gì rồi đấy, mục đích của mình bây giờ là làm sao để thu lại những gì mà ai đó đã gõ vào !  
Làm như hồi nãy, mình cần một trang trên là diemdanh.php có vùng text để gõ text vào, và một trang tên là cible.php để hiển thị ra cái gì mình đã gõ !

Code HTML



Code PHP



Chú ý : nếu mình muốn thu lại giá trị từ một form, mình dùng biến $\_POST['xxxx']  
Nếu muốn thu hồi giá trị từ một address, mình dùng biến $\_GET['xxxx']  
Bạn hãy thử cho quen với cách thu hồi biến kiểu form này đi nhé ! Chúc vui vẻ với những cái tên quái gở mà bạn sắp phát minh ra ! Nhưng hãy nhớ mục đích của mình là hiểu được cơ chế hoạt động của nó !  
Ở đây chỉ có mỗi một thứ cần hiểu là « action » chỉ ra trang web sẽ hiển thị ra những gì mình sắp gõ vào khi mình nhấp vào nút Send, còn tên của vùng text sẽ là tên biến được tạo !  
<input type="text" name="ten" />  
Tên của vùng text ở đây là “ten”. Trong trang cible.php, một biến $\_POST['ten'] sẽ được tạo và sẽ mang giá trị là những gì bạn gõ vào trong vùng text.

Khi dùng PHP, sớm muộn gì bạn cũng cần phải tính toán, công việc này còn sẽ được lặp đi lặp lại nhiều nữa là khác ! Trong chương trước tôi chỉ cho các bạn xem những tính toán cộng trừ nhân chia cở bản mà thôi.  
Ở đây tôi không hướng dẫn các bạn về toán học, mà chỉ cho các bạn xem làm sao sử dụng hàm trong PHP để thực hiện các tính toán một cách tự động và nhanh gọn thôi.

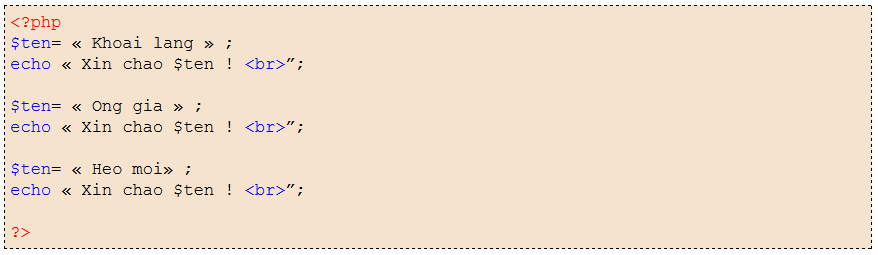
* Trong chương này:

|  |  |
| --- | --- |
| 1-5 | [Tự tạo hàm](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong5.php#taoham)  [Biến PHP thành cái đồng hồ biết nói](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong5.php#dongho) |

**Tự tạo hàm riêng**



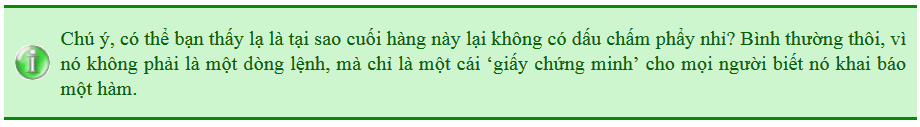
Một hàm là một tập hợp các cấu trúc lệnh (dòng lệnh) và cuối cùng cho ra một giá trị. Nếu bạn phải thực hiện những tính toán khá dài dòng, phức tạp và phải thực hiện nhiều lần cùng một tính toán như vậy, thay vì mỗi lần tính bạn viết một đống dòng lệnh, bạn chỉ cần viết một lần và đặt tên cho nó là một hàm, rồi sau đó gọi lại hàm này bất cứ lúc nào bạn muốn, không cần phải viết lại một đống dòng lệnh nữa !   
Tôi sẽ cho các bạn xem tại sao dùng hàm lại tiện lợi và hữu ích !  
  
Ví dụ 1 : nói « Xin chào ! » với ai đó  
Nếu diễn đàn của bạn có 1000 thành viên, chẳng lẽ để chào một người bạn lại viết một dòng ? Mệt nhỉ ! Nhưng yên tâm, hàm sẽ giúp bạn chào họ một cách tự động.  
Hãy xem đoạn mã sau :



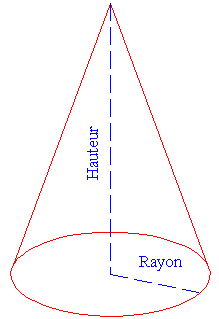
Bạn thấy đấy, hơi dài dòng mệt mỏi nhỉ! Mình hãy viết một hàm để nói xin chào nào!



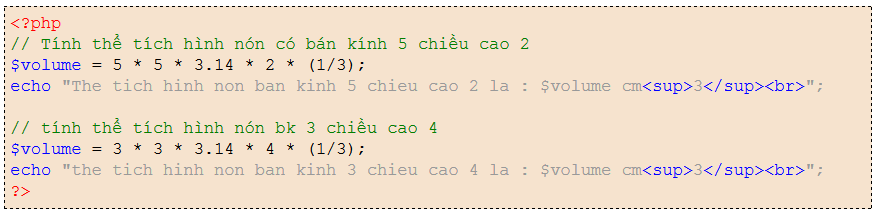
Xem nào, có gì mới ở đây nhỉ? Ở những dòng đầu tiên chỉ có một thứ mới, đó là hàm! 4 dòng đầu tiên cho phép mình tạo ra một hàm (gồm tên hàm, những gì nó có thể làm,…)  
Để tạo một hàm, bạn phải gõ vào chữ function (trong tiếng anh nghĩa là hàm), kế đến đặt tên cho hàm (trong ví dụ này, hàm có tên là XinChao)  
Cái hơi đặc biệt một chút nằm ở phía sau, những gì mình đặt trong dấu ngoặc đơn á! Bạn có thấy tôi đặt trong đó một biến không? Những gì mình đặt trong dấu ngoặc đơn được gọi là thông số của hàm! Tức là những thông tin mà hàm cần đến để làm việc bên trong.

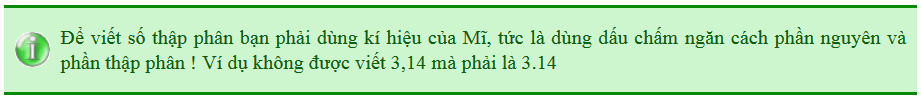


Tiếp theo, nội dung của hàm phải được đặt giữa hai dấu ngoặc móc ({}) để biết giới hạn của hàm, bắt đầu ở đâu, kết thúc chỗ nào!  
Ở đây tôi chỉ dùng có một dòng lệnh bên trong hàm (lệnh echo) để làm một ví dụ đơn giản, nhưng rồi bạn sẽ thấy mình có thể đặt rất nhiều dòng lệnh bên trong nữa!  
Vậy là một hàm đã được tạo, bạn không cần đụng đến nó nữa, sau này bạn chỉ cần gọi tên hàm kèm theo thông số của hàm là được! Ở đây hàm XinChao chỉ có một thông số chỉ tên của người mà mình muốn chào! Và đừng quên lần này phải có dấu chấm phẩy cuối cùng vì bây giờ nó là một câu lệnh rồi!   
Ví dụ : XinChao(“Khoai Tay”);  
Nào, bạn hãy thử xem, thử viết một trang chứa hàm này rồi nói xin chào với ai đó xem!  
  
Ví dụ 2 : Tính thể tích hình nón!  
Dzô, chạy với tốc độ nhanh hơn nào! Hãy nghĩ xem nếu tôi nhờ bạn tính giùm thể tích một hình nón, bạn sẽ tính thế nào đây?   
Hì, hẳn là bạn thừa biết công thức tính rồi phải không? Mình sẽ cung cấp cho hàm bán kính đáy và chiều cao của hình nón, rồi cho nó chiên xào nấu nướng gì bên trong đó, cuối cùng phải cho mình biết thể tích của nón là bao nhiêu!  
Khác với ví dụ trước, ví dụ này mình bắt hàm **cho ra một giá trị!**   
Rồi, tôi quên công thức rồi! Không sao, có internet mà, tôi đã tìm ra công thức tính thể tích hình nón rồi nè: **bán kính \* bán kính \* 3.14 \* chiều cao\* (1/3)**

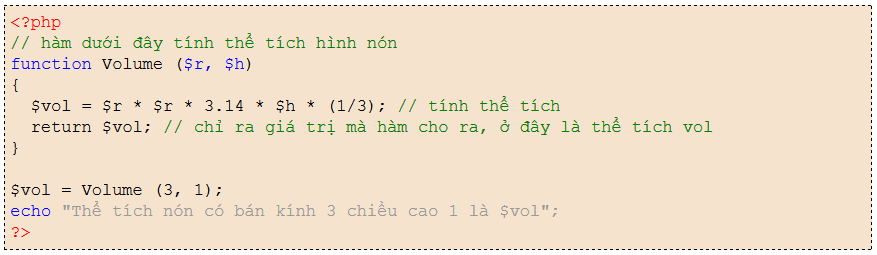


Nếu bạn đã theo dõi kĩ những phần từ trước đến giờ, bạn sẽ dễ dàng hiểu được đoạn code sau đây nói gì thôi:  
**Code : PHP**





Mình sẽ viết một hàm Volume chẳng hạn, với các thông số là bán kính r và chiều cao h, hàm này không in ra gì cả, chỉ đơn giản cho mình giá trị thể tích của hình nón thôi ! Hãy xem, có 2 cái mới ở đây đấy :  
**Code : PHP**



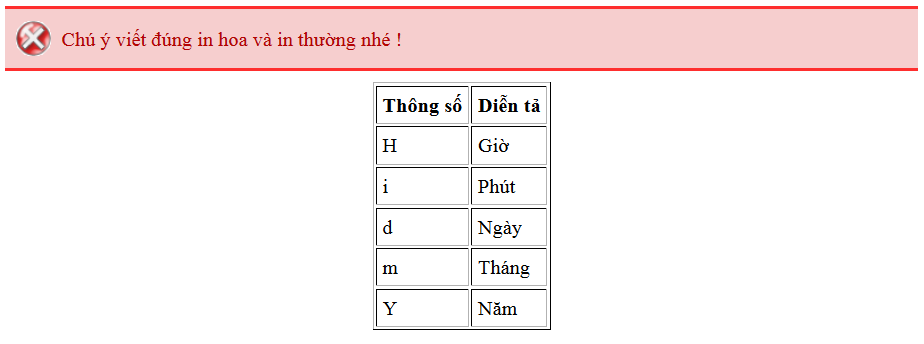
Hãy xem kĩ hàm, trong đó có cấu trúc : return $vol ;  
Cấu trúc này chỉ ra giá trị mà hàm sẽ cho ra, ở đây là thể tích ! Nếu bạn viết return 15 ; thì lần nào bạn gọi hàm này nó cũng chỉ cho ra 15 thôi. Thử xem.  
Như vậy mình có thể thấy rằng hàm không phải chỉ có một cách sử dụng, ở đây vì hàm cho ra một giá trị, nên mình phải gán nó vào một biến : **$vol =Volume(3,1) ;**  
Sau đó mình mới dùng echo để in ra giá trị mà $vol chứa ! Nếu bạn làm đúng, nó phải cho ra giá trị 9.42 (trong trường hợp này) !  
Cái mới thứ hai là có 2 thông số trong dấu ngoặc đơn ! Như bạn thấy đấy, một hàm có thể chứa nhiều thông số, bạn chỉ cần đặt dấu phẩy vào để ngăn cách các thông số là đủ !  
Khả năng sáng tạo của hàm có thể nói là vô hạn, thường thì rõ ràng là bạn không viết hàm chỉ để tính thể tích hình nón đâu nhỉ (hơi nhảm nhảm sao í). Những gì tôi muốn bạn hiểu ở đây là cách dùng hàm và sự tiện lợi của nó, nó giúp bạn tiết kiệm khá nhiều thời gian đấy !  
Sao bạn không thử tạo ra vài hàm khác chơi cho vui chút xíu trước khi mình qua phần khác nhỉ ?

**Biến PHP thành cái đồng hồ biết nói**

Bạn vừa xem cách tạo hàm, cách dùng hàm, cơ chế hoạt động cũng như biết hàm phục vụ cho cái gì ! Nhưng bạn vẫn chưa khám phá hết các hàm của PHP đâu, nó có hàng trăm « hàm ăn liền » lận đó !  
Tôi để ý rằng mỗi khi tôi viết một hàm gì đó, thì hầu như PHP đã có hàm đó sẵn rồi mà tôi không biết ! Rất nhiều và rất tiện !  
Bạn cần phải nhớ là có 2 loại hàm, một loại thực hiện những thao tác, không trả lại gì cả ! giống như thủ tục (procedure) trong Pascal á !   
Còn một loại (sau khi đã thực hiện hàng đống tính toán) trả lại một giá trị ! (mà mình dùng khá thường xuyên sau này)  
Sau đây là vài kiểu « hàm ăn liền » phục vụ các bạn đây :

* Một hàm cho phép tìm kiếm và thay thế từ ngữ trong một biến
* Một hàm cho phép gửi một file lên máy chủ
* Một hàm cho phép tạo hình thu nhỏ (avatar)
* Một hàm cho phép gửi mail tự động (cái này rất tiện để làm newsletter nhỉ)
* Một hàm cho phép sửa hình, như thêm vào text, vạch vài đường, vẽ một hình vuông hay hình méo gì đó vô hình
* Một hàm làm ẩn password
* Một hàm hiển thị ngày giờ,..

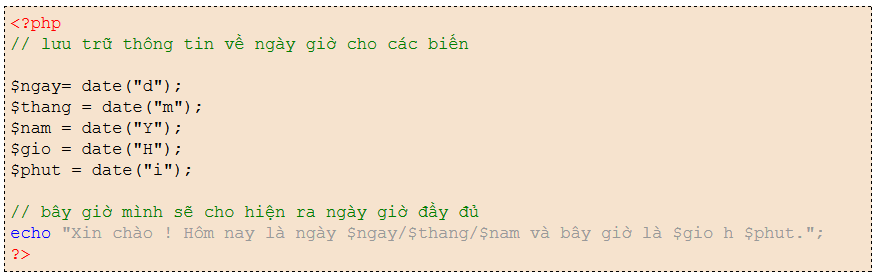
Vân vân và vân vân …  
Bây giờ mình thử nghịch cái hàm cho hiện ngày giờ nhé ! Hàm tên là date, đây là một loại « hàm ăn liền », nghĩa là mình không cần phải viết function date(..) làm gì, chỉ cần sử dụng bằng cách cho nó vài thông số mà mình cần là đủ !  
Sau đây là 5 thông số thường dùng cho hàm date này :



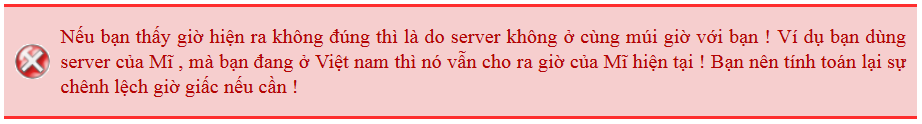
Và còn khoảng 30 thông số khác cho hàm date này nữa, mình sẽ gặp lại trong phần III khi bàn chi tiết hơn về hàm !  
Nếu bạn muốn in ra năm : (2007)  
**Code : PHP**



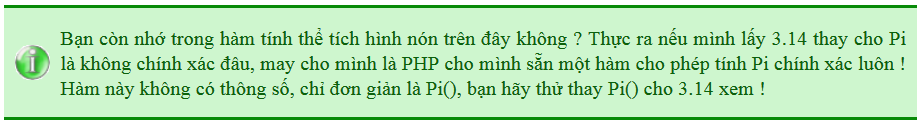
Hoặc ngày giờ đầy đủ:  
**Code : PHP**



Xong rồi ! Bạn cho chạy thử xem ! Chú ý là hàm cho ra ngày giờ lúc bạn chạy chương trình, nên đương nhiên mỗi lúc một khác nhau !



Chúng ta sắp kết thúc phần cơ bản về PHP trong vài chương nữa, những gì bạn đã học hôm nay về hàm chỉ là một phần rất nhỏ trong PHP, vẫn chưa đủ để bạn tạo một trang web cực kì hấp dẫn. Nhưng hãy kiên nhẫn, không lâu nữa bạn sẽ thấy nó hữu ích thế nào .



