



TRƯỜNG TRUNG CẤP NGHỀ KỸ THUẬT CÔNG NGHỆ HÙNG VƯƠNG

KHOA: Công Nghệ Thông Tin



TÀI LIỆU GIẢNG DẠY

MÔN : Lập trình Web PHP & MySQL

HIỆU TRƯỞNG

TRƯỞNG KHOA

GIÁO VIÊN BIÊN SỌAN

TÀI LIỆU LUU HÀNH NỘI BỘ
Tháng 06 năm 2012

MỤC LỤC

CHƯƠNG 1

1.1 Giới thiệu PHP
1.2. Cài đặt và cấu hình Wamp
1.3 Cài đặt và cấu hình Xampp
1.4 Cài đặt và cấu hình PHP Designer IDE
1.5 Cài đặt và cấu hình NetBeans IDE

CHƯƠNG 2

2.1 PHP Script.....
2.2 Ghi chú trong PHP
2.3 In nội dung bằng PHP
2.4 Kiểu dữ liệu và biến
2.5 Khai báo và sử dụng hằng
2.6 Dữ liệu mảng
2.7 Chuyển đổi kiểu dữ liệu
2.8 Phép toán và phát biểu có điều kiện trong php
2.9 Xử lý chuỗi
2.10 Làm việc với mảng dữ liệu
2.11 Kiểu DateTime

CHƯƠNG 3

3.1 Giới thiệu toán tử
3.2 Phát biểu có điều kiện If
3.3 Switch
3.4 While
3.5 For
3.6 Do-While

CHƯƠNG 4

4.1 Biến form
4.2 Form submit với phương thức GET

CHƯƠNG 5

5.1 Xử lý dữ liệu nhập trên form và thông báo lỗi với javascript.....
5.2 Regular Expression

CHƯƠNG 6

6.1 Session
6.2 Cookie

CHƯƠNG 7

7.1 Các hàm xử lý lỗi
7.2 Làm việc mảng dữ liệu
7.3 Kiểu DateTime

CHƯƠNG 8

8.1 Khai báo hàm trong PHP
8.2 Thống nhất kích thước của mọi trang PHP

CHƯƠNG 9

9.1 Hướng đối tượng trong PHP

CHƯƠNG 10

10.1 Giới thiệu CSDL MySQL
10.2 Cài đặt MySQL
10.3 Tạo CSDL người dùng.....
10.4 Kiểu dữ liệu trong MySQL

10.5 Câu lệnh SQL.....
10.6 Các hàm thông dụng trong MySQL.....
10.7 Các hàm về toán học
10.8 Phát biểu SQL dạng Select
10.9 Phát biểu SQL dạng Join
10.10 Tạo kịch bản SQL- SQL Script.....

CHƯƠNG 11

11.1 Stored procedure trong MySQL
11.2 Con trả trong stored procedure
11.3 Trigger.....

CHƯƠNG 12

12.1 Kết nối CSDL

CHƯƠNG 13

13.1 Liệt kê dữ liệu

CHƯƠNG 14

<u>14.1 Mô hình MVC.....</u>
<u>14.2 Friendly URL.....</u>

Tài liệu tham khảo

LỜI NÓI ĐẦU

1. Lời giới thiệu tài liệu giảng dạy:

- Giáo trình "Lập trình web PHP & MySQL" là tài liệu dùng trong giảng dạy và học tập của học viên đang theo học hệ trung cấp nghề Thiết kế web trường TCN KTCN Hùng Vương tại địa chỉ 161 - 165 Nguyễn Chí Thanh, P12, Q5, TpHCM.

- Ngày nay các ứng dụng website rất phổ biến và áp dụng rộng rãi trong rất nhiều lĩnh vực. Hầu hết các tổ chức hay doanh nghiệp đều xây dựng website riêng nhằm mục tiêu giới thiệu thương hiệu các hoạt động và hình thức kinh doanh trên website, một số doanh nghiệp xây dựng loại hình thương mại điện tử trên website. Trong lĩnh vực thiết kế website thì có nhiều ngôn ngữ hỗ trợ như Java, DotNet, PHP,... Trong đó PHP và hệ quản trị cơ sở dữ liệu MySQL được nhiều lập trình viên lựa chọn vì tính đơn giản và phổ biến cao. Rất nhiều hệ thống mã nguồn mở nổi tiếng và thông dụng được xây dựng trên nền tảng PHP và MySQL như : Joomla, Moodle, NukeViet, Magento, OpenCart, Wordpress..... Điều đó chứng tỏ PHP đang trở thành 1 ngôn ngữ lập trình web thông dụng và được sử dụng nhiều trên thế giới hiện nay.

- Giáo trình được chia làm 13 chương, đi từ cơ bản đến chuyên sâu cung cấp các kiến thức nền tảng để xây dựng lập trình hoàn chỉnh 1 website sử dụng ngôn ngữ lập trình nguồn mở PHP và cơ sở dữ liệu MySQL.

2. Mục tiêu môn học

- Hiểu rõ các kỹ thuật thiết kế giao diện và lập trình website hoàn chỉnh
- Cung cấp các kiến thức nền tảng về các kỹ thuật phát triển web hiện đại
- Hình thành và phát triển tư duy lập trình và tư duy phát triển ứng dụng web
- Xây dựng được hoàn chỉnh một ứng dụng web tuân thủ đầy đủ các bước trong quy trình phát triển phần mềm với 3 dạng web phổ biến:
 - + Website chuyên về trình bày thông tin – tin tức
 - + Website chuyên về bán hàng – thương mại điện tử
 - + Website chuyên về quản trị nội dung CMS

CHƯƠNG 1

1.1. Giới thiệu PHP

- PHP viết tắt của chữ Personal Home Page ra đời năm 1994 do phát minh của Rasmus Lerdorf, và nó tiếp tục được phát triển bởi nhiều cá nhân và tập thể khác, do đó PHP được xem như một sản phẩm của mã nguồn mở.
- PHP là kịch bản trình chủ (*server script*) chạy trên phía server (*server side*) như cách server script khác (*asp, jsp, cold fusion*).
- PHP là kịch bản cho phép chúng ta xây dựng ứng dụng web trên mạng internet hay intranet tương tác với mọi cơ sở dữ liệu như mySQL, PostgreSQL, Oracle, SQL Server và Access.
- Lưu ý rằng, từ phiên bản 4.0 trở về sau mới hỗ trợ session, ngoài ra PHP cũng như Perl là kịch bản xử lý chuỗi rất mạnh chính vì vậy bạn có thể sử dụng PHP trong những câu lệnh có yêu cầu về xử lý chuỗi.

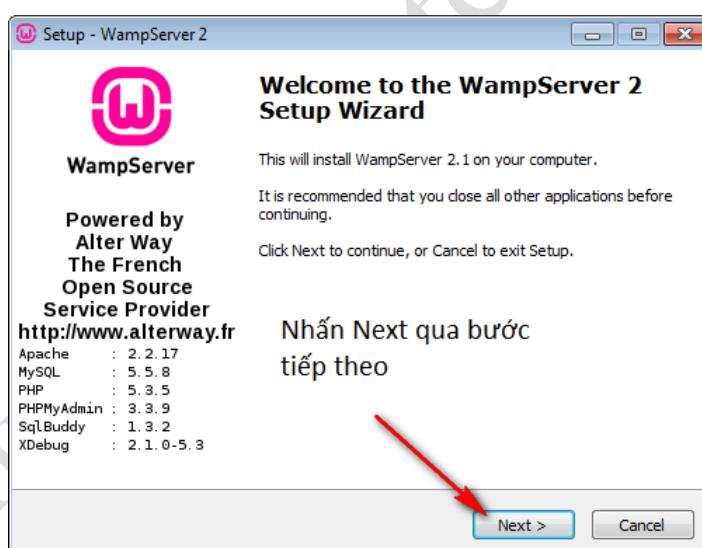
1.2. Cài đặt và tìm hiểu Wamp

- Cài đặt: Các bước cài đặt wampserver như sau:

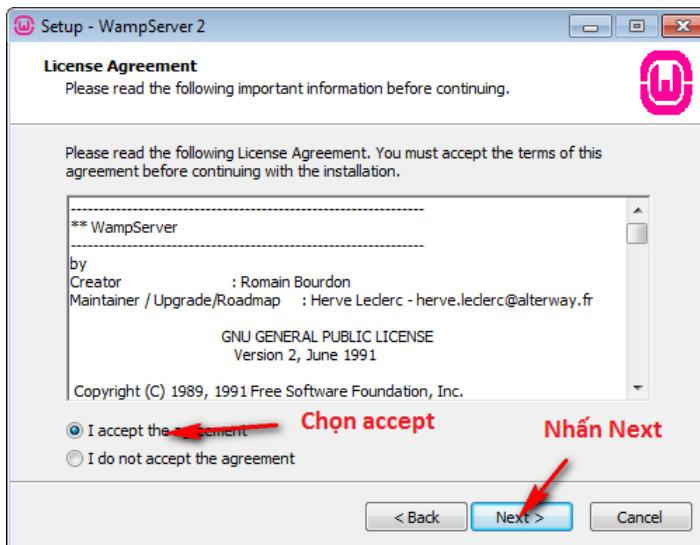
Bước 1: download chương trình wampserver 2.0 theo địa chỉ sau :

<http://www.wampserver.com/en/download.php>

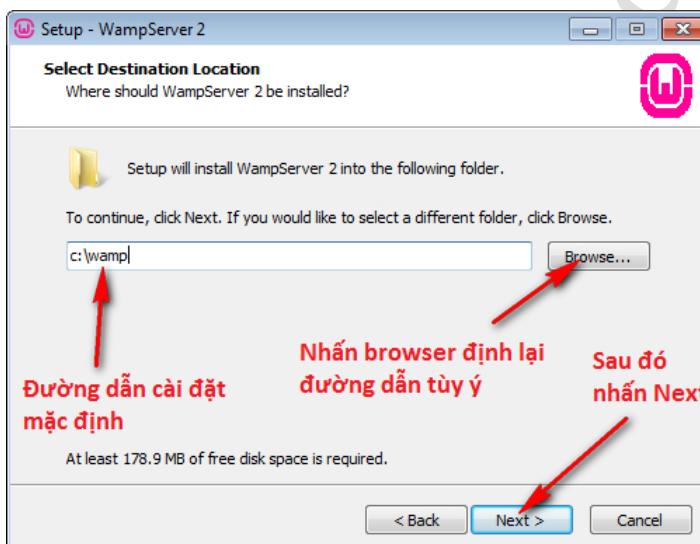
Bước 2: làm theo như hình chỉ dẫn



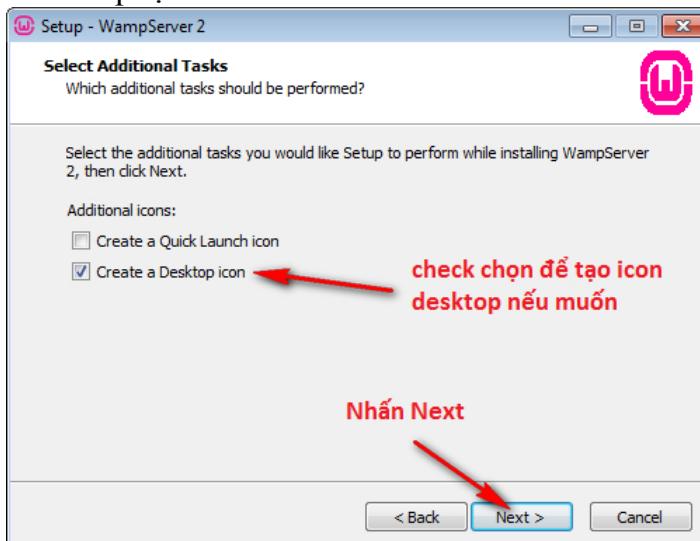
- Nhấn *Next* để tiếp tục.



- Click chọn mục ***I accept the agreement*** để đồng ý việc cài đặt
- Nhấn **Next** để tiếp tục.

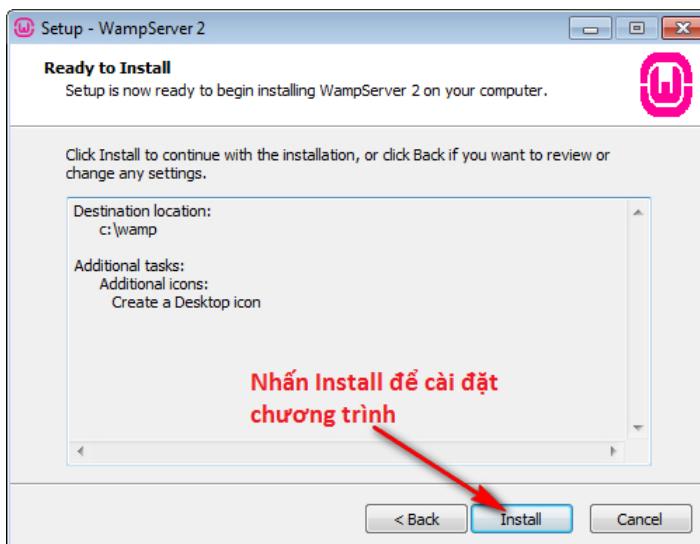


- Đường dẫn cài đặt mặc định sẽ là **c:\wamp**.
- Nếu muốn cài đặt tùy ý thì chọn **Browse..** để chọn nơi cài đặt khác.
- Sau đó nhấn **Next** để tiếp tục.

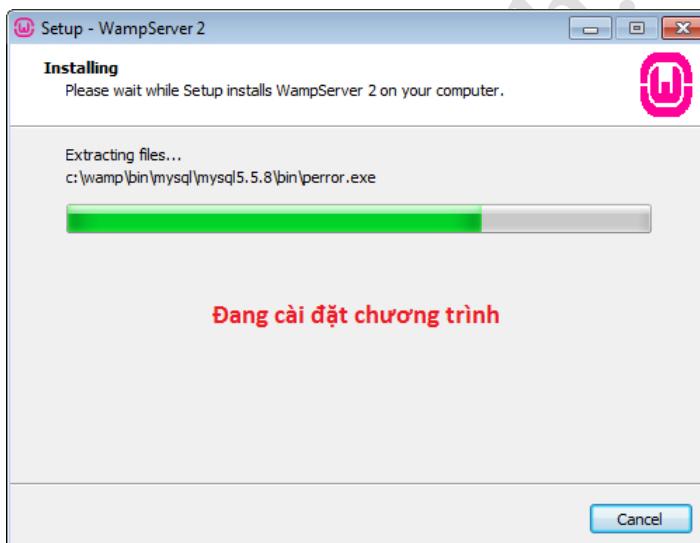


-Check vào **Create a Desktop icon** để tạo icon ngoài màn hình Desktop

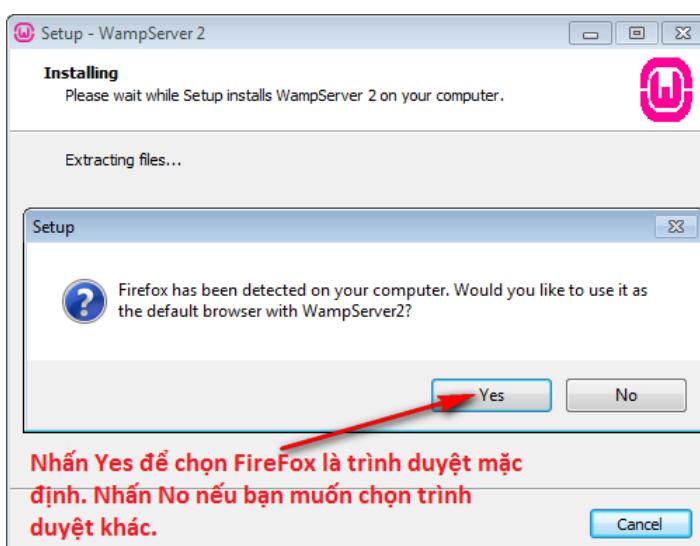
- Nhấn **Next** để tiếp tục.



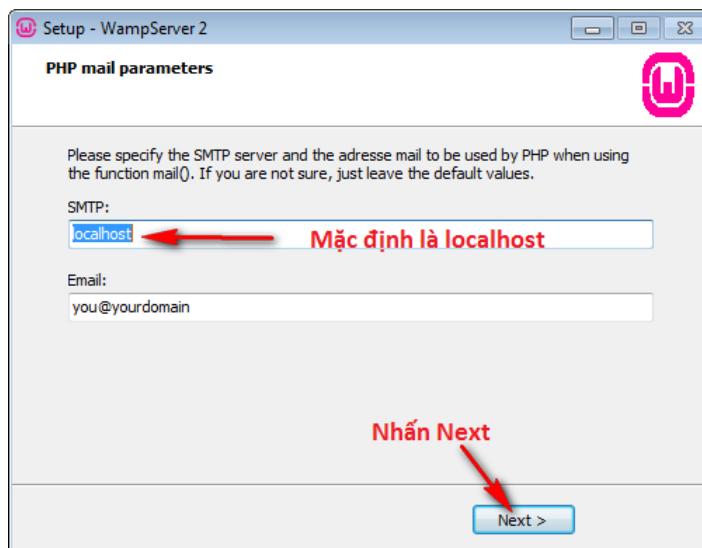
- Nhấn **Install** để bắt đầu quá trình cài đặt.



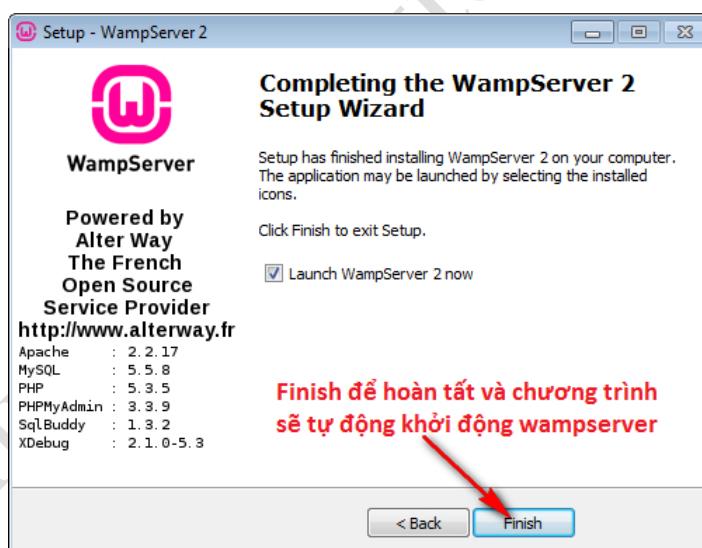
- Đang cài đặt **WampServer**



- Sau khi cài đặt xong. Chương trình sẽ hiện thị thông báo yêu cầu chọn trình duyệt để sử dụng. Ở đây mặc định sẽ là **firefox**, nếu chọn mặc định thì nhấn **Yes**, nếu không thì nhấn **No** và chọn đến trình duyệt tùy ý.



- SMTP mặc định là **localhost**
- Email mặc định là **you@yourdomain** (có thể thay đổi tùy ý)
- Nhấn **Next** để tiếp tục.

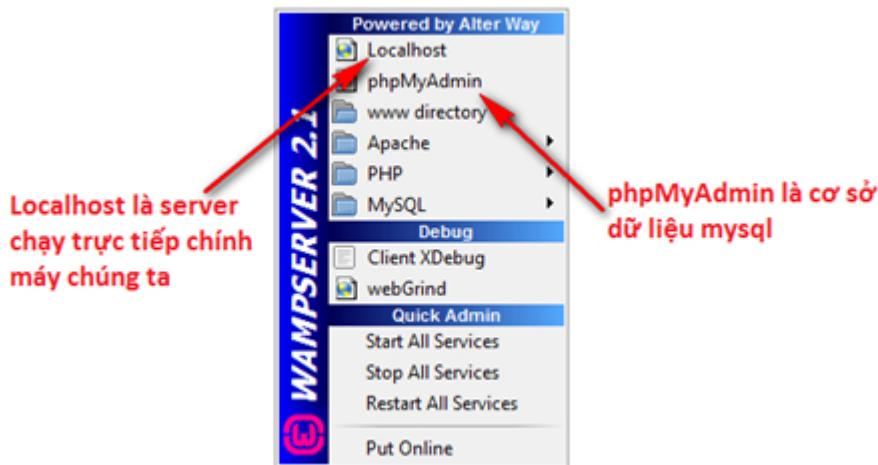


- Nhấn **Finish** để hoàn thành cài đặt.

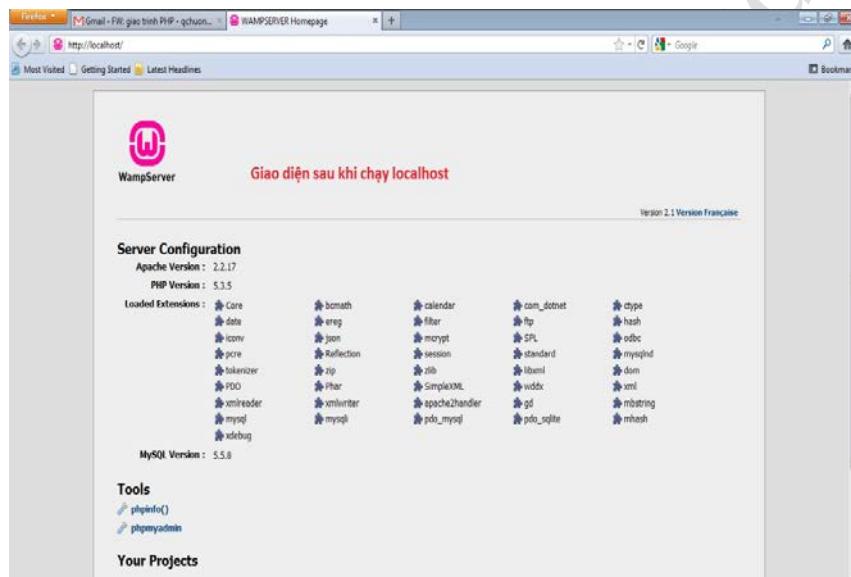
Biểu tượng sau khi wampserver khởi động



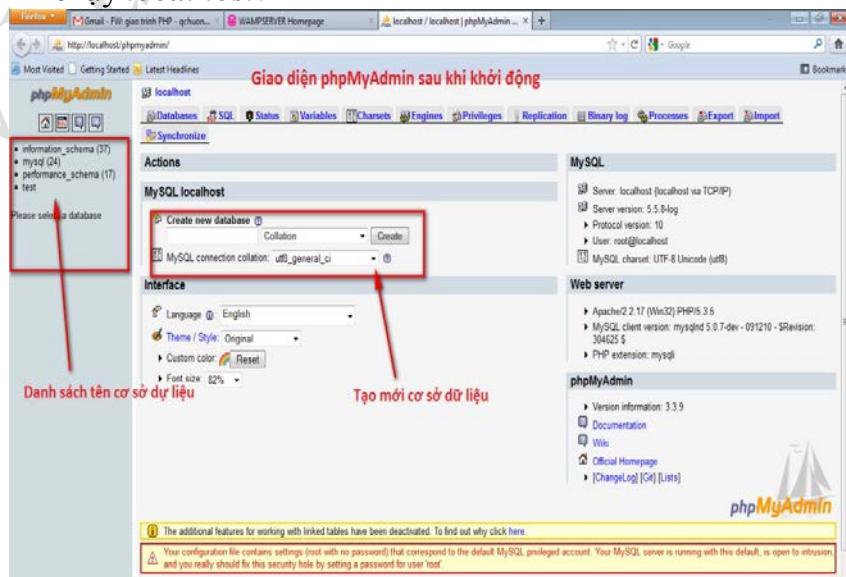
- Biểu tượng của WampServer sau khi khởi động chương trình.



- **localhost** là server chạy trực tiếp chính máy chúng ta.
- **PhpMyAdmin** là cơ sở dữ liệu.



- Giao diện khi chạy **localhost**.



- Giao diện cơ sở dữ liệu **phpMyAdmin**
- Bên trái là danh sách tên cơ sở dữ liệu có trong **phpMyAdmin**

- Bên phải là thông tin về MySQL, WebServer và version phpMyAdmin và các chức năng của phpMyAdmin

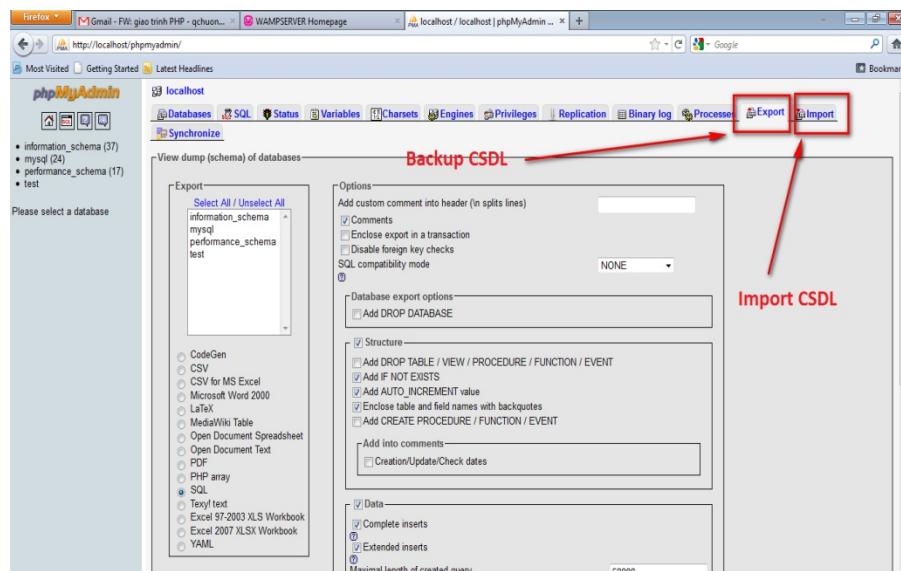
Tìm hiểu:

The screenshot shows the phpMyAdmin interface for the 'localhost' database. On the left, there's a sidebar with icons for Home, Databases, SQL, Status, Variables, Charsets, Engines, Privileges, Replication, Binary log, Processes, Export, and Import. Below this is a list of databases: information_schema (37), mysql (24), performance_schema (17), and test. A red box highlights this list with the label 'Danh sách Database'. In the center, there's a 'MySQL localhost' section with a 'Create new database' button and a dropdown for 'Collation'. Another red box highlights this area with the label 'Tạo mới Database'. To the right, a large red box encloses the 'MySQL' and 'Web server' sections, which provide detailed information about the MySQL server (version 5.5.8-log, protocol 10, user root@localhost) and the web server (Apache2.2.17, PHP 5.3.5). A third red box highlights the 'phpMyAdmin' section with the label 'Thông tin về MySQL, Web server and phpMyAdmin'. At the bottom, there are two warning messages: one about deactivated features and another about a security hole related to the root user.

- Bên trái là danh sách tên cơ sở dữ liệu có trong phpMyAdmin
- Bên phải là thông tin về MySQL, WebServer và version phpMyAdmin và các chức năng của phpMyAdmin
- Phần trên là menu chức năng như import, export hay tạo mới database

The screenshot shows the 'SQL' tab of the phpMyAdmin interface. At the top, there are tabs for Database, SQL, Status, Variables, Charsets, Engines, Privileges, Replication, Binary log, Processes, Export, and Import. The 'SQL' tab is highlighted with a red box. Below the tabs, there's a large text area labeled 'Tab SQL dùng để viết các câu lệnh truy vấn' (SQL tab for writing queries). This area contains a placeholder text 'Run SQL query/queries on server "localhost":'. At the bottom of this area, there's a 'Delimiter:' input field and a 'Go' button. A red arrow points from the label 'Khu vực viết lệnh' (Query writing area) to the main text input field.

- Thủ SQL dùng để viết query truy vấn cơ sở dữ liệu.



- Các thẻ export và import như đã nói ở phần trên là dung để export và import database.

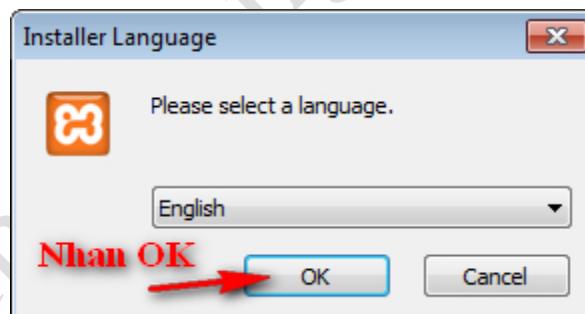
1.3. Cài đặt và tìm hiểu Xampp

Cài đặt:

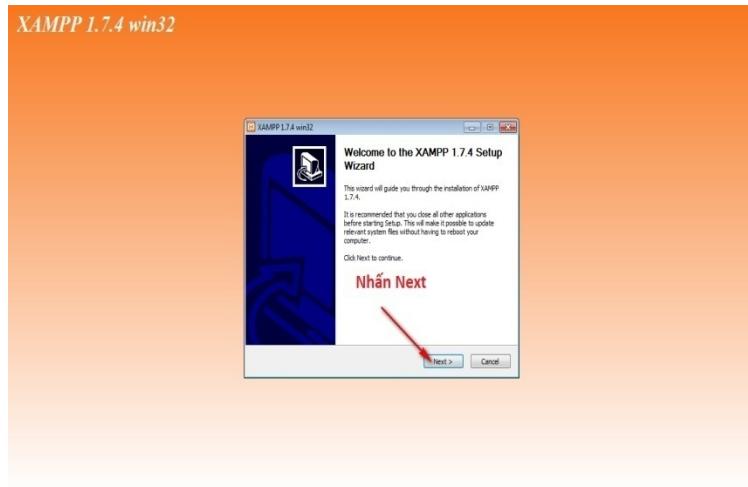
Bước 1: download chương trình xampp về theo địa chỉ sau:

<http://www.apachefriends.org/en/xampp-windows.html#641>

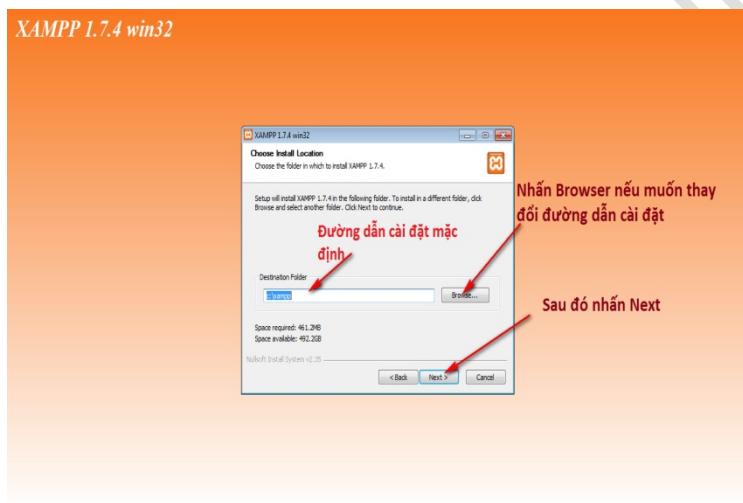
Bước 2: thực hiện theo hình chỉ dẫn sau:



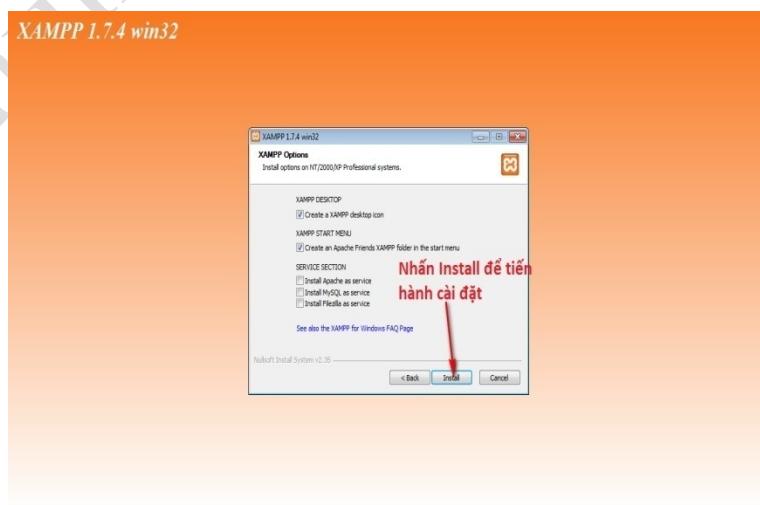
- Chạy file setup vừa download về.
- Nhấn OK để bắt đầu chương trình cài đặt.



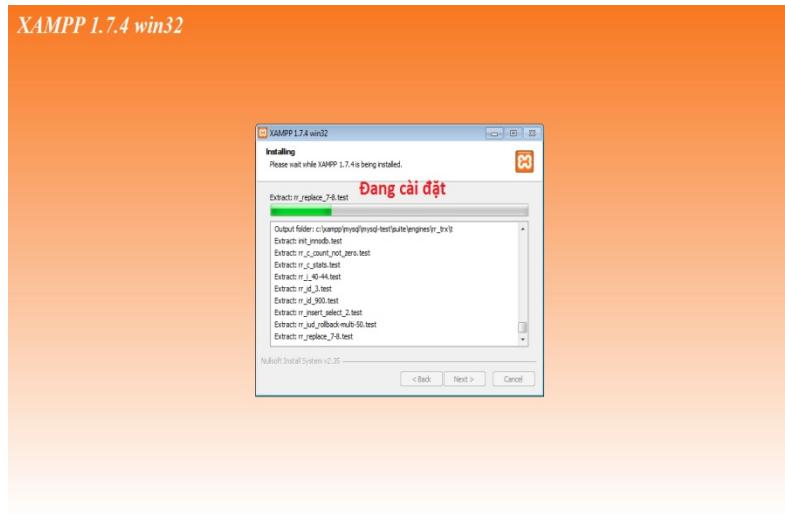
- Nhấn **Next** để tiếp tục.



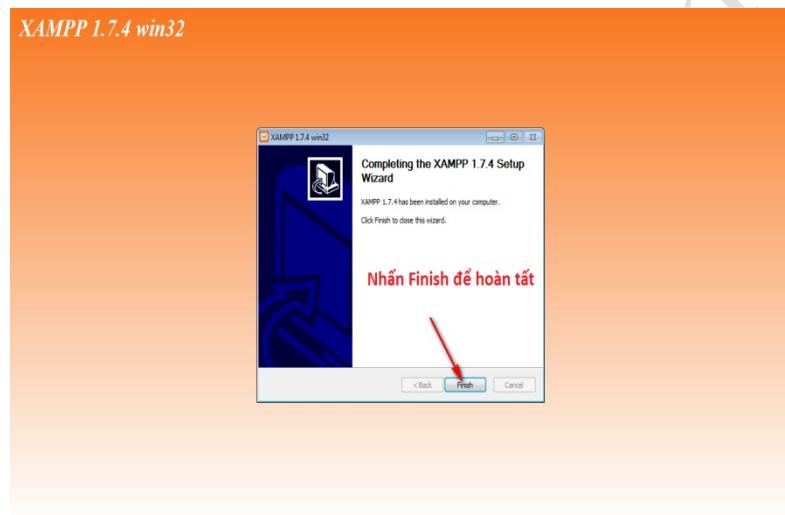
- Mặc định chương trình sẽ cài đặt mặc định là ổ C
- Nếu muốn cài đặt tùy ý thì chọn **Browse** để chọn đường dẫn khác tùy ý.