# Chương 6 : Điều kiện



Thỉnh thoảng mình cũng cần phải in ra những thứ khác nhau tùy theo một điều kiện nào đó, chẳng hạn nếu buổi sáng thì mình in ra câu « Chào buổi sáng ! » còn nếu buổi tối thì in ra « Chào buổi tối ! » ...  
Cũng từ đó mà nảy sinh ra cái vụ dùng điều kiện để ra lệnh cho PHP tùy trường hợp cụ thể. Ở ví dụ của chúng ta, chúng ta sẽ bảo PHP « nếu là buổi sáng thì chú nói « Chào buổi sáng », nếu tối thì chú « Chào buổi tối ! » nghe ! » Rồi bạn sẽ thấy điều kiện thực sự là cơ sở để tạo một trang web động, để xử lý tình huống tùy theo ngày, giờ, thành viên, …

* Trong chương này:

1-6

* + [Cấu trúc cơ bản: if … else](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong6.php#coban)
  + [Luân chuyển Switch](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong6.php#luanchuyen)

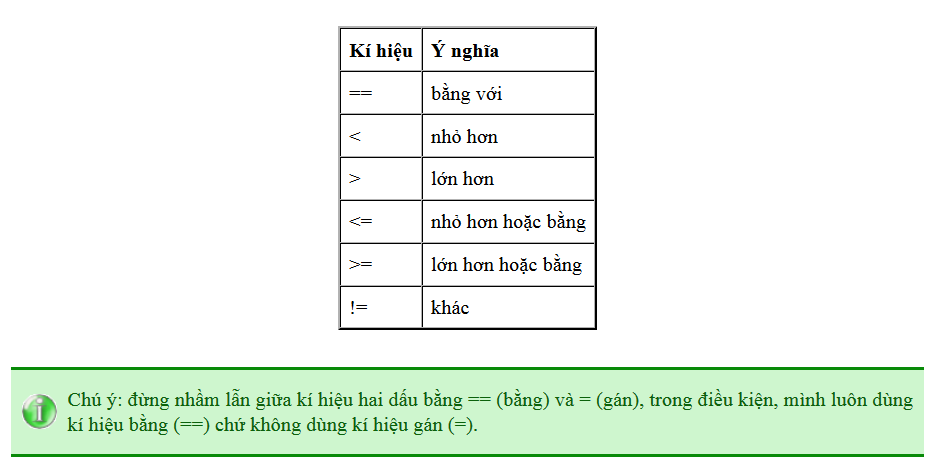
[precedent chapter](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong5.php)[Mục Lục](http://thanhtra.nguyen.free.fr/web/tuto-php/modau.php)[next chapter](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong7.php)

**Cấu trúc cơ bản : If … else**

Mình gọi đó là một cấu trúc bởi nó có một “dạng” hơi đặc biệt!  
Để nghiên cứu cấu trúc if…else, chúng ta hãy theo trình tự sau đây:

* Vài kí hiệu cần biết: Trước hết là cần phải nắm vài kí hiệu dùng trong so sánh. Phải ghi nhớ kĩ bởi chúng ta sẽ dùng nhiều khi sử dụng điều kiện
* Cấu trúc if…else: cái này hơi bự đa, ở đây mình sẽ xem cơ chế hoạt động của cấu trúc if…else như thế nào. Tôi khỏi nói các bạn cũng tự hiểu là không thể lơ là ở phần này.
* Nhiều điều kiện cùng lúc
* Trường hợp Boolean: cũng có lúc mình làm việc với những điều kiện dạng “nếu đúng thì… nếu sai thì…”; đó gọi là dùng các biến Boolean (biến có giá trị hoặc true hoặc false)
* Mẹo vặt

**Vài kí hiệu cần biết**:



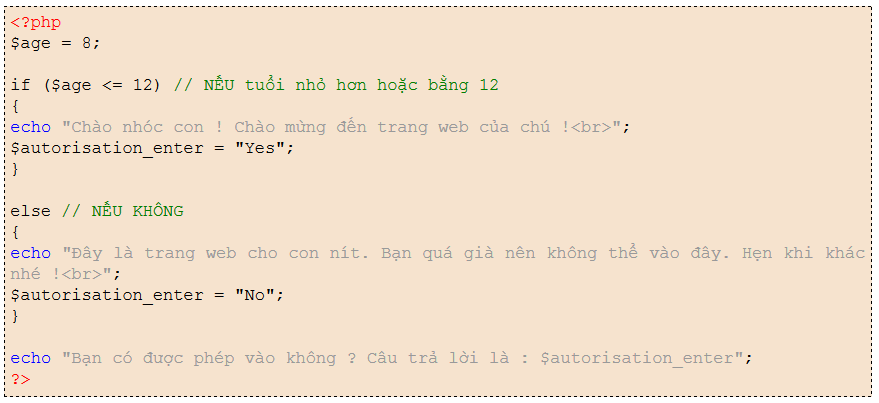
**Cấu trúc if … else**  
  
Sau đây là thứ tự cần tuân theo khi dùng cấu trúc if…else

* Để bắt đầu sử dụng điều kiện, mình ghi “if”
* Tiếp đó, mình thêm điều kiện cần dùng vào trong dấu ngoặc đơn ngay sau if
* Cuối cùng, mình đặt nội dung cần thực hiện nếu điều kiện đúng vào giữa hai dấu ngoặc móc ({})

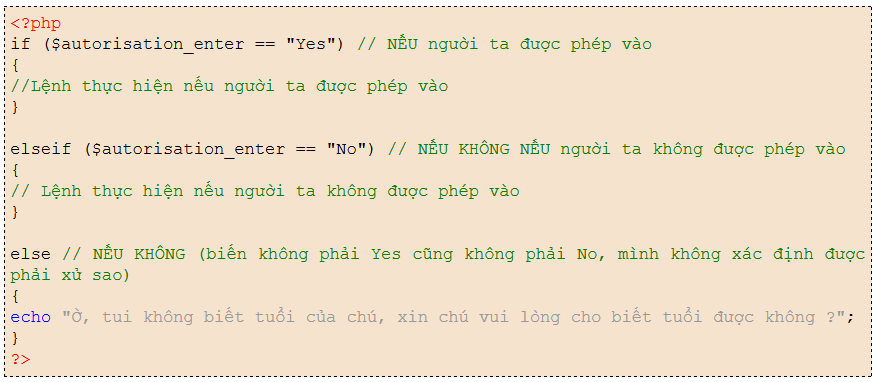
Và một ví dụ bao giờ cũng dễ hiểu hơn một tràng chữ nghĩa:  
**Code : PHP**



Ở đây,mình đã bảo PHP “nếu biến $age mà nhỏ hơn hoặc bằng 12 thì hãy in ra câu “chào nhóc con” nhé!”  
Để ý rằng trong hầu như tất cả trường hợp, mình đều dùng một biến để áp đặt cho nó một điều kiện nào đó!  
Ở ví dụ của mình, mình làm việc trên biến $age. Có hai khả năng của $age: nếu $age nhỏ hơn hoặc bằng 12 thì in ra cái gì đó, nếu không thì nhảy ra khỏi hai dấu ngoặc móc, không làm gì cả!  
Bây giờ hãy cải thiện ví dụ của mình chút xíu nhé, hãy thử in ra thứ gì khác nếu $age lớn hơn 12 xem nào!  
  
**Code : PHP**

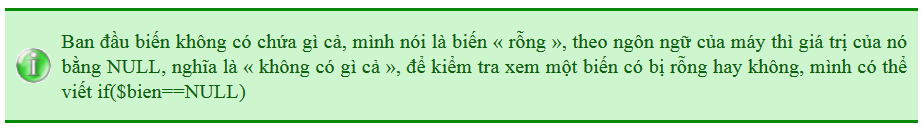


Bạn thử nói xem cái code trên đây hoạt động ra sao nào ?   
Trước hết, tôi đã đặt rất nhiều câu lệnh giữa các dấu ngoặc móc, đừng quên rằng bạn có thể đặt nhiều lệnh vào giữa các dấu ngoặc móc nhé ! Kế đến, bạn để ý rằng tôi có thêm từ « else » (trong tiếng anh nghĩa là « nếu không »), rõ ràng hơn chút , ý tưởng của mình là « nếu tuổi nhỏ hơn hoặc bằng 12 thì làm cái này, nếu không thì làm cái kia ! »  
Hãy thử sửa đổi một chút giá trị của biến $age và những lệnh bên trong để quen dần với cấu trúc này nhé !  
Nhớ kĩ là bạn có thể thêm bao nhiêu lệnh tùy mục đích và tùy thích ! Ở ví dụ trên tôi đã cho in ra vài thứ và đã gán cho biến $autorisation\_enter 2 giá trị khác nhau để xài cho ví dụ sau đây :  
  
**Code : PHP**

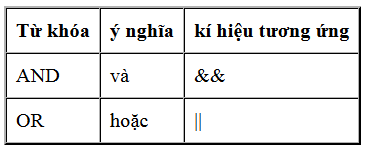


A, cuộc đời bắt đầu thêm rắc rối rồi đa !  
Cái mới ở đây là từ khóa elseif (nếu không nếu) theo thứ tự, PHP sẽ gặp những điều kiện sau đây :

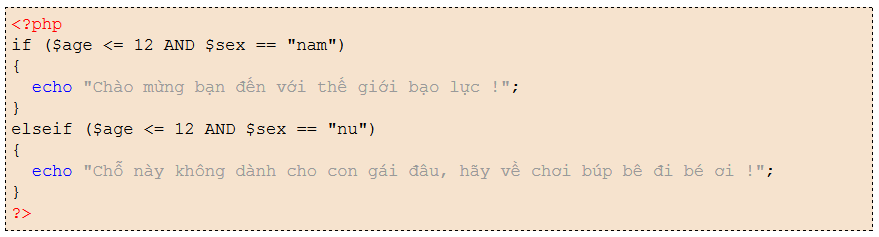
* nếu $autorisation\_enter bằng với yes, thì mình làm cái này
* nếu không, nếu $autorisation\_enter bằng với No thì mình làm cái kia
* nếu không, mình hỏi lại tuổi để biết người đó có được phép vào hay không



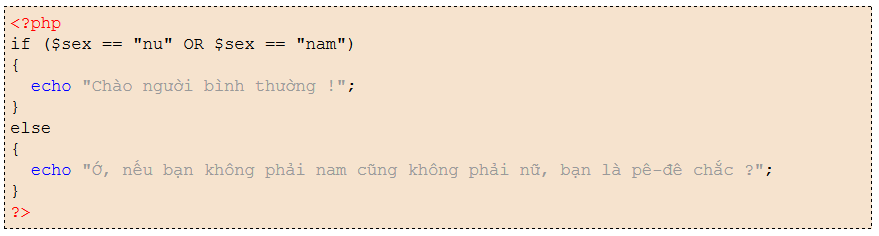
**Nhiều điều kiện cùng lúc :**  
  
Cố lên ! Còn chút xíu nữa là mình sẽ xong mục này ! Từ bây giờ mình sẽ thử làm cho cuộc đời thêm phức tạp chút nhé, từ từ rồi bạn cũng sẽ quen với từ phức tạp thôi, không phức tạp đâu có vui phải không nào ?  
Để sử dụng nhiều điều kiện cùng lúc, trước tiên bạn cần nắm vài từ khóa mới :



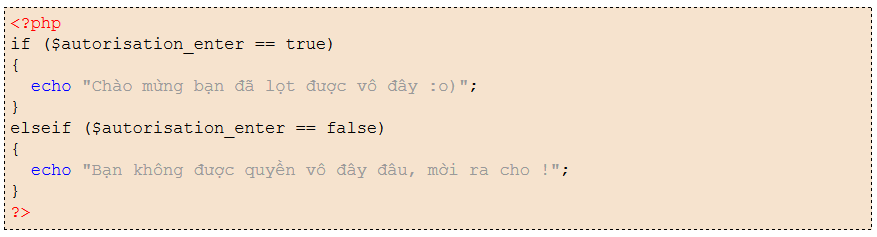
Ví dụ  
**Code : PHP**



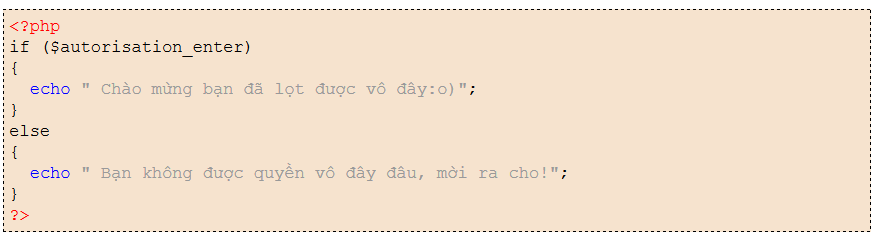
Tôi đã phát ngán và bạn cũng đã phát chán nghe giảng giải về cách hoạt động của đoạn mã trên rồi nhỉ ! Nếu bạn theo dõi kĩ từ nãy đến giờ thì đoạn mã bé tí tẹo này sao làm khó bạn được đúng không ?  
Ráng làm thêm một ví dụ về OR nữa, rồi tôi hứa là sẽ dừng lại mục nhỏ này ở đây :  
  
**Code : PHP**



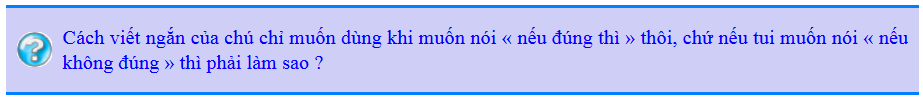
**Trường hợp boolean (biến logic)**  
Ở chương trước mình có nói về cái gì là biến logic (Boolean) rồi nhỉ? Biến logic hay Boolean chỉ có hai giá trị là “true” hoặc “false” thôi! Dùng Boolean trong phần điều kiện này khá tiện lợi đấy!  
Ở ví dụ trên, xém chút nữa là tôi đã xài Boolean rồi! Đố bạn biết là chỗ nào! Để tôi viết lại trong đoạn mã sau đây, rồi bạn coi thử có phải ý của bạn không nha :  
  
**Code : PHP**



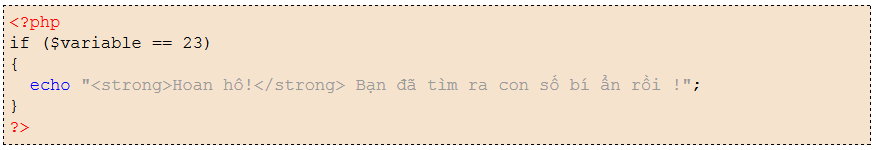
Một cách viết khác linh hoạt hơn khi dùng biến boolean, mình có thể bỏ đi ==true hoặc ==false, mà PHP vẫn hiểu  
Ví dụ :  
  
**Code : PHP**



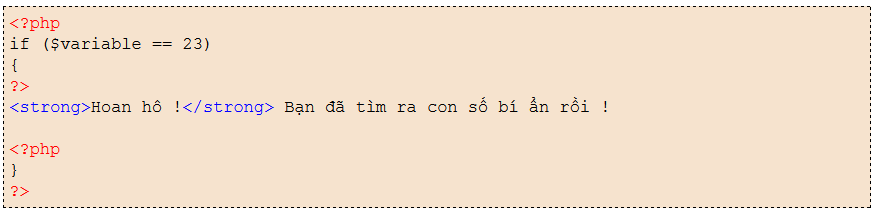
Cách viết ngắn này sẽ rất thuận tiện cho mình về cách đọc mã, chẳng hạn mã trên đây mình có thể đọc : nếu được phép vào thì nói « chào mừng … »



Có một kí hiệu cho phép kiểm tra « nếu điều kiện sai », bạn chỉ cần thêm dấu chấm than ( !) vào trước biến là ổn, chẳng hạn if( !$autorisation\_enter) được hiểu là « nếu không được phép vào », cách viết này tương đương với : if($autorisation\_enter==false) .  
**Mẹo vặt**  
Hai đoạn mã sau đây đều cho ra cùng một kết quả :  
  
**Code : PHP**



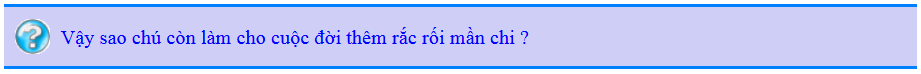
**Code : PHP**



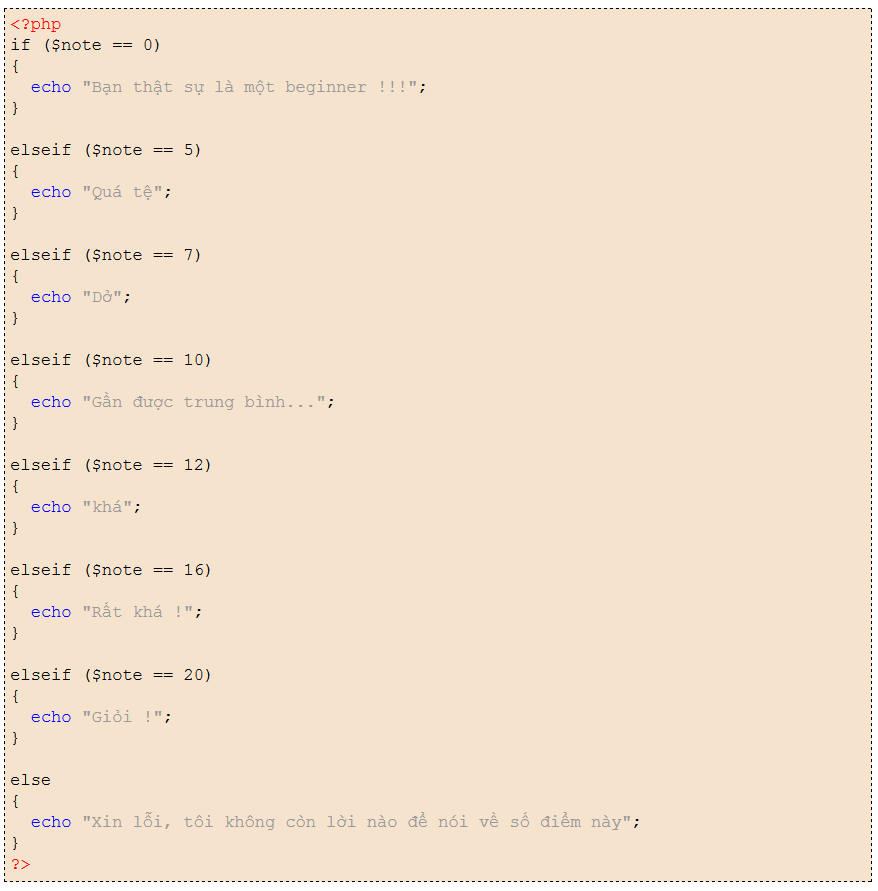
Như bạn thấy, ở code thứ 2 tôi đã không dùng lệnh echo để in text ra nữa, thay vào đó tôi chỉ cần mở dấu ngoặc móc ra, rồi đóng dấu tag PHP lại (dấu ?> á), sau đó tôi muốn cho in ra cái gì cũng được bằng HTML, điều này thật tiện lợi khi cái mình cần in ra không phải chỉ một dòng mà là cả một trang thì dùng echo sẽ bị hạn chế về trình bày (trang trí hoa lá cành này nọ) hoặc dễ bị lỗi. Và sau khi xong việc, tôi đóng dấu ngoặc móc lại (chú ý là dấu ngoặc ngày luôn luôn phải được đặt trong tag PHP (<?php } ?>) )

## Switch

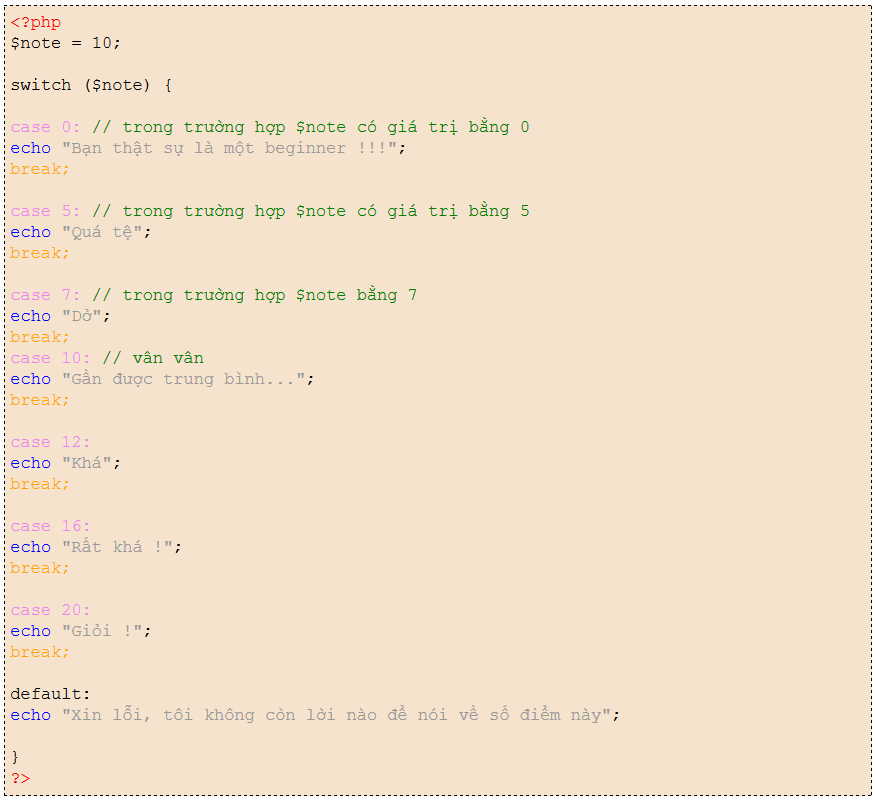
Theo lý thuyết, chỉ với if…elseif….else thôi cũng đủ để chúng ta xử lý các điều kiện rồi !

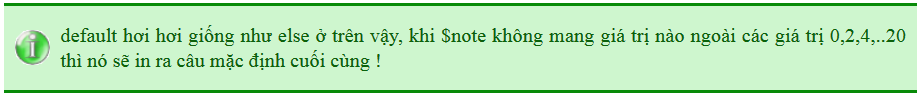


Đừng nóng, để cho các bạn thấy sự lợi hại của Switch, hãy xem ví dụ sau :  
  
**Code : PHP**



Bạn có thể ứng dụng cái này để làm bảng đánh giá kết quả trắc nghiệm được đấy. Bạn cũng thấy là cách viết này khá cồng kềnh, lại có sự lặp đi lặp lại phải không ? Hãy xem switch sẽ làm sao để giảm bớt cồng kềnh và lặp đi lặp lại này nhé :  
  
**Code : PHP**





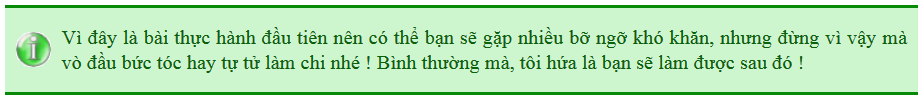
Lệnh break ; yêu cầu PHP ra khỏi vòng switch. Nếu không có break, PHP sẽ kiểm tra lần lượt hết tất cả các trường hợp mà mình đã viết, như vậy khá mất thời gian (mà có khi sẽ làm cho kết quả không đúng nữa), nên mình phải dùng break để sau khi thực hiện xong việc cần làm rồi thì nó phải nhảy ra, không kiểm tra mấy cái linh tinh vớ vẩn kia nữa !



Cái này còn tùy vào kĩ năng trình bày và sự khéo léo của bạn. Nếu gặp một điều kiện ngắn và đơn giản bạn nên dùng if, còn nếu gặp một chuỗi điều kiện cần phân tích thì mình nên dùng switch để làm cho code của mình được rõ ràng hơn !

# Chương 7 : Thực hành Tạo một trang web được bảo vệ bằng mật khẩu

Lời đầu tiên tôi xin chúc mừng bạn đã vượt qua 6 ải môn quan để đến với bài thực hành đầu tiên này !  
Đây không phải một chương như những chương khác, bạn sẽ không phải học thêm những gì mới mẻ mà sẽ ứng dụng những gì bạn đã học để viết những đoạn script PHP đầu tiên ! Học phải đi đôi với hành mới nhớ dai chứ nhỉ ?!  
Nói nhỏ nghe nè : bạn đã đủ trình độ để bảo vệ trang web của bạn bằng mật khẩu (password) rồi đấy ! Đó là nội dung mà hôm nay tôi sẽ hướng dẫn các bạn !



**Làm một trang được bảo vệ bằng mật khẩu**

**Bước 1 : Kiến thức cần có**

* Hiển thị text bằng lệnh echo
* Thao tác với biến (gán, hiển thị )
* Chuyển biến thông qua một vùng text
* Sử dụng điều kiện đơn giản (if … else)

Nếu những kiến thức này còn mơ hồ với bạn (có thể bạn đã quên) thì đừng ngại xem lại những chương liên quan nhé! Bạn sẽ thấy rằng bài thực hành này không đòi hỏi am hiểu nhiều điều phức tạp, nhưng vì là bài đầu tiên nên mình sẽ làm đơn giản ngắn gọn thôi!  
  
**Bước 2: chuẩn bị viết mã**

Để chuẩn bị viết script, tôi khuyên bạn nên viết ngoài giấy nháp trước, hơi mất thời gian một tí nhưng code của bạn sẽ có cấu trúc dễ nhìn hơn và bạn sẽ tránh được một số lỗi thông thường khi gõ vào máy!

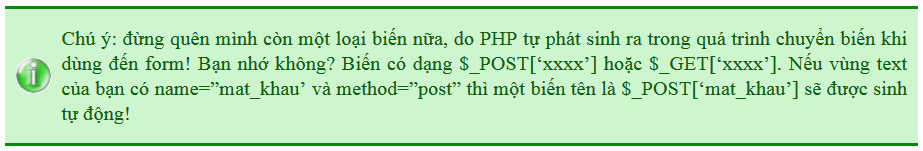


* Về vấn đề bạn đặt ra nè (tớ muốn làm gì đây?)
* Về sơ đồ code, nghĩa là bạn sẽ cắt code ra làm nhiều đoạn nhỏ để xử lý từng đoạn, dù sao viết một đoạn nhỏ vẫn dễ hơn làm một đoạn lớn chứ nhỉ?!
* Về những hàm và kiến thức PHP mà bạn sẽ cần đến
* Về các biến bạn sẽ sử dụng (hãy đặt tên gì cho biến, biến sẽ chứa cái gì,…)

Để ví dụ, bạn hãy theo dõi list sau đây cho bài thực hành của mình nhé:

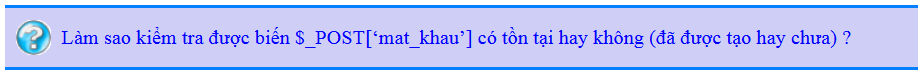
1. Vấn đề đặt ra: bạn đã viết một trang web chứa nội dung tuyệt mật (cho bộ quốc phòng hay trung tâm vũ trụ quốc tế NASA chẳng hạn), bạn sẽ dùng password để chỉ có bạn và bạn bè mới xem được nội dung trong đó!
2. Sơ đồ code: trước hết mình chỉ làm việc trên một trang thôi! Trang này chứa một đoạn text đại khái yêu cầu nhập vào mật khẩu, nếu mật khẩu đúng thì cho xem nội dung trang web, nếu sai thì yêu cầu nhập lại! (Có ‘mùi’ của điều kiện rồi nhỉ?!)  
   Có thể bạn đang nghĩ tới việc làm cái gì đó hơi phức tạp, và cũng có thể bạn sắp làm vài thứ càng phức tạp hơn nhưng chẳng để làm gì cả! He he, những gì tôi giới thiệu ở đây đã được đơn giản tối đa rồi, mình cũng chẳng cần làm gì phức tạp hơn việc sử dụng một điều kiện đơn giản!
3. Những hàm và kiến thức cần nắm: như tôi đã nhắc ở trên
4. Biến cần dùng: theo ý bạn, mình sẽ sử dụng bao nhiêu biến đây? 4? 6? 10?

Không đâu không đâu, chỉ một biến là đủ xài cho bài thực hành của mình rồi! Bạn có đoán ra biến đó để làm gì rồi nhỉ? Mình sẽ dùng một biến để chứa password nhập vào đó mà!  
Bạn đã nghĩ ra một cái tên cho biến chưa? Lưu ý đừng đặt tên quá dài, quá khó hiểu, đừng đặt trùng với các từ khóa, không chứa khoảng trắng v.v… (hình như mình nói vụ này trong chương Biến rồi thì phải), ở đây, bạn hãy ***tránh*** đặt tên biến là $pass hay $password vì đây là 2 ***từ khóa*** của PHP.  
Ở đây tôi đặt tên cho biến là $mat\_khau nhé! Dấu gạch underscore\_ thay cho khoảng trắng là tiện nhất!

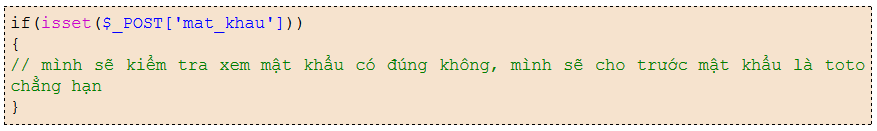


Vấn đề cuối cùng: lần đầu tiên bạn tải trang web về, bạn cần phải kiểm tra xem biến $\_POST[‘mat\_khau’] có tồn tại hay không? Tức là nó đã được tạo hay chưa? Nếu chưa thì bạn sẽ bị lỗi! Chẳng hạn bạn hỏi PHP rằng “có phải mat\_khau là toto hay không” trong khi biến này chưa ra đời, PHP sẽ báo lỗi!  
Vì vậy có 2 khả năng như sau:

* Khi bạn tải trang web về lần đầu tiên, bạn chưa có điền gì vào form nên biến $\_POST[‘mat\_khau’] chưa được tạo! Cho nên có hỏi PHP mật khẩu có đúng không cũng vô dụng! Nó đâu có biết đúng hay không!
* Khi bạn đã gõ vào mật khẩu rồi bấm nút “Send”, thì biến $\_POST[‘mat\_khau’] đã được tạo, lần này bạn mới kiểm tra xem mật khẩu có đúng hay không để cho phép xem nội dung hay không!



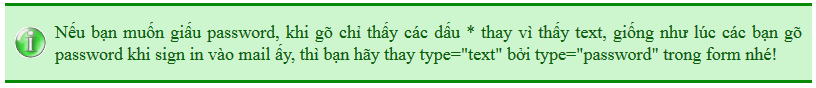
Mình sẽ dùng đến một hàm PHP tên là isset , hàm này có giá trị true nếu biến đã được tạo, false nếu chưa! Cấu trúc của isset như thế này: isset($\_POST[‘mat\_khau’]), nếu mình dùng if … else để kiểm tra điều kiện, mình có thể viết:



Xong rồi, bước chuẩn bị đã xong! Bắt tay nấu nướng nào!  
  
**Bước 3: Tới lượt bạn chơi!**  
  
Mình đã cùng nhau chuẩn bị công thức chế biến rồi, bạn thử nấu xem! Cũng đừng ngạc nhiên nếu code của bạn không chạy! Nhưng cũng đừng nản lòng nhé, nếu bạn đã theo trình tự tôi hướng dẫn, tôi tin là bạn sẽ thành công, người Việt mình vốn thông minh mà (ngoại trừ tôi!), bạn hãy chạy WAMP, mở notepad++ ra (hoặc text editor mà bạn đang dùng) rồi gõ code vào! Sau đó vào localhost chạy thử!  
Bạn hãy vô tư phát minh ra một password nào đó nhé! Nhưng đừng lấy password email thật của bạn đấy, tôi không đảm bảo rằng sẽ không có ai phát hiện ra đâu! Chúc may mắn! Tôi đi chơi một lát đây, lát nữa trỡ lại coi kết quả nha!  
  
**Bước 4: Sửa!**  
  
Dzô, xem bạn đã làm gì nào, mình cùng nhau sửa nhé! Bạn chỉ nên xem phần này khi bạn đã xong việc của bạn để so sánh với kết quả của tôi, hoặc bạn bị bí hoàn toàn!  
**Code : PHP**

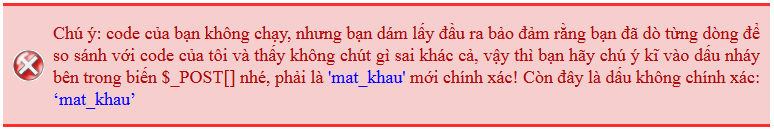


[Nhấp vào đây](http://thanhtra.nguyen.free.fr/web/Test/protection.php) để chạy thử đoạn code trên!



Đoạn code khá đơn giản hỉ? Tôi xin giải thích rõ hơn vài điểm trên này mà bạn có thể đặt câu hỏi:

* Mình bắt đầu bằng việc kiểm tra xem biến $\_POST[‘mat\_khau’] có tồn tại hay không, nếu tồn tại, mình tạo một biến tên là $mat\_khau để chứa password mà visitor gõ vào. Chú ý là biến $\_POST[‘mat\_khau’] là do PHP tạo chứ không phải mình, giá trị của nó chính là những gì mà visitor gõ vào (text). Biến mà mình tạo chỉ là $mat\_khau thôi! Nếu $\_POST[‘mat\_khau’] chưa tồn tại, đơn giản chỉ vì đây là lần đầu tiên tải trang web về, visitor chưa gõ gì vào cả! Mình sẽ tạo biến $mat\_khau rỗng (text rỗng được kí hiệu bằng hai dấu nháy "").  
  Đừng băn khoăn về isset, bạn chỉ cần nghĩ rằng viết như vậy chỉ là để kiểm tra xem $\_POST[‘mat\_khau’] có tồn tại hay không thôi!
* Tiếp theo, mình dùng điều kiện để kiểm tra xem $mat\_khau có đúng là mật khẩu mà mình đã giao trước hay không! Ở đây tôi chọn mật khẩu là *kangourou* đó!
* Tôi đã không hiển thị nội dung trang web bằng lệnh echo như thường lệ, bởi nó quá dài lại chứa nhiều tag rắc rối, nên như tôi đã chỉ cho các bạn biết một mẹo vặt để hiển thị nội dung dài dòng rắc rối mà không phải nhờ đến echo: chỉ cần đóng cái tag ?> lại, rồi làm việc với code HTML bình thường! Khi xong cái cần hiển thị, để trỡ lại làm việc với PHP mình phải mở tag php ra (<?php)
* Bạn để ý rằng thực ra có 2 trang trong 1 đoạn code ở trên! Bên ngoài tag php tôi đã dùng code HTML đơn thuần!
* Chuyện gì xảy ra khi mình chạy chương trình lần đầu tiên? Biến $\_POST[‘mat\_khau’] rỗng, có giá trị NULL. Do đó nó không bằng “kangourou”! Vì vậy những lệnh sau **else** sẽ chạy trước, nghĩa là trước hết bạn sẽ thấy trang yêu cầu nhập password! Bạn hãy cố ý gõ sai password một hai lần thử xem!