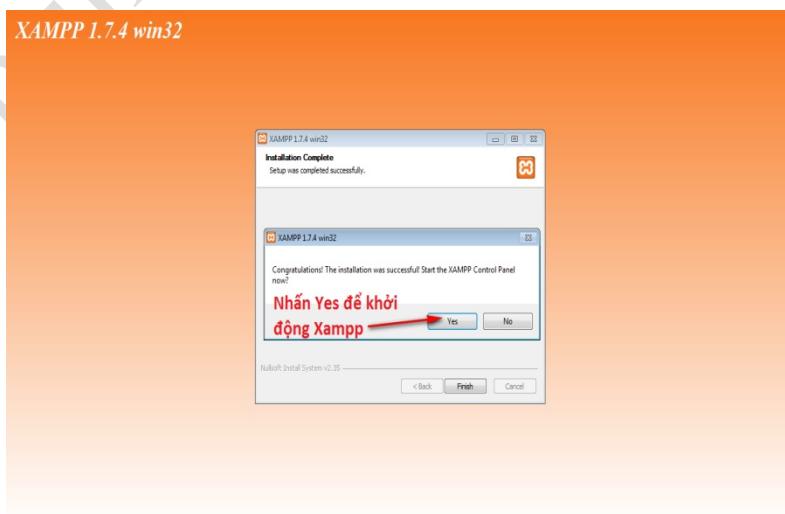
- Nhấn **Install** để tiến hành quá trình cài đặt **xampp**



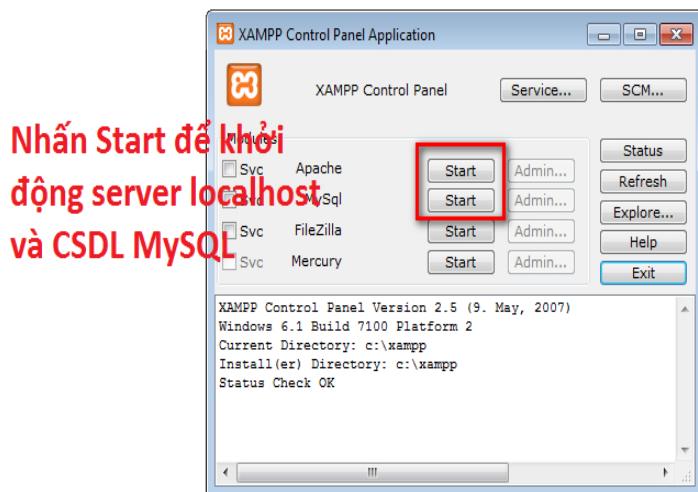
- Đang trong quá trình cài đặt **xampp**



- Nhấn **Finish** để hoàn thành cài đặt **xampp**

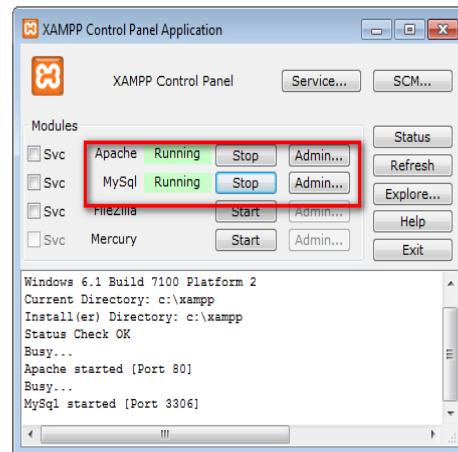


- Chương trình hiện thị thông báo có khởi động **xampp** sau khi cài đặt không. Nhấn **Yes** để khởi động chương trình.



- Hiện tại server và CSDL MySQL chưa được khởi động.
- Nhấn **Start** để khởi động server localhost và cơ sở dữ liệu MySQL

Server và CSDL MySQL đã được kích hoạt

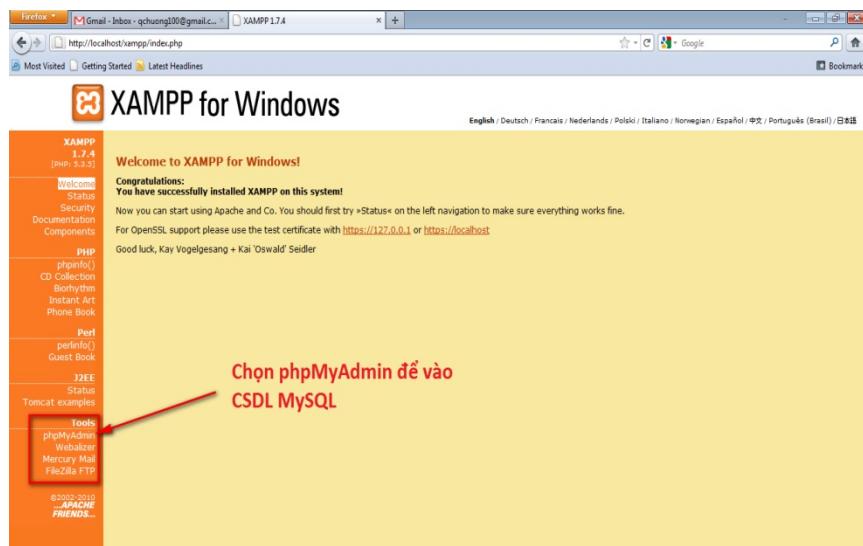


- Server và CSDL MySQL đã được khởi động.

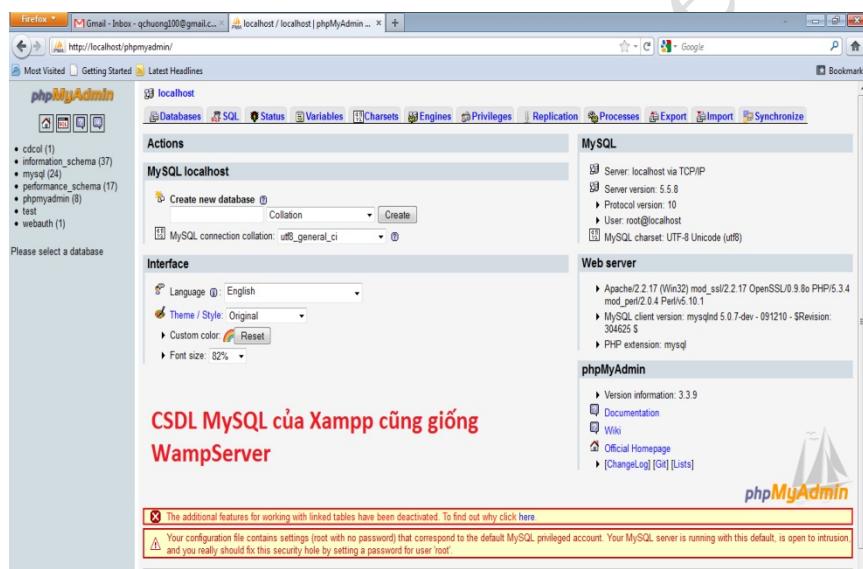
Chạy localhost nhấn Admin



- Để chạy localhost nhấn **Admin** như hình vẽ.



- Nếu muốn vào hệ thống CSDL MySQL thì chọn **phpMyAdmin** bên trái như hình vẽ.



Tìm hiểu:

- Bởi vì hệ quản trị csdl của **Xampp** cũng tương tự giống của **WampServer** nên chỉ cần xem lại hệ csdl wampserver đã viết ở trên.

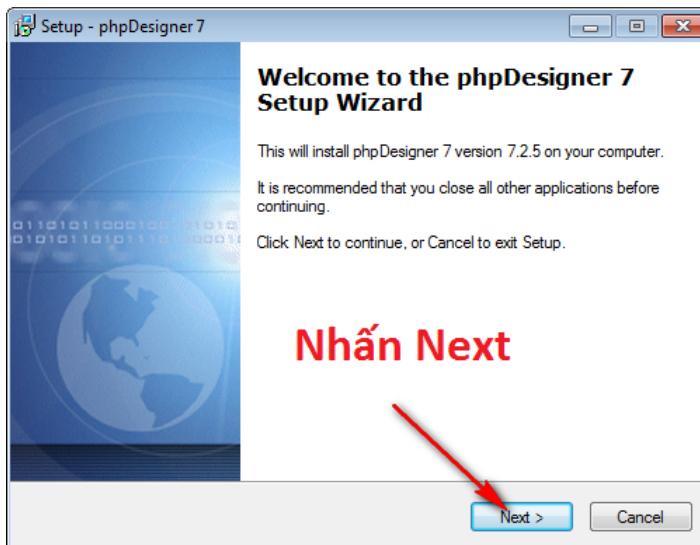
1.4. Cài đặt và cấu hình PHP Designer IDE

Bước 1: download phần mềm PHP Designer theo địa chỉ sau:

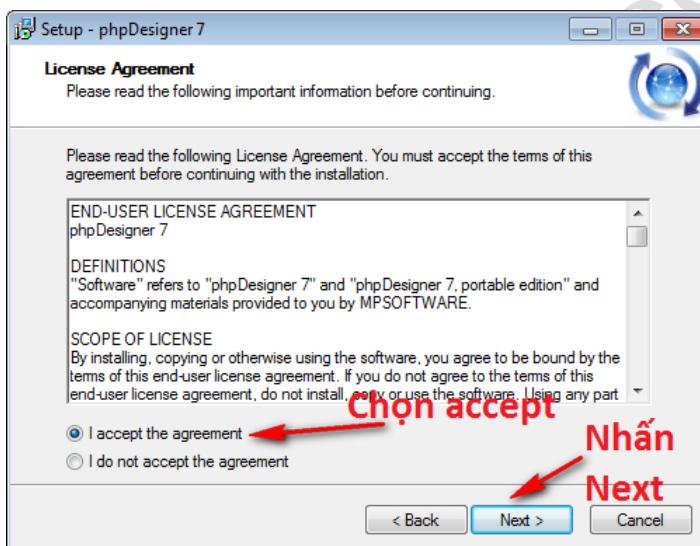
http://www.mpsoftwareweb.com/phpdesigner_7_2_5_setup.exe

Còn đây là link mediatr <http://www.mediafire.com/?aasic74i214eq6t>

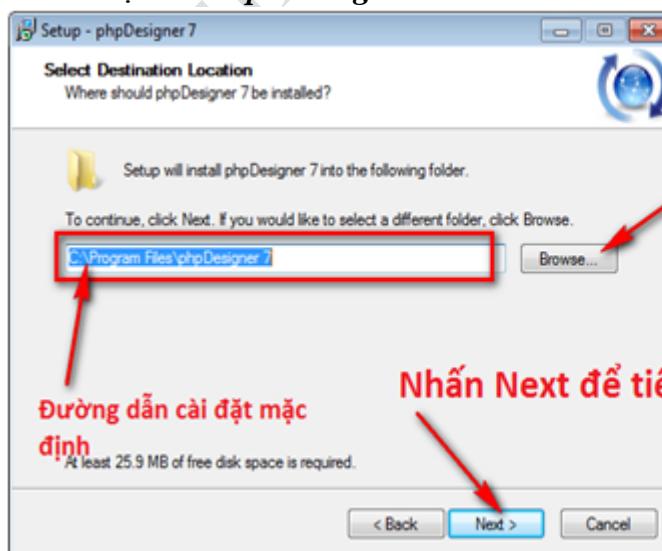
Bước 2: làm theo hình chỉ dẫn sau:



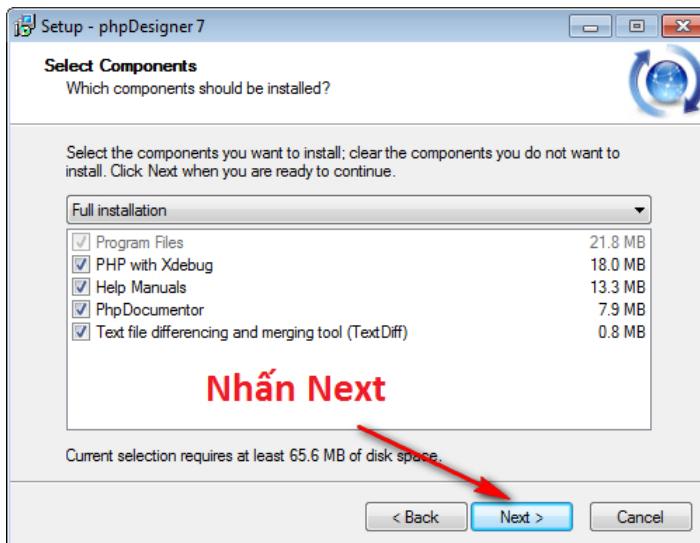
- Nhấn **Next** để tiếp tục.



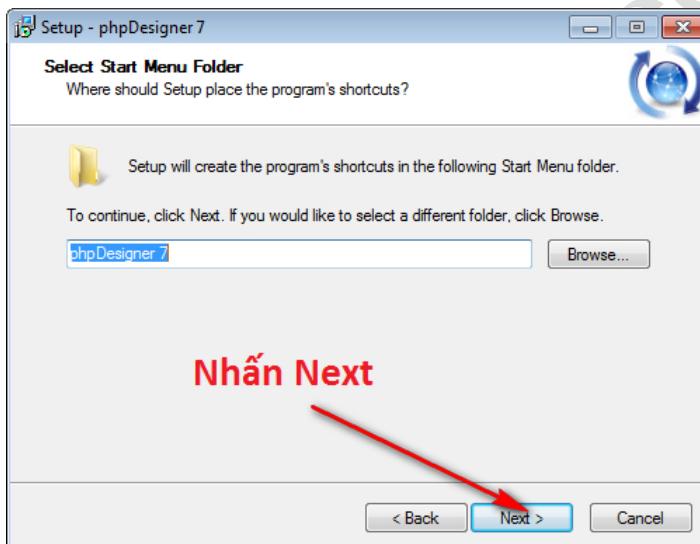
- Click chọn **I accept the agreement** và Nhấn **Next** để tiếp tục cài đặt.



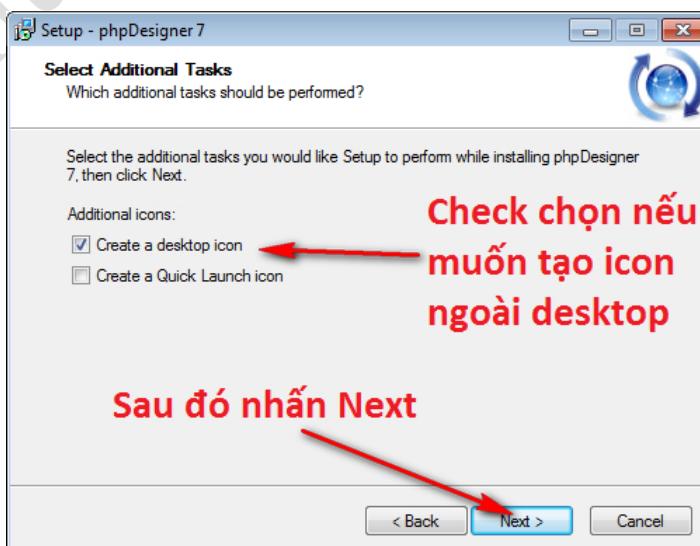
- Chương trình sẽ cài vào đường dẫn mặc định.
- Nhấn **Browse** để chọn lại đường dẫn tùy ý. Sau đó nhấn **Next** để tiếp tục cài đặt chương trình.



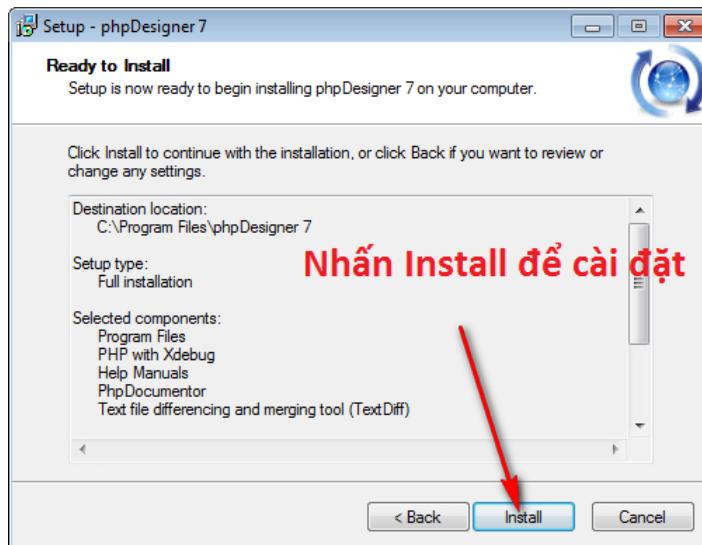
- Bước này nhấn **Next** để tiếp tục.



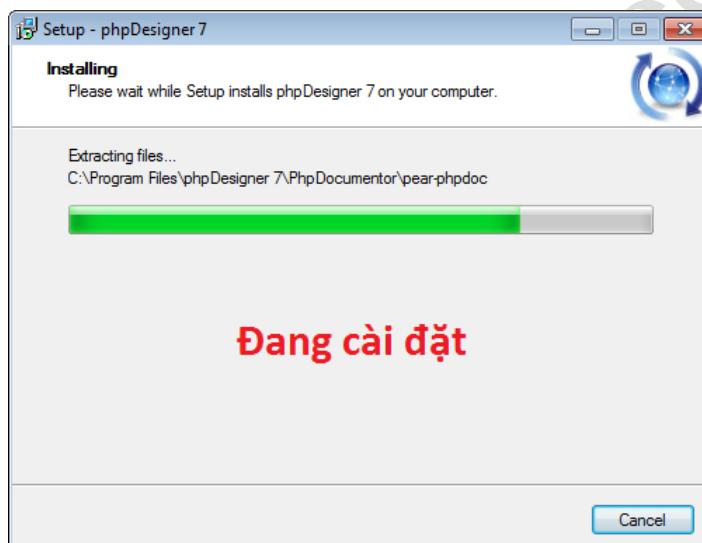
- Vẫn nhấn **Next** để tiếp tục việc cài đặt.



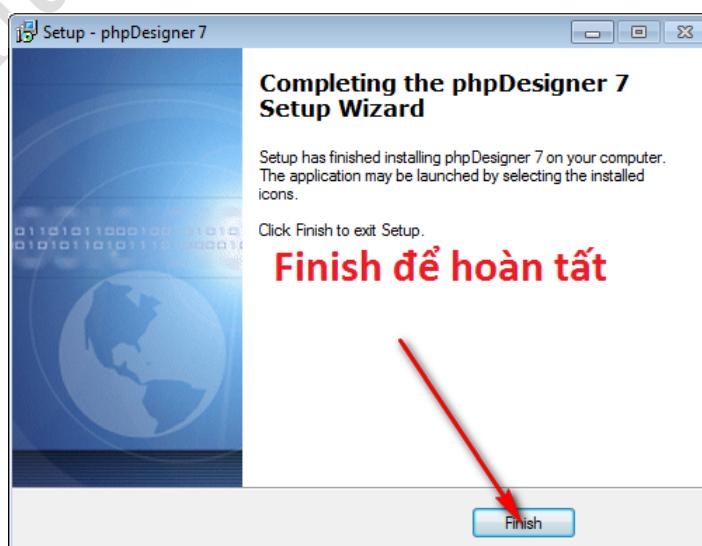
- Click check vào **Create a desktop icon** để tạo icon chương trình ngoài desktop.
- Sau đó nhấn **Next** để tiếp tục quá trình cài đặt.



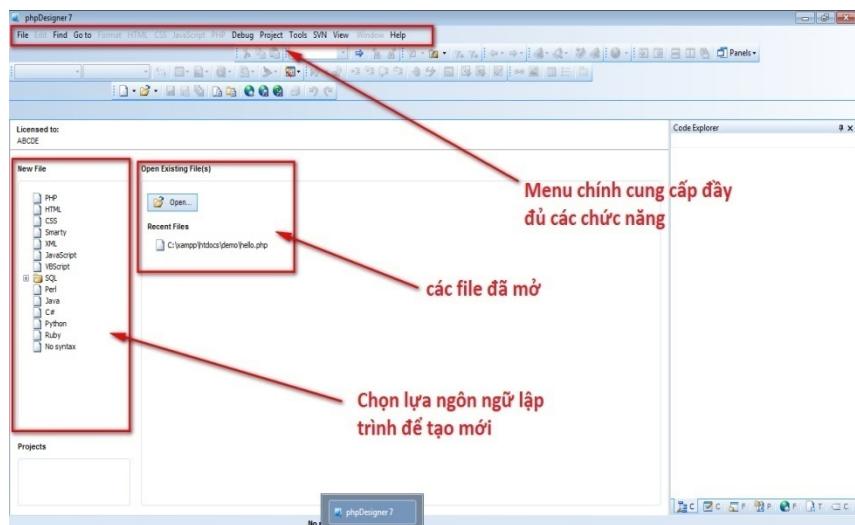
- Nhấn **Install** để bắt đầu quá trình cài đặt chương trình vào ổ cứng.



- Chương trình đang được cài đặt.



- Nhấn **Finish** để hoàn tất quá trình cài đặt.

Tìm hiểu:

- Khởi động chương trình **phpDesigner**. Giao diện chương trình sẽ như hình vẽ trên.
- Bên trái là danh sách các ngôn ngữ lập trình mà chương trình hỗ trợ. Chúng ta chọn 1 ngôn ngữ lập trình rồi double click vào ngôn ngữ đó để tạo mới.
- Chính giữa là phần hiển thị danh sách các file đã mở. Chúng ta có thể mở file bằng cách nhấn vào biểu tượng **Open**.
- Phía trên là menu chính cung cấp đầy đủ các chức năng của chương trình.

1.5. Cài đặt và cấu hình NetBeans IDE**Nội dung:****Cài đặt:**

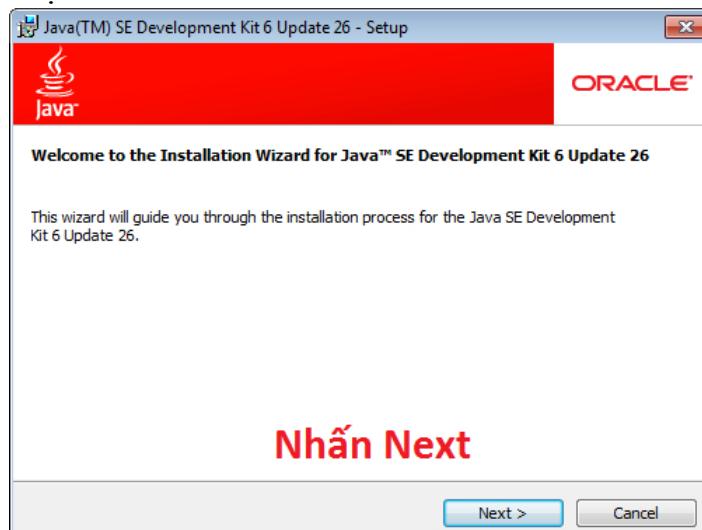
- Đầu tiên để cài đặt được NetBeans, máy tính cần phải có JDK. Download JDK theo địa chỉ sau: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Bước 1: download phần mềm NetBeansIDE 7.0 full theo địa chỉ sau:

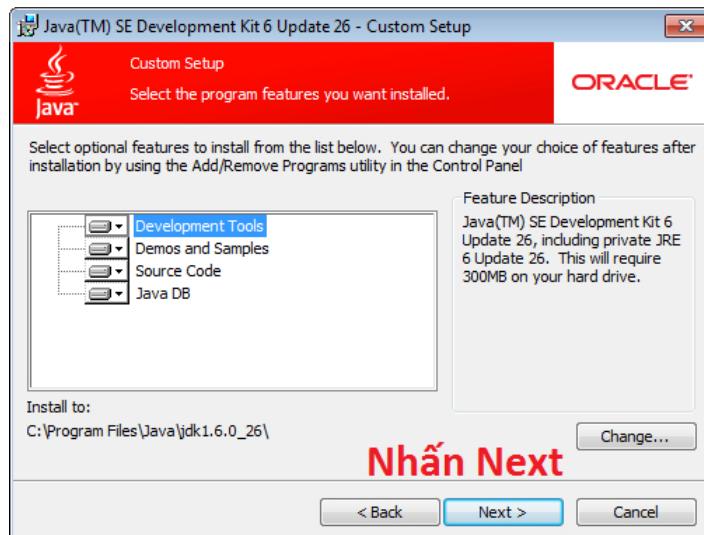
<http://netbeans.org/downloads/start.html?platform=windows&lang=en&option=all>
phiên bản này bao gồm tất cả ngôn ngữ lập trình như java, php, c++, hoặc download bản chỉ gồm PHP

<http://netbeans.org/downloads/start.html?platform=windows&lang=en&option=php>
Bước 2: làm theo hình chỉ dẫn sau:

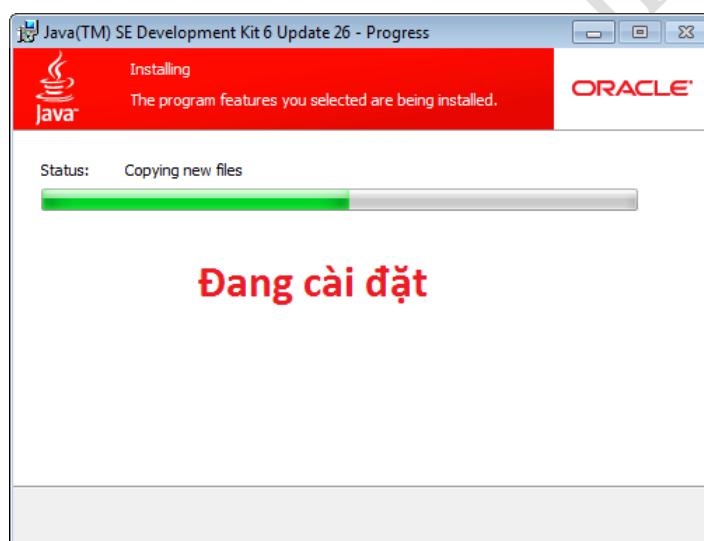
Hướng dẫn cài đặt JDK



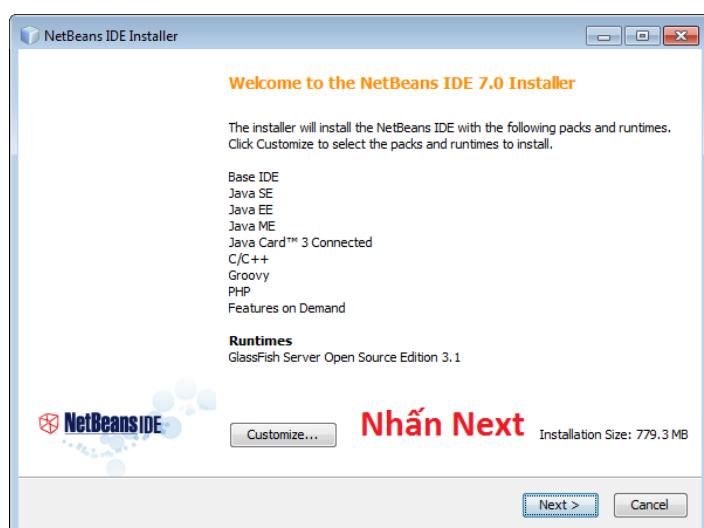
- Nhấn **Next** để tiếp tục quá trình.



- Nhấn **Change** để thay đổi đường dẫn tùy ý.
- Nhấn **Next** để bắt đầu cài đặt.

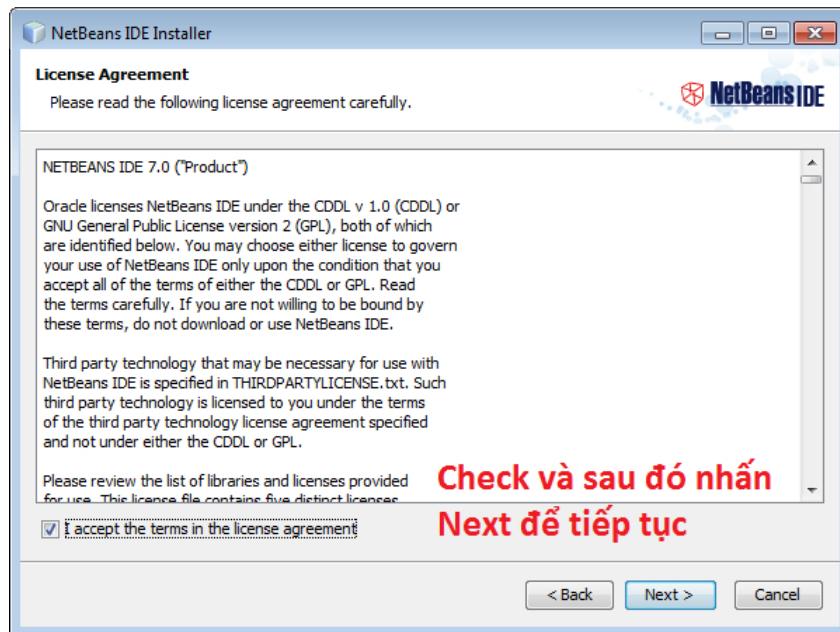


- Sau khi cài đặt JDK xong, chạy file cài đặt netbean 7.0

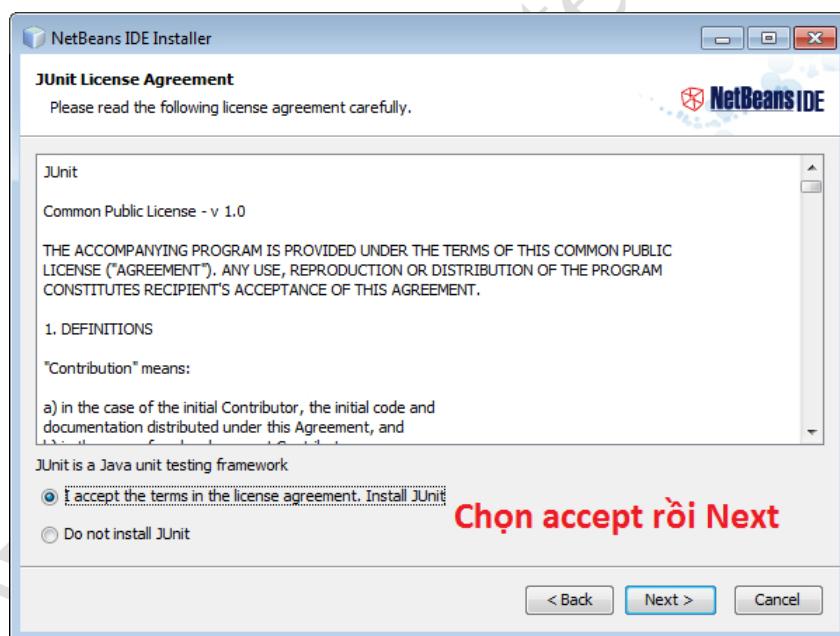


- Nhấn **Customize** để chọn ngôn ngữ mà bạn muốn cài đặt để lập trình.

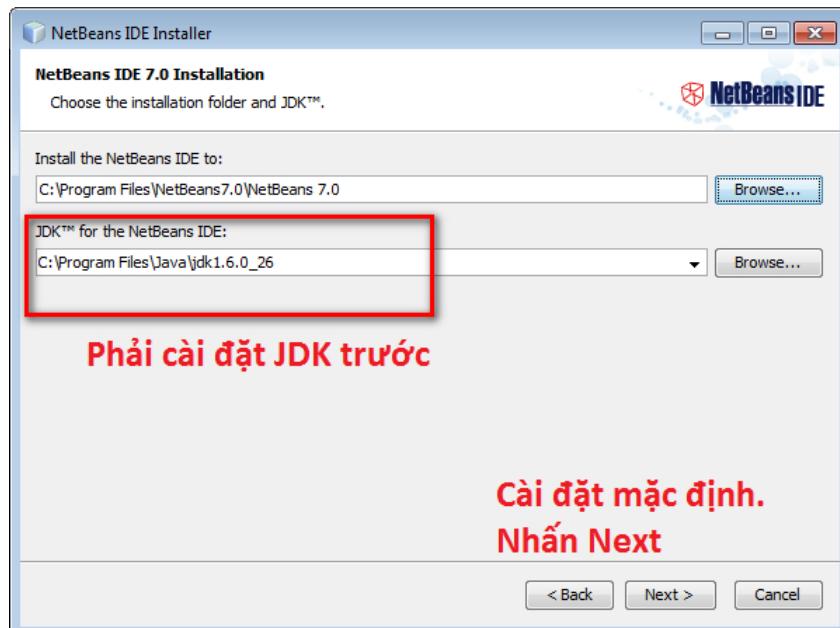
- Sau đó nhấn **Next** để tiếp tục.



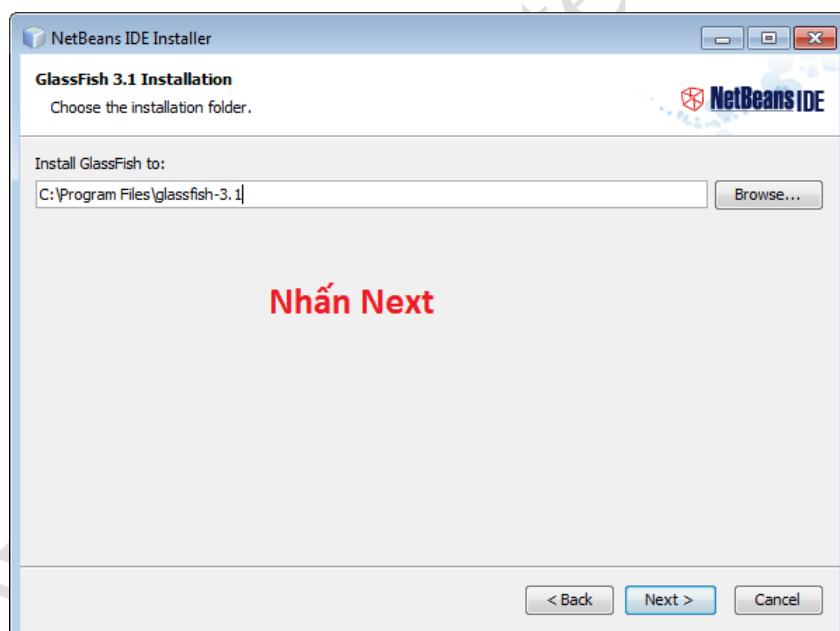
- Check vào **I accept the terms in the license agreement**.
- Nhấn **Next** để tiếp tục cài đặt.



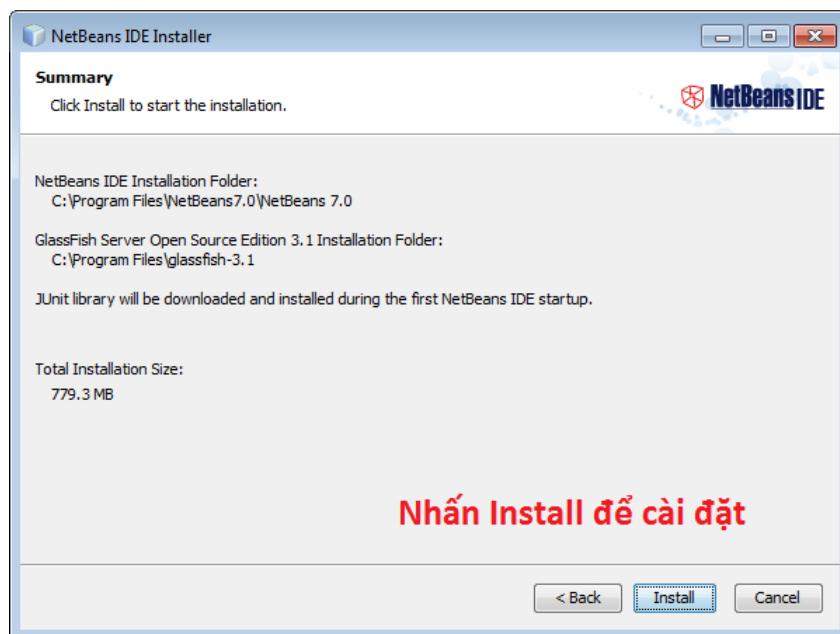
- Check vào **I accept the terms in the license agreement. Install JUnit**.
- Nhấn **Next** để tiếp tục cài đặt.



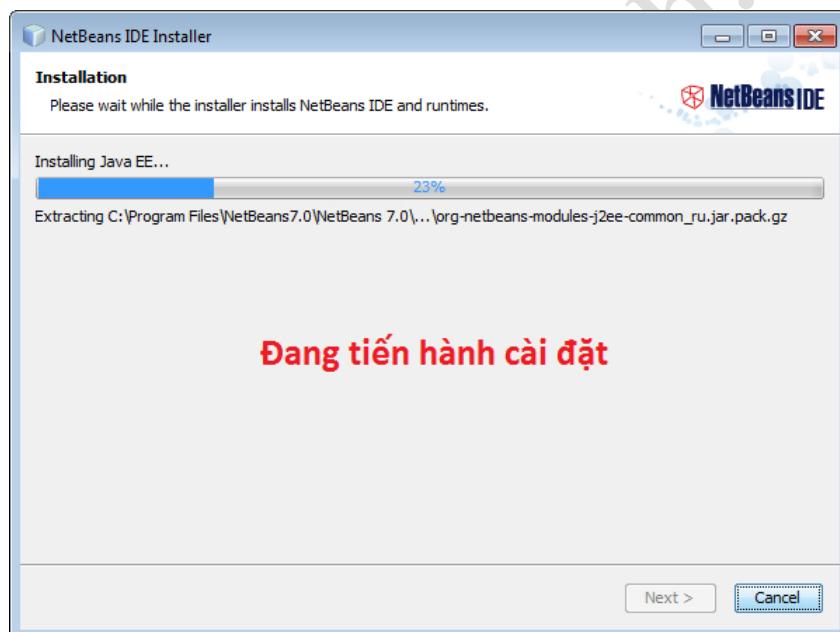
- **NetBeans IDE** tự động tìm kiếm JDK đã cài đặt trước. Nếu bạn chưa cài JDK thì không thể cài đặt NetBeans IDE.
- Nhấn **Browse** phía trên nếu muốn thay đổi đường dẫn cài đặt chương trình. Sau đó nhấn **Next** để tiếp tục quá trình.



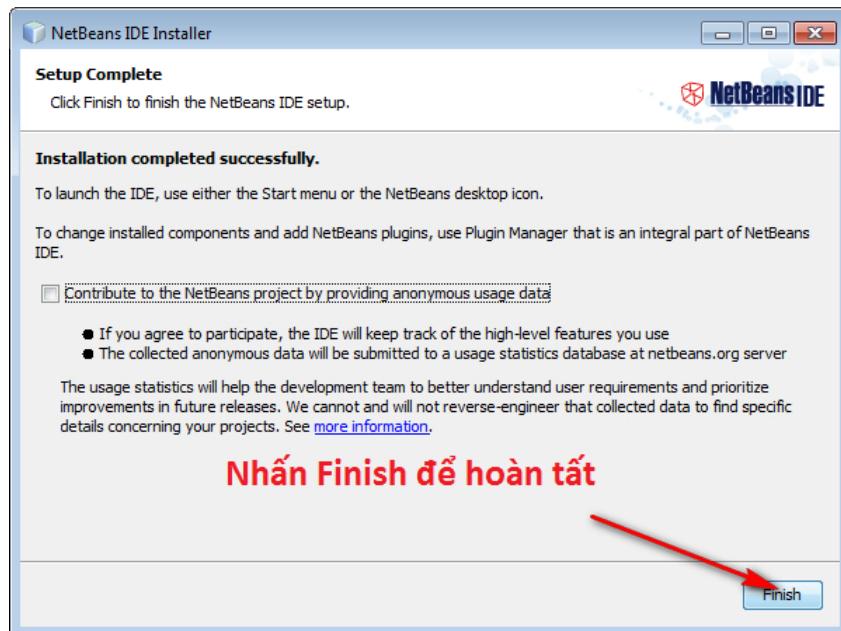
- Nhấn Next để tiếp tục.



- Nhấn **Install** để cài đặt chương trình.



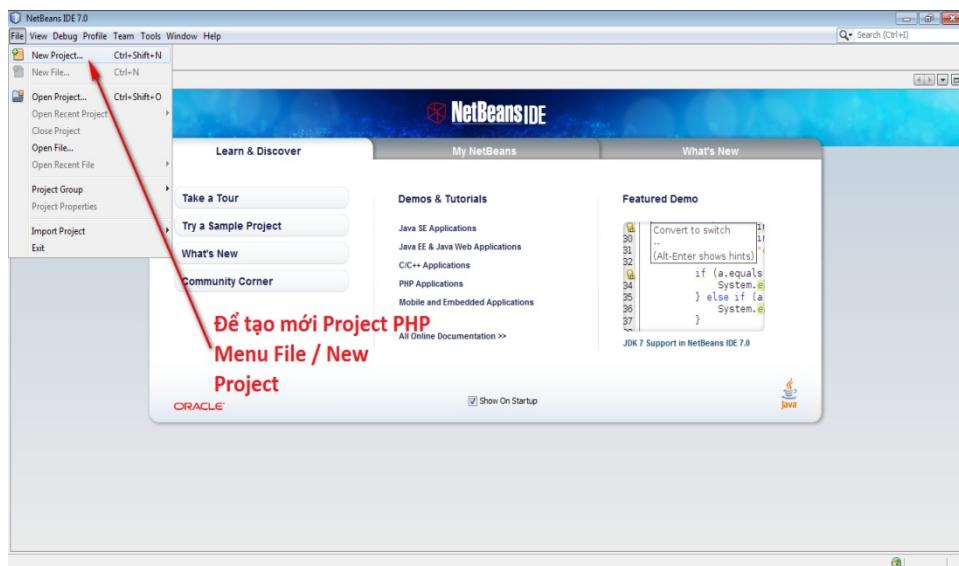
- Chương trình đang được cài đặt.



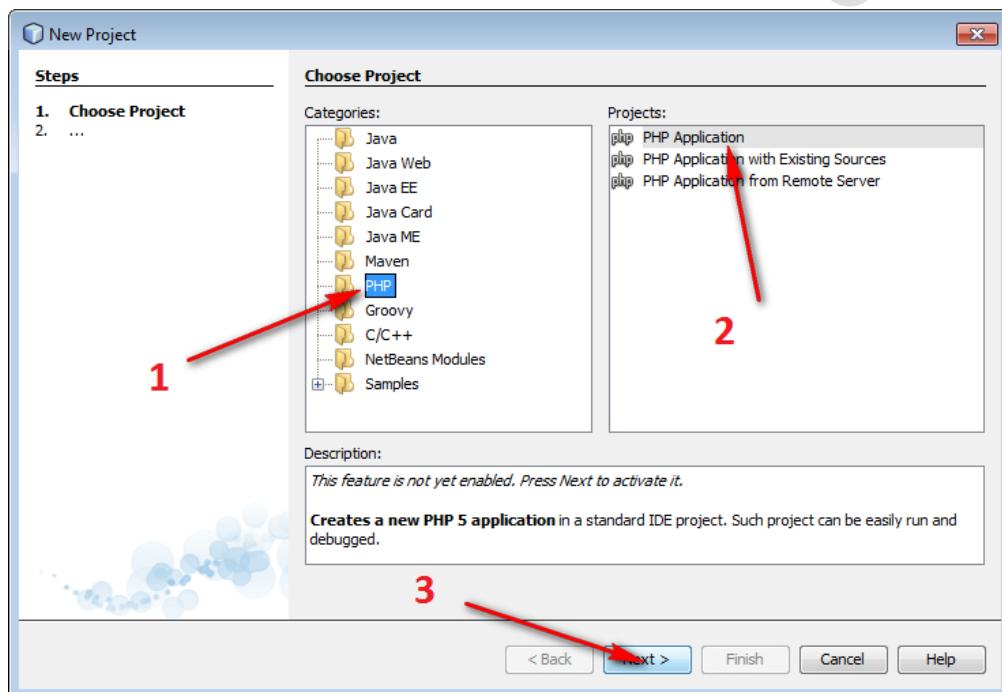
- Nhấn **Finish** để hoàn tất quá trình cài đặt NetBeans IDE.
- Sau đó khởi động **NetBeans IDE**.
- Giao diện **NetBeans IDE** sẽ như hình bên dưới.



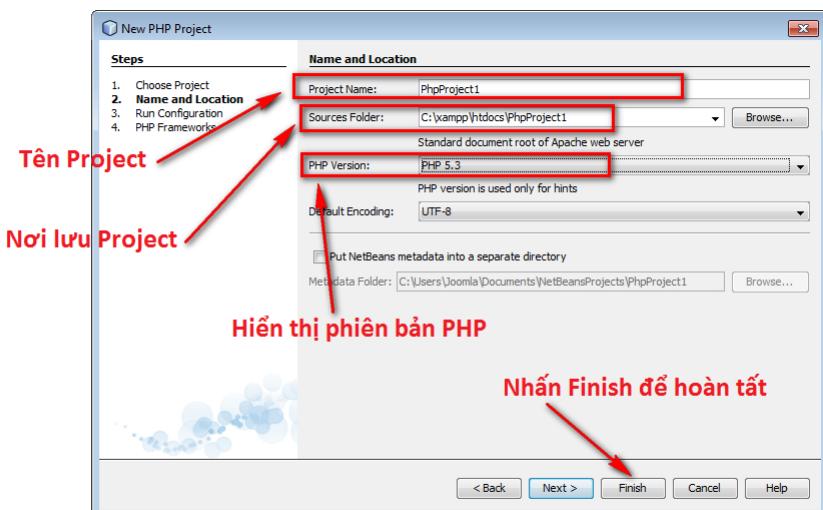
- Hướng dẫn cách tạo **Project PHP**



- Để tạo mới **Project PHP**, chúng ta vào menu **File** chọn **New Project**. Sau đó chương trình sẽ xuất hiện như hình sau:



- Nếu bạn lập trình PHP thì chọn ngôn ngữ PHP bên trái. Sau đó chọn **PHP Application** bên phải cửa sổ. Cuối cùng là nhấn **Next** để tiếp tục.



- Nhấn **Finish** để tạo mới một project lập trình ngôn ngữ PHP.

Câu hỏi ôn tập:

Câu 1: Làm sao để tạo mới database trong **Wamp/ Xampp** ?

Câu 2: Làm sao viết câu truy vấn trong **Wamp/Xampp** ?

Câu 3: Làm sao backup, import, export database trong **Wamp/Xampp** ?

Câu 4: So sánh ưu & khuyết điểm của **Wampp** và **Xampp** ?

CHƯƠNG 2

2.1. PHP script

2.1.1 Script

- Code PHP có thể được chèn lẩn lộn và bất cứ ở đâu trong body HTML, chính vì vậy, để phân biệt code PHP bạn sử dụng cặp dấu `<?=>`. Giá trị/Biểu thức/Biến?> .
Chúng ta có thể khai báo như sau :

```
<br>
1-Giá trị biến MemberNo : <?=$memberNo?>
2-Giá trị biến k : <?=$k?>
3-Giá trị cụ thể : <?="8000"?>
```

Ví dụ :

```
<HTML>
  <HEAD>
    <TITLE>Welcome to PHP World</TITLE>
  </HEAD>
  <BODY>
    Hello : <?="PHP World"?>
  </BODY>
</HTML>
```

2.1.2 Scriptlet

- Khác với cách khai báo Script, bạn chỉ được khai báo 1 biến. Scriptlet dùng trong trường hợp khi khai báo nhiều biến. Điều này có nghĩa là sử dụng cặp dấu trên như `<?php Khai báo ?>` với các khai báo PHP với cú pháp của C như sau:

```
<?php
$tenSinhVien=$studentName;
$column=$filed;
$totalSalary = 8000;
$studentNo="";
?>
-Khai báo trên là Scriptlet
<?= $paging ?>
-Khai báo này là Script
```

Lưu ý: Khi kết thúc mỗi câu trong script PHP phải dùng dấu ;

Ví dụ :

```
<HTML>
  <HEAD>
    <TITLE> Welcome to PHP World </TITLE>
  </HEAD>
<BODY>
<?php
$tenSinhVien=$studentName;
$column=$filed;
$totalSalary = 8000;
```

```

$studentNo="";
$mySchool="TCN KTCN Hùng Vương";
?>
Giá trị của mySchool : <?=$mySchool?>
</BODY>
</HTML>

```

Lưu ý rằng, nếu bạn muốn sử dụng script hay scriptlet như ASP thì bạn khai báo trong tập tin php.ini như sau:

```

asp_tags = On ;
Allow ASP-style <% %> tags. mặc định là Off

```

Khi đó, thay vì bạn khai báo cú pháp PHP là :

```

<?php
$tenSinhVien=$studentName;
$column=$filed;
$totalSalary = 8000;
$studentNo="";
?>

```

Thì bạn có thể khai báo như sau:

```

<%
$tenSinhVien=$studentName;
$column=$filed;
$totalSalary = 8000;
$studentNo="";
%>

```

2.2. Ghi chú trong PHP

- Ghi chú trong code PHP cũng tương tự như trong ngôn ngữ lập trình C. Để ghi chú 1 dòng trong PHP thì sử dụng một cặp dấu // hoặc #. Chẳng hạn chúng ta sẽ khai báo như sau :

```

<?php
//Ghi chú mot dong trong code PHP
#$tenSinhVien=$studentName;
//$column=$filed;
#$totalSalary = 8000;
//$studentNo="";
?>

```

- Ngoài ra, Nếu có nhiều dòng mà chúng ta cần ghi chú .Chúng ta nên sử dụng cặp dấu /* Nội dung ghi chú */.

Ví dụ :

```
<?php
    /*Ghi chu nhieu dong trong code PHP
    $tenSinhVien=$studentName;
    $column=$filed;
    $totalSalary = 8000;
    $studentNo=""*/?
?>
```

2.3. In nội dung bằng PHP

- Khác với các kịch bản như ASP, JSP, Perl, đối với code PHP để in ra giá trị của biến, biểu thức, hàm hoặc giá trị cụ thể thì bạn có thể sử dụng script như sau:

Giá trị của mySchool : <?=\$mySchool?>

Ngoài ra **echo** cũng được sử dụng khi in ra giá trị của biến, biểu thức, hàm, giá trị cụ thể trong PHP, chúng được khai báo như sau :

```
<?php
    $mySchool="TCN KTCN Hùng Vương";
    echo $mySchool;
?>
```

Ví dụ :

```
<HTML>
    <HEAD>
        <TITLE> Welcome to PHP World </TITLE>
    </HEAD>
<BODY>
    <?php
        $tenSV="Nguyen Van A";
        $tuoi="19";
        $soDT = "0975587553";
        /*Dùng echo để xuất ra thông tin */
        echo "Thông Tin Sinh Viên";
        echo $tenSV;
        echo $tuoi;
        echo $soDT;
    ?>
</BODY>
</HTML>
```

2.4: Khái niệm cú pháp trong PHP

- Cú pháp PHP được xây dựng trên nền tảng trong ngôn ngữ C, vì vậy các bạn làm quen với ngôn ngữ C thì có lợi thế trong việc tìm hiểu và lập trình PHP sau này.

Để lập trình tốt bằng ngôn ngữ PHP chúng ta cần chú ý những đặc điểm sau:

- Cuối câu lệnh có dấu ;
- Biến trong PHP có tiền tố là \$
- Mỗi phương thức đều bắt đầu { và đóng bằng dấu }
- Khi khai báo biến thì không có kiểu dữ liệu

- Nên có giá trị khởi đầu cho biến khai báo
- Phải có chi chú (comment) cho mỗi thuộc tính mới
- Sử dụng dấu // hoặc # để giải thích cho mỗi câu ghi chú
- Sử dụng /* và */ cho mỗi đoạn ghi chú
- Khai báo biến có phân biệt chữ hoa hay thường.

2.5: Khai báo biến trong PHP

- Khi khai báo biến trong C, chúng ta cần phải biết tuân thủ những quy định như : đầu tiên là kiểu dữ liệu , tiếp theo sau kiểu dữ liệu là tên biến và phải có giá trị khởi tạo. Nhưng trong PHP thì không cần khai báo phức tạp như vậy, chỉ cần sử dụng tiền tố \$ trước tên biến. Chúng ta có thể khai báo biến như sau :

```
$count = 0;
$queryStr = "SELECT * FROM tblSinhVien where status = 0";
$location = array("first", "last", "company");
$check = true;
```

Ví dụ :

```
<HTML>
  <HEAD>
    <TITLE>Welcome to PHP World</TITLE>
  </HEAD>
  <BODY>
    <h4>Khai bao bien</h4>
    <?php
      $soSV = 10;
      $test = true;
      $sql = "SELECT * FROM tblStudent";
      $location = array("first", "last", "company");
      $myarrs[] = array(3);
      $myarrs[0] = "SV001";
      $myarrs[1] = "SV002";
      $myarrs[2] = "SV003";
      echo $myarr[2]; echo "<br>";
      echo $myarrs[1];
    ?>
  </BODY>
</HTML>
```

2.6: Kiểu dữ liệu

Danh sách các kiểu dữ liệu cơ bản và thường dùng :

- | | |
|------------------|----------------------------------|
| - Boolean | True hay false |
| - Integer | giá trị lớn nhất xấp xỉ 2 tỷ |
| - Float | ~1.8e308 gồm 14 số lẽ |
| - String | Lưu chuỗi ký tự chiều dài vô hạn |
| - Object | Kiểu đối tượng |
| - Array | Mảng với nhiều kiểu dữ liệu |

2.6.1: Thay đổi kiểu dữ liệu

- Thay đổi kiểu dữ liệu (ép kiểu). Để thay đổi từ kiểu dữ liệu này sang kiểu dữ liệu khác, tương tự như trong ngôn ngữ lập trình C hay Java, bạn có thể sử dụng cách ép kiểu. Chẳng hạn, khai báo ép kiểu như sau:

```
<HTML>
  <HEAD>
    <TITLE>Welcome to PHP World</TITLE>
  </HEAD>
<BODY>
  <h4>Thay doi kieu du lieu(Ep Kiep)</h4>
  $i="T50T";
  echo $i."<br>";
  echo ($i + 20)."<br>";
  $i="200A";
  $j=(float)$i;
  echo $j."<br>";
  $j+=10;
  echo $j."<br>";
  $q=25;$p=4;
  echo "q/p: ".(float)$q/$p;
</BODY>
</HTML>
```

Lưu ý rằng PHP tự động nhận biết giá trị chuỗi đằng sau số, vì vậy các giá trị chuỗi sẽ không được chuyển sang kiểu số như trường hợp trên.

Ngoài phương pháp ép kiểu, hàm **settype** cũng có thể được sử dụng để chuyển đổi từ kiểu dữ liệu này sang một kiểu dữ liệu khác. Ví dụ chúng ta khai báo như sau :

```
<HTML>
  <HEAD>
    <TITLE>Welcome to PHP World</TITLE>
  </HEAD>
<BODY>
  <h4>Thay doi kieu du lieu(Settype)</h4>
  <?php
    $var="99-THD";
    $check=true;
    echo $var."<br>";
    echo $check."<br>";
    settype($var,"integer");
    echo $var."<br>";
    settype($check,"string");
    echo $check;
  ?>
</BODY>
</HTML>
```

2.6.2: Kiểm tra kiểu dữ liệu của biến

- Để kiểm tra kiểu dữ liệu của biến, chúng ta sử dụng những hàm sau (kết quả trả về là true = 1):

- **is_int** Kiểm tra biến có phải kiểu **Integer** không.
- **is_array** Kiểm tra biến có phải kiểu **Mảng** không.
- **is_bool** Kiểm tra biến có phải kiểu **Boolean** không.
- **is_double** Kiểm tra biến có phải kiểu **Double** không.
- Ngoài ra còn có những hàm khác như : **is_callable**, **is_float**, **is_integer**, **is_long**, **is_null**, **is_numeric**, **is_object**, **is_real**, **is_string**

Ví dụ :

```
<HTML>
  <HEAD>
    <TITLE>::Welcome to PHP</TITLE>
  </HEAD>
  <BODY>
    <h4>Kiem tra kieu du lieu cua bien</h4>
    <?php
      $record=5;
      $myarr = array("first", "last", "company");
      echo is_bool($record);
      echo "<br>";
      echo is_array($myarr);
    ?>
  </BODY>
</HTML>
```

2.6.3: Thay đổi kiểu dữ liệu biến

- Khi khai báo biến với một giá trị khởi tạo cho biến đó. Sau đó bạn muốn sử dụng giá trị của biến đó thành tên biến và có giá trị chính là giá trị của biến ban đầu mà ta khai báo, thì sử dụng cặp dấu **\$\$**. Chúng ta xem xét ví dụ sau :

```
<HTML>
  <HEAD>
    <TITLE>::Welcome to PHP</TITLE>
  </HEAD>
  <BODY>
    <h4>Thay doi kieu du lieu bien</h4>
    <?php
      $obj="math";
      echo $obj;
      echo "<br>";
      $$obj=10;
      echo $math;
    ?>
  </BODY>
</HTML>
```

Ở ví dụ trên, biến **\$obj** có giá trị là *math*, sau đó chúng ta muốn đổi *math* thành tên biến thì sử dụng **\$\$obj**.

2.6.4: Kiểu Array

Mảng 1 chiều

- Kiểu Array là một mảng các giá trị do người dùng định nghĩa. Thứ tự index trong một Array luôn bắt đầu từ 0 và tăng dần lên 1, cú pháp của kiểu Array được khai báo như sau :

```
//Mang gom cac kieu chuoi
$myarrs=array("SV001", "SV002", "SV003");
```

Hay chúng ta có thể khai báo như sau :

```
$myarrs[] = array(3);
$myarrs[0] = " SV001";
$myarrs[1] = " SV002";
$myarrs[2] = " SV003";
```

Ở ví dụ dưới đây, chúng ta sẽ minh họa cho việc sử dụng kiểu dữ liệu Array.

```
<HTML>
  <HEAD>
    <TITLE>::Welcome to PHP</TITLE>
  </HEAD>
  <BODY>
    <h4>Mang 1 chieu</h4>
    <?php
      $room[] = array(4);
      $room[0] = "Room001";
      $room[1] = " Room002";
      $room[2] = " Room003";
      $room[3] = " Room004";
      echo $room[0];
      echo $room[1];
      echo $room[2];
      echo $room[3];
      echo "<br>";
      $language = array("English", "VietNam", "Japan");
      echo $language[2];
    ?>
  </BODY>
</HTML>
```

Mảng 2 chiều

- Về cơ bản mảng 2 chiều trong php tương tự với trong C nhưng có một điểm khác đó là trong PHP không cần phải khai bao kiểu dữ liệu của mảng . Cú pháp khai báo mảng 2 chiều được minh họa trong ví dụ sau đây :

```
<HTML>
  <HEAD>
    <TITLE>Welcome to PHP</TITLE>
  </HEAD>
```

```

<BODY>
    <h4>Mang 2 chieu</h4>
    <?php
        $student[][]=array(2,3);
        $student[0][0]="SV001";
        $student[0][1]="SV002";
        $student[0][2]="SV003";
        $student[1][0]="SV004";
        $student[1][1]="SV005";
        $student[1][2]="SV006";
        echo $student[0][2];
    ?>
</BODY>
</HTML>

```

Ghi chú : ở ví dụ trên chúng ta khai báo một mảng 2 chiều có tên là **student** gồm 2 dòng và 3 cột, tương ứng sẽ có 6 phần tử trong mảng 2 chiều đó .

2.6.5 Kiểu đối tượng

- Trong PHP để khai báo một đối tượng, chúng ta sử dụng khái niệm Class như trong ngôn ngữ lập trình C hay Java. Ngoài ra các phương thức trong PHP còn được biết đến như một hàm(Function). Nếu một hàm trong Class bất kì có cùng tên với tên Class thì hàm đó cũng được gọi là hàm khởi tạo(Constructor). Khi ta khai báo 1 Class và Constructor. Thì khi khởi tạo một đối tượng, tự động Constructor được gọi. Sau đó gọi hàm trong Class đó (*nội dung chi tiết liên quan đến hướng đối tượng được trình bày trong chương 9*)

```

<HTML>
    <HEAD>
        <TITLE>Welcome to PHP World</TITLE>
    </HEAD>
    <BODY>
        <h4>Kieu doi tuong</h4>
        <?php
            class clsA
            {
                function clsA() {
                    echo "Day la ham khoi tao(Constructor)";
                    echo "<br />\n";
                }
                function B() {
                    echo "Learning PHP very well";
                    echo "<br />\n";
                }
            }
            //Khoi tao mot doi tuong thuoc clsA
            $b = new clsA();
            echo "<br>";
            //Goi phuong thuc B().
            $b->B();
        </?php>
    </BODY>
</HTML>

```

```
?>
</BODY>
</HTML>
```

2.6.6 Tầm vực của biến

- Tầm vực của biến phụ thuộc vào nơi khai báo biến, nếu biến khai báo bên ngoài của một hàm hoặc lớp thì sẽ có tầm vực trong cả trang PHP. Ngược lại trong trường hợp biến khai báo trong hàm hoặc lớp thì chỉ có hiệu lực trong hàm hoặc lớp đó.

Ví dụ: chúng ta có biến **\$var** khai báo bên ngoài hàm nhưng khi vào trong hàm thì biến **\$var** được khai báo lại, thì biến **\$var** trong hàm có tầm vực nhỏ hơn khác với biến **\$var** ngoài hàm có tầm vực lớn hơn. Tương tự như vậy, khi biến **\$k** khai báo trong hàm thì chỉ có tầm vực bên trong hàm cho dù chúng được khai báo lại bên ngoài như ví dụ sau đây :

```
<HTML>
  <HEAD>
    <TITLE>::Welcome to PHP</TITLE>
  </HEAD>
  <BODY>
    <h4>Tam vuc cua bien</h4>
    <?php
      //bien toan cuc
      $var = 100;
      $k=1000;
      function Test() {
        //bien cuc bo trong ham
        $k=10;
        $var=10;
        echo "var=$var";echo"<br>";
        echo "k=$k";echo"<br>";
        /* reference to local scope variable */
      }
      Test();
      echo "var=$var"; echo"<br>";
      echo "k=$k"; echo"<br>";
    ?>
  </BODY>
</HTML>
```

- Ngoài ra từ khóa **global** được sử dụng khi chúng ta muốn biến toàn cục được dùng bên trong hàm.

Ví dụ :

```
<HTML>
  <HEAD>
    <TITLE>Welcome to PHP World</TITLE>
  </HEAD>
  <BODY>
```

```

<h4>Sử dụng từ khóa Global</h4>
<?php
    // biến toàn cục
    $var = 100;
    function check() {
        global $var;
        $k=10;
        $var+=10;
        echo "<br>var=$var";
        echo "<br>k=$k";
    }
    check();
    echo "<br>var=$var";
    $k=1000;
    echo "<br>k=$k";
?
</BODY>
</HTML>

```

Ghi chú: khai báo biến \$var bên ngoài hàm check(), sau đó bên trong hàm chúng ta sử dụng từ khóa global cho biến \$var, khi đó biến \$var sẽ được sử dụng và giá trị đó có hiệu lực sau khi ra khỏi hàm.

2.7: Hằng trong PHP

2.7.1: Khai báo và sử dụng hằng

- Hằng là một giá trị không bao giờ bị thay đổi kể từ khi chúng được khai báo, bạn có thể sử dụng từ khóa **define** để khai báo hằng, xem ví dụ sau :

```

<HTML>
    <HEAD>
        <TITLE>Welcome to PHP World</TITLE>
    </HEAD>
    <BODY>
        <h4>Hằng trong PHP</h4>
        <?php
            define("day",24);
            function Test() {
                echo "<br>day=".day;
                echo "<br>day=".constant("day");
            }
            Test();
            echo "<br>day=".day;
            echo "<br>day=".constant("day");
?
        </BODY>
</HTML>

```

2.7.2: Kiểm tra hằng

- Khi sử dụng hằng, mà chúng ta không biết hằng đó có tồn tại hay chưa thì bạn sử dụng hàm **defined**. Ví dụ :

```
<HTML>
```

```

<HEAD>
    <TITLE>Welcome to PHP World</TITLE>
</HEAD>
<BODY>
    <h4>Kiem tra hang</h4>
    <?php
        define("week",7);
        //define("sec",60);
        function Test()
        {
            if(define("week"))
            {
                echo "<br>week=".week;
            }
            else
            {
                echo "<br>week is not defined";
            }
            if(define("sec"))
            {
                echo "<br>sec=".sec;
            }
            else
            {
                echo "<br>sec not defined";
            }
        }
        Test();
    ?>
</BODY>
</HTML>

```

Câu hỏi ôn tập:

Câu 1: In ra màn hình danh sách thông tin của sinh viên của các lớp. Biết có 2 lớp là a và b, trong mỗi lớp có 5 sinh viên, mỗi sinh viên có 1 mã sinh viên. (thông tin sinh viên ở đây là chỉ in mã sinh viên).

Câu 2: Viết ứng dụng máy tính cá nhân có chức năng thực hiện các phép tính (cộng, trừ, nhân, chia) đơn giản 2 chữ số. với các yêu cầu cơ bản sau:

- Tạo Form nhập dữ liệu
- Trong Form có 2 trường textbox để nhập các số thứ nhất và số thứ hai
- Trong Form có hộp selectbox để lựa chọn các phép toán (+, -, x, /)
- Trong Form có nút kết quả để khi bấm kết quả sẽ cho in ra kết quả ngay tại trang tính toán
- Nếu các số cần thực hiện phép tính chưa được nhập thì khi bấm kết quả sẽ thông báo “Các trường nhập số không được để trống”.
- Nếu nội dung nhập vào không phải là số thì sẽ thông báo “Các trường nhập vào phải là số”.

Để hoàn thành một ứng dụng đạt yêu cầu, cần có một số gợi ý như sau:

- Để kiểm tra dữ liệu nhập vào có phải là một chuỗi rỗng hay không, chúng ta sử dụng `if(bien_kiem_tra == "")`
- Để kiểm tra dữ liệu nhập vào có phải là một số hay không sử dụng hàm `isnumeric(bien_kiem_tra)`. Kết quả trả về TRUE nếu là số và FALSE nếu không phải số.

CHƯƠNG 3

3.1: Giới thiệu toán tử

- **Toán tử số học:** là phép toán cộng trừ nhân chia trong số học, ngoài ra còn có thêm phép chia lấy dư (%), phép tăng thêm 1, giảm đi 1.

Toán tử	Miêu tả	Ví dụ	Kết quả
+	Cộng	$x = 2$ $x+2$	4
-	Trừ	$x = 2$ $5-x$	3
*	Nhân	$x = 4$ $x*5$	20
/	Chia	$15/5$ $5/2$	3 2.5
%	Chia lấy dư	$5\%2$ $10\%8$ $10\%2$	1 2 0
++	Cộng thêm 1	$x = 5$ $x++$	$x = 6$
-	Giảm đi 1	$x = 5$ $x-$	$x = 4$

- **Toán tử gán:** là phép gán trong PHP, sử dụng kí tự “=” kết hợp các phép toán khác để tính toán và gán giá trị của toán hạng bên phải cho toán hạng bên trái.

Toán tử	Ví dụ	Giải thích
=	$x = y$	$x = y$
+=	$x += y$	$x = x + y$
-=	$x -= y$	$x = x - y$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
.=	$x .= y$	$x = x . y$
%=	$x \%= y$	$x = x \% y$

- **Toán tử so sánh:** là phép so sánh giữa 2 toán hạng với nhau, có thể bằng, khác, lớn hơn, nhỏ hơn,

Toán tử	Miêu tả	Ví dụ
==	So sánh bằng	$5 == 8$ trả về false
!=	Khác	$5 != 8$ trả về true
<>	Khác	$5 <> 8$ trả về true
>	Lớn hơn	$5 > 8$ trả về false
<	Nhỏ hơn	$5 < 8$ trả về true
>=	Lớn hơn hoặc bằng	$5 >= 8$ trả về false
<=	Nhỏ hơn hoặc bằng	$5 <= 8$ trả về true

- **Toán tử logic:** là phép toán kiểm tra logic (true/false) của 1 điều kiện cho trước.

Toán tử	Miêu tả	Ví dụ
&&, AND	Và	$x = 6$ $y = 3 (x < 10 \&\& y > 1)$ Trả về true

,OR	Hoặc	x = 6 y = 3 (x == 5 y == 5) Trả về false
!	Phủ định	x = 6 y = 3 !(x == y) Trả về true

Ví dụ:

```

1  <?php
2  //Toán tử logic trong PHP
3  $x = 6;
4  $y = 3;
5  if($x > 7 AND $y > 4)
6      echo "Thỏa mãn điều kiện đưa ra.";
7  else
8      echo "Không thỏa mãn điều kiện đưa ra.";
9  //Kết quả sẽ in ra màn hình là: Không thỏa mãn điều kiện đưa ra.
10 ?>

```

3.2: Phát biểu có điều kiện

- **Phát biểu if (nếu):** là phát biểu kiểm tra điều kiện nào đó. **Nếu** thỏa mãn và đúng với điều kiện **thì** sẽ 1 đoạn lệnh nào đó. **Nếu** không thỏa mãn điều kiện **thì** sẽ không thực thi đoạn lệnh bên trong.