# Thực hành kết thúc!

**Chương 8 : Vòng lặp**

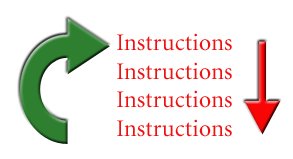
* Trong chương này:

|  |  |
| --- | --- |
| 1-8 | [Một vòng lặp đơn giản: while](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong8.php#while)  [Vòng lặp phức tạp hơn: for](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong8.php#for) |

**Một vòng lặp đơn giản : while**



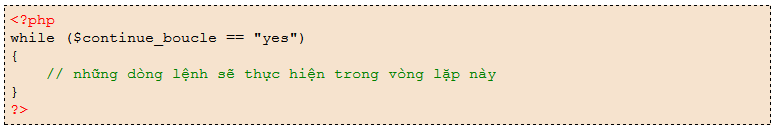
Một vòng lặp là một cấu trúc hoạt động cùng nguyên lý với điều kiện (if…else). Bạn sẽ thấy có nhiều điểm giống giống với chương điều kiện đấy !  
Cụ thể hơn, một vòng lặp cho phép lặp lại các dòng lệnh (hay các instruction) nhiều lần. Sơ đồ hóa như sau :



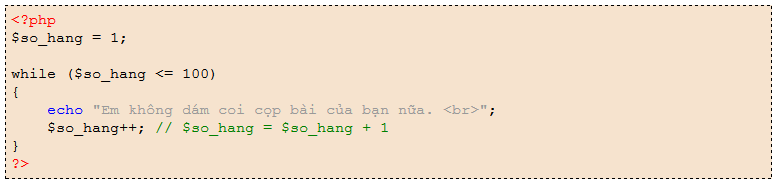
Đây là những gì xảy ra trong một vòng lặp :

* Như thường lệ, các dòng lệnh sẽ chạy lần lượt theo thứ tự từ trên xuống dưới (mũi tên đỏ)
* Cuối cùng, sau khi chạy dòng lệnh chót, nó sẽ chạy trỡ lại từ đầu (mũi tên xanh)
* Rồi nó lại bắt đầu chạy theo thứ tự từ trên xuống dưới (mũi tên đỏ)
* Rồi lại quay trỡ lại dòng đầu tiên sau khi chạy dòng chót (mũi tên xanh)

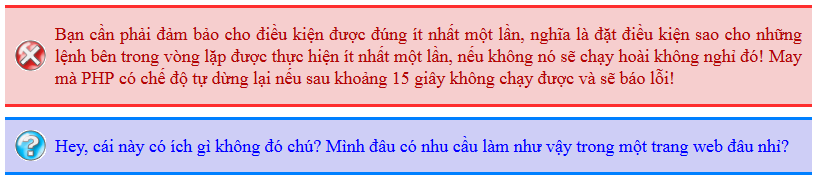
v.v…  
Điều hơi buồn cười ở sơ đồ này là vòng lặp sẽ chạy hoài chạy hoài, cứ xong dòng cuối nó lại trỡ lại dòng đầu, không dừng lại. Chính vì vậy mà cho dù mình dùng vòng lặp kiểu nào đi nữa (while hay for) mình cũng phải chỉ ra điều kiện ! Khi điều kiện còn đúng thì vòng lặp được tiếp tục, khi điều kiện không còn đúng nữa thì nhảy ra khỏi vòng lặp !  
Ví dụ một vòng lặp while đơn giản :  
  
**Code : PHP**



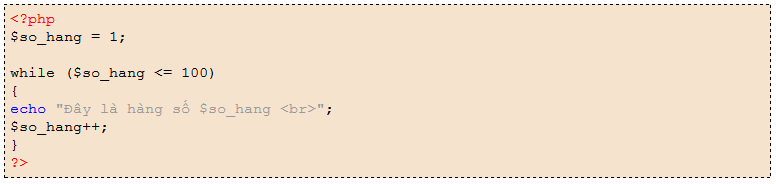
While có thể dịch là « khi mà », « khi mà continue\_boucle bằng yes thì chạy các dòng lệnh bên trong ;  
Những dòng lệnh bên trong vòng lặp được đặt giữa hai dấu ngoặc móc ! Mình bắt đầu gặp dấu ngoặc móc này khắp nơi rồi đa !  
Để làm quen với cách dùng vòng lặp, tôi xin đưa ra vài ví dụ cụ thể như sau :  
  
**Ví dụ 1** : Giả sử bạn bị phạt, bạn phải chép lại 100 lần câu « Em không dám coi cọp bài của bạn nữa ! » Tôi nhớ hồi nhỏ tôi đã phát khóc mỗi khi phải bỏ ra hàng giờ để chép phạt ! Bây giờ nhờ có PHP, tôi hết sợ bị phạt rồi ! Hehehe …  
  
**Code : PHP**



Cái vòng lặp này đặt điều kiện: khi mà $so\_hang nhỏ hơn hoặc bằng 100 thì in ra câu “Em không dám coi cọp bài của bạn nữa.”, Sau khi in xong dòng này, mình thực hiện một động tác làm tăng giá trị của $so\_hang lên 1 bằng cách dùng kí hiệu rút gọn ++ ($a++; tương đương với $a=$a+1;)  
Ban đầu $so\_hang có giá trị bằng 1, sau khi in xong nó được tăng lên 1, giá trị mới của nó là 2! Rõ ràng là 2<=100! Do đó điều kiện còn đúng nên tiếp tục in ra một câu nữa, in xong, lại được tăng lên 1 thành 3, và 3<=100 là đúng, cứ thế vòng lặp in đến hàng thứ 100, sau khi in xong hàng thứ 100 $so\_hang cũng sẽ được tăng lên 1 thành 101, lúc này điều kiện đã không còn đúng nữa (101>100 mà) nên vòng lặp sẽ dừng lại! Vậy là vèo một cái mình có 100 dòng rồi! Bây giờ dù cho cô giáo bắt chép 1000 dòng cũng đâu ngán ha! (nói vậy thôi, ai lại để cô giáo thấy làm chi rồi bị phạt, ha ha ha).

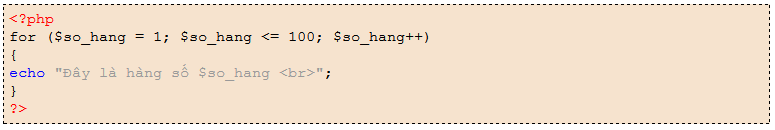


Đúng vậy! Thật khó nói cho bạn hiểu được cái này phục vụ vào đâu trong trang web, nhưng các bạn sẽ rõ trong phần II của bài viết này (về Cơ Sở Dữ Liệu). Tạm thời bạn hãy cố gắng hiểu cách sử dụng và cơ chế hoạt động của nó nhé!  
Bây giờ bạn hãy thử đoán trước xem ví dụ sau đây sẽ ra như thế nào nhé:  
**Ví dụ 2**:  
  
**Code : PHP**



## Một vòng lặp phức tạp hơn: For

Đừng hoảng nhé bạn, cái từ **phức tạp** này không có nghĩa là rắc rối khó nuốt đâu!   
Về cơ chế hoạt động của nó thì cũng giống với while và mục đích của nó thì cũng như while, chỉ có điều cách viết của nó hơi khác, ví dụ:  
  
**Code : PHP**



Mỗi một điểm khác ở hàng for: sau for là một cặp ngoặc đơn, trong đó mình đặt những gì?  
Đầu tiên là giá trị ban đầu của biến, rồi dấu chấm phẩy, kế đến là điều kiện của biến để vòng lặp tiếp tục lặp lại công việc, rồi dấu chấm phẩy, sau cùng là thao tác làm tăng giá trị của biến!  
Ví dụ này và ví dụ 2 ở trên cho ra cùng một kết quả!



While thì dùng lúc nào cũng được, for dùng khi biến có nhu cầu “tăng giá trị”! Nhưng mà không ai kề dao vào cổ bạn ép bạn phải dùng for nếu bạn chỉ thích while thôi đâu!

**Chương 9: Mảng (Array)**

Chúng ta sắp bắt tay vào một khía cạnh rất quan trọng của PHP : Array, các bạn sẽ thấy rằng bọn array giống như những biến phức hợp mà mình có thể tưởng tượng như một cái bảng (hay ma trận) chứa giá trị của biến trong đó !  
Chúng ta có thể làm những thứ rất đồ sộ với array, và cách sử dụng array không phải lúc nào cũng dễ nuốt ! Trên thực tế, có thể một chuyên gia nào đó về PHP sẽ ngạc nhiên vô cùng khi thấy mình cùng nhau nói về array trong phần ***Cơ bản*** của PHP ! Tôi hướng dẫn các bạn phần này sớm như vậy cũng có lý do : một khi đã hiểu về Array, các bạn sẽ không còn gặp khó khăn gì khi mình học về Cơ Sở Dữ Liệu (phần II của bài viết) !  
Vì sự phức tạp của nó, tôi đã quyết định cắt phần này làm 2, tại đây chúng ta sẽ nói về cơ bản, phần « lợi hại » hơn chúng ta sẽ gặp lại trong phần III của bài viết !

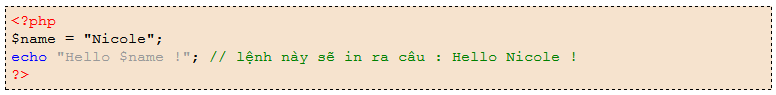
* Trong chương này:

|  |  |
| --- | --- |
| 1-9 | * + [Mảng đánh số](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong9.php#numerote)   + [Mảng kết hợp](http://thanhtra.nguyen.free.fr/web/tuto-php/chuong9.php#associatif) |

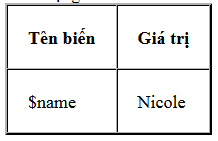
**Mảng kiểu bảng đánh số**



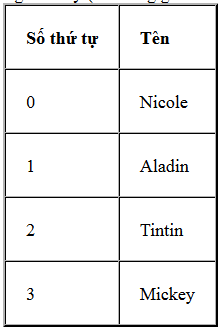
Một array là một biến, nhưng là một biến đặc biệt !  
Cho đến thời điểm này chắc các bạn cũng đã quen mặt với biến rồi ! Ví dụ một biến :  
  
**Code : PHP**



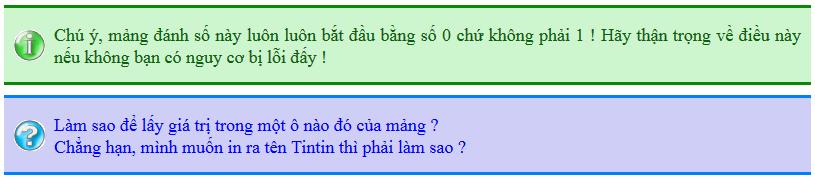
Biến $name ở đây có thể được cụ thể hóa dưới dạng như sau:



Chúng ta sẽ thấy rằng thực ra chúng ta có thể lưu cùng lúc nhiều giá trị cho một biến, chứ không chỉ lưu mỗi một giá trị Nicole duy nhất thôi đâu !  
Rất dễ tưởng tượng, ví dụ hãy xem cái bảng sau đây (là những gì mà biến $name chứa)



$name là một Array , cái mà tôi gọi là một biến kiểu bảng( hay mảng) ! Nó không chỉ chứa được 1 mà nhiều giá trị, mình có thể cho nó chứa bao nhiêu giá trị cũng được !  
Trong một mảng, giá trị được sắp xếp vào các « ô », mình tưởng tượng rằng mỗi ô được đánh một số (tạm gọi là chỉ số của ô), chẳng hạn ô chứa Nicole được đánh số 0, ô chứa Aladin được đánh số 1,... !



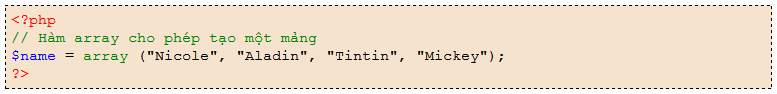
Để lấy một giá trị trong một ô nào đó của mảng, mình phải biết đó là ô số mấy (không phải thứ mấy), cấu trúc lấy giá trị trong một ô nào đó như sau: Ở đây mình muốn in ra giá trị nằm trong ô số 3 (vì Tintin nằm ở ô số 3 mà! Chú ý, tôi nói ô số 3 chứ không phải ô thứ 3 nhé!)  
  
**Code : PHP**



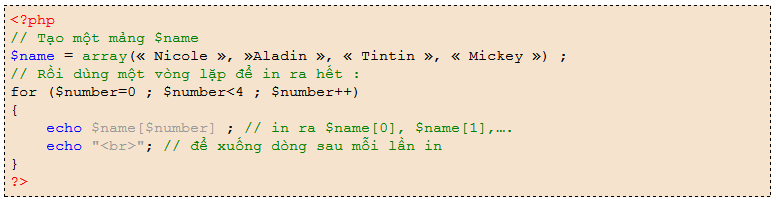
Nếu bạn không đặt số 3 trong ngoặc vuông thì nó không chạy mà sẽ báo lỗi « Array … gì đó » ! Cho nên khi làm việc với mảng, bắt buộc bạn phải chỉ ra ô số mấy và đặt số trong dấu ngoặc vuông !



Ví dụ tạo một mảng :  
  
**Code : PHP**

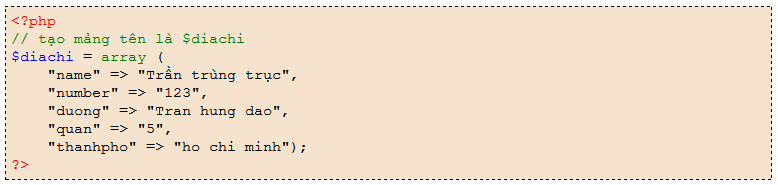


Chú ý vào thứ tự nhé! Thứ tự rất quan trọng trong việc sử dụng mảng, chẳng hạn trong ví dụ này, mình đã tạo ra một bảng gồm 4 ô, tên Nicole sẽ được tự động đặt vào ô số 0 của mảng, Aladin ô số 1,.., Mickey ô số 3!  
Vậy là các bạn đã xem hết ruột gan của tôi rồi đấy! Bạn đã biết làm sao tạo một mảng, làm sao lấy giá trị của mảng và in ra!  
Bây giờ tôi đề nghị các bạn viết một đoạn script để tóm tắt lại! Đoạn script phải cho phép in ra tất cả giá trị nằm trong mảng $name!  
Đầu tiên hãy tạo một biến kiểu mảng tên là $name, sau đó dùng vòng lặp (while hoặc for) để in tất cả giá trị trong mảng này ra! Ở đây tôi dùng for là thích hợp hơn cả!   
**Code : PHP**

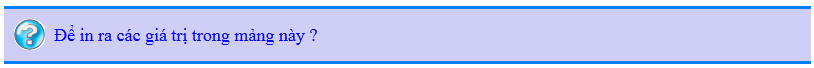


## Mảng kiểu kết hợp

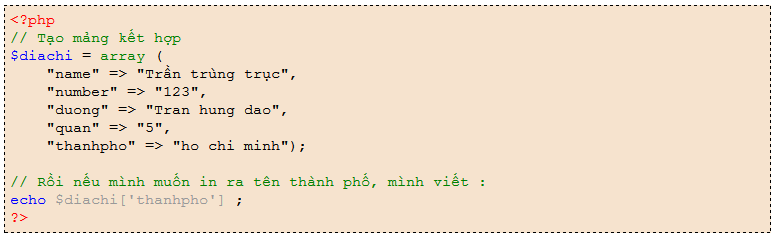
Nếu bây giờ tôi muốn lưu địa chỉ của ai đó, chẳng hạn :  
Tên : Trần Trùng Trục  
Số nhà :123  
Đường : Trần Hưng Đạo  
Quận : 5  
Thành phố : Hồ chí Minh  
Nếu mình vẫn dùng mảng theo kiểu bảng đánh số 0,1,2… như trên thì làm sao để biết 0 là tên, 1 là số nhà, 2 là đường,… ???  
Như vậy, thay vì ghi $diachi[0] = « Trần Trùng Trục », mình làm cách nào đó để ghi $diachi[tên] = « Trần Trùng Trục » chẳng hạn, như vậy sẽ dễ nhận biết hơn nhiều nhỉ ?  
Để làm như vậy, mình sẽ khai báo một mảng gọi là mảng kết hợp (associative table), dịch không sát lắm, nhưng tạm gọi vậy đi nhá ! Chúng ta vẫn dùng hàm array như cũ, nhưng bên trong có chút khác khác :  
  
**Code : PHP**



Chú ý : đây chỉ là một dòng lệnh khai báo (kết thúc bởi một dấu chấm phẩy duy nhất), mình có thể viết trên 1 dòng duy nhất, nhưng không có ai cấm mình viết thành nhiều dòng như trên, như vậy dễ nhìn hơn !  
Để ý rằng ở đây chúng ta đã dùng một mũi tên,(=>) ý nói "được kết hợp với", chẳng hạn viết "thanhpho"=> "ho chi minh" ý nói "thanhpho" thì được kết hợp với "ho chi minh" !