Syntax (cách viết):

```

if ( điều kiện )
{
    // TODO
}

```

Ví dụ:

```

1  <?php
2  /*
3  ví dụ 1: có biến x = 13
4  yêu cầu: in ra màn hình câu: "x thỏa mãn điều kiện"
5  Điều kiện: x > 10
6  */
7  //khai báo x = 13
8  $x = 13;
9  //kiểm tra điều kiện dùng phát biểu If
10 if($x > 10)
11     echo "x thỏa mãn điều kiện.";
12 //kết quả in ra màn hình câu: "x thỏa mãn điều kiện."
13 ?>

```

```

1  <?php
2  /*
3   ví dụ 2: có biến x = 5
4   yêu cầu: in ra màn hình câu: "x thỏa mãn điều kiện"
5   Điều kiện: x > 10
6  */
7  //khai báo x = 5
8  $x = 5;
9  //kiểm tra điều kiện dùng phát biểu If
10 if($x > 10)
11     echo "x thỏa mãn điều kiện.";
12 //kết quả là sẽ không có câu gì được in ra màn hình.
13 ?>

```

- **Phát biểu if - else:** cũng giống như phát biểu If. Phát biểu này cho phép ta có thể kiểm tra nhiều điều kiện khác nhau trong trường các phát biểu If phía trên không thỏa mãn điều kiện.

Syntax (cách viết):

```

if( điều kiện thứ 1 ) {
    //TODO
} else if( điều kiện thứ 2 ) {
    //TODO
} else {
    //TODO
}

```

Ví dụ:

```

1  <?php
2  /*
3   ví dụ: có 2 biến x = 2 và y = 5
4   yêu cầu: in ra màn hình câu: "x = 5" nếu x = 5
5   in ra màn hình câu: "y = 5" nếu y = 5
6   còn không thỏa mãn 2 điều kiện trên thì in câu: "không thỏa mãn điều kiện."
7  */
8  //khai báo x = 2 và y = 5
9  $x = 2;
10 $y = 5;
11 //kiểm tra điều kiện dùng phát biểu if-else
12 if($x == 5)
13     echo "x = 5";
14 else if($y == 5)
15     echo "y = 5";
16 else
17     echo "không thỏa mãn điều kiện.";
18 //kết quả in ra màn hình là: y = 5.
19 /*
20 Giải thích: vì $x = 2
21         => không thỏa điều kiện x = 5
22         vì $y = 5
23         => thỏa điều kiện nên in ra y = 5
24 sau đó sẽ không tiếp tục kiểm tra nữa sau khi đã thỏa mãn điều kiện.
25 */
26 ?>

```

3.3 Switch: cũng như giống với phát biểu If. Trong nhiều trường hợp, bạn có thể so sánh chung 1 biến với nhiều giá trị khác nhau và thực thi cái phần code khác nhau phụ thuộc trên cái giá trị mà được chọn.

Syntax (cách viết):

```
switch( biến ) {
    case giá trị 1: {
        //code thực thi
        break;
    }
    case giá trị 2: {
        //code thực thi
        break;
    }
    default :{
        //code thực thi nếu không thỏa các giá trị trên
    }
}
```

```
<?php
if ($i == 0) {
    echo "i equals 0";
} elseif ($i == 1) {
    echo "i equals 1";
} elseif ($i == 2) {
    echo "i equals 2";
}

switch ($i) {
    case 0:
        echo "i equals 0";
        break;
    case 1:
        echo "i equals 1";
        break;
    case 2:
        echo "i equals 2";
        break;
}
?>
```

```
<?php
switch ($i):
    case 0:
        echo "i equals 0";
        break;
    case 1:
        echo "i equals 1";
        break;
    case 2:
        echo "i equals 2";
        break;
    default:
        echo "i is not equal to 0, 1 or 2";
endswitch;
?>
```

3.4 While: cũng như mệnh đề if thì bắt buộc thỏa điều kiện trong while thì mới vào được vòng while để thực thi các lệnh.

Syntax :

```
While( Điều kiện ) {
    //đoạn code sẽ được thực thi.
}
```

```
1  <?php
2  $i = 0;
3  while($i < 10)
4  {
5      echo $i."<br />";
6      $i++;
7  }
?>
```

3.5 For: cách hoạt động của vòng lặp for cũng giống với while nhưng chỉ khác ở cách viết syntax.

Syntax:

```
For( điều kiện 1; điều kiện 2; điều kiện 3)
{
    //đoạn code sẽ được thực thi.
}
```

VD:

```
1 <?php
2 for($i = 0; $i < 10; $i++)
3 {
4     echo $i."<br />";
5 }
6 ?>
```

3.6 Do – While: giống với while nhưng do while thực thi đoạn code 1 lần rồi sau đó mới xét tới điều kiện trong while().

Syntax:

```
do {
    //đoạn code thực thi
}while(điều kiện);
```

***Chú ý:** sau while có dấu chấm “;”

VD:

```
1 <?php
2 $i = 0;
3 do {
4     echo $i."<br />";
5 }while($i < 10);
6 ?>
```

Câu hỏi ôn tập:

Câu 1: In ra màn hình danh sách các số nguyên tố từ 1 đến 30.

Câu 2: Tạo 1 form đăng ký thành viên với các thông tin sau:

- Tên Đăng Nhập
- Họ Tên
- Email
- Điện Thoại
- Năm Sinh (combobox cho chọn từ năm 1950-2000)

Dữ liệu được ràng buộc bằng javascript, sau khi nhập đầy đủ thì click nút đăng ký thì hệ thống thông báo đăng ký thành công.

CHƯƠNG 4

4.1 Biến Form:

- Biến Form trong PHP được hiểu như là một loại biến. Thay vì, nếu muốn sử dụng một biến nào đó thì chúng ta phải khởi tạo còn biến Form chính là thuộc tính **name** của bất kì một thẻ nhập nào trong Form hoặc tham số trên querystring.

4.1.1 Form submit với phương thức POST:

- Khi từ một trang người dùng **submit** đến 1 trang khác (*phương thức action của thẻ form*), nếu khai báo thuộc tính name của bất kì một thẻ nào trong form là student thì biến form đó được định nghĩa là **\$student**.

Ví dụ:

```
...
<form action=example.php method=post>
    <tr>
        <td>Name :</td>
        <td><input type=text name=studentName></td>
    </tr>
    <tr>
        <td>Gender :</td>
        <td>
            <input type=radio value=M name=gender> Male
            <input type=radio value=F name=gender> Female
        </td>
    </tr>
    <tr>
        <td>&nbsp</td>
        <td><input type=submit value=Submit></td>
    </tr>
</form>
...
```

- Khi người dùng nhập nội dung vào textbox của Student Name và chọn Giới Tính một trong hai radio Male hoặc Female, sau đó nếu chúng ta nhấp button Submit thì trang **resulExample.php** sẽ được gọi, trong trang này các giá trị nhập từ trang Example.php sẽ được lấy ra bằng cách sử dụng biến form.

Ví dụ :

```
<HTML>
    <HEAD>
        <TITLE>Welcome to PHP World</TITLE>
    </HEAD>
    <BODY>
        <h4>Biến Form</h4>
        <table>
            <tr>
                <td>Student Name :</td>
                <td> <?=$studentName?></td>
            </tr>
            <tr>
```

```

        <td>Gender :</td>
        <td><?=$gender?> </td>
    </tr>
</table>
</BODY>
</HTML>

```

- Ở ví dụ trên, chúng ta dễ dàng nhận thấy 2 biến \$studentName và \$gender là tên của hai thẻ input trong form của trang Example.php. Trong 2 Ví dụ phần này chúng ta sử dụng phương thức POST để truyền dữ liệu giữa 2 trang Example.php và resultExample.php (**method = post**).

4.1.2 Form submit với phương thức GET:

- Khác với phương thức POST không truyền biến form lên QueryString, phương thức GET có thể lấy giá trị của các tham số trên chuỗi QueryString bằng biến form.

```

...
<form action=Example.php method=get>
    <tr>
        <td>Khoa-Trung Tam :</td>
        <td>
            <select name=khoa>
                <option value=CNTT>Cong Nghe Thong Tin</option>
                <option value=CDT>Co Dien Tu</option>
                <option value=DL>Dien Lanh</option>
            </select>
        </td>
    </tr>
    <tr>
        <td>Country :</td>
        <td>
            <select name=country multiple>
                <option value=USA>United State</option>
                <option value=VN>Viet Nam</option>
                <option value=CAN>Canada</option>
                <option value=ITI>Italia</option>
            </select>
        </td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><input type=submit value=Submit></td>
    </tr>
</form>
...

```

Câu hỏi ôn tập:

Câu 1: Viết trang Information.php và tạo Form thông tin cho phép người dùng nhập vào thông tin cá nhân của mình với các thông tin như sau:

- họ tên : text
- giới tính : radio
- ngày sinh : text
- địa chỉ : textarea
- email : text

Và button Submit để chuyển sang trang PrintInfo.php với phương thức tùy ý.
PrintInfo.php phải hiển thị đầy đủ thông tin mà người dùng đã nhập vào.

Câu 2: Viết trang đăng nhập với các thông tin email, mật khẩu. Chương trình tự động validate dữ liệu sử dụng javascript, nếu nhập thông tin đúng email là:
admin@hungvuongtech.edu.vn với mật khẩu là hungvuong thì thông báo đăng nhập thành công, các trường hợp khác thông báo thất bại.

CHƯƠNG 5

5.1 Xử lý dữ liệu nhập trên form và thông báo lỗi với javascript.

B1: Xây dựng 1 form bất kỳ. Ở đây chúng ta sẽ xây dựng form đăng nhập để kiểm tra dữ liệu nhập vào có phù hợp hay không?

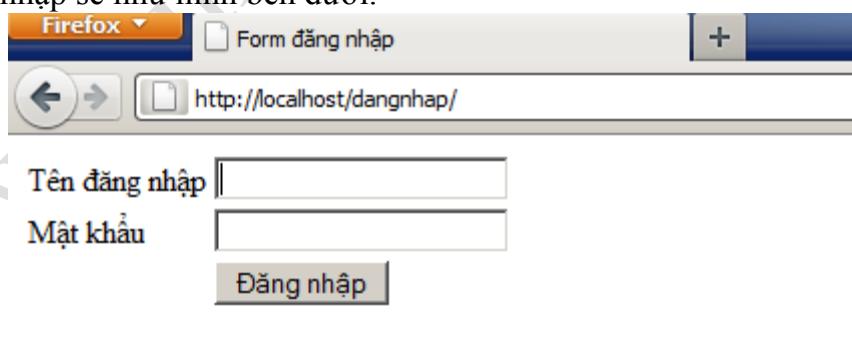
```

31 <body>
32 <form name="formDangNhap" action="home.php" method="post" onsubmit="return check();">
33 <table>
34   <tr>
35     <td>Tên đăng nhập</td>
36     <td><input name="tenDangNhap" id="tenDangNhap" type="text" /></td>
37   </tr>
38   <tr>
39     <td>Mật khẩu</td>
40     <td><input name="matKhau" id="matKhau" type="password" /></td>
41   </tr>
42   <tr>
43     <td></td>
44     <td><input name="btnDangNhap" id="btnDangNhap" type="submit" value="Đăng nhập" /></td>
45   </tr>
46 </table>
47 </form>
48 </body>

```

Giải thích: trong đoạn code trên cho phép người dùng đăng nhập vào hệ thống website khi nhập đúng tên đăng nhập và mật khẩu. Ở đây gọi hàm onsubmit với điều kiện trả về là phải thỏa mãn đoạn code trong hàm check() mà chúng ta sẽ viết tiếp theo trong phần sau.

Form đăng nhập sẽ như hình bên dưới.



Phần javascript chúng ta khai báo sai khi đóng thẻ title. Xem hình bên dưới.

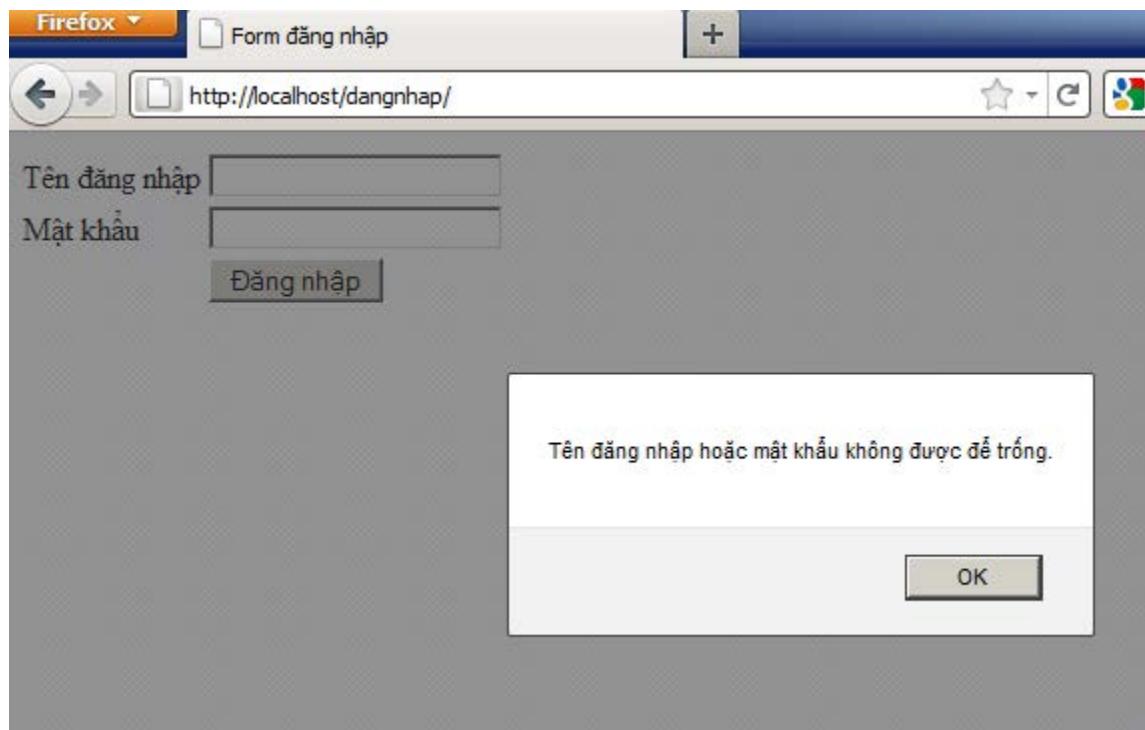
```

5   <title>Form đăng nhập</title>
6   <script type="text/javascript">
7   function check()
8   {
9     var tenDangNhap = document.getElementById("tenDangNhap");
10    var matKhau = document.getElementById("matKhau");
11    if(tenDangNhap.value == "" || matKhau.value == "")
12    {
13      alert("Tên đăng nhập hoặc mật khẩu không được để trống.");
14      return false;
15    }
16    else
17    {
18      if(tenDangNhap.value == "admin" && matKhau.value == "123")
19      {
20        alert("Đăng nhập thành công.");
21        return true;
22      }
23      else
24      {
25        alert("Đăng nhập thất bại.");
26        return false;
27      }
28    }
29  </script>

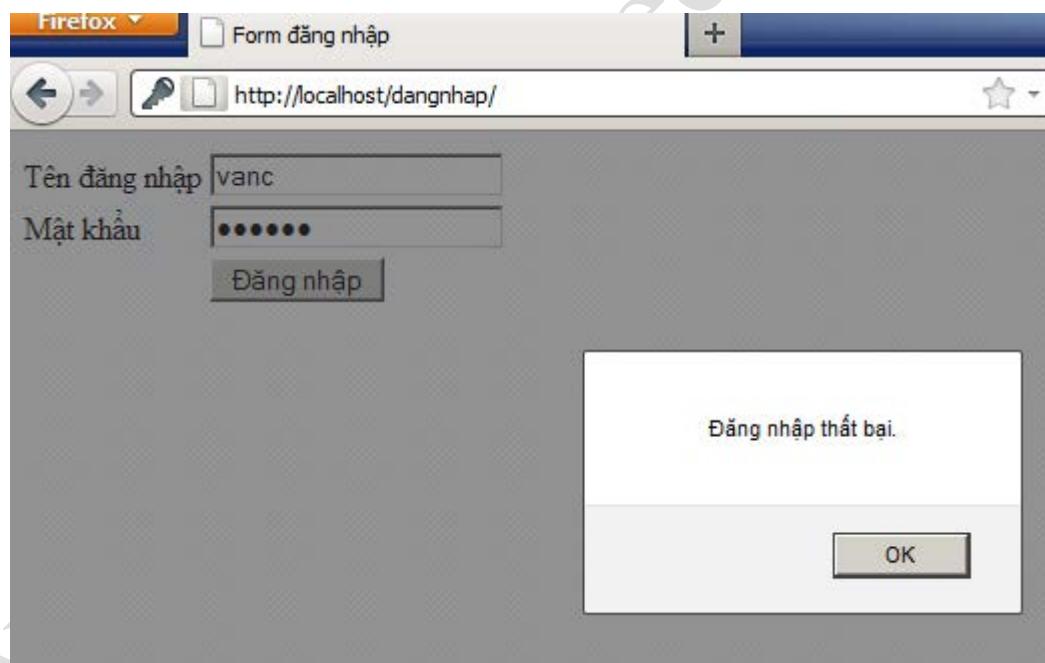
```

Giải thích:

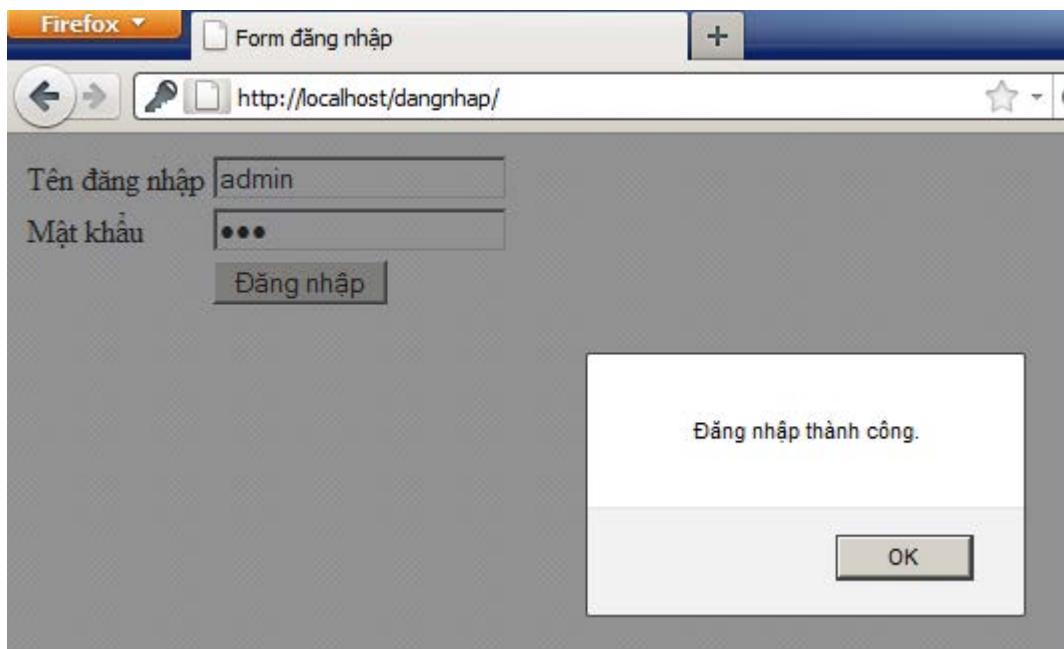
- + Từ khóa var là khai báo 1 biến mới.
- + Để lấy đối tượng HTML trong javascript, chúng ta dùng cú pháp:
`document.getElementById("tenDangNhap");`
- + Cú pháp này nhận vào id của đối tượng HTML mà chúng ta đã tạo id cho đối tượng HTML trong đoạn code tạo form đăng nhập ở phía trên.
(Chú ý: Chúng ta nên khai báo thuộc tính name và id cho đối tượng HTML giống nhau để tránh trường nhầm lẫn)
- + Sau đó để lấy được giá trị nhập vào từ đối tượng đó (tùy thuộc vào đối tượng mà thuộc tính lấy sẽ khác nhau). Ví dụ ở đây là đối tượng input với type là text nên để lấy được giá trị text thì chúng ta chỉ cần → `.value` như đoạn code javascript phía trên (VD: `tenDangNhap.value`)
- + Cú pháp return false hoặc true như sau:
 - Nếu return false tức điều kiện không thỏa mãn nên action form đăng nhập sẽ không chuyển qua trang home.php như code ví dụ.
 - Nếu return true đã thỏa mãn điều kiện nên lúc đó khi nhấn submit thì sẽ chuyển qua trang home.php.



Thông báo lỗi “*Tên đăng nhập hoặc mật khẩu không được để trống.*” sẽ xuất tên đăng nhập và mật khẩu đều để trống.

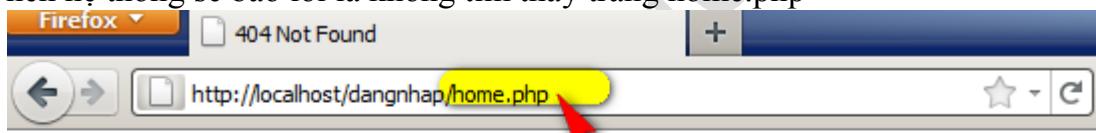


Còn đây là trường hợp lỗi đăng nhập thất bại khi người dùng cung cấp thông tin tên đăng nhập hoặc mật khẩu không chính xác.



Khi người dùng cung cấp thông tin chính xác (*tên đăng nhập là admin, mật khẩu là 123*) lúc này hệ thống kiểm tra thấy chính xác điều kiện thì xuất thông báo như hình trên.

Sau đó sẽ chuyển trang qua trang home.php. Vì trang home.php chúng ta chưa viết nên hệ thống sẽ báo lỗi là không tìm thấy trang home.php



Sau khi đăng nhập thành công thì sẽ chuyển trang theo action của form

The requested URL /dangnhap/home.php was not found on this server.

5.2 Regular Expression

+Sơ bộ về Regular expression (*thường gọi tắt là RegEx*).

+RegEx là gì? RegEx là một biểu thức thường dùng để mô tả một mẫu câu chuỗi (*còn gọi là string pattern*).

+Mẫu chuỗi là gì? Mẫu chuỗi là các chuỗi văn bản tuân thủ theo 1 quy luật sắp xếp mẫu nào đó. Ví dụ dễ hiểu nhất là email, email được đặt theo quy định là gồm 2 thành phần: phần username và phần domain, 2 phần này được quy định là cách nhau bởi dấu @. Phần domain thì có quy luật riêng theo cách đặt tên của domain.

+RegEx được dùng để làm gì và có tác dụng gì? RegEx được dùng để thiết lập để tạo ra 1 quy luật nào đó. RegEx có tác dụng là yêu cầu chương trình tìm kiếm theo quy luật vừa được thiết lập phía trên. (*Cụ thể là ví dụ về email đã được nhắc đến phía trên*).

+Cú pháp RegEx trong javascript: có 2 cách để tạo regular expression trong javascript:

- **Cách 1:** var reExample = /pattern/thamso;
- **Cách 2:** cách này tạo regular expression theo cấu trúc constructor RegExp() :
var reExample = new RegExp("pattern","thamso");

thamso : tham số này xác định cách thức tìm kiếm mẫu chuỗi và chứa các giá trị sau đây:

- o **g** : (“global”) lặp lại quá trình tìm kiếm cho đến khi kết thúc chuỗi.
- o **i** : so sánh không phân biệt hoa thường.
- o **m** : so sánh ở chế độ đa dòng.

+Các ký tự dùng trong mẫu trong pattern:

- \ : loại bỏ các ký tự đặc biệt.
- {**n**} : lặp 1 ký tự hoặc 1 biểu thức con trước đó n lần
- {**n,m**} : lặp 1 ký tự hoặc 1 biểu thức con trước đó từ n đến m lần
- {**n,j**} : lặp 1 ký tự hoặc 1 biểu thức con trước đó lớn hơn hoặc bằng n lần (**pattern**): tìm và bắt lấy /pattern/
- (?:**pattern**): tìm nhưng không bắt lấy /pattern/
- | : tìm thay phiên
- [**characters**]: tìm trong chuỗi chứa bất kỳ các ký tự characters
- [^**characters**]: tìm trong chuỗi không chứa bất kỳ các ký tự trong character.
- [a-z]: tìm 1 ký tự bất kỳ nằm trong tập hợp a đến z.
- * lặp 1 ký tự hoặc 1 biểu thức con trước đó lớn hơn bằng 0 lần
- + lặp 1 ký tự hoặc 1 biểu thức con trước đó lớn hơn bằng 1 lần
- ? lặp 1 ký tự hoặc 1 biểu thức con trước đó 0 hoặc 1 lần
- ^ : trả về chuỗi kết quả trong trường hợp chuỗi này nằm ở vị trí đầu của chuỗi gốc
- \$: trả về chuỗi kết quả trong trường hợp chuỗi này nằm ở vị trí cuối của chuỗi gốc.
- . : tìm bất cứ ký tự nào trừ ký tự xuống dòng
- \b : trả về chuỗi kết quả trong trường hợp chuỗi này nằm ở vị trí đầu của 1 từ trong chuỗi gốc. Nếu cần so sánh ở vị trí cuối từ hãy đặt \b ở vị trí cuối từ.
- \B : trả về chuỗi kết quả trong trường hợp chuỗi này không nằm ở vị trí đầu của 1 từ trong chuỗi gốc. (Tùy thuộc vào vị trí đặt \B ở đầu hay cuối của 1 từ.)
- ?: : trả về chuỗi kết quả nếu theo sau đó là chuỗi nào đó được chỉ định trước
- ?? : trả về chuỗi kết quả nếu theo sau đó không phải là chuỗi nào đó được chỉ định trước
- \w : tìm 1 ký tự dạng từ a-Z, 0-9 và dấu gạch dưới.
- \W : ngược lại với \w
- \d : tìm 1 ký tự thuộc tập ký tự từ 0-9
- \D : ngược lại với \d
- \s : tìm ký tự cách (khoảng trắng)
- \S : ngược lại với \s

Gộp nhóm các biểu thức:

- o () : tìm kiếm 1 nhóm các ký tự bên trong cặp dấu ngoặc và lưu vào chuỗi kết quả.
- o (?:) : tìm kiếm 1 nhóm các ký tự bên trong cặp dấu ngoặc nhưng không lưu vào chuỗi kết quả.
- o | : phép toán hoặc dùng để kết hợp các mệnh đề với nhau vào chung 1 biểu thức.

+Các phương thức có hỗ trợ RegEx trong javascript: ở đây ví dụ chúng ta đã có 1 chuỗi và đặt là var str = “chao buoi sang ban An”;

- o **str.search(/pattern/)**: phương thức này trả về vị trí đầu tiên tìm được của chuỗi này trong chuỗi pattern.

Vd: document.write(str.search(/buoi/));

- ⇒ Kết quả trả về sẽ là: 5
- o **str.replace(/pattern/,"chuỗi thay thế"):** phương thức này tìm và thay thế pattern tìm được trong chuỗi str và thay thế thành chuỗi thay thế.
Vd: document.write(str.replace(/sang/,"toi"));
⇒ Kết quả trả về sẽ là: chao buoi toi ban An
- o **str.split(pattern):** phương thức này sẽ tìm pattern đó chuỗi và sẽ dùng pattern đó để cắt chuỗi thành mảng.
Vd: document.write(str.split(/\s/g));
⇒ Kết quả trả về sẽ là: chao,buoi,sang,ban,An
- o **str.match(pattern):** trả về 1 mảng các thông tin nếu tìm thấy và false nếu không tìm thấy /pattern/ trong chuỗi str.
Vd: var str = “Watch out for the rock!”;
document.write(str.match(r?or?/g));
⇒ Kết quả là: o,or,ro
- o **/pattern/.test(str):** trả về true nếu pattern tìm thấy trong str và ngược lại trả về false.
Vd: var ketqua = /example/.test(“My Example”);
⇒ Kết quả trả về là: false vì javascript phân biệt hoa thường.
- o **/pattern/.exec(str):** trả về array không nếu /pattern/ tìm thấy trong chuỗi str và ngược lại.
Vd: document.write(/s(amp)le/i.exec(“Sample text”));
⇒ Kết quả trả về là: Sample,amp

Câu hỏi ôn tập:

- Viết trang Register.php với 2 thông tin như sau:

- Tên người dùng : text
- Mật khẩu : password
- Button Tạo mới

Khi click button sẽ chuyển đến trang xử lý Process.php.

Chú ý: kiểm tra như sau trước khi chuyển đến Process.php

- Các fiel không được để rỗng.
- Mật khẩu nhập vào phải ít nhất 5 kí tự và phải bao gồm chữ cái và số.

CHƯƠNG 6

6.1 Đối tượng Session

Tổng quan về session: Một cách khác quản lý người sử dụng là session. Session được hiểu là khoảng thời gian người sử dụng giao tiếp với 1 ứng dụng. Một session được bắt đầu khi người sử dụng truy cập vào ứng dụng lần đầu tiên, và kết thúc khi người sử dụng thoát khỏi ứng dụng. Mỗi session sẽ có được cấp một định danh (ID) khác nhau và nội dung được lưu trong thư mục thiết lập trong file php.ini (tham số session.save_path).

Khởi tạo session: việc khởi tạo session thì bắt buộc trang nào chúng ta đều phải start session lên thì mới có thể sử dụng session. Để khởi tạo 1 session, ta sử dụng cú pháp: session_start(); Đoạn code này phải nằm trên các kịch bản HTML hoặc những lệnh echo, printf.

Thiết lập giá trị cho session: ngoài việc cho phép bắt đầu thực thi session, ta phải đăng ký 1 giá trị cho session. Ta sử dụng cú pháp: session_register("Tên session")

VD: Chúng ta sẽ bắt đầu khởi tạo session và đăng ký 1 session có tên là userName

```

1  <?php
2      session_start();
3      session_register("userName");
4
5
6  ?>

```

Gán giá trị vào session và lấy giá trị từ session:

+Việc gán giá trị vào session được thực hiện như ví dụ sau:

```

1  <?php
2      session_start();
3      session_register("userName");
4      $_SESSION["userName"] = "Nguyen Van C";
5
6  ?>

```

+Việc lấy giá trị từ session như sau:

```

1  <?php
2      session_start();
3      echo $_SESSION["userName"];
4
5  ?>

```

+**Hủy session:** việc hủy session có 2 cách hủy session:

Hủy 1 session, cú pháp là: unset(\$_SESSION["tên session"]);

Hủy toàn bộ session, cú pháp là: session_destroy();

6.2 Cookie

+Tổng quan về cookie:

- Cookie là 1 đoạn dữ liệu được ghi vào đĩa cứng hoặc bộ nhớ của máy người sử dụng. Nó được trình duyệt gửi ngược lên lại server mỗi khi browser tải 1 trang web từ server.
- Những thông tin được lưu trữ trong cookie hoàn toàn phụ thuộc vào website trên server. Mỗi website có thể lưu trữ những thông tin khác nhau trong cookie, ví dụ thời điểm lần cuối ta ghé thăm website, đánh dấu ta đã login hay chưa, v.v... Cookie được tạo ra bởi website và gửi tới browser, do vậy 2 website khác nhau (cho dù cùng host trên 1 server) sẽ có 2 cookie khác nhau gửi tới browser. Ngoài ra, mỗi browser quản lý và lưu trữ cookie theo cách riêng của mình, cho nên 2 browser cùng truy cập vào 1 website sẽ nhận được 2 cookie khác nhau.

+Thiết lập cookie:

Để thiết lập cookie ta sử dụng cú pháp:

Setcookie("tên cookie", "giá trị", thời gian sống);

Tên cookie là tên mà chúng ta đặt cho phiên làm việc.

Giá trị là thông số của tên cookie.

Ví dụ:

Setcookie("userName", "admin", time() +3600)

Như ví dụ trên ta thấy với tên là username và giá trị là admin, có thời gian sống là 1 giờ tính từ thời điểm thiết lập.

Chú ý: Kịch bản cookie phải đặt trên mọi giá trị trả về bao gồm thẻ HTML và lệnh echo.

+Sử dụng cookie:

Để sử dụng lại cookie vừa thiết lập, chúng ta sử dụng .

Cú pháp: **\$_COOKIE["tên cookie"];**

Tên cookie là tên mà chúng ta thiết lập phía trên.

Ví dụ: page1.php

```

1  <?php
2      setcookie ("userName", "Nguyen Van C", time () + 3600);
3  ?>
4
5  <html>
6  <head>
7  <title>Test page 1</title></head>
8  <body>
9  <b><a href=page2.php>Click here</a></b>
10 </body>
11 </html>

```

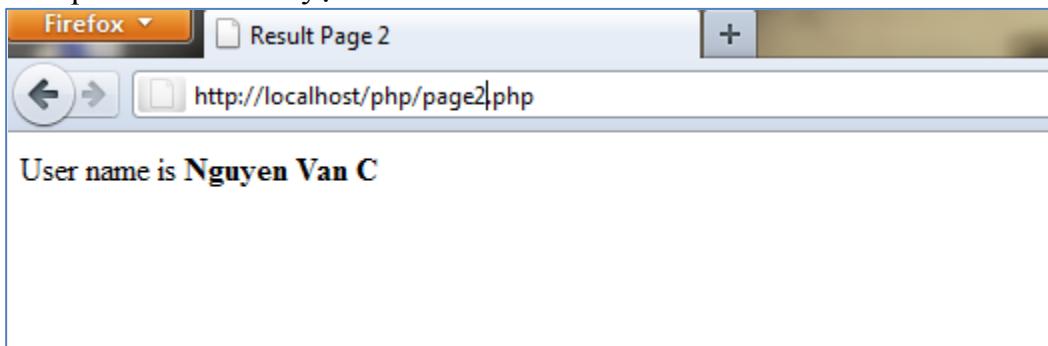
page2.php sẽ có code như sau:

```

1 <html>
2   <head><title>Result Page 2</title></head>
3   <body>
4
5   <?php
6     echo "User name is <b>".$_COOKIE['userName']."</b>";
7   ?>
8   </body>
9   </html>

```

Kết quả in ra trình duyệt là:



+**Hủy cookie:** Để hủy 1 cookie đã được tạo ta có thể dùng 1 trong 2 cách sau:

- Cú pháp: setcookie("Tên cookie")
Gọi hàm setcookie với chỉ duy nhất tên cookie mà thôi
- Dùng thời gian hết hạn cookie là thời điểm trong quá khứ.
Ví dụ: setcookie("userName","Nguyen Van C",time()-3600);

Câu hỏi ôn tập:

Câu 1: Viết trang login.php và lưu trữ thông tin vào biến session user, sau đó in ra thông tin đăng nhập thông qua session.

Câu 2: Xây dựng trang web theo mẫu sau:

 THẦN TÀI MAY... 15000 VND <input type="button" value="Thêm Vào Giỏ"/> <input type="button" value="Xem"/>	 Máy tính con ếch 65000 VND <input type="button" value="Thêm Vào GiỎ"/> <input type="button" value="Xem"/>	 Cân học toán... 85000 VND <input type="button" value="Thêm Vào GiỎ"/> <input type="button" value="Xem"/>
 Găng tay... 42000 VND <input type="button" value="Thêm Vào GiỎ"/> <input type="button" value="Xem"/>	 Lịch đồng hồ... 50000 VND <input type="button" value="Thêm Vào GiỎ"/> <input type="button" value="Xem"/>	 Đèn Hồ... 100000 VND <input type="button" value="Thêm Vào GiỎ"/> <input type="button" value="Xem"/>

Click vào nút thêm vào giỏ thì sử dụng session để lưu thông tin tên sản phẩm và giá sản phẩm. Thiết kế thêm 1 menu xem giỏ hàng, lúc đó hiển thị tất cả các sản phẩm được chọn để người dùng cập nhật và thanh toán.