Gần như giống với cách làm hồi nãy, nhưng lần này mình đặt "chữ" vào bên trong dấu ngoặc đơn thay vì đặt "số" vào trong đó như đã làm ! Và chú ý rằng « chữ » ở đây phải được đặt trong dấu nháy ! ('') ! Cái này là bắt buộc ! Chẳng hạn $diachi['thanhpho'] để trích tên thành phố ra !  
  
**Code : PHP**



Mảng kiểu kết hợp này rất quan trọng trong phần II khi mình nói về Cơ Sở Dữ Liệu, vì vậy bạn phải nhớ kĩ nhé !

## Phần II: Cơ sở dữ liệu

**Giới thiệu về MySQL**

Cuối cùng cũng tới được phần II rồi ! Thở một cái nào ! Giờ thì các bạn không còn là một beginner về PHP nữa, bạn đã có thể viết một trang web bằng PHP rồi đấy ! Nhưng đấy chỉ là về cơ bản, chưa có gì hấp dẫn cả đâu ! Bạn đã học những chức năng rời rạc của PHP, vấn đề là trong thực tế mình dùng những chức năng ấy như thế nào, dùng để làm gì cho trang web, làm sao kết hợp các chức năng đó lại để phục vụ cho mục đích của mình ! Bạn đã từng suy nghĩ làm sao để tạo một forum hay đơn giản chỉ là một cái blog cho riêng mình chưa nhỉ ? Tất nhiên ý tôi muốn nói là bạn tự viết, chứ không phải dùng các công cụ có sẵn để làm forum hay dùng các dịch vụ blog miễn phí như Yahoo 360 hay WorldPress, Opera,…  
Thế thì đừng chần chừ nữa, hãy bắt tay vào khám phá MySQL, bạn đồng hành của PHP ! Hai người bạn này mà hợp sức lại thì sẽ làm nên nghiệp lớn đấy !

* Trong chương này:

|  |  |
| --- | --- |
| 2-1 | [Ơ, mà MySQL là gì ta?](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql1.php#dinhnghia)  [Cấu trúc của một cơ sở dữ liệu](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql1.php#structure)  [Ê, tui có câu hỏi !](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql1.php#question) |

**MySQL là gì ?**

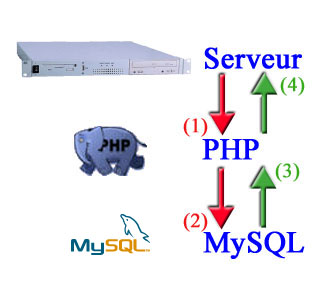
MySQL là một chương trình dùng để quản lý hệ thống cơ sở dữ liệu (CSDL), CSDL là một hệ thống lưu trữ thông tin ! Giống như một file text chứ gì ? Không, hoàn toàn không ! Điều quan trọng ở đây là CSDL là một hệ thống được sắp xếp, phân lớp rõ ràng ngăn nắp những thông tin mà mình lưu trữ.

Nếu tớ không thích sắp xếp gọn gàng ngăn nắp thì sao ? Tớ thích cứ bừa ra hỗn độn thế đấy ! Có ai kề dao vào cổ bắt tớ phải sắp xếp thông tin lưu trữ cho có trật tự lại không ?

Không ! Chả ai ép bạn sắp xếp lại cả, nhưng cuối cùng rồi chính bạn là người sẽ tự kề dao vào cổ mình thôi ! Hãy thử tưởng tượng nhé : khi bạn bước vào một căn phòng, bạn chứng kiến một cảnh tượng hỗn độn chưa từng thấy trên đời : nồi niêu xoong chảo bừa cả trên giường, chăn màn chiếu gối dồn cục ở xó bếp, bát đĩa đầy dầu mỡ ngỗn ngang trong bồn chưa được rửa, tủ chứa đầy sách bị đổ cả ra ngoài không còn lối đi, chiếc máy tính được tìm thấy dưới một lớp vỏ chai bia, nước ngọt, trái cây, vỏ kẹo,…thì bạn có thấy dễ chịu không ? Tôi thà không thèm trỡ về còn hơn là chui vô đó để ngủ ! Chẳng hạn trong hình này, đố bạn tìm được hộp cơm của tôi đấy, đố bạn biết tôi ngồi ở đâu để đánh máy vi tính !



Sự ngăn nắp thì có lợi như thế nào ? Tưởng tượng một cái tủ hồ sơ nhé ! Tất cả hồ sơ được cất trong đó, ngăn trên cùng chứa hồ sơ khách hàng, ngăn dưới chứa hồ sơ nhân sự trong công ty, ngăn dưới nữa chứa các catalogues sản phẩm của công ty. Và trong mỗi ngăn hồ sơ lại được sắp xếp theo một thứ tự nào đó, chẳng hạn khách hàng thì được sắp xếp theo mẫu tự ABC, nhân sự thì được sắp xếp theo cấp bậc trong công ty, và mỗi cấp bậc lại được sắp xếp theo mẫu tự ABC nữa, tên sản phẩm cũng được sắp xếp theo vần ABC hoặc theo lĩnh vực sử dụng (giải trí, ăn uống, du lịch, thời trang, thương mại,…). Như vậy, chuyện tìm ra một hồ sơ để xử lý không còn là vấn đề khó khăn nữa đúng không ?  
Thông tin cũng vậy, nếu bạn sắp xếp tốt thông tin mà bạn sưu tập và lưu trữ, thì sau đó bạn sẽ dễ dàng tìm ra thông tin cần thiết để xử lý !   
  
**PHP làm việc cùng với MySQL**  
  
Nói nhiều vậy đâu có ích gì phải không ? Nói thiệt đọc xong mấy đoạn trên tôi cũng chưa tưởng tượng ra rốt cụt thì MySQL là cái gì !  
Cho tới bây giờ mình chỉ biết mỗi PHP, dùng để ra lệnh cho server làm gì đó ! Và mình ra lệnh bằng tiếng PHP (ngôn ngữ PHP) ! Với MySQL, mình sẽ ra lệnh cho nó bằng « tiếng SQL » ! Đừng hoảng, tạm thời bạn hãy nhớ điều này, rồi tôi sẽ giải thích ngôn ngữ SQL là gì.  
Vấn đề là không hiểu sao mình không có cách nào để ra lệnh trực tiếp cho MySQL, có lẽ thằng MySQL không hiểu tiếng của mình nên nó nhờ thằng PHP làm thông dịch giúp ! Và vì vậy mà mình mới phải nhờ PHP làm trung gian để mình ra lệnh cho MySQL. Ví dụ mình bảo PHP như thế này : "mày đi mà nói với thằng MySQL rằng tao muốn lấy tài liệu số 4 hàng thứ 2" ! Đại khái như vậy.  
Sơ đồ hóa quá trình làm việc của PHP và MySQL như thế này :



Ở đây mình không thấy client đâu cả, mình chỉ quan tâm đến những gì server làm khi phát sinh một trang web từ php mà thôi.  
Đây là những gì diễn ra khi client yêu cầu server cho phép post một tin nhắn mới lên forum :

1. Server sử dụng PHP để thực hiện việc chuyển tải tin nhắn
2. PHP thực hiện việc cần làm của nó, rồi báo cáo là nó cần đến MySQL để lưu trữ tin nhắn đó, thế là PHP gọi « alo, anh MySQL cho em gửi nhờ tin nhắn này trong cơ sở dữ liệu nhé, để sau này cần thì em còn lấy lại »
3. MySQL nhận lời gửi thác của PHP, làm chút việc của nó, rồi trả lời lại PHP rằng « mọi việc OK, chú có thể lấy dữ liệu lúc nào cũng được » !
4. PHP quay lại nói với Server rằng MySQL đã chấp nhận yêu cầu, lệnh đã được gửi ! Server có thể in nội dung tin nhắn ra được rồi !

Đại khái quá trình làm việc song hành của PHP và MySQL là như vậy!   
Bây giờ chúng ta cùng ngâm cứu xem CSDL được sắp xếp tổ chức như thế nào nhé!

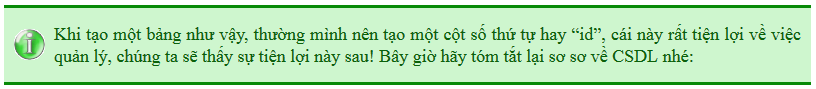
**Cấu trúc của một cơ sở dữ liệu**

Chú ý chú ý: cẩn thận ở phần này, những gì sau đây là vô cùng quan trọng để lãnh ngộ được ruột gan của CSDL!  
Tôi chưa yêu cầu bạn phải nhớ nằm lòng từ ngữ nào cả phải không? Vậy lần đầu tiên tôi xin các bạn hãy nhớ các thuật ngữ sau đây, bởi vì với CSDL mình cần phải dùng từ chính xác! Để dễ nhớ mình có thể tưởng tượng một cái tủ chứa đồ nhé, và những từ in đậm sau đây là những thuật ngữ mà bạn buộc phải nhớ và dùng chính xác :

1. CSDL, **data base** (gọi tắt là **base**)là một cái tủ mà mấy bà thư kí hay chứa thông tin tài liệu trong đó
2. Trong tủ có rất nhiều ngăn kéo, một ngăn kéo theo ngôn ngữ MySQL gọi là một **table**. Mỗi ngăn kéo chứa nhiều loại dữ liệu khác nhau, ví dụ một ngăn kéo chứa nick name, thông tin, bài viết của các thành viên trong forum chẳng hạn!   
   Một **table** cũng tương tự, nó chứa các loại dữ liệu khác nhau, mình có thể tưởng tượng nó như một cái bảng, trong đó các cột gọi là các **trường (fields)** còn các hàng gọi là **row** hay **record**. Ví dụ một **table** tên là visitors như sau:



Đây là bảng chứa dữ liệu của một table (hay một ngăn kéo)! Bảng gồm 4 trường (4 fields): **stt, nickname, email và age**! 4 record (4 hàng), thực tế một bảng có thể chứa hàng trăm, hàng nghìn record (row) như vậy! Mỗi record chứa thông tin của một người chẳng hạn!



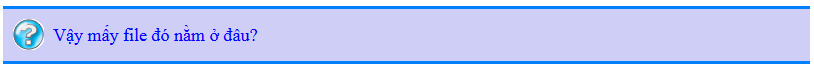
Một CSDL chứa nhiều table, ( nhiều bảng), mỗi table chứa dữ liệu được xếp vào các hàng (row, record) và các cột (fields)!   
Ví dụ một forum có thể có các table sau đây:

* Một table chứa tin tức mới (news)
* Một table chứa bài viết cho sổ lưu bút
* Một table chứa thông tin thành viên
* Một table chứa nội dung bài viết
* Một table news letters chứa email các thành viên đăng kí vào news letters
* …

**Help! Tớ có câu hỏi!**



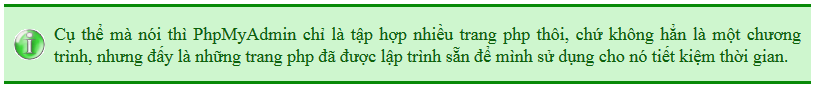
Lĩnh vực này hơi trừu tượng một chút, cái bảng mà tôi đã trình bày ở trên chỉ là một cách tưởng tượng để mình “nhìn” dữ liệu bên trong. Thực ra MySQL lưu dữ liệu ở một nơi nào đó, trong một file!



Ở trong một thư mục mà MySQL được cài đặt, trong bài viết này mình đã cài wamp, trong đó nó cài luôn Mysql, mặc định thì những file đó nằm ở : C:\wamp\mysql\data  
Nhưng mà tớ bảo này, mình chả cần đụng đến mấy file đó đâu, mọi thao tác lưu, đọc, truy xuất, sắp xếp này nọ mình đều dùng ngôn ngữ SQL để cho MySQL làm gì thì làm, mình không trực tiếp mò đến nó!

# Chương 2: PhpMyAdmin

Bây giờ chúng ta sẽ bắt đầu thao tác trên một CSDL, các bạn sẽ thấy một CSDL có thể chứa được những gì !  
Để làm việc, chúng ta cần đến một chương trình gọi là PhpMyAdmin, cái này được cài đặt cùng với WAMP, bây giờ mình có thể đem ra xài được rồi đấy !  
Nếu bạn có một cái host để chứa trang web của bạn trên mạng, bạn nên liên hệ với admin để biết địa chỉ đăng nhập và sử dụng PhpMyAdmin (nếu host đó có hỗ trợ), ví dụ mình dùng host ở free.fr thì địa chỉ đăng nhập vào CSDL của mình là [http://sql.free. fr](http://sql.free.fr) , mình cần phải có một username và password để đăng nhập vào !



Việc đầu tiên mình đề nghị các bạn là hãy mở chương trình (xin lỗi, mình gọi nó là chương trình quen rồi) PhpMyAdmin lên ! Sau khi đã chạy WAMP, các bạn click chuột trái vào icon WAMP bên dưới (phần bên phải của task bar, chỗ cạnh cái đồng hồ máy tính í), rồi chọn PhpMyAdmin (hàng thứ 2 trong menu)

* Trong chương này:

2-2

* + [Tạo một bảng](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql2.php#taobang)
  + [Sửa đổi bảng](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql2.php#modifier)
  + [Các thao tác khác](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql2.php#autres)

[precedent chapter](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql1.php)[Mục Lục](http://thanhtra.nguyen.free.fr/web/tuto-php/modau.php)[next chapter](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql3.php)

**Tạo một bảng**

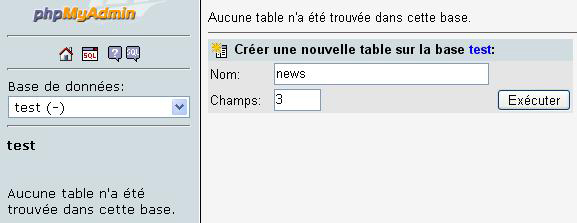
Khi bạn đã nhấp vào Phpmyadmin, trang chủ PhpMyAdmin hiện ra trông giống như thế này :



Bạn có 2 chỗ quan trọng cần lưu ý :

* (1) List các CSDL : phần bên trái, trong menu xổ xuống có chứa tên các CSDL, những con số trong ngoặc đơn là số bảng (table) trong CSDL đó.
* (2) Creat a data base (Créer une base de données) : tạo một CSDL ! Bạn gõ vào tên của CSDL cần tạo rồi nhấp vào Creat (Créer)

Tạm thời đã có vài CSDL được tạo sẵn như MySQL,Test,.. ; nhưng xin đừng đụng vào các CSDL đó (trừ Test) vì chúng là những thành phần quan trọng để PhpMyAdmin hoạt động. Mình chỉ mổ xẻ CSDL Test mà thôi. Hãy mở CSDL này ra bằng cách click vào Test trong menu xổ xuống !  
Sau khi mở CSDL Test ra, bên trái bạn sẽ thấy rằng Test có một table đã được tạo sẵn ! Mình hãy tự tạo thêm một table mới nhé. Bạn hãy nhìn xuống phía dưới của phần bên phải, trong khung Creat a new table (Créer une nouvelle table), gõ tên table vào, rồi gõ số cột (hay số fields) vào trong ô bên phải.  
Chẳng hạn mình muốn tạo một table news gồm 3 cột, mình làm như sau :



Gõ xong tên table và số cột, mình nhấp vào Exécuter (execute hay run gì đó trong tiếng Anh) !  
Cái table chưa được tạo ngay lập tức mà mình còn phải xác định tên của từng cột và kiểu dữ liệu mà mỗi cột chứa nữa ! Vì để test nên mình sẽ làm 3 cột đơn giản thế này :

* Cột đầu tiên tên là « id », người ta luôn luôn để cột đầu tiên là id (giống như số thứ tự) để làm « số đặc trưng » cho dữ liệu, như vậy mỗi hàng đều sẽ được đánh số ! Chẳng hạn news số 1, news số 2,…
* Cột hai tên là « titre », chứa tựa (tiêu đề) của news
* Cột cuối cùng mình đặt tên là « contenu » chứa nội dung của news

Lưu ý là mình chưa phải hướng dẫn các bạn làm một CSDL cho bản tin news thật đâu nhé ! Cái đó mình sẽ hướng dẫn sau, bây giờ mình chỉ làm chơi thôi, để làm quen với các thao tác trên CSDL đã !   
Sau khi điền vào rồi, cái bảng của các bạn phải trông giống thế này :



Mình điền vào bên trái tên của cột, ở giữa là kiểu dữ liệu mà cột đó sẽ chứa, bên phải là kích thước tối đa của dữ liệu.