CHƯƠNG 7

7.1 Các hàm xử lý chuỗi :

Định dạng chuỗi:

- Chuyển chuỗi từ chữ thường thành hoa: hàm strtoupper. Vd: strtoupper("chuong")
=> kết quả: CHUONG
- Chuyển chuỗi từ chữ hoa thành thường: hàm strtolower. Vd: strtolower("CHUONG")
=> kết quả: chuong
- Viết hoa kí tự đầu tiên: hàm ucfirst. Vd: ucfirst("chuong") => kết quả: Chuong
- Viết hoa kí tự đầu của mỗi từ: hàm ucwords. Vd: ucwords("quoc chuong") => kết
quả: Quoc Chuong

Kết hợp hay tách chuỗi:

- Hàm explode: hàm dùng để tách chuỗi. Hàm này trả về 1 mảng.

Vd:

hàm explode sẽ tìm trong mail qchuong100@gmail.com cứ có 1
kí tự @ là sẽ cắt thành và lưu vào thành 1 phần tử của mảng
data. Tức ở đây sẽ có 2 phần tử trong mảng data.

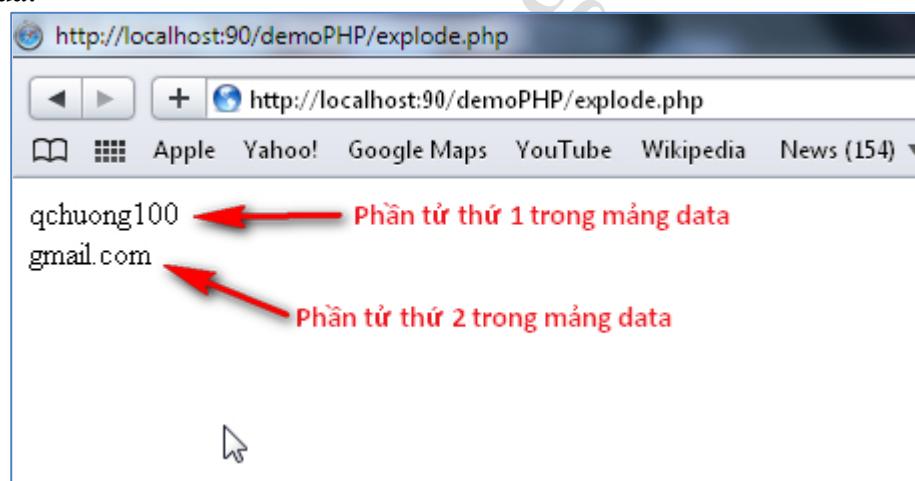
```

1  <?php
2  $data = explode("@", "qchuong100@gmail.com");
3  foreach($data as $i => $value)
4      echo $data[$i]. "<br/>";
5  ?>

```

foreach dùng để chạy và in ra hết toàn bộ các
phân tử trong mảng data lấy \$i làm giá trị phân
tử

Kết quả:



- Hàm implode thì ngược lại với hàm explode. Hàm này ghép nối 2 phần tử của
mảng lại và nối giữa 2 phần tử này là 1 kí tự hoặc 1 chuỗi nào đó.

Vd: như nối 2 phần của email trên vd của explode đã tách ra bằng @

```

1  <?php
2  $data = explode("@", "qchuong100@gmail.com");
3  foreach($data as $i => $value)
4      echo $data[$i]. "<br/>";
5
6  echo "<br />";           $data là mảng gồm 2 phần tử  
được tách ra bởi hàm explode
7
8  echo implode("@", $data); email hồi nay tách ra thành 1. =>  
email như cũ: qchuong100  
@gmail.com
9  ?>

```

Hàm substr: hàm cắt chuỗi theo điều kiện. Hàm này như sau:

substr(tham số 1, tham số 2);

Chú thích: tham số 1 là chuỗi cần cắt, tham số 2 là bắt đầu cắt từ đâu. Chuỗi sẽ được lấy từ tham số 2 trở về sau chuỗi đó.

substr(tham số 1, tham số 2, tham số 3)

Chú thích: tham số 1 là chuỗi cần cắt, tham số 2 là bắt đầu cắt. Tham số 3 là số lượng kí tự cần lấy.

Ví dụ:

```

Chú ý số thứ tự của
chuỗi bắt đầu bằng 0
12
13 =<?php
14 echo substr("good morning", 3);
15 echo "<br />".substr("good morning", 2, 5);
16 ?>
    
```

hàm substr 2 tham số
substr cắt chuỗi "good morning" và bắt đầu cắt từ vị trí thứ 3

hàm substr 3 tham số này cắt chuỗi này từ vị trí thứ 2 và số lượng kí tự sẽ lấy là 5

Và kết quả là:

Hàm strlen: hàm này sẽ trả về chiều dài của chuỗi.

Ví dụ:

```

20
21 =<?php
22 echo strlen("good morning");
?>
    
```

Kết quả trả về là 12.

Hàm strcmp(), strcasecmp(), strnatcmp(): để so sánh chuỗi ta có thể dùng hàm strcmp(). Nếu hàm này trả về 0 khi 2 chuỗi này bằng nhau. Nếu chuỗi 1 lớn hơn chuỗi 2 thì hàm sẽ trả về giá trị lớn hơn 0, ngược lại thì sẽ nhỏ hơn 0.

Syntax của strcmp(): strcmp(chuỗi 1, chuỗi 2);

Chú ý: hàm so sánh chuỗi strcmp() này có phân biệt chữ hoa và chữ thường. Trong trường hợp ngược lại thì sử dụng hàm strcasecmp(), strnatcmp().

Hàm thay thế chuỗi: str_replace(), substr_replace().

Ví dụ:

```

hàm str_replace tìm và thay thế từ you thành từ Jack
trong chuỗi $str. Ở đây sẽ có 2 từ được thay thế.
31 $str = "hello Jack, now are you? you look so strong in the morning.";
32 echo str_replace("you", "Jack", $str);
33 echo "<br /><br />";
34 echo substr_replace($str, "good", 25); ← Hàm substr_replace sẽ tìm vị trí thứ 25 trong chuỗi $str và
35 ?>     thay thế toàn bộ từ từ vị trí thứ 25 trở đi bằng từ good.
    
```

hello Jack, how are Jack? Jack look so strong in the morning.

hello Jack, how are you? good

7.2 Làm việc với mảng dữ liệu

Mảng 1 chiều: Cách khai báo mảng 1 chiều như sau:

Syntax:

Khai báo 1 mảng rỗng: \$arr = array();

Khai báo 1 mảng có sẵn phần tử:

\$arr = array(phan tử 1, phần tử 2, phần tử 3,...);

Ví dụ:

```

40  <?php
41  $arr = array("1", "2", "3");
42  foreach($arr as $i => $value)
43      echo $arr[$i]."<br />";
44 ?>

```

Kết quả in ra là:

```

1
2
3

```

Mảng 2 chiều: thực chất ra cũng là mảng 1 chiều nhưng được gọi là 2 chiều là vì trong mảng 1 chiều có chứa thêm 1 mảng 1 chiều.

\$arr = array(array(),array(),...);

Ví dụ:

```

40  <?php
41  $arr = array("1", "2", "3");           ← so sánh mảng 1 chiều và mảng 2 chiều
42  foreach($arr as $i => $value)
43      echo $arr[$i]."<br />";
44
45  echo "<br /><br />";
46  $arr2 = array(array("a", "b", "c"), array("e", "d"));
47  foreach($arr2 as $s => $value)          ← mảng 2 chiều
48      foreach($arr2[$s] as $r => $val)    ← hàm count là hàm đếm số lượng
49          echo $arr2[$s][$r]."<br />";     ← phần tử có trong mảng
50
51 ?>

```

← số lượng phần tử trong mảng arr2 là: ".count(\$arr2);|

7.3 Kiểu datetime

- Để làm việc với kiểu dữ liệu Date và Time, ta sử dụng hàm của PHP có sẵn. Chẳng hạn, muốn trình bày chuỗi ngày tháng, ta dùng hàm date với các tham số như ví dụ sau:

```

<html>
<head>
    <title>Date and Time</title>
</head>
<body>
<h4>Ngay hiện tại</h4>
<?php
    echo date("j-S-F-Y");
    echo "<br>";
    echo date("M/Y");
    echo "<br>";
    echo "Days of ".date("M")." is ".date("t");
    echo "<br>";
?
</body>
</html>

```

Kết quả:

Ngay hiện tại

20-th-April-2005
 Apr/2005
 Days of Apr is 30

Các tham số được quy định trong bảng sau:

Kí tự	Ý nghĩa
a	am/pm
A	AM/PM
d	Day (01-31) trong tháng với 2 số, nếu ngày 1-9 sẽ có kèm thêm số 0
D	Day (Mon-Sun) trong tuần với 3 ký tự
F	Tháng (January-December) trong năm với tên tháng đầy đủ dạng text
g	Hour (1-12) trong ngày, 1 hoặc 2 số (không kèm thêm số 0 nếu giờ từ 1-9)
G	Hour (0-23) trong ngày, 1 hoặc 2 số (không kèm thêm số 0 nếu giờ từ 1-9)
h	Hour (01-12) trong ngày, 2 số (kèm 0 nếu giờ từ 01-09)
H	Hour (00-23) trong ngày, 2 số (kèm 00 nếu giờ từ 00-09)
i	Minutes (01-59) đã trôi qua (kèm 00 nếu phút từ 00-59)
j	Day (1-31) 1 hoặc 2 số (không kèm 0 nếu ngày từ 1-9)
I	Day (Monday-Sunday) trong tuần dạng text
L	Năm nhuận trả về 1, ngược lại trả về 0
m	Month (01-12) trong năm 2 số (kèm 00 nếu tháng từ 01-09)
M	Month (Jan-Dec) trong năm 3 ký tự
n	Month (1-12) 1 hoặc 2 số (không kèm 0 nếu tháng từ 1-9)

s	Second (01-59) đã trôi qua (kèm 00 nếu giây từ 00-59)
S	Thêm 2 ký tự st, nd, rd hay th theo sau ngày dạng 2 ký tự số
t	Trả về tổng số ngày trong tháng từ 28-31
T	Ký tự Timezone của server với 3 ký tự chặng hạn như EST
U	Tổng số second từ 1 January 1970 tới hôm nay ứng với UNIX Time Stamp
w	Day (0-6) của tuần, 0 ứng với Sunday và 6 ứng với Saturday
y	Năm định dạng 2 con số (11)
Y	Năm định dạng 4 con số (2011)
z	Ngày trong năm, 1 hoặc 2 con số (0-365)
X	Timezone hiện tại tính bằng giây từ -43200 đến 43200

Câu hỏi ôn tập:

-Viết trang Register.php với các thông tin như sau:

- Tên đăng nhập
- Mật khẩu
- Xác nhận mật khẩu
- Giới tính
- Ngày sinh
- Địa chỉ
- Button Register

Yêu cầu:

- Các fiel không được rỗng.
- Mật khẩu và xác nhận mật khẩu phải giống nhau
- Ngày sinh theo chuẩn dd-mm-yyyy
- Sau đó lưu tất cả thông tin vào mảng (1 hoặc 2 chiều) và lưu vào session register.
- Cuối cùng in ra thông tin thông qua trong session.

CHƯƠNG 8

8.1 Khai báo HÀM trong PHP:

Cú pháp:

```
function TenFunction( Các tham số nếu có)
```

```
{
```

```
    //các mã lệnh thực thi
```

```
}
```

***Lưu ý:** Tên hàm phải bắt đầu bằng chữ cái hoặc dấu gạch dưới.

Ví dụ: viết hàm tính tổng 2 số.

```
function fnTinhTong2So($a,$b)
```

```
{
```

```
    return $a+$b;
```

```
}
```

Kiểm tra hàm

```
echo "Tổng 2 số = ".fnTinhTong2So(2,5);
```

⇒ Kết quả ra là: Tổng 2 số = 7

8.2 Thông nhất kích thước của mọi trang PHP

+ Khi xây dựng ứng dụng web chuyên nghiệp, điều đầu tiên bạn nên quan tâm là sự thống nhất kích thước của các phần trên trang web. Điều này có nghĩa khi người dùng thay đổi trang web khi duyệt các phần như top,bottom,left,right có kích thước như nhau.

+Tùy theo bố cục của mỗi người mà trang web được chia theo bao nhiêu đó phần. Ở đây chúng ta sẽ lấy ví dụ trang web gồm 5 phần :

- Top
- Bottom
- Left
- Right
- Body

Phần Top: thường trình bày các thuộc tính như quảng cáo (banner), logo (biểu tượng công ty), menu của ứng dụng web đó và 1 số thông tin khác.

Phần Left: là trình bày thông tin của các menu phụ hay còn gọi là menu của các menu chính, bên cạnh các menu con này trang web thường có các liên kết về liên hệ, quảng cáo, mailing list (đăng ký email), gửi đến bạn bè(send to friend),...

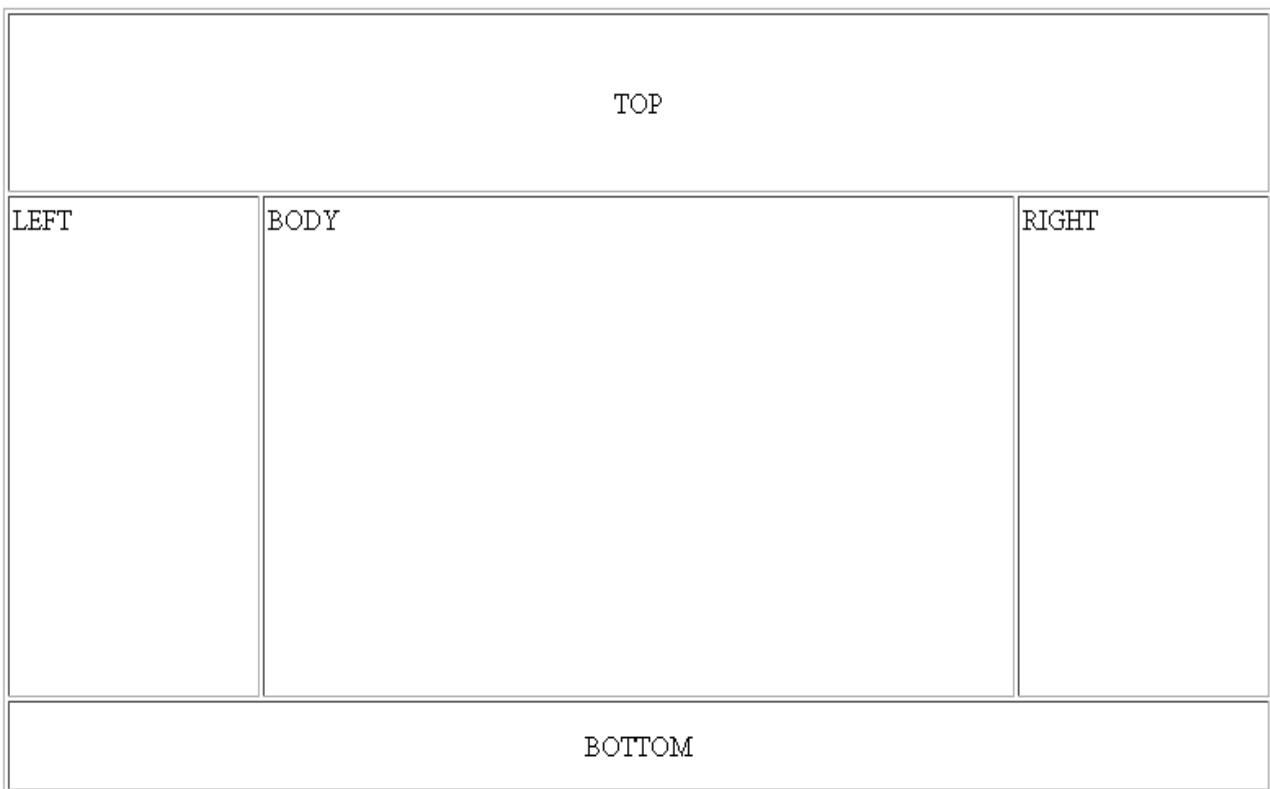
Phần Right: thường là phần giới thiệu về các thông tin đặc biệt và quảng cáo, chẳng hạn đối với ứng dụng bán sách, phần right thường là danh sách các nhóm sách bán chạy, sắp phát hành,....

Phần Bottom: là phần thường trình bày thông tin liên lạc với công ty, người quản trị website và bản quyền. Ngoài ra, phần bottom đôi khi là danh sách các menu con khác.

Phần Body: thể hiện nội dung chính của website

Nói chung là tùy thuộc ý tưởng thiết kế bố cục và thiết kế website của mỗi người thì trang website sẽ có sự khác nhau. Tùy theo bố cục của mỗi người mà website có thể có các phần nêu trên.

Vd: đây là bố cục của 1 website gồm 5 phần như nói trên.



Hình ảnh trên chỉ mang tính chất minh họa cho bộ cục của website gồm 5 phần: Top, Bottom, Left, Right, Body

Còn đây là đoạn code thể hiện bộ cục 5 phần của 1 website như hình trên

```
<!DOCTYPE html PUBLIC "-//W3C//Dtd XHTML 1.0 transitional//EN"
"http://www.w3.org/tr/xhtml1/Dtd/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Thống nhất kích thước trang PHP</title>
</head>

<body bottommargin="0" leftmargin="0" topmargin="0" rightmargin="0">
<table cellspacing="2" cellpadding="2" width="778" border="1" align="center">
    <tr height="100">
        <td align="center" colspan="3">TOP</td>
    </tr>
    <tr height="280">
        <td valign="top" width="20%">LEFT</td>
        <td valign="top" width="60%">BODY</td>
        <td valign=top width="20%">RIGHT</td>
    </tr>
    <tr height="50">
        <td colspan="3" align="center">BOTTOM</td>
    </tr>
</table>
</body>
</html>
```

***Chú ý:** Sử dụng 2 hàm sau để nhúng trang:

- Hàm **include**(“Tên trang muốn nhúng vào”);
- Hàm **require**(“Tên trang muốn nhúng vào”);

Câu hỏi ôn tập:

Câu 1: Thiết kế layout website sử dụng các tập tin dùng chung theo mẫu sau:



Chương 9: Hướng đối tượng trong PHP

9.1 Hướng đối tượng trong PHP

+**Định nghĩa:** Lập trình hướng đối tượng (gọi tắt là OOP viết tắt của từ Object Oriented Programming) là kĩ thuật lập trình hỗ trợ công nghệ hướng đối tượng. OOP được xem là giúp tăng năng suất, đơn giản hóa độ phức tạp khi bảo trì cũng như là việc mở rộng phần mềm bằng cách cho phép lập trình viên tập trung vào các đối tượng phần mềm ở bậc cao hơn.

Những đối tượng trong một ngôn ngữ OOP là các kết hợp giữa mã và dữ liệu mà chúng được nhìn nhận như là 1 đơn vị duy nhất. Mỗi đối tượng có 1 tên riêng biệt và tất cả các tham chiếu đến đối tượng đó được tiến hành qua tên của nó. Như vậy, mỗi đối tượng có khả năng nhận vào các thông báo, xử lý dữ liệu bên trong của nó, gửi ra hay trả lời đến các đối tượng khác hay đến môi trường.

+Các tính chất cơ bản của lập trình hướng đối tượng:

Một đối tượng bao gồm 2 thành phần:

1. Các thuộc tính (hay còn gọi là các biến)
2. Các phương thức (hay còn gọi là các hàm)

Thuộc tính là tập hợp các dữ liệu đặc trưng của 1 đối tượng. Vd: xe máy có đặc trưng như:

1. Xe máy thuộc hãng xe nào
2. Model đời xe máy
3. Màu xe
4. Tốc độ của 1 chiếc xe máy
5.

=> Đó là các thuộc tính của 1 chiếc xe máy.

Phương thức là các phương tiện hành động của 1 đối tượng được dùng để sử dụng 1 đối tượng. Vd như trên nhưng phương thức của xe máy là chạy và dừng,....

Lập trình hướng đối tượng là một phương pháp lập trình có các tính chất chính sau:

Tính trừu tượng (abstraction): Đây là khả năng của chương trình bỏ qua hay không chú ý đến một số khía cạnh của thông tin mà nó đang trực tiếp làm việc lên, nghĩa là nó có khả năng tập trung vào những cốt lõi cần thiết. Mỗi đối tượng phục vụ như là một "động tử" có thể hoàn tất các công việc một cách nội bộ, báo cáo, thay đổi trạng thái của nó và liên lạc với các đối tượng khác mà không cần cho biết làm cách nào đối tượng tiến hành được các thao tác. Tính chất này thường được gọi là sự trừu tượng của dữ liệu.

Tính trừu tượng còn thể hiện qua việc một đối tượng ban đầu có thể có một số đặc điểm chung cho nhiều đối tượng khác như là sự mở rộng của nó nhưng bản thân đối tượng ban đầu này có thể không có các biện pháp thi hành. Tính trừu tượng này thường được xác định trong khái niệm gọi là lớp trừu tượng hay lớp cơ sở trừu tượng.

Tính đóng gói (encapsulation) và che dấu thông tin (information hiding): Tính chất này không cho phép người sử dụng các đối tượng thay đổi trạng thái nội tại của một đối tượng. Chỉ có các phương thức nội tại của đối tượng cho phép thay đổi trạng thái của nó. Việc cho phép môi trường bên ngoài tác động lên các dữ liệu nội tại của một đối tượng theo cách nào là hoàn toàn tùy thuộc vào người viết mã. Đây là tính chất đảm bảo sự toàn vẹn của đối tượng.

Tính đa hình (polymorphism): Thể hiện thông qua việc gửi các thông điệp (message). Việc gửi các thông điệp này có thể so sánh như việc gọi các hàm bên trong của một đối tượng. Các phương thức dùng trả lời cho một thông điệp sẽ tùy theo đối tượng mà thông điệp đó được gửi tới sẽ có phản ứng khác nhau. Người lập trình có thể định nghĩa một đặc tính (chẳng hạn thông qua tên của các phương thức) cho một loạt các đối tượng gần nhau nhưng khi thi hành thì dùng cùng một tên gọi mà sự thi hành của mỗi đối tượng sẽ tự động xảy ra tương ứng theo đặc tính của từng đối tượng mà không bị nhầm lẫn.

Thí dụ khi định nghĩa hai đối tượng "hinh_vuong" và "hinh_tron" thì có một phương thức chung là "chu_vi". Khi gọi phương thức này thì nếu đối tượng là "hinh_vuong" nó sẽ tính theo công thức khác với khi đối tượng là "hinh_tron".

Tính kế thừa (inheritance): Đặc tính này cho phép một đối tượng có thể có sẵn các đặc tính mà đối tượng khác đã có thông qua kế thừa. Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại. Tuy nhiên, không phải ngôn ngữ định hướng đối tượng nào cũng có tính chất này.

Lớp (class)

Một lớp được hiểu là một kiểu dữ liệu bao gồm các thuộc tính và các phương thức được định nghĩa từ trước. Đây là sự trừu tượng hóa của đối tượng. Một đối tượng sẽ được xác lập khi nó được thực thể hóa từ một lớp. Khác với kiểu dữ liệu thông thường, một lớp là một đơn vị (trừu tượng) bao gồm sự kết hợp giữa các phương thức và các thuộc tính. Để có một đối tượng (mà có thể được xem như là một biến) hoạt động được thì việc thực thể hóa sẽ có thể bao gồm việc cài đặt các giá trị ban đầu của các thuộc tính cũng như việc đăng ký bộ nhớ, mà công việc này thường được giao cho các phương thức gọi là "máy khởi tạo" (constructor) hay hàm dựng. Ngược lại khi một đối tượng thuộc về một lớp không còn sử dụng nữa thì cũng có thể có một phương thức để xử lý gọi là "máy hủy diệt" (destructor) hay hàm hủy.

Như vậy, để có được các đối tượng thì người lập trình OOP cần phải thiết kế lớp của các đối tượng đó bằng cách xây dựng các thuộc tính và các phương thức có các đặc tính riêng.

Mỗi một phương thức hay một thuộc tính đầy đủ của một lớp còn được gọi tên là một thành viên (member) của lớp đó.

Lớp con (subclass)

Lớp con là một lớp thông thường nhưng có thêm tính chất kế thừa một phần hay toàn bộ các đặc tính của một lớp khác. Lớp mà chia sẻ sự kế thừa gọi là lớp phụ mẫu (parent class).

Lớp trừu tượng hay lớp cơ sở trừu tượng (abstract class)

Lớp trừu tượng là một lớp mà nó không thể thực thể hóa thành một đối tượng thực dụng được. Lớp này được thiết kế nhằm tạo ra một lớp có các đặc tính tổng quát nhưng bản thân lớp đó chưa có ý nghĩa (hay không đủ ý nghĩa) để có thể tiến hành viết mã cho việc thực thể hóa. (xem thí dụ)

Thí dụ: Lớp "hinh_phang" được định nghĩa không có dữ liệu nội tại và chỉ có các phương thức (hàm nội tại) "tinh_chu_vi", "tinh_dien_tich". Nhưng vì lớp hình phẳng này chưa xác

định được đầy đủ các đặc tính của nó (cụ thể các biến nội tại là tọa độ các đỉnh nếu là đa giác, là đường bán kính và toạ độ tâm nếu là hình tròn, ...) nên nó chỉ có thể được viết thành một lớp trừu tượng. Sau đó, người lập trình có thể tạo ra các lớp con chẳng hạn như là lớp "tam_giac", lớp "hinh_tron", lớp "tu_giac",.... Và trong các lớp con này người viết mã sẽ cung cấp các dữ liệu nội tại (như là biến nội tại r làm bán kính và hằng số nội tại Pi cho lớp "hinh_tron" và sau đó viết mã cụ thể cho các phương thức "tinh_chu_vi" và "tinh_dien_tich").

Phương Thức (method)

Là hàm nội tại của một lớp (hay một đối tượng). Tùy theo đặc tính mà người lập trình gán cho, một phương pháp có thể chỉ được gọi bên trong các hàm khác của lớp đó, có thể cho phép các câu lệnh bên ngoài lớp gọi tới nó, hay chỉ cho phép các lớp có quan hệ đặc biệt như là quan hệ lớp con, và quan hệ bạn bè (friend) được phép gọi tới nó. Mỗi phương pháp đều có thể có kiểu trả về, chúng có thể trả các kiểu dữ liệu cổ điển hay trả về một kiểu là một lớp đã được định nghĩa từ trước. Một tên gọi khác của phương pháp của một lớp là hàm thành viên.

Người ta còn định nghĩa thêm vài loại phương pháp đặc biệt:

Hàm dựng (constructor) là hàm được dùng để cài đặt các giá trị ban đầu cho các biến nội tại và đôi khi còn dùng để khai báo về việc sử dụng bộ nhớ.

Hàm hủy (destructor) là hàm dùng vào việc làm sạch bộ nhớ và hủy bỏ tên của một đối tượng sau khi đã dùng xong, trong đó có thể bao gồm cả việc xóa các con trỏ nội tại và trả về các phần bộ nhớ mà đối tượng đã dùng.

Trong một số trường hợp thì hàm hủy hay hàm dựng có thể được tự động hóa bởi ngôn ngữ OOP như là trường hợp của Visual C++, C#.

Tiện ích (utility) là các hàm chỉ hoạt động bên trong của một lớp mà không cho phép môi trường bên ngoài gọi tới. Các hàm này có thể là những tính toán trung gian nội bộ của một đối tượng mà xét thấy không cần thiết phải cho thế giới bên ngoài của đối tượng biết là gì.

Thuộc tính (attribute)

Thuộc tính của một lớp bao gồm các biến, các hằng, hay tham số nội tại của lớp đó. Ở đây, vai trò quan trọng nhất của các thuộc tính là các biến vì chúng sẽ có thể bị thay đổi trong suốt quá trình hoạt động của một đối tượng. Các thuộc tính có thể được xác định kiểu và kiểu của chúng có thể là các kiểu dữ liệu cổ điển hay đó là một lớp đã định nghĩa từ trước. Như đã ghi, khi một lớp đã được thực thể hóa thành đối tượng cụ thể thì tập hợp các giá trị của các biến nội tại làm thành trạng thái của đối tượng. Giống như trường hợp của phương pháp, tùy theo người viết mã, biến nội tại có thể chỉ được dùng bên trong các phương pháp của chính lớp đó, có thể cho phép các câu lệnh bên ngoài lớp, hay chỉ cho phép các lớp có quan hệ đặc biệt như là quan hệ lớp con, (và quan hệ bạn bè (friend) trong C++) được phép dùng tới nó (hay thay đổi giá trị của nó). Mỗi thuộc tính của một lớp còn được gọi là thành viên dữ liệu của lớp đó.

Thực thể (instance)

Thực thể hóa (instantiate) là quá trình khai báo để có một tên (có thể được xem như là một biến) trở thành một đối tượng từ một lớp nào đó.

Một lớp sau khi được tiến hành thực thể hóa để có một đối tượng cụ thể gọi là một thực thể. Hay nói ngược lại một thực thể là một đối tượng riêng lẽ của một lớp đã định trước. Như các biến thông thường, hai thực thể của cùng một lớp có thể có trạng thái nội tại khác nhau (xác định bởi các giá trị hiện có của các biến nội tại) và do đó hoàn toàn độc lập nhau nếu không có yêu cầu gì đặc biệt từ người lập trình.

Công cộng (public)

Công cộng là một tính chất được dùng để gán cho các phương pháp, các biến nội tại, hay các lớp mà khi khai báo thì người lập trình đã cho phép các câu lệnh bên ngoài cũng như các đối tượng khác được phép dùng đến nó.

Thí dụ: Trong C++ khai báo public: int my_var; thì biến my_var có hai tính chất là tính công cộng và là một integer cả hai tính chất này hợp thành đặc tính của biến my_var khiến nó có thể được sử dụng hay thay đổi giá trị của nó (bởi các câu lệnh) ở mọi nơi bên ngoài lẫn bên trong của lớp.

Riêng tư (private)

Riêng tư là sự thể hiện tính chất đóng mạnh nhất (của một đặc tính hay một lớp) <http://hocvui.net>. Khi dùng tính chất này gán cho một biến, một phương pháp thì biến hay phương pháp đó chỉ có thể được sử dụng bên trong của lớp mà chúng được định nghĩa. Mọi nỗ lực dùng đến chúng từ bên ngoài qua các câu lệnh hay từ các lớp con sẽ bị phủ nhận hay bị lỗi.

Bảo tồn (protected)

Tùy theo ngôn ngữ, sẽ có vài điểm nhỏ khác nhau về cách hiểu tính chất này. Nhìn chung đây là tính chất mà khi dùng để áp dụng cho các phương pháp, các biến nội tại, hay các lớp thì chỉ có trong nội bộ của lớp đó hay các lớp con của nó (hay trong nội bộ một gói như trong Java) được phép gọi đến hay dùng đến các phương pháp, biến hay lớp đó.

So với tính chất riêng tư thì tính bảo tồn rộng rãi hơn về nghĩa chia sẻ dữ liệu hay chức năng. Nó cho phép một số trường hợp được dùng tới các đặc tính của một lớp (từ một lớp con chẳng hạn).

Lưu ý: Các tính chất công cộng, riêng tư và bảo tồn đôi khi còn được dùng để chỉ thị cho một lớp con cách thức kế thừa một lớp cha mẹ như trong C++.

Đa kế thừa (multiple inheritance)

Đây là một tính chất cho phép một lớp con có khả năng kế thừa trực tiếp cùng lúc nhiều lớp khác.

Vài điểm cần lưu ý khi viết mã dùng tính chất đa kế thừa:

Khi muốn có một sự kế thừa từ nhiều lớp phụ mẫu thì các lớp này cần phải độc lập và đặc biệt tên của các dữ liệu hay hàm cho phép kế thừa phải có tên khác nhau để tránh lỗi "ambiguity". Bởi vì lúc đó phần mềm chuyển dịch sẽ không thể xác định được là lớp con sẽ thừa kế tên nào của các lớp phụ mẫu.

Không phải ngôn ngữ OOP loại phân lớp nào cũng hỗ trợ cho tính chất này.

Ngoài các khái niệm trên, tùy theo ngôn ngữ, có thể sẽ có các chức năng OOP riêng biệt được cấp thêm vào.

Khai báo class và thể hiện của class trong PHP: 1 lớp bao gồm các thuộc tính và phương thức.

```
class tên_lớp
{
    //danh sách các biến, hằng, lớp, ... (thuộc tính)
    //danh sách các hàm (phương thức)
}
```

Trong đó, các lớp được khai báo thông qua từ khóa Class, các thuộc tính được khai báo dưới dạng các biến, còn các phương thức được xây dựng dưới dạng các hàm.

Các thuộc tính và phương thức có thể thiết lập các tính chất đặc biệt như : private, public, ...

Các tính chất này được đặt trước các khai báo thuộc tính và phương thức.