Mỗi cột có thể chứa text, số, ngày tháng,… kiểu text, kiểu số hay kiểu ngày tháng gọi là kiểu dữ liệu ! Sau đây là vài kiểu dữ liệu thường dùng mà bạn cần phải biết :

* INT : kiểu số nguyên, bên cạnh INT còn có vài kiểu nữa tùy theo kích thước tối đa của số nguyên, ví dụ TINYINT thì số tối đa có thể chứa là 255, SMALLINT chứa được đến số 30 000, MEDIUMINT (8 000 000), INT (2 000 000 000), BIGINT (cực kì lớn)
* TEXT : kiểu text
* DATE : kiểu ngày tháng dưới dạng YYYY-MM-DD hoặc YY-MM-DD hoặc YYMMDD (ngày tháng kiểu Mỹ)
* TIME : kiểu giờ, dưới dạng HH :MM :SS hoặc HHMMSS hoặc HHMM hoặc HH
* DATETIME : hỗn hợp ngày tháng với giờ, dưới dạng YYYY-MM\_DD HH :MM :SS
* BLOB : kiểu này hơi đặc biệt, rất hiếm khi dùng đến, nó cho phép lưu trữ file vào CSDL, mình sẽ không nói về thằng này, nhưng bạn cũng cần phải biết sự tồn tại của nó trên đời chứ nhỉ !

Vẫn còn vài option chưa chọn ở phần bên phải của màn hình cho mỗi cột :



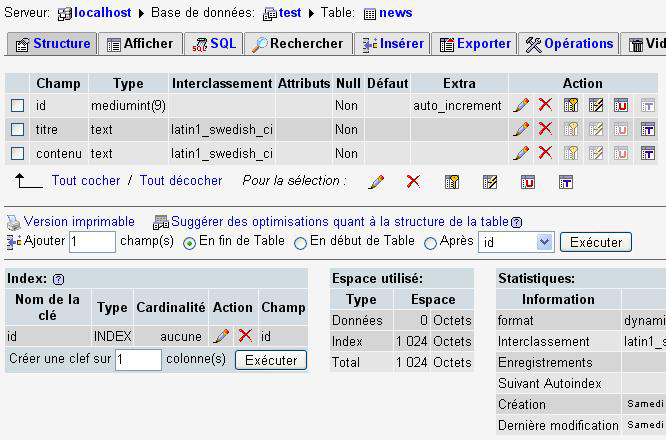
Tôi đã để chế độ **auto\_increment** ở cột **Extra** cho **id**, tôi khuyên bạn nên để như vậy cho tất cả các trường id, nó cho phép tăng tự động số thứ tự của hàng khi có một hàng mới được nhập vào.  
Tôi cũng đã đặt « index » cho trường id (hình tia chớp í), bạn cũng nên để như vậy cho mọi trường id, nó giúp gia tốc cho việc tìm kiếm trong CSDL sau này !  
Các trường khác mình để mặc định !  
Xong, nhấp vào Save (Sauvegarder), thế là một table đã được tạo.

**Chỉnh sửa một table**

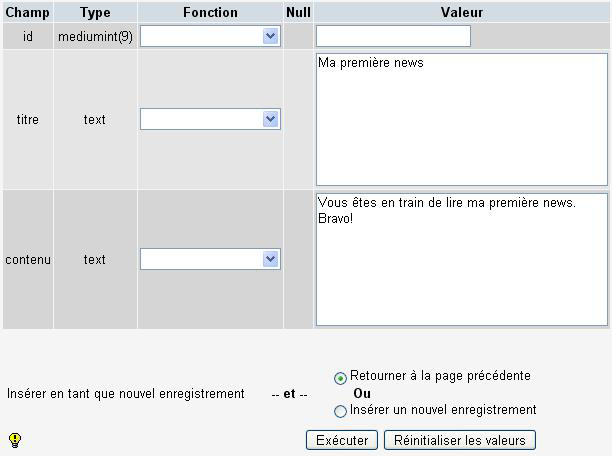
Bây giờ bạn có thể thấy ở bên trái, một bảng mới đã được tạo phía dưới tên CSDL Test !

  
*Lưu ý rằng trong trường hợp của bạn, con số trong dấu ngoặc đơn bên cạnh chữ Test sẽ là 2, vì bạn vừa tạo một bảng, nó có sẵn một bảng! Cái mình dùng khác với của bạn í mà!*

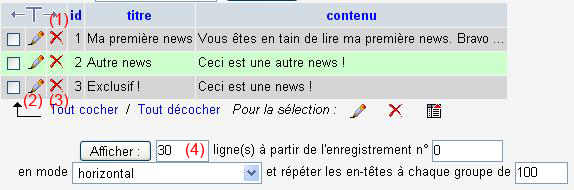
Nếu bạn nhấp vào chữ news, bên phải sẽ hiện ra cấu trúc của bảng  
Nếu bạn nhấp vào cái hình bênh cạnh (bên trái) chữ news, bên phải sẽ hiện ra nội dung của bảng !  
Tạm thời cái bảng của mình đang trống rỗng, mình chưa có đưa dữ liệu gì vào cả mà, cho nên nếu bạn cho hiện ra cấu trúc của bảng, bạn phải thấy nó giống thế này :



Chả có gì để động vào cả, nếu có chăng thì cũng chỉ là mấy cái lớp khó hiểu Exporter (export) afficher,SQL,… mà mình sẽ tìm hiểu sau ! Bây giờ hãy điền dữ liệu vào đã chứ, để nhập dữ liệu vào, bạn hãy nhấp nút **Inssérer (insert)**, rồi làm theo tôi nhé :



Mình không cần đánh số vào trường id vì như tôi đã nhắc rằng nó sẽ tự động tăng số thứ tự mỗi lần mình nhập vào một hàng mới, bạn chỉ cần nhập vào tiêu đề và nội dung của news là được, sau đó nhấp Exécuter (execute hay run).  
Làm 2, 3 lần như thế để nhập thêm news, và cũng không cần phải nhập số thứ tự ở cột id .  
Bây giờ để xem nội dung của bảng, mình nhấp vào **Afficher** (**show** hay **display** gì đó trong tiếng Anh), hoặc nhấp vào hình cái bảng bên trái chữ news như tôi có đề cập phía trên đấy !  
Bảng của bạn sẽ hiện ra giống như vầy :



Giải thích các mục được đánh số màu đỏ trên hình :

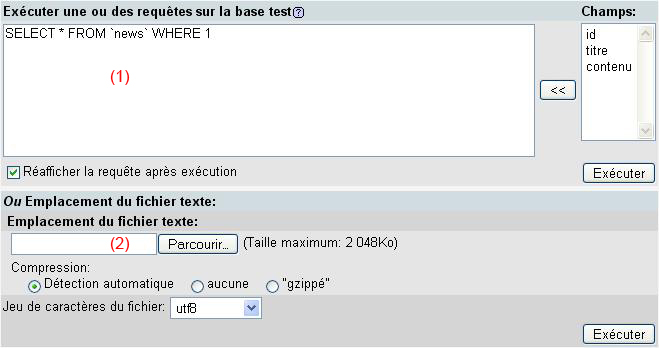
1. Hiển thị toàn bộ nội dung text : như bạn thấy, nếu text mình nhập vào quá dài, thì MySQL sẽ tự động cắt ra, chỉ thể hiện một phần của text thôi. Để xem toàn bộ nội dung text bạn hãy nhấp vào cái chữ T in hoa ấy !
2. Hình cái bút chì : nhấp vào đây để sửa đổi nội dung nhập vào
3. Xóa nội dung của hàng
4. Hiển thị nội dung của n hàng kể từ hàng thứ x ! Nếu bảng của mình có cả trăm hàng, thì MySQL chỉ cho hiện ra một khoảng 30 hàng thôi, mình muốn xem bao nhiêu hàng kể từ hàng thứ mấy thì mình điền vào ô số 4 đấy !

**Các thao tác khác**

Mình chia phần này làm 5 phần nhỏ để nói về cái menu phía trên cái bảng, theo thứ tự như sau :

* SQL
* Exporter (Export)
* Opération (operation)
* Vider (Empty)
* Supprimer (Delete)

**SQL**  
Click vào lớp SQL : http://thanhtra.nguyen.free.fr/web/img/mysql/sql.jpg  
  
Màn hình sẽ hiện ra :



Tại chỗ này mình sẽ viết các trình đơn để yêu cầu MySQL làm gì đó ! Bạn có 2 cách để chạy một trình đơn :

1. (1) Trong vùng text, bạn gõ vào nội dung trình đơn (bằng ngôn ngữ SQL), ở đây mặc định đã có một dòng SELECT \* FROM ‘news’ WHERE 1  
   Câu này có nghĩa là : hiển thị tất cả nội dung của bảng « news »
2. (2) Bên dưới, bạn có một nút Parcourrir (Browse) để tìm file đã viết sẵn trình đơn, mình chỉ cần đi tìm file đó trong ổ cứng để nạp vào đây, MySQL sẽ chạy theo yêu cầu của trình đơn đó.

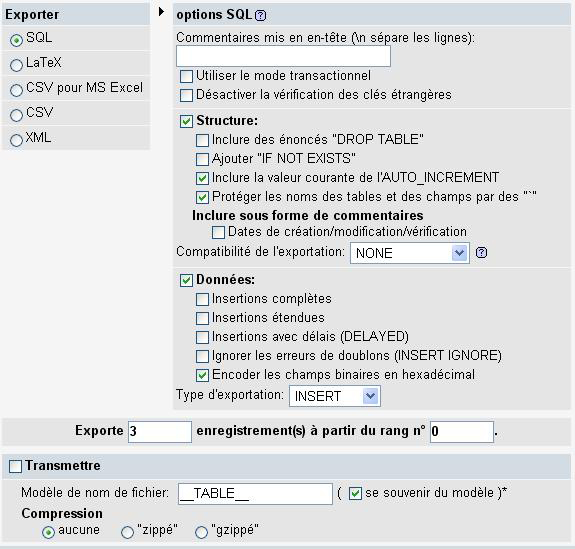
Nhấp Exécuter (execute) như thường lệ để chạy!



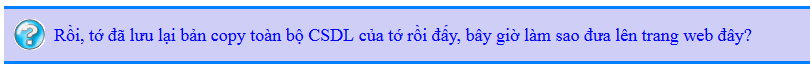
**Exporter**  
  
Mình còn 4 công cụ nữa : http://thanhtra.nguyen.free.fr/web/img/mysql/exporter.jpg  
  
Tiếng Anh : **Export, operation, empty, delete**  
Mình quan tâm tới công cụ **Exporter (export)**, nó cho phép lưu dữ liệu vào ổ cứng dưới dạng một file text chứa đầy các trình đơn trong đó!



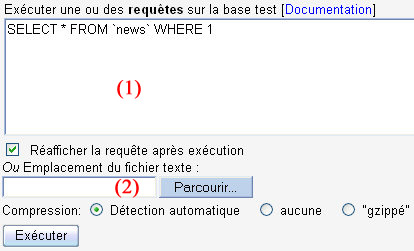
Để truyền lên mạng internet nếu bạn có một cái host nào đấy trên mạng, vì nãy giờ mình chỉ làm việc với CSDL trên máy tính của mình thôi, nếu truyền được file này lên mạng thì bạn sẽ có một CSDL y chang cái trong máy và trang web của mình có thể sử dụng được!  
Tạo một bản copy để phục hồi CSDL trong trường hợp trang web của bạn bị Hacker viếng thăm và xóa sạch toàn bộ CSDL!  
Màn hình bên phải trông như thế này:

  
*(tùy phiên bản mà thứ tự có thể thay đổi)*

Tôi đề nghị các bạn nên để các lựa chọn như mặc định, như vậy cũng đủ xài rồi!  
Phân biệt kĩ: **Structure** chỉ là cấu trúc của bảng, tức là chỉ có tên của các cột và kiểu dữ liệu mà cột chứa! Còn **Données (Data)** mới là cái quan trọng nhất (nội dung của bảng! Vì vậy để tạo một bản copy hoàn chỉnh CSDL, mình cần phải chọn cả **Structure và Data.**  
Nhấp vào exxécuter (execute), bạn sẽ bị yêu cầu download về một file, đừng ngại gì mà không down, bình thường mà, và cũng nên ngó một chút xem file chứa gì trong đó nhé! Bạn sẽ ngạc nhiên khi thấy file đó chứa đầy các trình đơn viết bằng ngôn ngữ SQL, cái mà tôi sắp hướng dẫn các bạn đấy!



Bạn phải vào địa chỉ của trang PhpMyAdmin (liên hệ admin để biết địa chỉ trang này), ví dụ nếu bạn xài host của Free.fr thì địa chỉ là : <http://phpmyadmin.free.fr/PhpMyAdmin>  
Nó sẽ yêu cầu nhập username và password để đăng nhập vào tài khoản của mình.  
Khi đã vào đến nơi, bạn nhấp vào lớp SQL nó sẽ ra như thế này:

  
*(trông quen nhỉ, đã thấy hồi nãy rồi thì phải)*

Đấy, bây giờ chính là lúc mình dùng đến công cụ thứ 2 để nạp file copy CSDL trong máy mình đưa lên web đấy!  
  
**Operation**  
  
Hãy xem khung **option of this table**, mình có các chức năng sau:

* Đổi tên bảng: (change the name of this table)
* Phía trên phần đổi tên là phần di chuyển table sang CSDL khác (déplacer la table vers..; remove this table to…)
* Copy : để sao lưu table này trong một csdl khác

**Empty (vider)**  
  
Xóa toàn bộ nội dung dữ liệu đã nhập, chỉ còn trơ lại cái sườn với tên của các cột thôi.  
  
**Delete (Supprimer)**  
  
Xóa toàn bộ cả nội dung lẫn cái sườn, nghĩa là cả cái bảng sẽ biến mất vĩnh viễn khi mình nhấp lên nút này ! Cho nên phải hết sức cẩn thận, suy nghĩ 2 lần trước khi quyết định ra tay hủy diệt nó nhé, bởi mình không có cách nào lấy lại được những gì đã mất nữa rồi !