Vd: tạo 1 lớp sinhvien có các thuộc tính public như masv, tensv,...

```
class sinhvien
{
    public $masv;
    public $tensv;
}
```

Để khai báo 1 thể hiện của 1 lớp, chúng ta dùng từ khóa new như cú pháp sau:

```
$ten_thuc_the = new Ten_lop();
```

Để truy cập vào các thuộc tính hay phương thức của lớp đó, ta dùng toán tử -> với cú pháp như sau:

```
$ten_thuc_the->ten_thuoc_tinh;
```

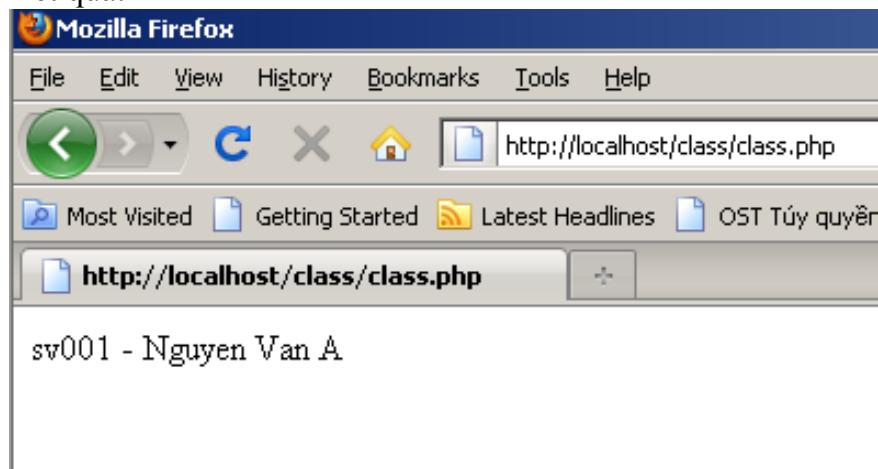
Ví dụ sau sẽ tạo 1 lớp như ví dụ trên và sẽ thiết lập thêm các thuộc tính của lớp này:

```

1  <?php
2  class sinhvien
3  {
4      public $masv;
5      public $tensv;
6  }
7  $sv = new sinhvien();
8  $sv->masv = "sv001";
9  $sv->ten = "Nguyen Van A";
10 echo $sv->masv." - ".$sv->ten;
11 ?>

```

Kết quả:



+Cú pháp tạo phương thức:

```
[Kiểu truy xuất] function Ten_Phuong_Thuc(danh sách các tham số)
{
    //xử lý
}
```

Vd: tạo thêm phương thức thiết lập thông tin cho sinh viên có 2 tham số nhận vào là masv và tensv với tên phương thức là thiếtLậpThôngTin()

```

1 <?php
2 class sinhvien
3 {
4     public $masv;
5     public $tensv;
6
7     public function thietLapThongTin($ma,$ten)
8     {
9         $this->masv = $ma;//this là gọi chính lớp này
10        $this->tensv = $ten;
11    }
12 }
13 $sv = new sinhvien();
14 $sv->thietLapThongTin("sv001","Nguyen Van A");
15 echo $sv->masv." - ".$sv->tensv;
16 ?>
```

Hoặc có thể sử dụng construct được thiết lập của PHP. Cú pháp hàm khởi tạo như hình như:

```

1 <?php
2 class sinhvien
3 {
4     public $masv;
5     public $tensv;
6
7     public function __construct($ma,$ten)
8     {
9         $this->masv = $ma;
10        $this->tensv = $ten;
11    }
12 }
13 $sv = new sinhvien("sv001","Nguyen Van A");
14 echo $sv->masv.", ".$sv->tensv;

```

Chúng ta có thể đưa tham số vào như hình hoặc tùy theo trường hợp mà sử dụng

function __construct(danh sách tham số nếu có)

```

{
    //mã thực thi
}

```

Lập trình hướng đối tượng tập trung vào việc đóng gói các thuộc tính và phương thức của 1 đối tượng nào đó. Trong lập trình hướng đối tượng thì các thành viên của 1 lớp cần được xác định xem chúng có thể được truy xuất từ đâu. Có 3 cách truy xuất như sau:

1. Kiểu truy xuất public (công khai): nếu phương thức hay thuộc tính nào mang giá trị này thì nó được truy xuất tự do bên ngoài lớp và bên trong nó.
2. Kiểu truy xuất private (riêng tư): thì chỉ truy xuất được chính bên trong lớp đó
3. Kiểu truy xuất protected (bảo vệ): truy xuất được chính bên trong lớp đó và lớp thừa kế từ lớp đó.

Vd ta xây dựng lớp thành viên để quản lý thành viên,trong đó biến ID thành viên là không cần thiết xuất ra trình duyệt và đó là thông tin quan trọng trong cơ sở dữ liệu nên ta khai báo nó ở dạng private để tránh sự thay đổi giá trị không mong muốn do sự sơ xuất trong lập trình.Cách dùng các khoá bạn xem trong VD sau:

```

<?php
/**
 * Define MyClass
 */
class MyClass
{
    public $public = 'Public';
    protected $protected = 'Protected';
    private $private = 'Private';

    function printHello()
    {
        echo $this->public;
        echo $this->protected;
        echo $this->private;
    }
}

$obj = new MyClass();
echo $obj->public; // Works
echo $obj->protected; // Fatal Error
echo $obj->private; // Fatal Error
$obj->printHello(); // Shows Public, Protected and Private

```



```

/**
 * Define MyClass2
 */
class MyClass2 extends MyClass
{
    // We can redeclare the public and protected method, but not private
    protected $protected = 'Protected2';

    function printHello()
    {
        echo $this->public;
        echo $this->protected;
        echo $this->private;
    }
}

$obj2 = new MyClass2();
echo $obj->public; // Works
echo $obj2->private; // Undefined
echo $obj2->protected; // Fatal Error
$obj2->printHello(); // Shows Public, Show Protected2, not show Private
?>

```

Lưu ý: phương thức, thuộc tính protected trong lớp cơ sở chỉ được thừa kế trong nội bộ của lớp thừa kế, do đó chỉ các hàm trong lớp thừa kế mới có quyền gọi chúng, còn đối tượng của lớp thừa kế thì không thể gọi các \$ và hàm protected trong lớp cơ sở.

+Thuộc tính hằng, thuộc tính tĩnh và phương thức tĩnh: Trong PHP, thuộc tính hằng, phương thức tĩnh và thuộc tính tĩnh tồn tại ngay cả khi chưa có đối tượng nào được gọi. Thuộc tính hằng dùng để tạo ra một hằng số và giá trị của nó không thể thay đổi. Còn thuộc tính và phương thức tĩnh tồn tại ngay sau khi nó được định nghĩa (nghĩa là không cần gọi lớp vẫn có thể dùng dc).

Lưu ý: Phương thức tĩnh không được tác động và truy xuất đến các thuộc tính không tĩnh. Có thể truy xuất đến thuộc tính tĩnh. Có vẻ hơi trừu tượng, chúng ta hãy đi vào VD sau:

```

<?php
class NGUOI
{
var $hoten;
var $tuoi;
static $quoctich="vietnam";
const viet_nam="Dang cap";
function dat_ten($hoten1,$tuoi1)
{
$this->hoten=$hoten1;
$this->tuoi=$tuoi1;
}
function lay_ten()
{
return $this->hoten;
}
function lay_tuoi()
{
return $this->tuoi;
}
public static function vidu()
{
    echo self::$quoctich.'<br>';
    echo self::viet_nam;
}
}

echo NGUOI::$quoctich.'<br>';//thuoc tinh tinh
echo NGUOI::viet_nam.'<br>'; //hang
echo NGUOI::vidu();           //phuong thuc tinh
-->

```

Như vậy để sử dụng hằng, thuộc tính tĩnh và phương thức tĩnh ta dùng cú pháp sau:
tên_lớp::tên_hằng/thuộc tính tĩnh/phương thức tĩnh.Nếu muốn truy cập thuộc tính tĩnh hoặc
hằng trong một hàm của chính lớp đó thì thay vì dùng \$this-> ta có thể dùng self::tên
hằng/thuộc tính tĩnh/phương thức tĩnh.

Khi nào thì dùng đến những thứ này?Trong một project web sẽ có lúc bạn phải dùng đến nó
ví dụ như hàm kiểm tra sự chính xác của email, username nhập vào trước khi cho nó vào
lớp...

+**Dẫn xuất, thừa kế, lớp cơ sở, chạy đè phương thức:**

Nếu lớp B thừa kế từ lớp A thì lớp B gọi là lớp dẫn xuất, lớp A là lớp cơ sở.Lớp dẫn xuất có
thể sử dụng tất cả các phương thức và thuộc tính public ,protected của lớp cơ sở.

Khi nào thì cần dùng thừa kế? Khi lớp B sử dụng một số phương thức và ttính mà lớp A đã
định nghĩa sẵn thì ta nên cho B thừa kế của A, chỉ khai báo trong B thêm những cái mới mà
trong A không có.

Ví dụ: trên đây ta có class nguoi, giờ nếu ta khai báo class công nhân. Thì mỗi công nhân
cũng phải có tên tuổi, mà hai yếu tố này ta đã khai báo, “xào nấu” sẵn trong class
nguoi.Vậy thì ta hãy cho class cong_nhan thừa kế class nguoi đi, khởi khai báo thêm chỉ
cho mệt!!!

Để cho class cong_nhan thừa kế từ class nguoi ta theo cú pháp sau: class cong_nhan
extends nguoi { ... }

```

2   class sinhvien
3   {
4       var $hoten;
5       var $tuoi;
6       const quoctich="vietnam";
7       public function __construct($hoten1="Nguyen Van A", $tuoi1=19){
8           $this->hoten=$hoten1;
9           $this->tuoi=$tuoi1;
10      }
11      function lay_ten(){
12          return $this->hoten;
13      }
14      function lay_tuoi(){
15          return $this->tuoi;
16      }
17      function chebien(){
18          $this->hoten="Tên tôi là: ".$this->hoten;
19          $this->tuoi++;
20      }
21  }

```

```

class cong_nhan extends NGUOI
{
//không cần khai báo tên tuổi nữa
var $ten_cong_ty;
    public function __construct($hoten2,$tuoi2,$cty)
    {
        $this->hoten=$hoten2;
        $this->tuoi=$tuoi2;
        $this->ten_cong_ty=$cty;
    }
    public function in()
    {
        printf("Ban ten <b>%s</b> <br>", $this->hoten);
        printf("Ban <b>%s</b> tuoi<br>", $this->tuoi);
        printf("Dang lam tai cong ty <b>%s</b> <br>", $this->ten_cong_ty);
        printf("Quoc tich: <b>%s</b>", parent::quoctich);
    }
}
$e=new cong_nhan("ngoc lan",19,"Intel Viet Nam");
echo $e->in();

```

Trong lớp dẫn xuất mà muốn truy xuất giá trị của thuộc tính tĩnh, hằng trong lớp cơ sở ta dùng cú pháp parent:: tên hằng/thuộc tính tĩnh/phương thức tĩnh.

Khi trong lớp dẫn xuất mà có phương thức trùng tên với phương thức trong lớp cơ sở thì sẽ ưu tiên chạy phương thức trong lớp dẫn xuất gọi là chạy đè phương thức. Ví dụ đơn giản như trong lớp cơ sở có hàm *** dùng để đặt tên là “chó” nhưng trong lớp dẫn xuất cũng có hàm *** và hàm này lại đặt tên là “mèo”. vậy khi chạy thì kết quả sẽ là “mèo”.

+Truy cập phương thức trong lớp cơ sở:

```

<?php
class cho
{
    public $tencho;
    public $tuoicho;
    public function cho($ten,$tuoi) //đây là một cách khác của hàm khởi tạo.
    {
        $this->tencho=$ten;
        $this->tuoicho=$tuoi;
    }
    public function in()
    {
        printf("<b>%s</b>, hiện đang <b>%s</b> tuổi!", $this->tencho, $this->tuoicho);
    }
}

class meo extends cho
{
    public $tenmeo;
    public function __construct($ten_meo,$ten_cho,$tuoi_cho)
    {
        $this->cho($ten_cho,$tuoi_cho); //1 cách khác để gán giá trị cho các thuộc tính
        //lớp dẫn xuất
        $this->tenmeo=$ten_meo;
    }
    public function printt()
    {
        printf("Con meo này tên <b>". $this->tenmeo."</b> có ban cho tên là ");
        cho::in(); // hoặc: parent::in();
    }
}
$meo=new meo("MIMI", "LULU", 1);
$meo->printt();
?>

```

Trong lớp dẫn xuất, nếu ta muốn truy cập các phương thức (kể cả phương thức tĩnh) trong lớp cơ sở thì theo cú pháp: tên_lớp_cơ_sở::tên_hàm hoặc parent::tên_hàm.

Câu hỏi ôn tập:

- Tạo form register.php với thông tin:

- Tên người dùng
- Mật khẩu

Nhưng lần này sử dụng hướng đối tượng với đối tượng là người dùng bao gồm các thông tin trên.

Viết thêm các phương thức: kiểm tra người dùng đã tồn tại chưa (vì không có database nên tạm thời dùng mảng tạo sẵn thông tin của vài người dùng được kiểm tra) và các phương get và set thông tin người dùng vào trong 1 đối tượng.

CHƯƠNG 10

10.1 Giới thiệu CSDL MySQL

MySQL là 1 phần mềm quản trị CSDL mã nguồn mở, miễn phí nằm trong nhóm LAMP(Linux- Apache – MySQL - PHP)

10.2 Cài đặt MySQL

Phần cài đặt này đã được thực hiện khi chúng ta cài đặt wampserver.

10.3 Tạo CSDL & người dùng

Quản lý người dùng:

Server: localhost

- information_schema (28)
- mysql (23)
- qlsinhvien (4)
- quanlysinhvien (1)

Please select a database

Actions

MySQL localhost

- Create new database
- MySQL connection collation: utf8_general_ci

Interface

- Language: English
- Theme / Style: Original
- Custom color: Reset
- Font size: 82%

MySQL

- Server: localhost (localhost via TCP/IP)
- Server version: 5.1.33-community-log
- Protocol version: 10
- User: root@localhost
- MySQL charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.2.11 (Win32) PHP/5.2.9-2
- MySQL client version: 5.0.51a
- PHP extension: mysqli

phpMyAdmin

- Version information: 3.1.3.1
- Documentation
- Wiki
- Official Homepage
- [ChangeLog] [Subversion] [Lists]

Your configuration file contains settings (root with no password) that correspond to the default MySQL privileged account. Your MySQL server is running with this default, is open to intrusion, and you really should fix this security hole by setting a password for user 'root'.

Your PHP MySQL library version 5.0.51a differs from your MySQL server version 5.1.33. This may cause unpredictable behavior.

http://localhost/phpmyadmin/server_privileges.php?token=667c9546ee7e7408787ab280b13f96fa

Server: localhost

- information_schema (28)
- mysql (23)
- qlsinhvien (4)
- quanlysinhvien (1)

Please select a database

User overview

User	Host	Password	Global privileges ¹	Grant
Any	%	--	USAGE	No
root	127.0.0.1	No	ALL PRIVILEGES	Yes
root	localhost	No	ALL PRIVILEGES	Yes

Add a new User

mật khẩu

Click Add a new User để tạo mới người dùng trong MySQL

Note: phpMyAdmin gets the users' privileges directly from MySQL's privilege tables. The content of these tables may differ from the privileges the server uses, if they have been changed manually. In this case, you should **reload the privileges** before you continue.

¹ Note: MySQL privilege names are expressed in English

[Open new phpMyAdmin window](#)

+Tạo mới người dùng trong MySQL và Cấp quyền cho người dùng:

Add a new User

Login Information

User name:	Use text field: <input type="text"/>	nhập tên đăng nhập vào MySQL
Host:	Any host	nhập tên host
Password:	Use text field: <input type="password"/>	nhập mật khẩu
Re-type:	<input type="password"/>	nhập lại mật khẩu
Generate Password:	<input type="button" value="Generate"/> <input type="button" value="Copy"/>	

Database for user

None
 Create database with same name and grant all privileges
 Grant all privileges on wildcard name (username\%)

cấp quyền chỉnh sửa cấu trúc database và dữ liệu cho user

Global privileges (Check All / Uncheck All)

Note: MySQL privilege names are expressed in English

Data	Structure	Administration	Resource limits
<input type="checkbox"/> SELECT <input type="checkbox"/> INSERT <input type="checkbox"/> UPDATE <input type="checkbox"/> DELETE <input type="checkbox"/> FILE	<input type="checkbox"/> CREATE <input type="checkbox"/> ALTER <input type="checkbox"/> INDEX <input type="checkbox"/> DROP <input type="checkbox"/> CREATE TEMPORARY TABLES <input type="checkbox"/> SHOW VIEW <input type="checkbox"/> CREATE ROUTINE <input type="checkbox"/> ALTER ROUTINE <input type="checkbox"/> EXECUTE <input type="checkbox"/> CREATE VIEW <input type="checkbox"/> EVENT <input type="checkbox"/> TRIGGER	<input type="checkbox"/> GRANT <input type="checkbox"/> SUPER <input type="checkbox"/> PROCESS <input type="checkbox"/> RELOAD <input type="checkbox"/> SHUTDOWN <input type="checkbox"/> SHOW DATABASES <input type="checkbox"/> LOCK TABLES <input type="checkbox"/> REFERENCES <input type="checkbox"/> REPLICATION CLIENT <input type="checkbox"/> REPLICATION SLAVE <input type="checkbox"/> CREATE USER	Note: Setting these options to 0 (zero) removes the limit. MAX QUERIES PER HOUR <input type="text" value="0"/> MAX UPDATES PER HOUR <input type="text" value="0"/> MAX CONNECTIONS PER HOUR <input type="text" value="0"/> MAX USER CONNECTIONS <input type="text" value="0"/>

cấp quyền chỉnh sửa dữ liệu cho user

cấp quyền sử dụng quản lý MySQL cho user

giới hạn việc sử dụng tài nguyên MySQL của user

Nhấn Go để thực thi

+Xóa quyền user

Server: localhost

Databases **SQL** **Status** **Variables** **Charsets** **Engines** **Privileges**

User overview

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
User	Host	Password	Global privileges ¹		Grant																
<input type="checkbox"/> Any	%	--	USAGE		No																
<input type="checkbox"/> root	127.0.0.1	No	ALL PRIVILEGES		Yes																
<input type="checkbox"/> root	localhost	No	ALL PRIVILEGES		Yes																
<input type="checkbox"/> user	localhost	Yes	ALL PRIVILEGES		Yes																
<input type="checkbox"/> Check All / Uncheck All																					
Để edit và chỉnh sửa thông số cho người dùng																					
<input type="checkbox"/> Add a new User																					
<input type="checkbox"/> Remove selected users																					

User 'user'@'localhost': Edit Privileges

Global privileges (Check All / Uncheck All)

Note: MySQL privilege names are expressed in English

Data	Structure	Administration	Resource limits
<input checked="" type="checkbox"/> SELECT <input checked="" type="checkbox"/> INSERT <input checked="" type="checkbox"/> UPDATE <input checked="" type="checkbox"/> DELETE <input checked="" type="checkbox"/> FILE	<input checked="" type="checkbox"/> CREATE <input checked="" type="checkbox"/> ALTER <input checked="" type="checkbox"/> INDEX <input checked="" type="checkbox"/> DROP <input checked="" type="checkbox"/> CREATE TEMPORARY TABLES <input checked="" type="checkbox"/> SHOW VIEW <input checked="" type="checkbox"/> CREATE ROUTINE <input checked="" type="checkbox"/> ALTER ROUTINE <input checked="" type="checkbox"/> EXECUTE <input checked="" type="checkbox"/> CREATE VIEW <input checked="" type="checkbox"/> EVENT <input checked="" type="checkbox"/> TRIGGER	<input checked="" type="checkbox"/> GRANT <input checked="" type="checkbox"/> SUPER <input checked="" type="checkbox"/> PROCESS <input checked="" type="checkbox"/> RELOAD <input checked="" type="checkbox"/> SHUTDOWN <input checked="" type="checkbox"/> SHOW DATABASES <input checked="" type="checkbox"/> CHECK TABLES <input checked="" type="checkbox"/> REFERENCES <input checked="" type="checkbox"/> REPLICATION CLIENT <input checked="" type="checkbox"/> REPLICATION SLAVE <input checked="" type="checkbox"/> CREATE USER	Note: Setting these options to 0 (zero) removes the limit. MAX QUERIES PER HOUR: 0 MAX UPDATES PER HOUR: 0 MAX CONNECTIONS PER HOUR: 0 MAX USER_CONNECTIONS: 0

có thể chỉnh sửa lại việc cấp quyền cho user

Database-specific privileges

Database Privileges Grant Table-specific privileges Action

None

Add privileges on the following database: Use text field: 1

Change password

No Password
Password: _____ Re-type: _____
Password Hashing: MySQL 4.1+
MySQL 4.0 compatible

thay đổi mật khẩu

Change Login Information / Copy User

Login Information

User name: Use text field: user
Host: Use text field: localhost
Password: Do not change the pass
Re-type:
Generate Password: Generate Copy

thay đổi thông tin đăng nhập vào CSDL MySQL

Create a new user with the same privileges and ...

... keep the old one.
... delete the old one from the user tables.
... revoke all active privileges from the old one and delete it afterwards.
... delete the old one from the user tables and reload the privileges afterwards.

chọn delete the old one from the user tables để xóa user này

nhấn Go để thực thi

Go

10.4 Kiểu dữ liệu trong MySQL

Numeric

+ Các kiểu dữ liệu số nguyên:

Kiểu dữ liệu	Kích thước	Miền giá trị
Tinyint	1 byte	-127 => 128 hay 0..255
Smallint	2 bytes	-32768 => 32767 hay 0..65535
Mediumint	3 bytes	-8388608 => 838860 hay 0..16777215
Int	4 bytes	-2 ³¹ => 2 ³¹ - 1 hay 0..2 ³² - 1
Bigint	8 bytes	-2 ⁶³ => 2 ⁶³ - 1 hay 0..2 ⁶⁴ - 1

+ Kiểu dữ liệu true/false:

Kiểu dữ liệu	Kích thước	Miền giá trị
Bool/Boolean	1 byte	Có hai giá trị là True và False

+ Kiểu dữ liệu dạng số thập phân: decimal và numeric

Kiểu dữ liệu	Kích thước	Miền giá trị
Decimal/Numeric	4 bytes	Có hai giá trị là True và False

+Kiểu dữ liệu kiểu số thực:

Kiểu dữ liệu	Kích thước	Miền giá trị
Float	4 bytes	-3.402823466E+38 => -1.175494351E - 38; 0; 1.175494351E - 38 => 3.402823466E + 38
Double	8 bytes	-1.7976931348623157E + 308 => -2.2250738585072014E - 308; 0; 2.2250738585072014E - 308 => 1.7976931348623157E + 308

+Kiểu dữ liệu dạng ngày giờ (DateTime)

Kiểu dữ liệu	Kích thước	Miền giá trị
Date	'1000-01-01' => '9999-12-31'	Ngày với định dạng yyyy-mm-dd
Datetime	'1000-01-01 00:00:00' => '9999-12-31 23:59:59'	Ngày giờ với định dạng yyyy-mm-dd hh:mm:ss
Time	'00:00:00' => '23:59:59'	Giờ với định dạng hh:mm:ss
Year[(2 4)]	4 ký số: '1901' => '2155' 2 ký số: '1970' => '2069'	Năm với định dạng 2 ký số hoặc 4 ký số
Timestamp [(Kích cỡ định dạng)]	'1970-01-01 00:00:01'	Timestamp trình bày dưới dạng yyyy-mm-dd hh:mm:ss

+Các Kiểu dữ liệu chuỗi

Kiểu dữ liệu	Kích thước	Miền giá trị
Char	1 => 255 ký tự	Chuỗi cố định
Varchar	1 => 255 ký tự	Chuỗi động
Tinyblob	1 => 2^8 -1 bytes (255 bytes)	Kiểu đối tượng nhị phân cỡ 255 ký tự
Tinytext	1 => 2^8 -1 ký tự (255 ký tự)	Kiểu đối tượng chuỗi kích cỡ 255 ký tự
Blob	1 => 2^{16} -1 bytes (65535 bytes)	Kiểu blob cỡ 65535 ký tự
Text	1 => 2^{16} -1 ký tự (65535 ký tự)	Kiểu chuỗi dạng văn bản cỡ 65535 ký tự
Mediumblob	1 => 2^{24} -1 bytes (16777215 bytes)	Kiểu blob vừa cỡ 16777215 ký tự
Mediumtext	1 => 2^{24} -1 ký tự (16777215 ký tự)	Kiểu chuỗi dạng văn bản vừa 16777215 ký tự
Longblob	1 => 2^{32} -1 bytes (4GB)	Kiểu blob lớn khoảng 4GB ký tự

Longtext	$1 \Rightarrow 2^{32}-1$ ký tự (4GB)	Kiểu chuỗi dạng văn bản lớn khoảng 4GB ký tự
----------	---	---

10.5 Câu lệnh SQL

Nhóm phát biểu SQL

SQL dạng thay đổi và định nghĩa CSDL

Phát biểu SQL dạng CREATE : phát biểu này dùng để tạo mới 1 CSDL và bảng.

Tạo CSDL - Create Database: dùng để tạo mới cơ sở dữ liệu. Cú pháp như sau:

create database TenDatabaseMuonTao

Tạo bảng-Create table theo cú pháp sau:

Create table

```
(  
    TenCot KieuDulieu,  
    ...  
)
```

Chú ý: Cuối cột cuối cùng không có dấu phẩy

Một số qui định khi thiết kế table: (có thể thêm nhiều giá trị theo sau nhưng phải theo thứ tự như sau)

Tên cột

Kiểu dữ liệu

Khóa chính hoặc khóa ngoại(Primary key/Foreign key)

Giá trị mặc định(default)

Số tự động (auto_increment)

Null/Not Null

Thay đổi cấu trúc đối tượng với **ALTER**: là cú pháp thay đổi.

+Đổi tên table: **alter table TenTable rename TenTableMoi**

+Thêm cột vào table:

Alter alter TenTable add column TenCot KieuDuLieu

Nếu muốn thêm 1 cột mới vào cột đầu tiên thì ta sử dụng cú pháp:

Alter table TenTable add column TenCot KieuDuLieu first

Nếu muốn thêm 1 cột mới vào sau 1 cột nào đó thì ta sử dụng cú pháp:

Alter table TenTable add column TenCot KieuDuLieu after TenCotKhac

Xóa cột trong table: ta sử dụng cú pháp

Alter table TenTable drop column TenCot

Phát biểu SQL dạng **DROP**: là cú pháp dùng để **xóa** bảng hoặc database

+**Xóa database:** cú pháp

Drop database TenDatabase

+**Xóa bảng table:** cú pháp

Drop table TenTable

Phát biểu SQL thao tác dữ liệu:

Khái niệm cơ bản về select:

+ Thao tác dữ liệu là điều mà chúng ta sẽ làm thường xuyên. Để thao tác lấy dữ liệu thì chúng ta phải sử dụng cú pháp **select**, cú pháp này sẽ lấy dữ liệu theo yêu cầu mà chúng ta muốn lấy.

Vd: select * => select toàn bộ dữ liệu, ta dùng *

Select TenCot => select cụ thể tên của cột mà ta muốn lấy.

Phát biểu Select với mệnh đề From: khi sử dụng phát biểu select thì phát biểu này thường đi cùng từ khóa **From**. Từ khóa **From** này chỉ ra nguồn bảng mà chúng ta muốn select dữ liệu.

Vd: ta có bảng SinhVien gồm các cột: id, name, age và yêu cầu là select ra tên và tuổi của sinh viên từ bảng SinhVien. Cú pháp như sau:

Select name, age from SinhVien

Phát biểu SQL dạng Select với mệnh đề Where: phát biểu select đi chung với mệnh đề **where** để chỉ ra rõ điều kiện mà chúng ta muốn select. Ví dụ ta sử dụng lại bảng SinhVien như vd ở trên và đã có nhập sẵn dữ liệu. Trong dữ liệu đã nhập thì bảng SinhVien này chỉ có 2 sinh viên là có tuổi 18. Vậy giờ chúng ta chỉ viết câu **select** để lấy sinh viên chỉ 18 tuổi. Cú pháp sẽ như sau:

Select * from SinhVien where age = 18

*Chú ý: Nếu có từ khóa **where**紧跟 sau thì câu select sẽ lọc để thỏa điều kiện đi theo sau mệnh đề **where***

Mệnh đề Order By: là mệnh đề dùng để sắp xếp dữ liệu tăng dần hoặc giảm dần theo 1 cột nào đó mà ta quy định.

Vd1: Lấy toàn bộ danh sách sinh viên và sắp xếp tăng dần theo tuổi: **Select * from SinhVien order by age asc**

*Chú ý: cú pháp **order by TenCot KieuXep**. Ở đây **asc** là tăng dần, ngược lại **desc** là giảm dần.*

Vd2: Lấy danh sách sinh viên theo điều kiện là tuổi phải lớn hơn 18 và sắp xếp giảm dần theo id.

Select * from SinhVien where age > 18 order by id desc

Mệnh đề Group By: Mệnh đề **group by** được thêm vào SQL bởi vì các hàm tập hợp (như Sum) trả về 1 tập hợp các giá trị trong cột mỗi khi chúng được gọi. Và nếu không có **group by** ta không thể nào tính được tổng của các giá trị theo từng nhóm riêng lẻ trong cột. Cú pháp **group by** như sau:

Select TenCot, SUM(TenCot) from TenBang Group By TenCot

Vd: ta có bảng Sales như sau:

Company	Amount
W3Schools	5500
IBM	4500
W3Schools	7100

Query 1: Select company, sum(amount) from sales

Câu query trên sẽ trả về kết quả như hình bên dưới:

Company	SUM(Amount)
W3Schools	17100
IBM	17100
W3Schools	17100

Query 2: select company, sum(amount) from sales group by company

Câu query trên sẽ trả về kết quả như hình bên dưới:

Company	SUM(Amount)
W3Schools	12600
IBM	4500

Mệnh đề HAVING: được thêm vào SQL vì mệnh đề where không áp dụng được đối với các hàm tập hợp (như sum). Nếu không có **HAVING** ta không thể nào kiểm tra được điều kiện đối với các hàm tập hợp.

Cú pháp HAVING:

```
Select TenCot, Sum(TenCot)
from TenBang
group by TenCot
Having Sum(TenCot) DieuKienGiaTri
```

VD: sử dụng lại bảng sales ở ví dụ trên.

```
Select company,sum(amount)
From sales
Group by company
Having sum(amount) > 10000
```

Company	SUM(Amount)
W3Schools	12600

10.6 Các hàm thông dụng trong MySQL

Các hàm trong Group By

+MySQL – Count : hàm này trả về số lượng mẫu tin trong câu truy vấn trên bảng.

Company	Amount
W3Schools	5500
IBM	4500
W3Schools	7100

Vd: đếm số mẫu tin trên bảng Sales này

```
Select count(*) from sales
```

+MySQL – AVG : hàm trả về giá trị bình quân của cột hay trường (column) trong câu truy vấn.

Vd : select avg(amount) from sales

+MySQL – MIN: hàm trả về giá trị nhỏ nhất của cột hay trường trong câu truy vấn.

Vd: select min(amount) from sales

+MySQL – MAX: hàm trả về giá trị lớn nhất của cột hay trường trong câu truy vấn .

Vd : select max(amount) from sales

+MySQL – SUM : hàm trả về tổng giá trị của trường (column), cột trong câu truy vấn.

Các hàm xử lý chuỗi:

+Hàm ASCII : hàm trả về giá trị mã ASCII của ký tự bên trái của chuỗi.

Vd: select ASCII('TOI') => Kết quả là 84

+Hàm Char : hàm này chuyển đổi kiểu mã ASCII từ số nguyên sang dạng chuỗi.

Vd: select char(35) => kết quả : #

+**Hàm Upper:** hàm này chuyển đổi chuỗi sang dạng kiểu chữ hoa.

Vd: select upper('jason') =>kết quả: JASON

+**Hàm Lower:** hàm này chuyển đổi chuỗi sang dạng kiểu chữ thường.

Vd: select lower('JASON') =>kết quả: Jason

+**Hàm Len:** hàm này trả về chiều dài của chuỗi.

Vd: select len('I love you') =>kết quả là 10

+**Hàm LTrim:** hàm này loại bỏ khoảng trắng bên trái của chuỗi.

Vd: select ltrim(' jason') =>kết quả : jason

+**Hàm RTrim:** hàm này loại bỏ khoảng trắng bên phải của chuỗi

Vd: select rtrim('jason ') =>kết quả : jason

+**Hàm LEFT:** hàm này trả về chuỗi bên trái tính từ đầu chuỗi đến vị trí thứ n.

Vd: select left('abcdefghkop',4) =>kết quả là: abcd

+**Hàm RIGHT:** hàm này trả về chuỗi bên phải tính từ cuối chuỗi cho đến vị trí thứ n.

Vd: select right('abcdefghkop',4) =>kết quả là: hkop

+**Hàm Instr:** hàm này trả về vị trí bắt đầu của con trong chuỗi cần kiểm tra.

Vd: select instr('def','abcdefghkop') =>kết quả là: 4

Các hàm xử lý thời gian

+**Hàm curdate():** hàm trả về ngày tháng năm hiện hành của hệ thống. vd:

select curdate() =>kết quả sẽ ra ngày tháng năm hiện tại trên máy tính của bạn.

+**Hàm curtime():** hàm trả về giờ phút giây hiện hành của hệ thống.

Vd: select curtime() =>kết quả sẽ ra giờ phút giây hiện tại trên máy tính của bạn.

+**Hàm dayofmonth:** hàm này trả về ngày thứ mấy trong tháng.

Vd: select dayofmonth(curdate())

10.7 Các hàm về toán học

Bảng sales

Company	Amount
W3Schools	5500
IBM	4500
W3Schools	7100

Bảng users

<i>Id</i>	<i>UserName</i>	<i>Password</i>
1	admin	admin
2	user1	
3	user2	user2

+AND:

Vd: select * from sales where company = ‘IBM’ **AND** amount = 4500

+**OR:**

Vd: select * from sales where amount = 7100 **OR** amount = 5500

+**NOT:**

Vd: select * from users where password **is not null**

+**NOT IN:**

Vd: select * from users where id **not in** (‘10’,’20’)

+**BETWEEN:**

Vd: select * from users where id **between 1 and 3**

+**LIKE:**

Vd: select * from users where username **like** ‘%min’

+**NOT LIKE:**

Vd: select * from users where username **not like** ‘%min’

10.8 Phát biểu SQL dạng Select với As: phát biểu này cho phép đặt tên alias cho cột mà chúng ta select ra.

Bảng users

<i>Id</i>	<i>UserName</i>	<i>Password</i>
1	admin	admin
2	user1	
3	user2	user2

Vd: select id as MaNguoiDung, username as TenNguoiDung, password as MatKhau from users

Kết quả ra như hình sau:

	<i>MaNguoiDung</i>	<i>TenNguoiDung</i>	<i>MatKhau</i>
<input type="checkbox"/>	1	admin	admin
<input type="checkbox"/>	2	user1	
<input type="checkbox"/>	3	user2	user2

Phát biểu SQL dạng Select với Limit: phát biểu SQL dạng **SELECT** cho phép truy vấn chỉ 1 số mẫu tin tính từ vị trí thứ n đến vị trí thứ m trong table theo 1 tiêu chuẩn sắp xếp nào đó. Để có thể làm được điều này , trong phát biểu dạng SQL dạng **SELECT** chúng ta chỉ cần dùng chỉ định từ khóa **LIMIT** với số lượng mẫu tin cần lấy là từ vị trí n đến vị trí m.

Vd: Giả sử chúng ta có rất nhiều người dùng trong bảng users khoảng 100 người. Chúng ta sẽ lấy khoảng 10 người đầu tiên ra.

Select * from users **limit** 0,10

⇒ Kết quả trả về sẽ là 10 người dùng đầu tiên trong bảng users.

*Lưu ý: chúng ta có thể sử dụng kết hợp với các mệnh đề như *where*, *order by* để truy vấn đúng kết quả theo ý muốn.

Phát biểu SQL dạng Select với Distinct: phát biểu này cho phép ta xử lý tình huống sau: Nếu có 1 hay nhiều bảng kết nối với nhau thì lúc đó sẽ xảy ra tình trạng trùng lặp mẫu tin nhưng trong trường hợp bạn chỉ muốn lấy ra 1 mẫu tin trong tập mẫu tin trùng

lắp ấy. Lúc đó, chúng ta sử dụng phát biểu SQL dạng **SELECT** với định khóa **DISTINCT**.

Vd: select **distinct** id,total,amount from ProductDetails

Nhập dữ liệu bằng phát biểu dạng **INSERT**:

- Khi cần thêm mẫu tin vào bảng (table) trong CSDL MySQL ,bạn có nhiều cách để thực hiện công việc này,nhưng để sử dụng các phát biểu SQL mang tính chuyên nghiệp bạn cần sử dụng phát biểu **INSERT**.Khi thêm dữ liệu ,cần chú ý kiểu dữ liệu giống hoặc tương ứng với kiểu dữ liệu đã khai báo của cột (column) ,nếu không phù hợp thì lỗi sẽ phát sinh

Insert vào bảng lấy giá trị cụ thể: cấu trúc như sau:

INSERT INTO TenBang (cột 1,cột 2,...) VALUES (giá trị 1,giá trị 2,...)

Vd: insert into users (id,username,password) values ('5','user3','user3')

Chú ý: số lượng giá trị insert vào phải tương đương với số cột mà ta chỉ định sẽ insert vào bảng.

Insert vào bảng lấy giá trị từ bảng khác: cấu trúc sẽ như sau:

INSERT INTO TENBANG1 SELECT * FROM TENBANG2

Vd: ta muốn copy toàn bộ user bên bảng users sang 1 bảng khác với tên user2. Câu truy vấn như sau:

Insert into user2 select * from users

Insert vào bảng lấy giá trị cụ thể,bảng khác: để insert vào table với giá trị từ 1 bảng khác ta dùng cấu trúc như sau:

INSERT INTO TenBang1 (DanhSachCot) SELECT (DanhSachCot) FROM TenBang2 WHERE CONDITION

Vd: ta muốn copy toàn bộ user bên bảng users với các giá trị cụ thể như id,username,password sang 1 bảng khác với tên user2. Câu truy vấn như sau:

Insert into user2 (id,username,password) select id,username,password from users

Phát biểu SQL dạng Update: phát biểu này dùng để cập nhật lại dữ liệu đã tồn tại trong bảng. Khi **UPDATE** dùng để cập nhật dữ liệu cho 1 mẫu tin chỉ định nào đó thường thì lệnh này được dùng chung với mệnh đề **WHERE**. Nếu cập nhật tất cả các mẫu tin trong bảng thì ta có thể bỏ mệnh đề **WHERE** đi. Cấu trúc như sau:

UPDATE TENBANG SET <cot1>=<giatri>,<cot2>=<giatri>,...

WHERE DIEUKIEN

Phát biểu SQL dạng Delete: phát biểu này dùng để xóa 1 dòng dữ liệu đã tồn tại trong bảng. Cú pháp như sau:

DELETE FROM TENBANG WHERE DIEUKIEN

10.9 Phát biểu SQL dạng Join

Khái niệm quan hệ: Trong cơ sở dữ liệu mysql thì các bảng table luôn có quan hệ với nhau bởi các ràng buộc khóa chính khóa ngoại.

Bảng Users

Id	UserName	Password
0	admin	admin
1	user1	user1
2	user2	user2

3	user3	user3
---	-------	-------

Bảng UserDetails

Id	IdUser	Address	Age
0	0	123 Nguyễn Trãi	25
1	1	456 Lê Lai	26
2	2	789 Lê Lợi	28
3	3	910 Nguyễn Đình Chiểu	30

Ta thấy 2 bảng này quan hệ với nhau bởi cột IdUser bên bảng UserDetails với cột Id bên bảng Users.

Khái niệm về mệnh đề Join: Join là sự kết hợp từ nhiều bảng table, cho phép ta lấy nhiều dữ liệu từ nhiều nguồn (table) khác nhau mà các bảng đó có mối liên hệ bởi các ràng buộc khóa chính (primary key) và khóa ngoại (foreign key). 1 ví dụ về join như sau:

Bảng Author

Id	Name
1	Anna
2	George
3	Sean
4	Mary

Bảng Books

BookId	AuthorID	Title
1	1	Book 1
2	1	Book 2
3	2	Book 3
4	3	Book 4

Đây là cách join bình thường 2 bảng, ta sẽ áp dụng 2 bảng trên dựa vào cột AuthorID. Câu query như sau:

Select * from Author, Books where Author.Id = Books.AuthorID

Kết quả như sau:

Id	Name	BookId	AuthorID	Title
1	Anna	1	1	Book 1
1	Anna	2	1	Book 2
2	George	3	2	Book 3
3	Sean	4	3	Book 4

Có 3 loại join: inner join, left join, right join.

Mệnh đề inner join: inner join cũng giống cách join bình thường nhưng cú pháp khác và tốc độ join nhanh hơn nhiều so với cách bình thường. Cú pháp như:

Select * from Author inner join Books on Author.Id = Books.AuthorID

Kết quả thì cũng y như bảng kết quả phía trên.

Mệnh đề left join: left join là cách join ưu tiên cho các cột có giá trị trong bảng bên trái join với bảng bên phải. Cú pháp như sau:

Select * from Author left join Books on Author.Id = Books. AuthorID

Kết quả như sau:

Id	Name	BookId	AuthorID	Title
1	Anna	1	1	Book 1
1	Anna	2	1	Book 2
2	George	3	2	Book 3
3	Sean	4	3	Book 4
4	Mary	NULL	NULL	NULL

Mệnh đề right join: right join cũng giống với left join nhưng ngược lại là bảng bên phải. Cú pháp như sau:

Select * from Author right join Books on Author.Id = Books. AuthorID

Kết quả như sau:

Id	Name	BookId	AuthorID	Title
1	Anna	1	1	Book 1
1	Anna	2	1	Book 2
2	George	3	2	Book 3
3	Sean	4	3	Book 4

Phép toán hợp (union): toán tử UNION cho phép ta hợp các kết quả của hai hay nhiều truy vấn thành một tập kết quả duy nhất. Join thì kết nối dữ liệu theo chiều ngang còn union thì kết nối dữ liệu theo chiều dọc. Cú pháp như sau:

<Query 1> UNION <Query 2>

Bảng Product

<i>Id</i>	<i>Product</i>
0	Mp3
1	Television
2	Laptop

Bảng ProductTemp

<i>Id</i>	<i>Product</i>
6	Mp4
7	Mobile Phone
8	USB

Vd: chúng ta dùng union để nối 2 bảng Product và ProductTemp lại với nhau.

Select * from Product UNION Select * from ProductTemp

Kết quả sẽ trả về danh sách cả 2 loại sản phẩm trong cùng 1 recordset.

<i>Id</i>	<i>Product</i>
0	Mp3
1	Television
2	Laptop

6	<i>Mp4</i>
7	<i>Mobile Phone</i>
8	<i>USB</i>

Lưu ý:

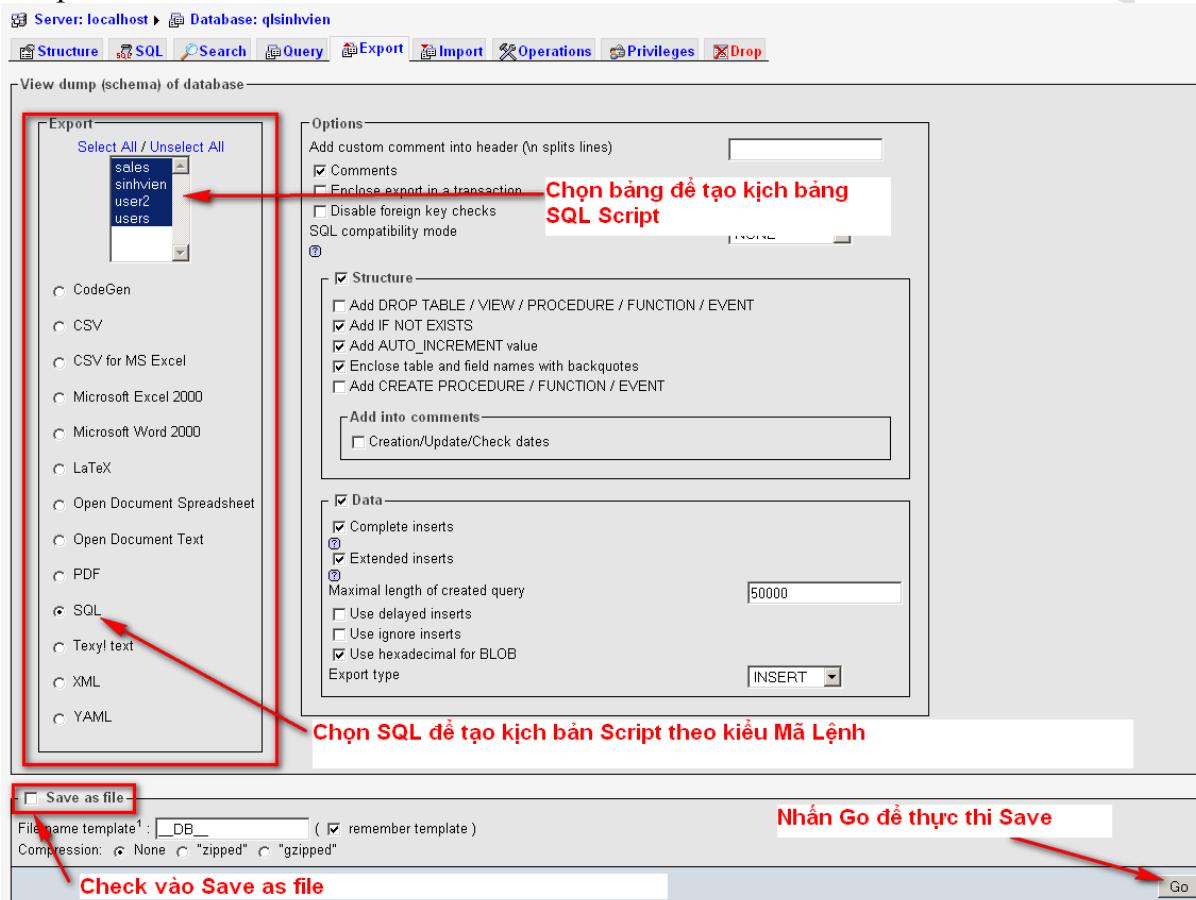
+ Tất cả những truy vấn trong UNION phải cùng số cột hay trường. Nếu truy vấn thứ nhất có hai cột thì truy vấn thứ hai được sử dụng UNION cũng phải có hai cột tương tự.

+ Kiểu dữ liệu trong các cột của truy vấn 2 tương thích với kiểu dữ liệu các cột tương ứng trong truy vấn thứ nhất.

+ Trong UNION chúng ta có thể kết hợp nhiều câu truy vấn lại với nhau.

10.10 Tạo kịch bản SQL- SQL Script:

thực hiện theo các bước sau để tạo kịch bản SQL Script như sau:

**Câu hỏi ôn tập:****Bảng Users**

Id	UserName	Password
0	admin	admin
1	user1	user1
2	user2	user2
3	user3	user3

Bảng UserDetails

Id	IdUser	Address	Age
0	0	123 Nguyễn Trãi	25

1	1	456 Lê Lai	26
2	2	789 Lê Lợi	28
3	3	910 Nguyễn Đình Chiểu	30

- Tạo bảng như trên.
- Tất cả dùng lệnh: (tập làm quen với lệnh)
- Thêm vào từng bảng vài user mới.
- Cập nhật lại password của bất kỳ user.
- Delete 1 user bất kỳ.
- Select tất cả dữ liệu có trong bảng user và userdetail.
- Select thông tin như sau:username,password,address,age (sử dụng lệnh join) của admin.

CHƯƠNG 11

11.1 Stored procedure trong MySQL

+**Giới thiệu:** stored procedure được định nghĩa như 1 tập các khai báo sql được lưu trữ ngay trong cơ sở dữ liệu (Database) và sau đó được triệu gọi bởi 1 chương trình, 1 trigger hay thậm chí là 1 stored procedure khác. Hầu hết các hệ quản trị CSDL đều hỗ trợ stored procedure đệ quy. Tuy nhiên, MySQL lại không hỗ trợ tốt tính năng này. Để sử dụng được stored procedure theo kiểu đệ quy, chúng ta cần kiểm tra phiên bản của MySQL (version 5.0 trở lên).

+**Ưu điểm của stored procedure:**

- Stored procedure làm tăng khả năng thực thi của ứng dụng.
- Stored procedure làm giảm lưu lượng thông tin giao tiếp giữa ứng dụng và database server.
- Stored procedure có thể tái sử dụng sang nhiều ứng dụng.

+**Khuyết điểm:**

- Làm cho database server phải tồn tại nhiều tài nguyên về cả bộ nhớ lẫn xử lý.

Stored Procedure đơn giản với chức năng là lấy toàn bộ danh sách sinh viên được lưu trong bảng sinhvien.

```
DELIMITER //
CREATE PROCEDURE GetAllSV()
BEGIN
    SELECT * FROM sinhvien;
END //
DELIMITER ;
```

Trong đoạn code trên, phát biểu DELIMITER được dùng để thay thế cú pháp phân cách chuẩn từ dấu chấm phẩy thành dấu khác (ở đây là dấu 2 gạch). Chúng ta có thể viết nhiều dòng SQL trong 1 stored procedure bằng việc sử dụng dấu chấm phẩy làm dấu phân cách dòng. Sau từ khóa END lại thấy có dấu 2 gạch để thông báo cho trình biên dịch biết rằng đã hết stored procedure. Cuối cùng là trả về dấu chấm phẩy.

⇒ Cú pháp stored procedure như sau:

```
DELIMITER //
CREATE PROCEDURE PROCEDURE_NAME()
BEGIN
    //các dòng lệnh
END //
DELIMITER ;
```

+**Cách gọi stored procedure trong MySQL:**

⇒ Cú pháp gọi như sau:

CALL PROCEDURE_NAME()

+Cách khai báo biến trong stored procedure: ta dùng từ khóa **DECLARE**

DECLARE [Tên Biến] [Kiểu Dữ Liệu]

+Cách gán giá trị cho 1 biến: ta dùng từ khóa **SET**:

SET [Tên biến] = [Giá trị]

+Stored Procedure có thêm các tham số như sau:

```
DELIMITER //
CREATE PROCEDURE STOREPROCEDURE_NAME([Kiểu tham số] [Tên Biến 1] [kiểu dữ liệu], ([Kiểu tham số] [Tên Biến 2] [kiểu dữ liệu], ...)
BEGIN
    //Các dòng lệnh
END //
DELIMITER ;
```

***Lưu ý:** Kiểu tham số: gồm 3 loại như sau.

1. **IN**: tham số nhận vào stored procedure
2. **OUT**: tham số đầu ra trả về 1 giá trị.
3. **INOUT**: tham số này là sự kết hợp của 2 tham số kia. Tham số này nhận vào 1 giá trị và trả về 1 giá trị mới ở đầu ra.

+Cách gọi **stored procedure có tham số** như sau:

CALL STOREPROCEDURE_NAME([Tham số 1],[Tham số 2], ...)

***Lưu ý:**

- Tùy thuộc vào stored procedure có bao nhiêu tham số thì gọi stored procedure cũng bấy nhiêu tham số.
- Kiểu dữ liệu của các tham số bắt buộc phải giống với kiểu dữ liệu của các tham số khi viết stored procedure.
- Khi muốn lấy tham số đầu ra, chúng ta sử dụng SELECT để select tên tham số cần lấy.

+Mệnh đề điều kiện **IF**: cú pháp như sau:

```
IF expression THEN commands
[ELSEIF expression THEN commands]
[ELSE commands]
END IF;
```

***Lưu ý:**

1. **Expression**: là điều kiện cần kiểm tra.
2. **Commands**: là đoạn lệnh sẽ thực thi nếu thỏa điều kiện expression.

+Mệnh đề **CASE**: cú pháp như sau:

```
CASE
  WHEN expression THEN commands
```

...
WHEN expression **THEN** commands
ELSE commands
END CASE;

***Lưu ý:**

1. **Expression:** là điều kiện cần kiểm tra.
2. **Commands:** là đoạn lệnh sẽ thực thi nếu thỏa điều kiện expression.

+ Vòng lặp **WHILE**: cú pháp như sau:

WHILE expression **DO**
 Commands
END WHILE

***Lưu ý:** **Commands** chỉ được thực thi khi thỏa mãn điều kiện expression trong vòng lặp **WHILE**. Ngược lại thì không thực hiện đoạn commands đó.

+ Vòng lặp **REPEAT**: cú pháp như sau:

REPEAT
 Statements;
UNTIL expression
END REPEAT

+Vòng lặp **Loop, Leave,Iterate**:

1. Phát biểu leave cho phép bạn rời khỏi vòng lặp
2. Phát biểu Iterate cho phép bạn bắt đầu vòng lặp trở lại

```
DELIMITER //
DROP PROCEDURE IF EXISTS LOOPLoopProc$$
CREATE PROCEDURE LOOPLoopProc()
BEGIN
  DECLARE x INT;
  DECLARE str VARCHAR(255);
  SET x = 1;
  SET str = "";
loop_label: LOOP
  IF x > 10 THEN
    LEAVE loop_label;
  END IF;
  SET x = x + 1;
  IF (x mod 2) THEN
    ITERATE loop_label;
  ELSE
    SET str = CONCAT(str,x,',');
  END IF;
END LOOP;
```

```

SELECT str;
END//
```

Stored procedure trên chỉ tạo dựng 1 chuỗi với các số chẵn. Đầu tiên, định nghĩa 1 nhãn loop. Nếu biến $x > 10$ thì vòng lặp kết thúc thông qua biểu thức leave. Nếu x lẻ thì iterate sẽ bỏ qua các dòng lệnh bên dưới nó và tiếp tục vòng lặp mới. Ngược lại thì block code trong phát biểu else sẽ thực thi để tạo chuỗi với các số chẵn.

11.2 Con trỏ trong stored procedure: con trỏ được dùng để duyệt theo từng hàng trong bảng kết quả trả về từ câu truy vấn và xử lý các hàng đó. Từ MySQL phiên bản 5.0 trở lên đều hỗ trợ con trỏ với các thuộc tính sau đây:

- **Read only:** bạn không thể update con trỏ.
- **Non-scrollable:** con trỏ chỉ có thể di chuyển theo 1 hướng và không thể bỏ qua bất cứ 1 hàng nào, trượt tới hay trượt lùi trong bảng kết quả.
- **Asensitive:** chúng ta nên tránh việc update 1 bảng table trong khi con trỏ vẫn đang được mở trong chính bảng đó. Nếu không sẽ cho ra kết quả sai.

B1: Khai báo con trỏ với phát biểu DECLARE:

DECLARE [Tên con trỏ] CURSOR FOR [Câu truy vấn];

B2: Mở con trỏ sử dụng phát biểu OPEN:

OPEN [Tên con trỏ];

*Lưu ý: chúng ta phải mở con trỏ trước khi duyệt trong bảng kết quả.

B3: Nhận dữ liệu ở dòng hiện tại trong bảng kết quả và di chuyển con trỏ đến dòng kế tiếp bằng phát biểu FETCH:

FETCH [Tên con trỏ] INTO [Danh sách các biến];

Cuối cùng đóng con trỏ và giải phóng bộ nhớ sử dụng phát biểu CLOSE:

CLOSE [Tên con trỏ];

VD: Ví dụ này tổng quát toàn bộ những kiến thức về stored procedure đã học phía trên

```

82  DELIMITER //
83  CREATE PROCEDURE SP_CURS2(IN MASV VARCHAR(100),OUT KETQUA NVARCHAR(100))
84  BEGIN
85      DECLARE HOTEN NVARCHAR(100);
86      DECLARE NO_MORE_ROW INT;
87      DECLARE CURSO01 CURSOR
88      FOR
89          SELECT NAME FROM SINHVIEN WHERE ID = MASV;
90          DECLARE CONTINUE HANDLER FOR NOT FOUND
91              SET NO_MORE_ROW = 1;
92          OPEN CURSO01;
93          MY_LOOP:LOOP
94              FETCH CURSO01 INTO HOTEN;
95              IF NO_MORE_ROW THEN
96                  SET KETQUA = N'KHÔNG CÓ';
97                  CLOSE CURSO01;
98                  LEAVE MY_LOOP;
99              END IF;
100             SET KETQUA = HOTEN;
101             CLOSE CURSO01;
102             LEAVE MY_LOOP;
103         END LOOP MY_LOOP;
104     END //
105     DELIMITER ;

```

*Chú thích: Mục đích của stored procedure SP_CURS2 này là xuất ra tên sinh viên có mã sinh viên do người dùng nhập vào. SP_CURS2 nhận vào 2 tham số:

1. Đầu vào nhận vào tham số mã sinh viên.
2. Đầu ra trả về tham số tên sinh viên.

```

DECLARE CONTINUE HANDLER FOR NOT FOUND
    SET NO_MORE_ROW = 1;

```

Dòng này sẽ kiểm tra xem có thỏa mãn câu select phía trên không ?
Nếu không thỏa mãn thì sẽ vào đoạn code sau:

```

IF NO_MORE_ROW THEN
    SET KETQUA = N'KHÔNG CÓ';
    CLOSE CURSO01;
    LEAVE MY_LOOP;
END IF;

```

Nếu thỏa mãn thì sẽ vào:

```

SET KETQUA = HOTEN;
CLOSE CURSO01;
LEAVE MY_LOOP;

```

11.3 Trigger

+**Định nghĩa:** trigger là 1 đối tượng được định danh trong CSDL và được gắn chặt với 1 sự kiện xảy ra trên 1 bảng nào đó. Điều này có nghĩa là trigger sẽ tự động thực thi khi xảy ra 1 sự kiện nào đó trên bảng. Những sự kiện đó là : INSERT, UPDATE, DELETE. Trigger chỉ được tích hợp từ MySQL phiên bản 5.0 trở lên.

+**Tạo mới 1 trigger:**

```

CREATE TRIGGER TriggerName
TriggerTime TriggerEvent
ON TableName FOR EACH ROW
BEGIN
    STATEMENTS
END

```

*Chú thích:

- TriggerName: là tên của trigger.
- TriggerTime: là thời điểm trigger được kích hoạt. Ở đây có thể là Before(Trước)/After(Sau).
- TriggerEvent: là các sự kiện Insert,Update,Delete.
- TableName: là tên của bảng mà chúng ta muốn tạo trigger.
- Statements: là các câu lệnh sẽ được thực thi.

+**Xóa Trigger:** **DROP TRIGGER [Tên trigger];**

```

CREATE TABLE test1(a1 INT);
CREATE TABLE test2(a2 INT);
CREATE TABLE test3(a3 INT NOT NULL AUTO_INCREMENT PRIMARY KEY);
CREATE TABLE test4(
    a4 INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    b4 INT DEFAULT 0
);

DELIMITER |

CREATE TRIGGER testref BEFORE INSERT ON test1
FOR EACH ROW BEGIN
    INSERT INTO test2 SET a2 = NEW.a1;
    DELETE FROM test3 WHERE a3 = NEW.a1;
    UPDATE test4 SET b4 = b4 + 1 WHERE a4 = NEW.a1;
END;

DELIMITER ;

INSERT INTO test3 (a3) VALUES
    (NULL), (NULL), (NULL), (NULL), (NULL),
    (NULL), (NULL), (NULL), (NULL);

INSERT INTO test4 (a4) VALUES
    (0), (0), (0), (0), (0), (0), (0), (0);

```

và cuối cùng là insert vào bảng test 1:

```
INSERT INTO test1 VALUES (1), (3), (1), (7), (1), (8), (4), (4);
```

Và kết quả sẽ như những hình sau:

```
mysql> SELECT * FROM test1;
+---+
| a1 |
+---+
| 1 |
| 3 |
| 1 |
| 7 |
| 1 |
| 8 |
| 4 |
| 4 |
+---+
8 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM test2;
+---+
| a2 |
+---+
| 1 |
| 3 |
| 1 |
| 7 |
| 1 |
| 8 |
| 4 |
| 4 |
+---+
8 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM test3;
+---+
| a3 |
+---+
| 2 |
| 5 |
| 6 |
| 9 |
| 10 |
+---+
5 rows in set (0.00 sec)

mysql> SELECT * FROM test4;
+---+---+
| a4 | b4 |
+---+---+
| 1 | 3 |
| 2 | 0 |
| 3 | 1 |
| 4 | 2 |
| 5 | 0 |
| 6 | 0 |
| 7 | 1 |
| 8 | 1 |
| 9 | 0 |
| 10 | 0 |
+---+---+
10 rows in set (0.00 sec)
```

Câu hỏi ôn tập:

- Với database như câu hỏi ôn tập trong chương 11.
- Tạo store procedure select tất cả các user.
- Tạo store procedure select user có độ tuổi phù hợp từ 25 đến dưới 30 với tham số đầu vào là username.

CHƯƠNG 12

12.1 Kết nối CSDL

+Tạo CSDL QuanLySinhVien và tạo bảng sinhvien như sau:

	id	name	address	age
<input type="checkbox"/>		sv01	Nguyễn Văn Hoa	16
<input type="checkbox"/>		sv02	Lê Thị Thúy	16
<input type="checkbox"/>		sv03	Trần Văn Tuấn	16
<input type="checkbox"/>		sv04	Lê Tuấn Anh	16
<input type="checkbox"/>		sv05	Nguyễn Quốc Trung	16
<input type="checkbox"/>		sv06	Phước Lâm	16

+Kết nối CSDL QuanLySinhVien: Cú pháp kết nối như sau

mysql_connect(host,tên_truy_cập,mật_khẩu);

Trong đó:

- **host** là chuỗi chứa tên (hoặc địa chỉ IP) của máy chủ cài đặt MySQL.
- **tên_truy_cập** là chuỗi chứa tên truy cập CSDL cần kết nối
- **mật_khẩu** là chuỗi chứa mật khẩu tương ứng với tên truy cập

Vd:

```
<?php
$conn = mysql_connect("localhost","root","");
if(!$conn)
{
    echo "Không thể kết nối đến MySQL.";
}
```

+Để kết nối đến 1 CSDL nào đó, ta dùng hàm **mysql_select_db**:

mysql_select_db(tên CSDL);

Vd:

```
<?php
$conn = mysql_connect("localhost","root","");
if(!$conn)
{
    echo "Không thể kết nối đến MySQL.";
}
$db = mysql_select_db("QuanLySinhVien");
if(!$db)
{
    echo "Không thể kết nối đến CSDL QuanLySinhVien.";
}
?>
```

+Để đóng kết nối đến CSDL, ta dùng hàm **mysql_close**. Hàm này có nhiệm vụ đóng kết nối đến CSDL có tên định danh được tạo ra bởi hàm **mysql_connect()**.

Cú pháp: **mysql_close(tên kết nối);**

Vd:

```
<?php
$conn = mysql_connect("localhost","root","");
if(!$conn)
{
    echo "Không thể kết nối đến MySQL.";
}
mysql_close($conn);
?>
```

Truy vấn dữ liệu

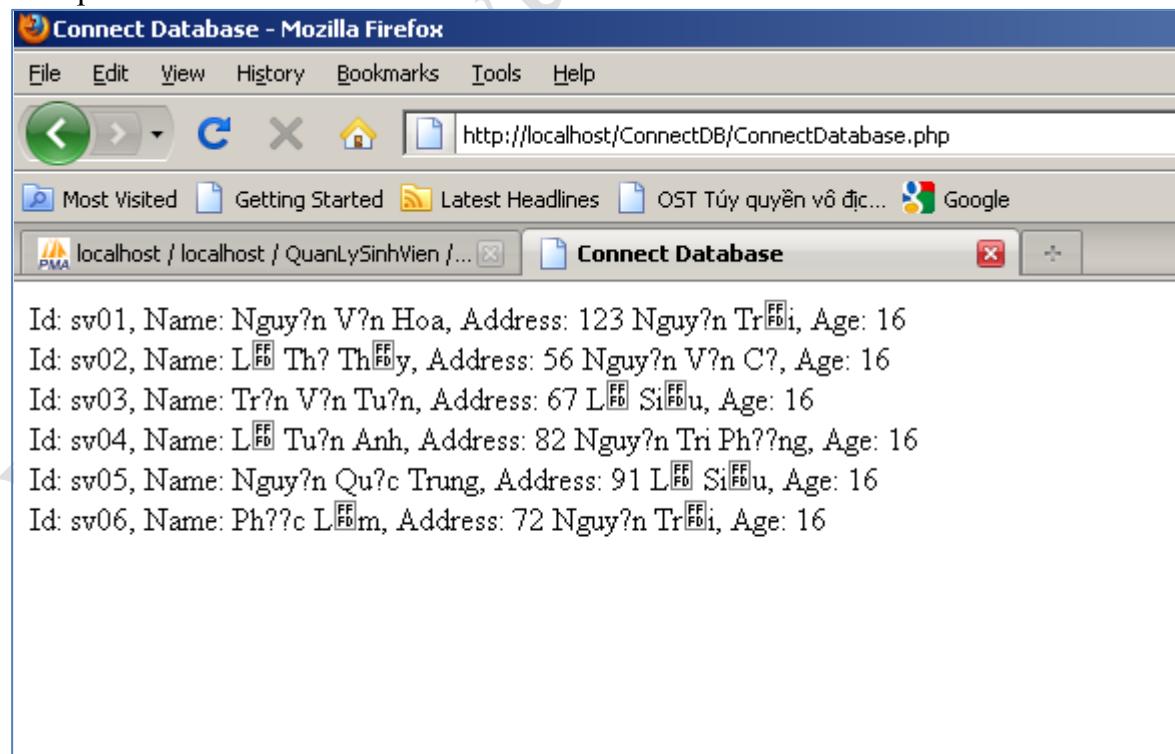
+Để đọc bảng quản lý sinh viên dùng **mysql_query(...)**

+Để duyệt các mẫu tin trong mảng sinh viên dùng **mysql_fetch_object(...)** và cứ mỗi dòng thì xuất ra thông tin của 1 sinh viên.

Vd:

```
<?php
$conn = mysql_connect("localhost","root","");
if(!$conn)
{
    echo "Không thể kết nối đến MySQL.";
}
$db = mysql_select_db("QuanLySinhVien");
if(!$db)
{
    echo "Không thể kết nối đến CSDL QuanLySinhVien.";
}
$result = mysql_query("select * from sinhvien");
if(mysql_num_rows($result) > 0) //kiểm tra có dữ liệu hay không
{
    while($row = mysql_fetch_object($result))//đọc từng phần tử trong mảng
    {
        //xuất thông tin sinh viên
        echo "Id: ".$row->id.", Name: ".$row->name.", Address: ".$row->address.", Age: ".$row->age."<br />";
    }
}
else
{
    echo "Không có dữ liệu";
}
?>
```

Kết quả:

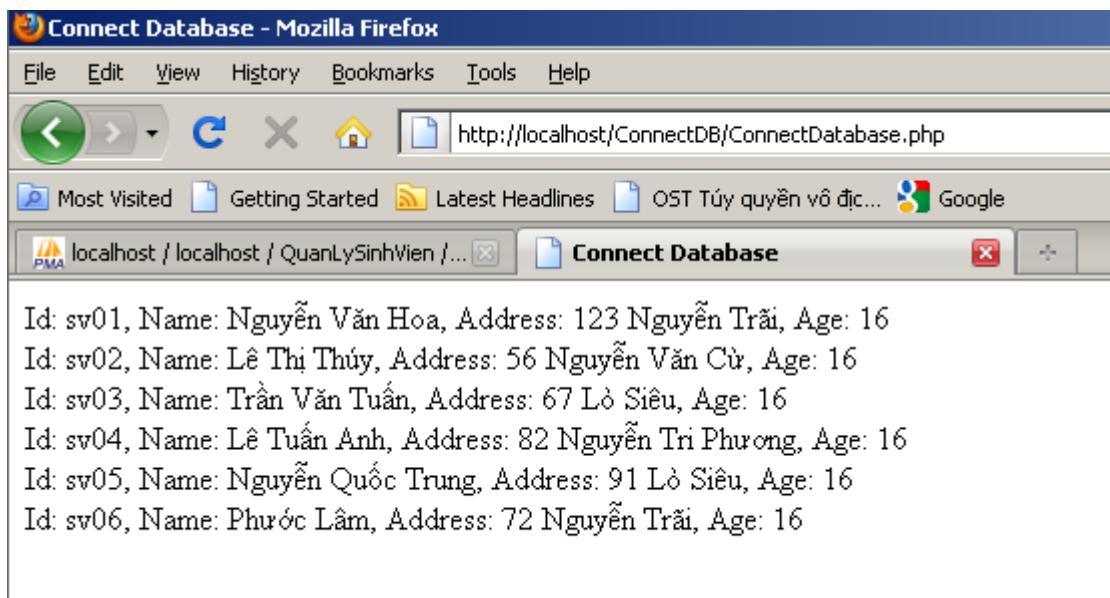


Lưu ý: Nếu xuất ra thấy lỗi như hình trên thì chỉ cần dùng thêm đoạn code **set names 'utf8'** vào là sẽ hiện thị được Tiếng Việt nếu dữ liệu của bạn là Tiếng Việt.

```

    }
    mysql_query("set names 'utf8'");
    $result = mysql_query("select * from sinhvien");
    if(mysql_num_rows($result) > 0) //kiểm tra có dữ liệu
    {
        while($row = mysql_fetch_object($result))
        {
            echo "Id: " . $row->id . ", Name: " . $row->name . ", Address: " . $row->address . ", Age: " . $row->age . "  
";
        }
    }
}

```



Thêm mẫu tin:

```

<?php
//kết nối MySQL và CSDL QuanLySinhVien
$conn = ...
//thêm 1 sinh viên mới vào CSDL bảng sinhvien
mysql_query("insert into sinhvien(id,name,address,age)
values ('13','Nguyễn Văn Tú','892 Nguyễn Trãi','16')");

?>

```

Kết quả 1 dòng mới được insert vào bảng sinhvien trong CSDL MySQL

	id	name	address	age
<input type="checkbox"/>		sv01	Nguyễn Văn Hoa	16
<input type="checkbox"/>		sv02	Lê Thị Thúy	16
<input type="checkbox"/>		sv03	Trần Văn Tuấn	16
<input type="checkbox"/>		sv04	Lê Tuấn Anh	16
<input type="checkbox"/>		sv05	Nguyễn Quốc Trung	16
<input type="checkbox"/>		sv06	Phước Lâm	16
<input type="checkbox"/>		13	Nguyễn Văn Tú	16

Cập nhật mẫu tin:

	id	name	address	age
<input type="checkbox"/>		sv01	Nguyễn Văn Hoa	16
<input type="checkbox"/>		sv02	Lê Thị Thúy	16
<input type="checkbox"/>		sv03	Trần Văn Tuấn	16
<input type="checkbox"/>		sv04	Lê Tuấn Anh	16
<input type="checkbox"/>		sv05	Nguyễn Quốc Trung	16
<input type="checkbox"/>		sv06	Phước Lâm	16
<input type="checkbox"/>		13	Nguyễn Văn Tú	16

Chúng ta sẽ cập nhật lại thông tin cho sinh viên mới vừa thêm vào có id là 13 với chi tiết cập nhật như sau:

- Address sẽ sửa lại là 192 Lò Siêu
- Age sẽ sửa lại là 17

```
<?php
//kết nối MySQL và CSDL QuanLySinhVien
$conn = ...
//cập nhật sinh viên
mysql_query("update sinhvien
            set address = '192 Lò Siêu', age = '17'
            where id = '13'");
?>
```

Kết quả trong CSDL MySQL thì thông tin sinh viên có id = 13 đã thay đổi

	id	name	address	age
<input type="checkbox"/>		sv01	Nguyễn Văn Hoa	16
<input type="checkbox"/>		sv02	Lê Thị Thúy	16
<input type="checkbox"/>		sv03	Trần Văn Tuấn	16
<input type="checkbox"/>		sv04	Lê Tuấn Anh	16
<input type="checkbox"/>		sv05	Nguyễn Quốc Trung	16
<input type="checkbox"/>		sv06	Phước Lâm	16
<input type="checkbox"/>		13	Nguyễn Văn Tú	17

Xóa mẫu tin: Bây giờ, chúng ta sẽ tiến hành xóa sinh viên có id là 13.

```
<?php
//kết nối MySQL và CSDL QuanLySinhVien
$conn = ...
//xóa sinh viên có id là 13
mysql_query("delete from sinhvien
            where id = '13'");
?>
```

Kết quả sinh viên có id là 13 đã bị xóa sau khi thực thi đoạn code trên.

		id	name	address	age
<input type="checkbox"/>	 	sv01	Nguyễn Văn Hoa	123 Nguyễn Trãi	16
<input type="checkbox"/>	 	sv02	Lê Thị Thúy	56 Nguyễn Văn Cừ	16
<input type="checkbox"/>	 	sv03	Trần Văn Tuấn	67 Lò Siêu	16
<input type="checkbox"/>	 	sv04	Lê Tuấn Anh	82 Nguyễn Tri Phương	16
<input type="checkbox"/>	 	sv05	Nguyễn Quốc Trung	91 Lò Siêu	16
<input type="checkbox"/>	 	sv06	Phước Lâm	72 Nguyễn Trãi	16

Câu hỏi ôn tập:

-Tạo form register.php với thông tin:

- Tên người dùng
- Mật khẩu

Sử dụng hướng đối tượng với đối tượng là người dùng bao gồm các thông tin trên.