**Chương 3 : Đọc dữ liệu**

Chương này khá dài nhưng rất thú vị! Hơn nữa tầm quan trọng của CSDL là nằm ở chương này và chương sau, cho nên nhất thiết các bạn phải đọc kĩ!http://thanhtra.nguyen.free.fr/web/img/emotion/leluoi.gif

* Trong chương này:

|  |  |
| --- | --- |
| 2-3 | [Kết nối vào CSDL](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql3.php#connect)  [Lấy dữ liệu](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql3.php#recuperer)  [Tiêu chuẩn lựa chọn](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql3.php#critere)  [Đếm số hàng trong bảng dữ liệu](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql3.php#compter) |

**Kết nối vào cơ sở dữ liệu**

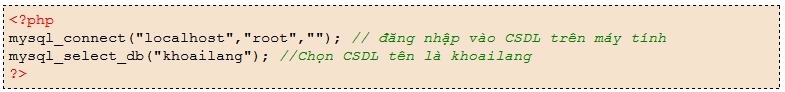
Để có thể làm việc trên CSDL, việc đầu tiên là phải kết nối (connect) vào CSDL.  
Các bạn còn nhớ không, tôi đã có nhắc rằng để ra lệnh cho MySQL mình không thể nói trực tiếp với nó mà phải nhờ PHP làm "thông dịch viên" trung gian. Vấn đề là trước hết PHP không thể bảo ngay "MySQL ơi, làm ơn cho tớ lấy link phim Tây Du Kí nhé !" được. Vì đầu tiên MySQL đòi user name và password, phải có user name và password mới vào được CSDL của mình chứ, nếu không thì ai cũng có thể vào được rồi xóa hết hoặc thay đổi hết thì sao.http://thanhtra.nguyen.free.fr/web/img/emotion/eooi.gif  
Để theo dõi chi tiết việc kết nối và thao tác trên CSDL, mình sẽ học lần lượt các cách để :

* Kết nối vào CSDL (lấy chìa khóa mở cửa vào nhà)
* Chọn CSDL cần làm việc (vào nhà và chọn nơi làm việc)
* Ngắt kết nối với CSDL sau khi đã làm việc xong (khóa cửa lại, ra khỏi nhà)

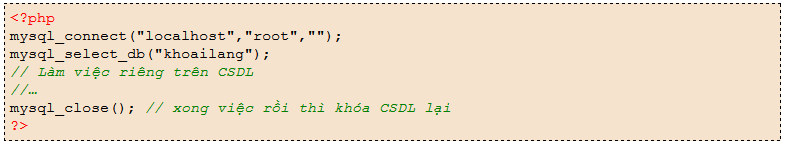
**Kết nối (connection)**  
  
Để connect vào CSDL chúng ta dùng một hàm PHP: *mysql\_connect*  
Hàm này cần có 3 thông số:

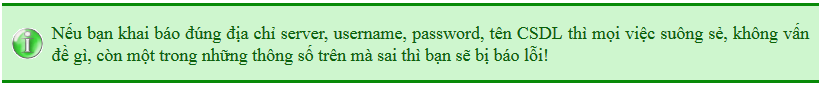
* Tên (địa chỉ) của máy chủ chứa CSDL (server), nếu bạn test trên máy tính của bạn thì địa chỉ này là "localhost"
* User name: liên hệ admin để biết username đăng nhập vào CSDL, nếu bạn có một cái host miễn phí, thì thông thường username này trùng với username bạn đăng nhập vào FTP, nếu bạn test trên máy của bạn thì user name là "root"
* Password: 99% là giống với password bạn dùng để kết nối FTP. Test trên máy thì để "" (không có password)

Ví dụ test trên máy với WAMP: mysql\_connect("localhost","root","");  
  
**Chọn CSDL làm việc**  
  
Sau khi đã đăng nhập vào CSDL rồi, bạn cần phải lựa chọn tên CSDL mà bạn cần làm việc (nếu bạn có nhiều CSDL). Đối với các host miễn phí (như FREE.FR chẳng hạn) thì nó chỉ cho mình một CSDL thôi, vậy cũng quá đủ rồi! Và tên của CSDL này thường là trùng với username đăng nhập vào MySQL của bạn (do server tạo tự động).  
Hàm PHP để lựa chọn CSDL: *mysql\_select\_db* (chữ db là viết tắt của DataBase)  
Ví dụ nếu tên của CSDL của tôi là **khoailang**, thì tôi làm như sau:  
**Code PHP:**

****

**Ngắt kết nối**   
  
Sau khi đã làm việc với CSDL xong, việc cuối cùng của bạn là phải ngắt kết nối! Giống như căn nhà của bạn vậy, bạn muốn vào thì phải có chìa khóa (ở đây là username và password), rồi vào làm việc riêng, sau khi xong việc bạn muốn ra khỏi nhà hay đi đâu đó thì phải khóa cửa lại!  
Hàm để ngắt kết nối (đóng CSDL lại) : mysql\_close(); không cần thông số bên trong  
Cấu trúc tóm tắt như sau:  
**Code PHP:**

****

****

**Lấy dữ liệu**

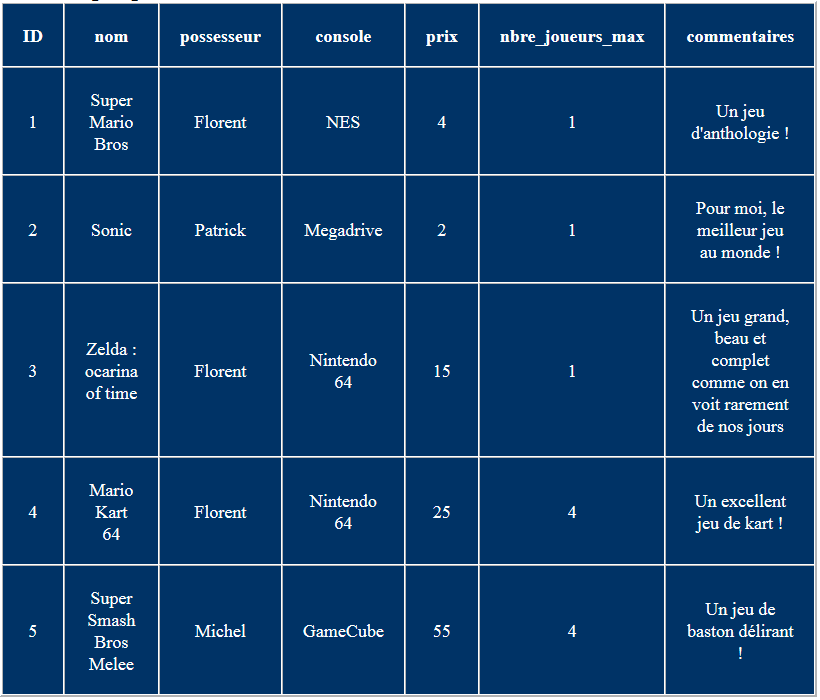
Thường thường sau khi đã có một trang web (và có CSDL) thì mình phải nhập CSDL vào, sau đó đọc CSDL và truy xuất dữ liệu mà mình cần dùng.  
Bây giờ tôi sẽ hướng dẫn các bạn làm sao để đọc dữ liệu, còn việc ghi dữ liệu tôi sẽ nói sau.  
Nhưng để đọc CSDL thì đầu tiên mình cần phải có sẵn một CSDL đầy đủ để mà đọc chứ nhỉ, không có gì cả thì lấy gì mà đọc?!  
May cho các bạn là hôm nay tôi hơi bị siêng năng, nên đã chuẩn bị sẵn cho các bạn một bảng dữ liệu về video games (jeux\_videos trong tiếng Pháp). Các bạn hãy tải bảng này về trước đã:

[Download bảng jeux\_videos (2,5 Kb)](http://www.siteduzero.com/uploads/fr/ftp/mateo21/jeux_videos.sql)

File có phần mở rộng là .sql   
Bạn định mở file ra để đọc hả ? Nói nhỏ nghe nè : chả có gì thú vị để đọc trong đó đâu, chỉ là một đám rừng toàn là text.  
Việc cần làm bây giờ là nạp cái bảng này vào PHPMyAdmin. Hãy theo trình tự sau đây nhé :

* Mở **PhpMyAdmin** ra
* Chọn CSDL **test** trong menu đổ xuống bên trái
* Chọn tab **Importer (import)** trên hàng menu lớn bên trái
* Nhìn xuống dưới, nhấp nút **parcourrir (Browse)**
* Một hộp thoại hiện ra, bạn hãy đi tìm nơi đặt file **jeux\_videos.sql** mà bạn đã tải hồi nãy, nhấp chọn nó
* Cuối cùng, nhấp **exécuter (execute)**

Xong rồi http://thanhtra.nguyen.free.fr/web/img/emotion/yeah.gif, bạn sẽ thấy bên trái, dưới Test có một bảng mới xuất hiện tên là jeux\_videos đấy!  
Nếu bạn nhấp vào hình cái bảng bên cạnh chữ jeux\_videos thì bảng dữ liệu sẽ hiện ra bên phải, sau đây là 5 dòng đầu tiên trong bảng dữ liệu:



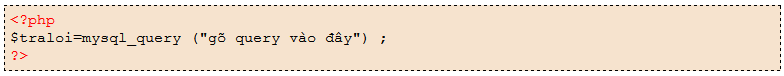
Tạm thời, hãy khoan sửa đổi gì cái bảng này nhé !



Mục đích của chúng ta bây giờ là : viết một trang php cho phép in ra tất cả dữ liệu chứa trong bảng trên !  
  
**Viết một query**   
  
Đã đến hồi quan trọng của buổi học hôm nay rồi, bây giờ mình sẽ yêu cầu MySQL làm vài thứ bằng ngôn ngữ SQL. Viết một yêu cầu gọi là thực hiện một query . Mình sẽ nhờ MySQL in ra nội dung của cái bảng trò chơi điện tử (jeux\_videos) mà hồi nãy mình có đề nghị bạn tải về đấy !  
Để viết một query chúng ta sử dụng hàm PHP : *mysql\_query*

* Hàm này chứa một thông số, đó là những gì mà PHP sắp yêu cầu MySQL làm (bằng ngôn ngữ SQL)
* Hàm này trả về một giá trị, vì vậy mình cần phải có một biến để chứa những gì mà MySQL trả lời.

Mình luôn viết như thế này :  
  
**Code PHP:**



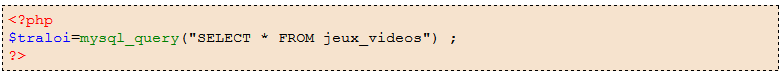
$traloi chứa những gì mà MySQL trả lời theo đơn yêu cầu của mình.  
Bây giờ chúng ta hãy xem làm sao để hỏi MySQL những gì có trong bảng jeux\_videos nhé.  
  
**Query SQL đầu tiên của bạn**   
  
Như đã nói, SQL là một ngôn ngữ cho phép mình giao tiếp với MySQL, đây là query SQL đầu tiên :  
  
**Code SQL :**

C:\Users\Chu Quang Vien\Desktop\12_7.PNG

Dòng này có thể hiểu là : *chọn tất cả các dữ liệu trong bảng jeux\_videos.*  
Mình cùng phân tích vài từ khóa trong này nhé :

* SELECT : trong ngôn ngữ SQL, việc đầu tiên là mình phải xác định kiểu (loại) thao tác mà mình sẽ làm với MySQL. Ở chương này chúng ta chỉ xem mỗi một thao tác CHỌN thôi (SELECT). SELECT cho phép hiển thị tất cả nội dung trong một bảng.
* \* : ngay sau SELECT, chúng ta phải xác định rằng chúng ta muốn lấy dữ liệu trong cột (trường, fields) nào. Dấu \* ám chỉ "lấy hết cho tao ! Không chừa lại thứ gì cả! http://thanhtra.nguyen.free.fr/web/img/emotion/haha.gif». Giả sử bạn muốn chọn cột "nom" (tên) và "possesseur" (người sở hữu) thì bạn gõ : SELECT nom, possesseur FROM jeux\_videos
* FROM là một từ nối (từ liên kết), ý nói "ở trong" cái bảng nào ! Sau FROM mình sẽ chỉ ra tên của cái bảng mà mình muốn lấy dữ liệu.
* Jeux\_videos chính là tên của cái bảng mà mình muốn lấy dữ liệu !

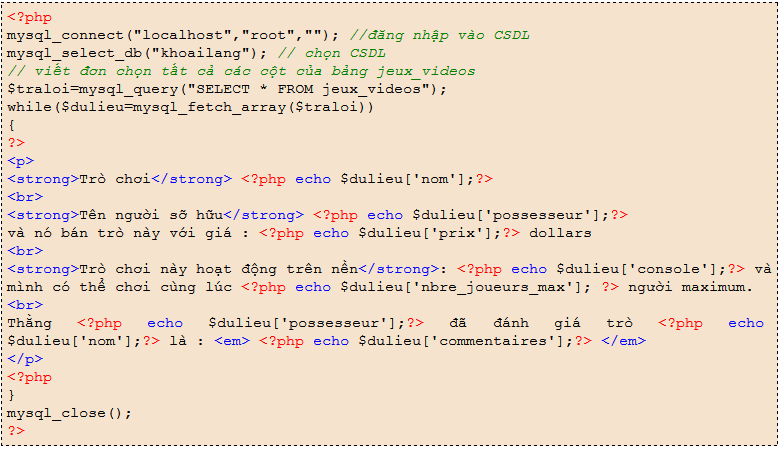
Mổ xẻ xong rồi, bây giờ mình hãy đặt lá đơn này vào hàm mysql\_query đi ! http://thanhtra.nguyen.free.fr/web/img/emotion/yeah.gif  
  
**Code PHP :**



Thế là biến $traloi của mình bây giờ đã chứa câu trả lời của MySQL rồi đấy!



**In (hiển thị) kết quả của một query**  
  
Vấn đề là cái biến $traloi của mình chứa một thứ gì đó không thể bung ra được, nghĩa là không giống như một biến bình thường chứa số hay chứa text mà mình có thể dùng lệnh echo để in ra, mà nó chứa một thứ rất hỗn độn vô trật tự.  
Bạn tưởng tượng nếu cái bảng của mình có 10 cột, 200 dòng, thì cái biến $traloi chứa 2000 thông tin! Bây giờ làm sao lấy ra 2000 thông tin đó đây? Loại biến nào lại có kiểu bảng giống như vậy nhỉ? Từ "cái bảng" giúp tôi nhớ đến Array! http://thanhtra.nguyen.free.fr/web/img/emotion/smile.gif  
PHP có sẵn một hàm giúp tạo ra một array từ $traloi: mysql\_fetch\_array($traloi)  
Đây là một mảng kiểu bảng kết hợp (kiểu thứ nhất là bảng đánh số như mình đã học ở chương Array rồi đấy), nếu bạn muốn lấy dữ liệu ở cột "console" chẳng hạn, thì bạn phải sử dụng một mảng khác: $dulieu['console'] ($dulieu là một biến array luôn đấy)  
Tiếp theo, mình cần dùng một vòng lặp để in hết các hàng ra! Mỗi lần các bạn dùng hàm mysql\_fetch\_array, bạn sẽ truy nhập vào hàng kế tiếp của bảng. Vòng lặp sẽ tự động lặp lại cho đến khi nào không còn hàng nào nữa trong bảng của mình thì thôi. Đây là điểm thuận lợi khi mình không biết bảng của mình có bao nhiêu hàng!  
Đây là những gì tôi đã làm để in ra nội dung của cái bảng:  
  
**Code PHP:**



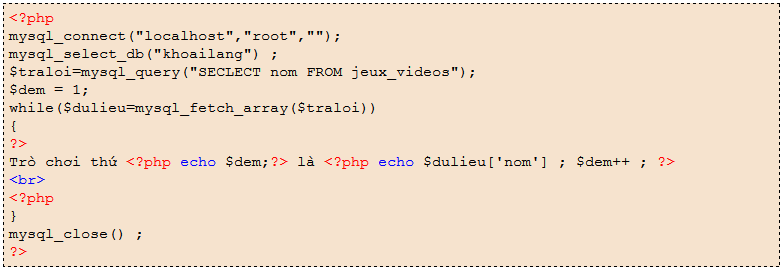
Bạn nhìn thấy gì? Nó in ra khoảng 50 kết quả, không gì lạ bởi vì cái bảng của mình chứa khoảng 50 hàng mà.



$traloi là một biến chứa câu trả lời của MySQL, là một mớ hỗn độn vô trật tự có kiểu dữ liệu là resource (hiếm gặp từ này). Nhờ vào hàm mysql\_fetch\_array mà mình tạo được mảng $dulieu ! Mảng này chứa 1 dòng trong bảng dữ liệu của mình, khi mình dùng hàm mysql\_fetch\_array một lần nữa thì mảng $dulieu sẽ chứa hàng thứ 2 trong bảng dữ liệu. Tưởng tượng thế này nhé :  
Lần đầu tiên gọi : $dulieu= mysql\_fetch\_array($traloi) ;   
Thì $dulieu có dạng một mảng như thế này : {'nom' :'Super mario Bros', 'possesseur' :'Florent','prix' :4, 'console' :'NES', 'nbre\_joueurs\_max' :1,'commentaires' :'un jeu d'anthologie'}   
Sau khi in hết dòng này, vòng lặp **while** cho phép gọi tiếp hàm mysql\_fetch\_array một lần nữa, và lần này mảng $dulieu tự động chứa hàng thứ 2 của bảng jeux\_videos và có cấu trúc như trên !

C:\Users\Chu Quang Vien\Desktop\12_12.PNG

Vậy đó! Bây giờ bạn có thể cho in ra bất cứ thứ gì bạn muốn, không ai ép bạn phải in ra cả cái bảng đâu, ví dụ bạn có thể chỉ in ra tên của trò chơi bằng cách chọn cột 'nom':  
  
**Code PHP:**



**Lỗi lầm vây kín …**http://thanhtra.nguyen.free.fr/web/img/emotion/haha.gif  
  
Khi một query SQL của bạn bị treo, thường thì PHP sẽ báo là có lỗi ở dòng chứa hàm **mysql\_fetch\_array**. Điều này thực ra không chính xác đâu, có thể bạn đã bị lỗi đâu đó phía trên kia, chứ không phải chính xác tại dòng mysql\_fetch\_array đâu. Để biết rõ chi tiết lỗi gì, ở đâu, bạn nên tập thói quen chèn thêm dòng này vào cùng dòng với mysql\_query nhé : or die(mysql\_error());  
Ví dụ với trường hợp mình đang làm :

C:\Users\Chu Quang Vien\Desktop\12_14.PNG

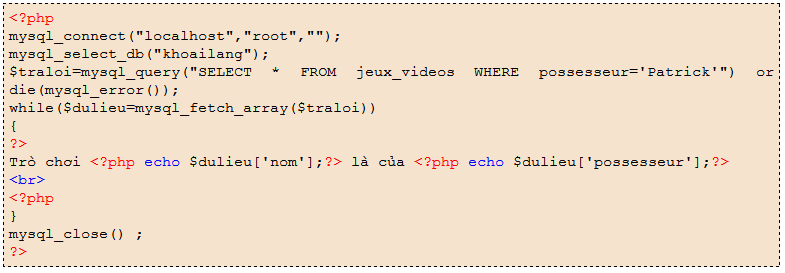
Dòng code vừa thêm vào sẽ không làm gì cả nếu bạn không có lỗi, trái lại nếu có lỗi đâu đó, nó sẽ báo cho mình biết chi tiết hơn!http://thanhtra.nguyen.free.fr/web/img/emotion/smile.gif

## Tiêu chuẩn lựa chọn

Ở đây chúng ta chỉ quan tâm tới các query query SQL thôi. Bạn sẽ thấy, bằng cách thay đổi điều kiện và tiêu chuẩn chọn lựa, mình có thể sắp xếp dữ liệu một cách gọn gàng ngăn nắp dễ dàng!  
Các từ khóa sẽ đề cập: WHERE, ORDER BY, LIMIT  
  
**WHERE**  
  
Where giúp mình lọc bớt dữ liệu theo tiêu chuẩn mà mình đặt ra. Chẳng hạn bây giờ mình chỉ muốn lấy ra các trò chơi của Patrick (trong bảng jeux\_videos), query của mình sẽ thêm WHERE possesseur='Patrick' vào sau từ khóa FROM:  
  
**Code SQL:**

C:\Users\Chu Quang Vien\Desktop\12_16.PNG

Dòng này được hiểu là : chọn tất cả các cột trong bảng jeux\_videos sao cho cột possesseur chứa Patrick.  
  
**Code PHP:**



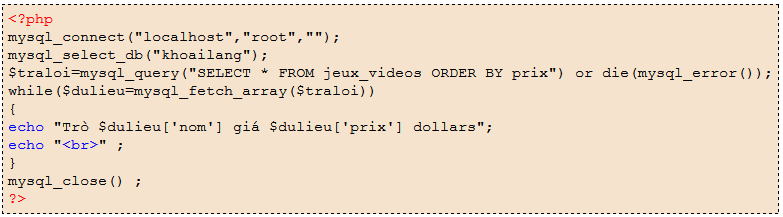
Nếu thích bạn có thể thay đổi một chút, chẳng hạn bạn muốn xem tất cả trò chơi của Michel, bạn hãy thay WHERE possesseur='Patrick' bằng WHERE possesseur='Michel' thử xem !  
Mình cũng có thể đặt điều kiện vào trong query nữa, chẳng hạn mình muốn xem những trò nào của Patrick mà nó bán với giá thấp hơn 20 dollars :  
  
**Code SQL :**

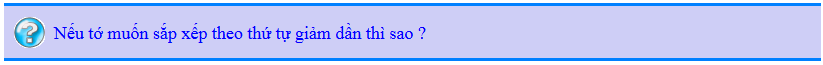
C:\Users\Chu Quang Vien\Desktop\12_18.PNG

**ORDER BY**  
  
ORDER BY cho phép chúng ta sắp xếp lại kết quả thu được cho có trật tự! Chẳng hạn mình sẽ sắp xếp lại bảng trò chơi điện tử jeux\_videos theo thứ tự tăng dần của giá cả (prix):  
  
**Code SQL:**

C:\Users\Chu Quang Vien\Desktop\12_!9.PNG

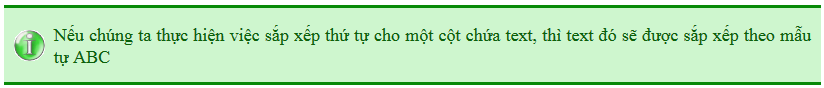
Ứng dụng :  
**Code PHP:**





Thì thêm **DESC** vào cuối dòng lệnh SQL trên :  
  
**Code SQL :**

C:\Users\Chu Quang Vien\Desktop\12_22.PNG

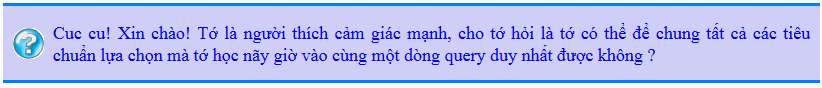


**LIMIT**  
  
LIMIT cho phép chúng ta giới hạn được kết quả xuất ra, chẳng hạn chúng ta chỉ cần lấy 20 dòng đầu tiên ra thôi thì chúng ta sẽ thêm từ LIMIT vào cuối query, kèm theo 2 con số, ngăn cách bằng dấu phẩy:  
  
**Code SQL:**

C:\Users\Chu Quang Vien\Desktop\12_24.PNG

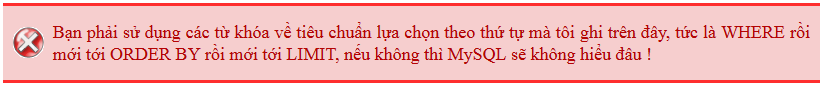


Đầu tiên, số 0 chỉ ra rằng MySQL sẽ đọc dữ liệu kể từ hàng số 0 (tức hàng thứ 1), cách đánh số của nó giống như cách đánh số trong mảng vậy!   
Số thứ 2 cho biết SỐ LƯỢNG hàng cần phải đọc! Ở đây nó sẽ đọc từ hàng số 0 đến hàng số 19 để được 20 hàng!  
Ví dụ, nếu mình đặt:  
LIMIT 0,10: in ra 10 hàng đầu tiên (từ hàng thứ 1 đến hàng thứ 10)   
LIMIT 5,10: in ra 10 hàng, từ hàng thứ 6 đến hàng thứ 15  
LIMIT 10,2: in ra 2 hàng, từ hàng thứ 11 đến hàng thứ 12



Nghe kĩ đây bạn !http://thanhtra.nguyen.free.fr/web/img/emotion/hihi.gif  
Đây là món tả pín lù mà bạn vừa đề nghị nè :  
  
**Code SQL :**

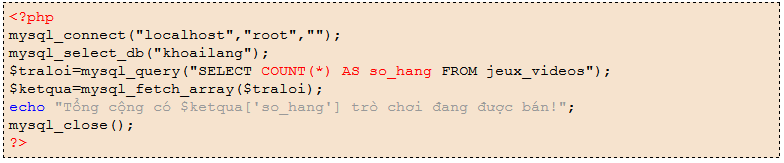
C:\Users\Chu Quang Vien\Desktop\12_28.PNG



Bạn thử dịch ngôn ngữ SQL trên đây sang tiếng Việt thử xem ! Rồi test thử dùm tôi nhé ! Tôi lười quá nên chưa test! http://thanhtra.nguyen.free.fr/web/img/emotion/leluoi.gif

## Đếm số hàng trong bảng dữ liệu

Trước khi kết thúc chương này, mình hãy thử làm một động tác mà đôi khi cũng thú vị: đếm số hàng trong bảng dữ liệu! Ví dụ bạn có thể ghi ra câu "Tạm thời có 50 trò chơi được bán" nếu như bạn đếm được rằng có 50 hàng (mỗi hàng tương ứng với một trò chơi) trong bảng jeux\_videos!  
Để đếm, mình dùng code như sau:  
  
**Code PHP:**



Từ khóa **COUNT** dùng để đếm xem có bao nhiêu hàng trong bảng jeux\_videos rồi cho ra giá trị trong mảng $ketqua['so\_hang']  
Bạn cũng có thể thêm tiêu chuẩn lựa chọn vào query trên, chẳng hạn đếm xem có bao nhiêu trò chơi của Florent chẳng hạn! Thử xem!

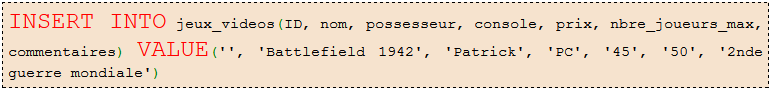
# Chương 4 : Ghi dữ liệu

* Trong chương này:

|  |  |
| --- | --- |
| 2-4 | [Thêm dữ liệu](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql4.php#add)  [Sửa đổi DL](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql4.php#modifier)  [Xóa DL](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql4.php#delete) |

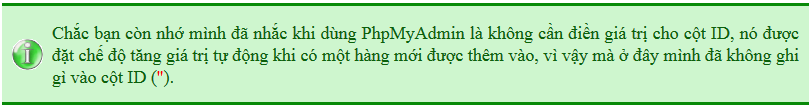
## Thêm dữ liệu

Nhiệm vụ của bạn bây giờ là (nếu bạn chấp nhận) thêm một hàng nữa vào cuối bảng jeux\_videos mà mình làm việc ở chương trước !  
Công việc này chỉ cần một cú nhấp chuột trong phpmyadmin là xong, nhưng ở đây tôi muốn các bạn thêm vào bằng mã PHP chứ không dùng PhpMyAdmin !  
Để thêm một hàng vào bảng dữ liệu jeux\_videos, chúng ta sử dụng code SQL sau đây :  
  
**Code SQL :**

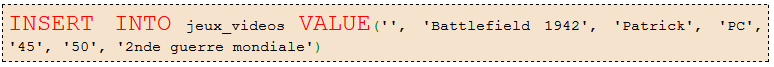


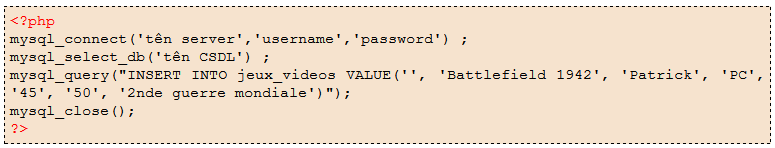
Giải thích :

* Trước hết các bạn dùng **INSERT INTO** để thông báo ‘tớ sắp chèn thêm một hàng vào đấy !’
* Tiếp theo, các bạn chỉ ra tên của cái bảng mà bạn định chèn, ở đây là bảng **jeux\_videos**, rồi đặt vào trong ngoặc đơn tên các cột của bảng đó.
* Cuối cùng, bạn viết **VALUE** rồi ghi giá trị cần chèn vào theo đúng thứ tự với các cột để khỏi nhầm lẫn.



Nếu thích các bạn có thể không cần ghi ra tên các cột ngay sau tên bảng, code vẫn chạy đúng, nhưng như vậy sẽ thiếu sự rõ ràng khi mình đọc code :  
  
**Code SQL:**

Sau đây là code PHP áp dụng :  
**Code PHP :**

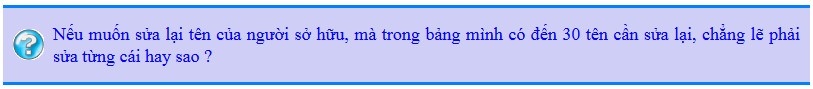


## Sửa đổi dữ liệu

Lúc nãy bạn đã thêm trò chơi BattleField vào bảng jeux\_videos, theo đó thì trò chơi này có thể chơi cùng lúc 50 người maximum và giá của nó là $45, nhưng bây giờ trò đó chỉ có thể chơi 32 người maximum cùng lúc, và giá cả cũng đã giảm còn 10 đô thôi, làm thế nào để cập nhật tin này bây giờ ? Làm sao mà sửa lại những giá trị này nhỉ ?  
No problem ! Đoạn code sau đây giúp bạn sửa lại dễ dàng :  
**Code SQL :**

C:\Users\Chu Quang Vien\Desktop\13_5.PNG

Trước hết, từ khóa **UPDATE** cho biết mình sắp sửa đổi một dữ liệu  
Tiếp theo : tên của bảng cần sửa  
Từ khóa **SET** ngay sau đó dùng để thay đổi giá trị của dữ liệu cần sửa đổi  
Dùng phép gán để thay đổi giá trị  
**WHERE** để chỉ ra ID của hàng chứa dữ liệu cần sửa (ở đây trò Battlefield ở hàng có ID là 51)  
Nếu bạn không biết ID của nó là bao nhiêu, bạn có thể vào PhpMyAdmin để xem !  
Nếu không mình cũng có thể chỉ ra WHERE nom='Battlefield192' cũng được, không nhất thiết phải là ID, miễn là làm sao bạn chỉ ra được cái hàng mà bạn cần sửa nội dung.



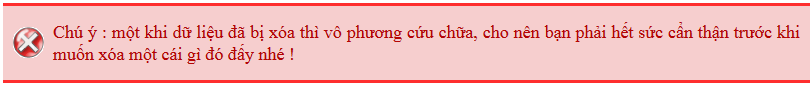
Ồ không đâu bạn, nếu bạn làm như thế này :  
**Code SQL :**

C:\Users\Chu Quang Vien\Desktop\13_7.PNG

Thì nó sẽ tìm tất cả các hàng chứa Michel và sửa lại thành Florent hết ! Vì vậy hãy chắc chắn rằng cái tên bạn chọn không bị lặp lại ở hàng khác, nếu không thì « cái không cần sửa cũng bị sửa luôn ! », hàng nào cũng có thể có chỗ nào đó giống nhau, chỉ có ID là khác nhau, vì vậy mà đại đa số người ta dùng ID để chỉ ra hàng cần sửa !

## Xóa dữ liệu

Thao tác cuối cùng : xóa dữ liệu bằng từ khóa DELETE !

**Code SQL :**

C:\Users\Chu Quang Vien\Desktop\13_9.PNG

Cấu trúc khá đơn giản nhỉ?  
Sau **DELETE** mình dùng từ khóa **FROM** để chỉ ra xóa ở đâu (ở bảng nào) và chỉ ra tên của bảng, rồi **WHERE** để chỉ ra hàng cần xóa! Bạn cũng có thể viết WHERE ID='51'. Kết quả cũng vậy, cả hàng nhà nó đi đời cả lũ! http://thanhtra.nguyen.free.fr/web/img/emotion/haha.gif

# Chương 5 : Thực hành tạo trang mini chat

* Trong chương này:

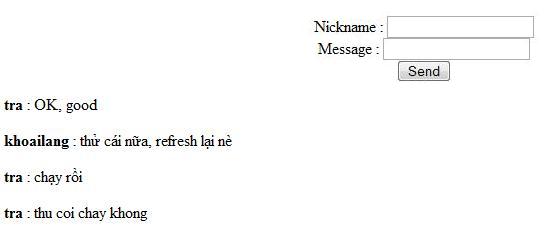
|  |  |
| --- | --- |
| 2-4 | [Thực hành](http://thanhtra.nguyen.free.fr/web/tuto-php/mysql5.php#thuchanh) |

## Tạo chương trình mini chat trên web

Tuy không vui bằng Shoutbox mà các trang web "lớn" hay dùng, nhưng dù sao cũng giúp bạn tiêu khiển tí chút với vốn kiến thức PHP mà bạn đã học được !   
  
**Bước 1 : Kiến thức cần có**

* Chuyển biến thông qua một trình đơn (qua một form)
* Đọc dữ liệu trong bảng CSDL
* Ghi dữ liệu vào bảng CSDL

**Bước 2: Chuẩn bị viết Script**  
Nếu bạn không muốn suy nghĩ mà đọc thẳng vào bài giải dưới đây thì … không vui lắm! Hãy suy nghĩ cơ chế hoạt động của nó một tí nhé, mình sẽ mô tả cái trang chat của mình như thế này:



* Một vùng text để nhập nickname
* Một vùng text để gõ vào nội dung chat
* Một nút Send để gửi
* Bên dưới là nội dung chat (mình sẽ cho hiển thị 10 nội dung chat mới nhất, theo thứ tự mới nhất trước, từ từ xuống dưới là nội dung cũ dần, cũ dần …)

Cơ chế hoạt động của nó như thế này: sau khi nhấn Send, nickname và nội dung chat của bạn sẽ được ghi vào bảng dữ liệu, giả sử trước đó mình tạo một table mới tên là **minichat** có 3 cột: ID, nickname, message, rồi mình lại lôi nội dung từ bảng dữ liệu này cho hiện ra trên trang chat! Để hiển thị nội dung mới nhất ở trên cùng, mình thêm vào trong query SQL "ORDER BY id DESC"!  
Xong, lí thuyết chỉ đơn giản thế thôi, khi bắt tay vào nấu nướng bạn sẽ thấy mùi vị của nó như thế nào mà tùy bạn thêm mắm hay thêm muối!  
  
**Bước 3: Hoàn chỉnh code**  
Dạo này mình hơi lười, nên không thay đổi màu sắc cho nó dễ nhìn, nhưng không sao, chỉ cần bạn copy paste vào notepad++ là nó sẽ tự động đổi màu ý mà!



**Bước 5 : Hoàn thiện code**  
Nếu dừng tại đây thì trông cái trang chat nó chán chán sao ấy nhỉ, chả màu mè hoa lá cành chi cả ! Bạn thử trang trí thêm cho đẹp đi ! Hay là có lỗi gì đó xuất hiện thì bạn thử sửa nhé ! Nếu không được, hãy post câu hỏi của bạn trong trang [lưu bút](http://thanhtra.nguyen.free.fr/web/luubut.php) nhé !