Viết thêm các phương thức: kiểm tra người dùng đã tồn tại chưa và các phương get và set thông tin người dùng vào trong 1 đối tượng.

Kết nối database để viết các phương thức kiểm tra.

CHƯƠNG 13

Liệt kê dữ liệu

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Hiển thị danh sách sinh viên</title>
</head>

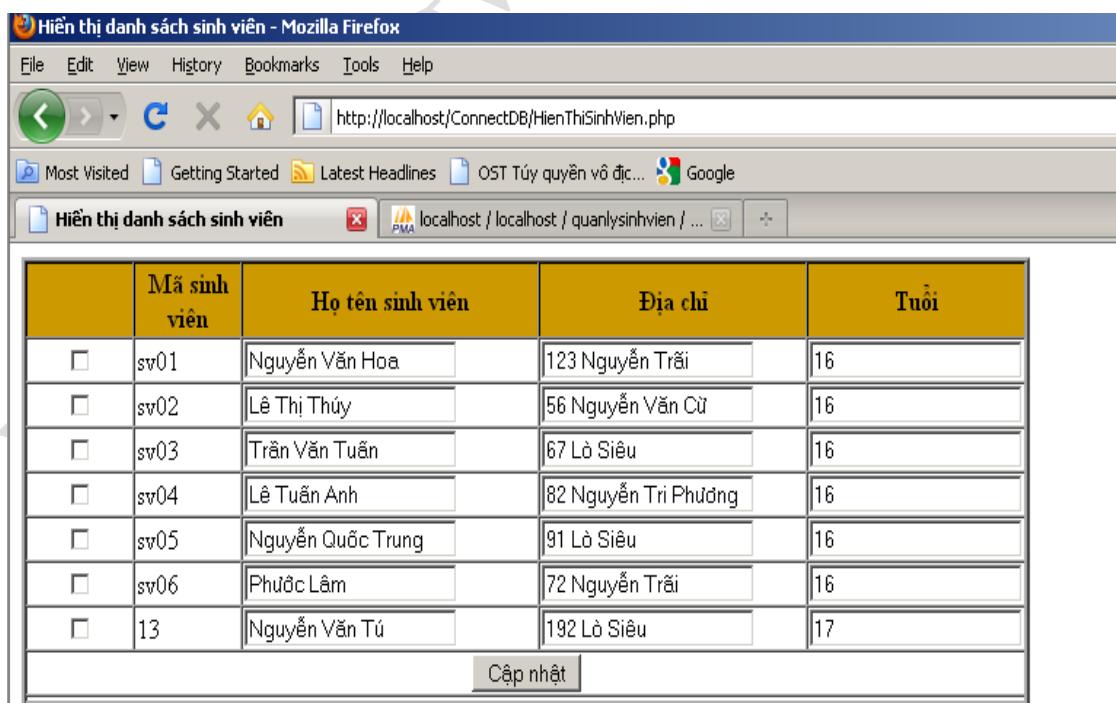
<body>
<?php
//kết nối MySQL và CSDL QuanLySinhVien
$conn = mysql_connect("localhost","root","");
if(!$conn)
{
    echo "Không thể kết nối đến MySQL.";
}
//select database QuanLySinhVien
$db = mysql_select_db("QuanLySinhVien");
if(!$db)
{
    echo "Không thể kết nối đến CSDL QuanLySinhVien.";
}
mysql_query("set names 'utf8'");
$result = mysql_query("select * from sinhvien");
?>
<table border="1" style="border:ridge; border-spacing:0">
<tr bgcolor="#CC9900">
    <th width="70">Id</td>
    <th width="200">Name</td>
    <th width="180">Address</td>
    <th width="70">Age</td>
</tr>
<?php
while($row = mysql_fetch_object($result))
{
    echo "<tr>";
    echo "<td>".$row->id."</td>";
    echo "<td>".$row->name."</td>";
    echo "<td>".$row->address."</td>";
    echo "<td>".$row->age."</td>";
    echo "</tr>";
}
?>
</table>
</body>
</html>
```



A screenshot of a Mozilla Firefox browser window. The title bar says "Hiển thị danh sách sinh viên - Mozilla Firefox". The address bar shows the URL "http://localhost/ConnectDB/HienThiSinhVien.php". The page content displays a table with four columns: Id, Name, Address, and Age. The data is as follows:

Id	Name	Address	Age
sv01	Nguyễn Văn Hoa	123 Nguyễn Trãi	16
sv02	Lê Thị Thúy	56 Nguyễn Văn Cừ	16
sv03	Trần Văn Tuấn	67 Lò Siêu	16
sv04	Lê Tuấn Anh	82 Nguyễn Tri Phương	16
sv05	Nguyễn Quốc Trung	91 Lò Siêu	16
sv06	Phước Lâm	72 Nguyễn Trãi	16
13	Nguyễn Văn Tú	192 Lò Siêu	17

Cập nhật nhiều mẫu tin: chúng ta sẽ thêm vào table những checkbox, chỉnh lại giao diện 1 vài phần và thêm button để cập nhật mẫu tin.



A screenshot of a Mozilla Firefox browser window. The title bar says "Hiển thị danh sách sinh viên - Mozilla Firefox". The address bar shows the URL "http://localhost/ConnectDB/HienThiSinhVien.php". The page content displays a table with five columns: Mã sinh viên, Họ tên sinh viên, Địa chỉ, and Tuổi. The first column contains checkboxes. The data is the same as the previous table. A "Cập nhật" button is at the bottom of the table.

Mã sinh viên	Họ tên sinh viên	Địa chỉ	Tuổi
<input type="checkbox"/>	sv01	123 Nguyễn Trãi	16
<input type="checkbox"/>	sv02	56 Nguyễn Văn Cừ	16
<input type="checkbox"/>	sv03	67 Lò Siêu	16
<input type="checkbox"/>	sv04	82 Nguyễn Tri Phương	16
<input type="checkbox"/>	sv05	91 Lò Siêu	16
<input type="checkbox"/>	sv06	72 Nguyễn Trãi	16
<input type="checkbox"/>	13	192 Lò Siêu	17
Cập nhật			

Trang ConnectDB.php

```
<?php
//kết nối MySQL và CSDL QuanLySinhVien
$conn = mysql_connect("localhost","root","");
if(!$conn)
{
    echo "Không thể kết nối đến MySQL.";
}
//select database QuanLySinhVien
$db = mysql_select_db("QuanLySinhVien");
if(!$db)
{
    echo "Không thể kết nối đến CSDL QuanLySinhVien.";
}
mysql_query("set names 'utf8'");
?>
```

Trang HienThiSinhVien.php

```
<?php
include("ConnectDB.php"); // Nhúng 1 file php vào
$result = mysql_query("select * from sinhvien");
?>
<form action="XuLy.php" method="post">
<table border="1" style="border:ridge; border-spacing:0">
<tr bgcolor="#CC9900">
    <th width="70"></td>
    <th width="70">Mã sinh viên</td>
    <th width="200">Họ tên sinh viên</td>
    <th width="180">Địa chỉ</td>
    <th width="70">Tuổi</td>
</tr>
<?php
while($row = mysql_fetch_object($result))
{
    echo "<tr>";
    echo "<td><center><input name='ckbox[]' type='checkbox' value='".$row->id."</></center></td>";
    echo "<td>".$row->id."</td>";
    echo "<td><input name='txtName' value='".$row->name."</td>";
    echo "<td><input name='txtAddress' value='".$row->address."</td>";
    echo "<td><input name='txtAge' value='".$row->age."</td>";
    echo "</tr>";
}
```

```

<tr>
    <td colspan="5"><center><input id="btnUpdate" type="submit" value="Cập nhật" /></center></td>
</tr>
<tr>
    <td colspan="5">
        <?php
            if(isset($_GET["status"]))
            {
                if($_GET["status"] == "ok")
                    echo "<center><p style='background-color:#CC3300; color:#FFFFFF'>Cập nhật thành công!</p><center>";
            }
        ?>
    </td>
</tr>
</table>
</form>

```

Trang XuLy.php

```

<?php
if(isset($_POST["ckbox"]))
{
    $ck = $_POST["ckbox"];
    include("ConnectDB.php");
    foreach($ck as $c)
    {
        $result = mysql_query("update sinhvien
                                set name = '".$_POST["txtName"][$c]."',
                                    address = '".$_POST["txtAddress"][$c]."',
                                    age = '".$_POST["txtAge"][$c]."'
                                where id = '".$c."'");
    }
    header("location:HienThiSinhVien.php?status=ok");
}
else
    header("location:HienThiSinhVien.php");
?>

```

***Chú ý:** Dữ liệu chỉ cập nhật được khi người dùng check vào checkbox. Nếu người dùng sửa dữ liệu mà không check vào checkbox thì sẽ không cập nhật được.
Để test xem đoạn code trên có hoạt động cập nhật nhiều mẫu tin được không? Chúng ta sẽ thêm vào số 2 chỉ để test mà thôi. Sửa lại như hình sau:

ta sẽ sửa 3 dòng này

Nhấn cập nhật

	Mã sinh viên	Họ tên sinh viên	Địa chỉ	Tuổi
<input type="checkbox"/>	sv01	Nguyễn Văn Hoa	123 Nguyễn Trãi	16
<input checked="" type="checkbox"/>	sv02	Lê Thị Thúy 2	56 Nguyễn Văn Cừ 2	16 2
<input type="checkbox"/>	sv03	Trần Văn Tuấn	67 Lò Siêu	16
<input checked="" type="checkbox"/>	sv04	Lê Tuấn Anh 2	82 Nguyễn Tri Phương 2	16 2
<input type="checkbox"/>	sv05	Nguyễn Quốc Trung	91 Lò Siêu	16
<input checked="" type="checkbox"/>	sv06	Phước Lâm 2	72 Nguyễn Trãi 2	16 2
<input type="checkbox"/>	13	Nguyễn Văn Tú	192 Lò Siêu	17

Thông báo cập nhật thành công xuất hiện dưới button cập nhật

Kết quả như hình:

Cập nhật thành công!

	Mã sinh viên	Họ tên sinh viên	Địa chỉ	Tuổi
<input type="checkbox"/>	sv01	Nguyễn Văn Hoa	123 Nguyễn Trãi	16
<input type="checkbox"/>	sv02	Lê Thị Thúy 2	56 Nguyễn Văn Cừ 2	16 2
<input type="checkbox"/>	sv03	Trần Văn Tuấn	67 Lò Siêu	16
<input type="checkbox"/>	sv04	Lê Tuấn Anh 2	82 Nguyễn Tri Phương 2	16 2
<input type="checkbox"/>	sv05	Nguyễn Quốc Trung	91 Lò Siêu	16
<input type="checkbox"/>	sv06	Phước Lâm 2	72 Nguyễn Trãi 2	16 2
<input type="checkbox"/>	13	Nguyễn Văn Tú	192 Lò Siêu	17

Và trong CSDL MySQL, 3 sinh viên có id sv02, sv04, sv06 cũng đã update lại thông tin ta vừa cập nhật.

	id	name	address	age		
<input type="checkbox"/>			sv01	Nguyễn Văn Hoa	123 Nguyễn Trãi	16
<input checked="" type="checkbox"/>			sv02	Lê Thị Thúy 2	56 Nguyễn Văn Cừ 2	16 2
<input type="checkbox"/>			sv03	Trần Văn Tuấn	67 Lò Siêu	16
<input checked="" type="checkbox"/>			sv04	Lê Tuấn Anh 2	82 Nguyễn Tri Phương 2	16 2
<input type="checkbox"/>			sv05	Nguyễn Quốc Trung	91 Lò Siêu	16
<input checked="" type="checkbox"/>			sv06	Phước Lâm 2	72 Nguyễn Trãi 2	16 2
<input type="checkbox"/>			13	Nguyễn Văn Tú	192 Lò Siêu	17

Câu hỏi ôn tập:

Hãy liệt kê danh sách user và user detail lên trang php dưới dạng table.
(Thêm-xóa-sửa user).

CHƯƠNG 14

14.1 Mô hình MVC

Mô hình MVC = Model - View – Controller

Model : thao tác với CSDL

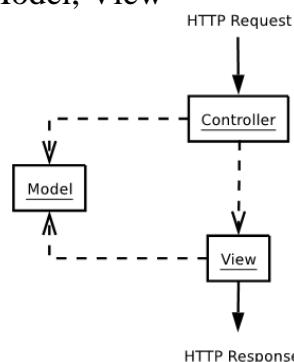
View : hiển thị nội dung

Controller : xử lý yêu cầu bên client

Mô hình MVC được sử dụng rất phổ biến vì tính logic và sự linh hoạt của nó giúp chúng ta có thể quản lý Project một cách dễ dàng và thuận tiện cho việc nâng cấp hệ thống.

Hiện nay có rất nhiều loại mô hình MVC như là:

+ Mô hình 3 lớp : mô hình này không được gọi là mô hình MVC nhưng có tách lớp ra 3 vai trò khác nhau: Controller, Model, View

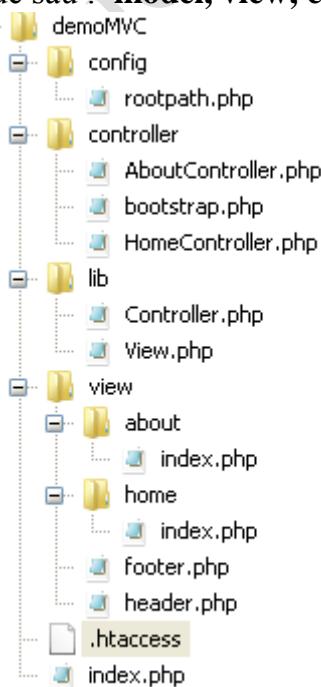


+ Mô hình MVC: mô hình chuẩn MVC thì có rất nhiều ở đây chúng ta sẽ làm theo cách mô phỏng từ Zend url → Bootstrap → Controller [→Model] → View

Đầu tiên chúng ta tạo 1 project mới và 1 vài thư mục như sau trên host (wamp, xamp, v.v...)

Tên project là **demoMVC**

Trong project ta tạo những thư mục sau : **model, view, controller, config, lib**.



Trong folder **lib** mình sẽ tạo 2 file php(**View.php, Controller.php**) như sau:

View.php

```
<?php
    class View{
        public function __construct(){}
        public function render($filename){
            include "view/header.php";
            include "view/$filename.php";
            include "view/footer.php";
        }
    }
```

Controller.php

```
<?php
    class Controller extends View{
        protected $view;
        public function __construct(){
            $this->view = new View();
        }
    }
```

Trong folder controller tạo 3 file(**bootstrap.php**, **HomeController.php**, **AboutController.php**) như sau

bootstrap.php

```
1 <?php
2     class Bootstrap {
3         public function __construct(){
4             $url = isset($_GET["url"]) ? $_GET["url"] : "";
5             if ($url != "") {
6                 $ar_url = explode("/", $url);
7                 $controller = $ar_url[0];
8                 $this->load($controller);
9             } else {
10                 $this->load("home");
11             }
12         }
13
14         private function load($controller){
15             $controllername = ucfirst($controller)."Controller";
16             $path = "controller/$controllername.php";
17             if (file_exists($path)) {
18                 include $path;
19                 new $controllername();
20             } else {
21                 include "controller/HomeController.php";
22                 new HomeController();
23             }
24         }
25     }
}
```

HomeController.php

```
1 <?php
2     class HomeController extends Controller{
3         public function __construct() {
4             parent::__construct();
5             $this->index();
6         }
7
8         public function index(){
9             $this->view->render("home/index");
10        }
11    }
```

AboutController.php

```

1 <?php
2     class AboutController extends Controller{
3         public function __construct() {
4             parent::__construct();
5             $this->index();
6         }
7
8         public function index(){
9             $this->view->render("about/index");
10        }
11    }

```

Trong folder config tạo file rootpath.php

```

1 <?php
2     defined("PATH") || define("PATH", "http://localhost/demoMVC");

```

Trong folder view :

- + header.php
- + footer.php

... có bao nhiêu controller thì tạo bấy nhiêu folder hoặc ngược lại, ví dụ:

HomeController thì phải có folder **home** trong view (view/home/index.php)

Tương tự như demo trên thì có 2 controller **HomeController**, **AboutController**, vậy ta tạo 2 folder là **home**,**about**

Trong mỗi folder chứa 1 file **index.php** chứa nội dung chính của chính nó

view/home/index.php

<h1>đây là trang Home</h1>

view/about/index.php

<h1>đây là trang About</h1>

Và việc cuối cùng là cấu hình .htaccess, mở file .htaccess ghi nội dung như sau

```

1 RewriteEngine On
2 RewriteCond %{REQUEST_FILENAME} !-d
3 RewriteCond %{REQUEST_FILENAME} !-f
4 RewriteRule ^(.*)$ index.php?url=$1 [QSA,L]

```

Lưu ý để có thể chạy được RewriteEngine ta phải bật lên trong file **httpd.conf**

Nếu như sài xamp thì đường dẫn như sau **xampp/apache/conf/httpd.conf**, chuột phải chọn Edit

Tìm dòng **#LoadModule rewrite_module modules/mod_rewrite.so**

mặc định mới cài là như vậy

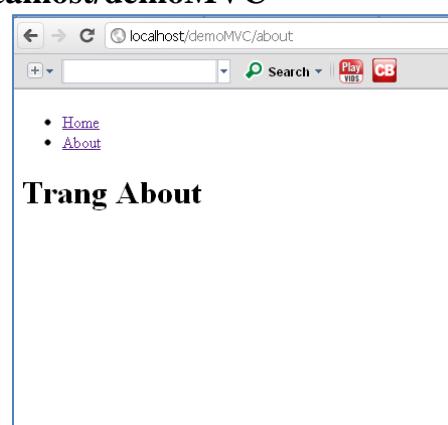
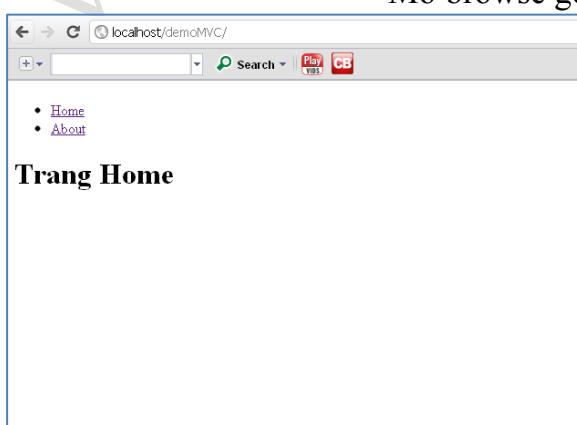
#LoadModule rewrite_module modules/mod_rewrite.so

Bỏ đi # để khởi động module rewrite

LoadModule rewrite_module modules/mod_rewrite.so

Sau khi bỏ đi thì phải **restart** lại **apache**, nếu đã bấm thõi bắt đầu chạy chương trình

Mở browser gõ **localhost/demoMVC**



Như chúng ta thấy bootstrap sẽ nhận url = tất cả những ký tự sau “localhost/demoMVC/”
Ví dụ : localhost/demoMVC/about/aaa/bbb/ccc
thì url = about/aaa/bbb/ccc
việc này được cấu hình trong file .htaccess

RewriteEngine On	bật chức năng rewrite
RewriteCond % {REQUEST_FILENAME} !-d	kiểm tra file
RewriteCond % {REQUEST_FILENAME} !-f	kiểm tra file
RewriteRule ^(.*)\$ index.php?url=\$1 [QSA,L]	[QSA,L] là gì nghiên cứu sau nhé

Như ta thấy RewriteRule chia ra làm 2 cụm cách nhau = khoảng trắng

A : ^(.*)\$ đây là 1 regex, lấy tất cả ký tự

B : index.php?url=\$1 \$1 = (.*)

Lấy ví dụ hiện tại localhost/demoMVC/about , sau localhost/demoMVC/ là about sẽ là **cụm A** và được thay thế \$1

Trên thanh url là localhost/demoMVC/about nhưng thực tế khi chạy xuống hệ thống nó sẽ chạy = cụm B => localhost/demoMVC/index.php?url=about

Như vậy khi chạy hệ thống dù có link wa trang nào nữa cũng chỉ có 1 trang duy nhất là index.php chỉ có url là thay đổi.

Ta phân tích lần lượt từ index.php

```
<?php
    session_start(); //chưa dùng, sau này sẽ dùng
    include "lib/View.php"; //include file View.php
    include "lib/Controller.php"; //include file Controller.php
    include "config/rootpath.php"; //include file rootpath.php
    include "controller/bootstrap.php"; //include file bootstrap.php
    new Bootstrap(); //chạy hệ thống
?>
```

Khi dòng new Bootstrap(); thực hiện, hệ thống sẽ chạy vào class Bootstrap và mặc định của class khi khởi tạo sẽ chạy vào Constructor : __construct hoặc phương thức cùng tên class.

Lúc này ta phân tích đến bootstrap.php

Gồm 2 phương thức : __construct và load

```
1 <?php
2     class Bootstrap {
3         public function __construct() {
4             $url = isset($_GET["url"]) ? $_GET["url"] : "";
5             if ($url != "") {
6                 $ar_url = explode("/", $url);
7                 $controller = $ar_url[0];
8                 $this->load($controller);
9             } else {
10                 $this->load("home");
11             }
12         }
13
14         private function load($controller) {
15             $controllername = ucfirst($controller). "Controller";
16             $path = "controller/" . $controllername . ".php";
17             if (file_exists($path)) {
18                 include $path;
19                 new $controllername();
20             } else {
21                 include "controller/HomeController.php";
22                 new HomeController();
23             }
24         }
25     }
}
```

Phương thức load dùng để load controller tương ứng với biến truyền vào. Vd: \$controller = “about” thì sẽ vào class AboutController...

Phương thức __construct dùng để xử lý và phân tích đường dẫn để đưa Controller phù hợp để xử lý.

\$url = isset(\$_GET["url"]) ? \$_GET["url"] : ""; biến url sẽ nhận giá trị trên đường dẫn có tên là url, nếu trên đường dẫn ko có biến này thì url = "".

TH1 : \$url khác "" thì sẽ tách \$url thành 1 mảng và qui định ký tự đầu tiên luôn là ký tự mà controller sẽ xử lý. Vd : menu **about** có đường dẫn là

http:localhost/demoMVC/about/thongtin => \$url = “about/thongtin” thì **about** tương ứng trung cho **AboutController**

\$ar_url = {"about", "thongtin"}; thì \$ar_url[0] = "about" khi đưa xuống hàm load sẽ load vào **AboutController**.

TH2 : \$url bằng "" nghĩa là khi load mặc định, ta cho về trang chủ ở đây là **HomeController**.(load trang home)

Xét **TH1** nếu như rơi vào **AboutController** thì sẽ load trang **About**

```

1 <?php
2     class AboutController extends Controller{
3         public function __construct() {
4             parent::__construct();
5             $this->index();
6         }
7
8         public function index(){
9             $this->view->render("about/index");
10        }
11    }

```

Mặc định khi new 1 class nào đó sẽ tự động nhảy vào hàm Constructor (hàm khởi tạo).

Như trong file **bootstrap** sau khi phân tích xong \$url ta dc. **AboutController**, khi new **AboutController** thì sẽ chạy vào hàm **__construct** của **AboutController**

Parent::__construct(); //chạy Constructor trong class Controller

\$this->index(); //dùng để load trang mặc định của controller này...

Hàm **index()** sử dụng **render** của class **View**, phương thức **render()** dùng để include dữ liệu vào trang html(hiển thị)

```

<?php
class View{
    public function __construct(){}
    public function render($filename) {
        include "view/header.php";
        include "view/$filename.php";
        include "view/footer.php";
    }
}

```

Như ta thấy chỉ việc đưa vào đường dẫn file nội dung là sẽ tự động chèn header và footer vào, như ví dụ trên thì **\$filename = “about/index”**.

Include “view/about/index.php”; đây là file nội dung.

Như vậy là xong phần cơ bản về MVC

Lúc này ta nghiệm lại sẽ thấy vai trò của mỗi class sẽ khác nhau như

View : dùng để chèn dữ liệu vào file html

Controller(AboutController, HomeController) : dùng để xử lý yêu cầu bên client và hiển thị dữ liệu tương ứng

Bootstrap : sẽ phân tích đường dẫn và đưa ra controller

Model : dùng để giao tiếp CSDL(chưa sử dụng).

url → Bootstrap → Controller [→Model] → View

Tiếp theo sẽ kết nối CSDL

Muốn kết nối CSDL thì trước tiên phải có CSDL, có thể import từ file **demomvc.sql** trong thư mục demomvc/db

Hoặc tạo 1 table **account** gồm 4 field **id, ten_dang_nhap, mat_khau, vai_tro**; insert 4 hoặc 5 record

Đặt tên database là demomvc

Trong folder config ta tạo 1 file **dbconfig.php**, file này dùng để chứa thông tin kết nối đến

```

1 <?php
2     $server = "localhost";
3     $username = "root";
4     $password = "";
5     $database = "demomvc";

```

CSDL

Trong folder lib ta tạo 2 file: **Model.php** và **Database.php**, 2 file này chứa phương thức để kết nối CSDL

Database.php

```

1 <?php
2     class Database{
3         protected $conn;
4         public function __construct() {}
5
6         public function connect(){
7             global $server;
8             global $username;
9             global $password;
10            global $database;
11            $this->conn = mysql_connect($server, $username, $password);
12            if ($this->conn) {
13                if (!mysql_select_db($database)) {
14                    echo "Cannot connect database ".$database." !!!";
15                    die;
16                }
17            } else {
18                echo "Cannot connect server ".$server." !!!";
19                die;
20            }
21        }
22
23        public function disconnect(){
24            $this->conn = null;
25        }
26
27        public function executeQuery($sql){
28            $array = new ArrayObject();
29            $rs = mysql_query($sql);
30            if (mysql_num_rows($rs) > 0) {
31                while ($row = mysql_fetch_object($rs)) {
32                    $array->append($row);
33                }
34            }
35            return $array;
36        }
37
38        public function executeNonQuery($sql){
39            mysql_query($sql);
40        }
41    }

```

Model.php

```

1 <?php
2     class Model{
3         protected $db;
4         public function __construct(){
5             $this->db = new Database();
6         }
7     }

```

Trong folder controller tạo 1 file AccountController.php

```

1 <?php
2     class AccountController extends Controller{
3         public function __construct(){
4             parent::__construct();
5             $this->index();
6         }
7
8         public function index(){
9             include "model/AccountModel.php";
10            $accountModel = new AccountModel();
11            $this->view->listAccount = $accountModel->getAllAccount();
12            $this->view->render("account/index");
13        }
14    }

```

Trong folder view tạo 1 folder có tên là **account**, và trong folder **account** tạo file **index.php** có nội dung như sau

```

1 <table>
2     <?php
3         foreach ($this->listAccount as $account) {
4     ?>
5             <tr>
6                 <td><?= $account->id ?></td>
7                 <td><?= $account->ten_dang_nhap ?></td>
8                 <td><?= $account->mat_khau ?></td>
9                 <td><?= $account->vai_tro ?></td>
10            </tr>
11        <?php
12        ?
13    ?>
14 </table>

```

Edit file **index.php (demomvc/index.php)**

```

1 <?php
2     session_start();
3     include "lib/View.php";
4     include "lib/Controller.php";
5     include "lib/Database.php";
6     include "lib/Model.php";
7     include "config/rootpath.php";
8     include "config/dbconfig.php";
9     include "controller/bootstrap.php";
10    new Bootstrap();
11 ?>

```

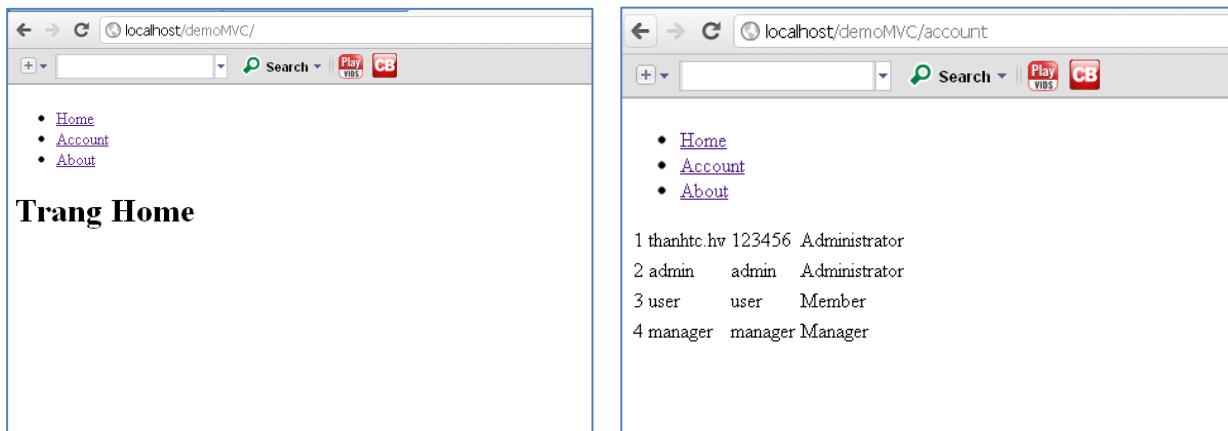
Mở file **header.php** trong folder view thêm 1 dòng menu Account

```

1 <!DOCTYPE html>
2 <html>
3     <head>
4         <title>Demo MVC</title>
5         <meta content="text/html" charset="UTF-8" />
6     </head>
7     <body>
8         <div id="menu">
9             <ul>
10                 <li><a href="<?php echo PATH; ?>/home">Home</a></li>
11                 <li><a href="<?php echo PATH; ?>/account">Account</a></li>
12                 <li><a href="<?php echo PATH; ?>/about">About</a></li>
13             </ul>
14         </div>
15         <div id="content">
16

```

Sau đó chúng ta tiến hành chạy localhost/demoMVC và xem kết quả



Vậy là ta kết nối CSDL thành công

Khi sang trang Account, hệ thống sẽ đẩy sang AccountController xử lý, AccountController chạy hàm index và kết nối CSDL lấy dữ liệu trả về 1 biến là listAccount

```

1 <?php
2     class AccountController extends Controller{
3         public function __construct(){
4             parent::__construct();
5             $this->index();
6         }
7
8         public function index(){
9             include "model/AccountModel.php";
10            $accountModel = new AccountModel();
11            $this->view->listAccount = $accountModel->getAllAccount();
12            $this->view->render("account/index");
13        }
14    }

```

Bên trang nội dung của account sẽ nhận biến listAccount và đoạn code.

\$this->listAccount

```

1 <table>
2     <?php
3         foreach ($this->listAccount as $account) {
4             ?>
5                 <tr>
6                     <td><?= $account->id ?></td>
7                     <td><?= $account->ten_dang_nhap ?></td>
8                     <td><?= $account->mat_khau ?></td>
9                     <td><?= $account->vai_tro ?></td>
10                </tr>
11            <?php
12            ?
13        ?>
14    </table>

```

14.2 Friendly URL

Trong phần mô hình MVC, chúng ta cũng đã tiếp cận với việc rewrite url rồi, cũng chính là nội dung của Friendly URL

Friendly url là gì? Nó sẽ là chủ đề của bài viết này. url seo friendly là yếu tố SEO tích cực mà các SEOer không bao giờ bỏ qua, chúng tôi sẽ cho các bạn thấy tầm quan trọng của nó và như thế nào là đường dẫn thân thiện cũng như cách để tạo ra đường dẫn thân thiện để đọc bài viết 1 cách hoàn chỉnh hơn

1 đường link friendly url phải rõ ràng và có chủ đề bài viết trên đó lấy ví dụ như

<http://vnexpress.net/gl/the-gioi/anh/2012/05/chien-hamcac-nuoc-do-ve-quanh-bien-dong/>

như đường link trên là 1 friendly url seo

chien-hamcac-nuoc-do-ve-quanh-bien-dong là chủ đề trang này

gl/the-gioi/anh/2012/05 .v.v... đều nói lên 1 điều gì đó cho người đọc biết và chủ yếu là cho google search ra theo từ khóa cách làm thì ta sử dụng Rewrite mod như trên để thay thế đường dẫn.

Vd:

localhost/demomvc/index.php?idtitle=5

đây là đường dẫn ko friendly url và thậm chí ko có tiêu chí để seo. Thay vì để idtitle=5 thì ta để tên của title như là friendly-url thay thế cho id và đặt biệt là không được có dấu localhost/demomvc/index.php?title=friendly-url

để rút ngắn lại như sau

localhost/demomvc/friendly-url

ta viết regex để nhận biết

RewriteRule ^(.*)\$ index.php?title=\$1

Rất đơn giản.

Trong trường hợp viết lại như sau

Localhost/demomvc/article/friendly-url

RewriteRule ^article/([a-zA-Z0-9_-]+)\$ index.php?title=\$1

Localhost/demomvc/article/title/friendly-url

RewriteRule ^article/tite/([a-zA-Z0-9_-]+)\$ index.php?title=\$1

hoặc

RewriteRule ^article/ ([a-zA-Z_-]+)([a-zA-Z0-9_-]+)\$ index.php?title=\$2