

**BỘ CÔNG THƯƠNG
TRƯỜNG CAO ĐẲNG CÔNG NGHIỆP HUẾ**



DƯƠNG NGỌC DUY

**BÀI GIẢNG
LẬP TRÌNH WEB CƠ BẢN**

KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

LƯU HÀNH NỘI BỘ

HUẾ, 2017

Lập trình PHP&MYSQL

Chương 1. Tổng quan về lập trình website

Chương 2. Ngôn ngữ lập trình PHP

Chương 3. Hệ quản trị CSDLMySQL

Chương 4. Lập trình website động với PHP&MySQL

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và Cài đặt

1.2 Kiến trúc www

MỤC TIÊU

1. Nắm được kiến thức cơ bản về lập trình website
2. Biết được cách cài đặt, cấu hình xampp, appserv. Biết được cách build một website
3. Nắm được kiến trúc www

3

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 các phần mềm cần thiết

- Xampp / Appser
- Dreamwaver (CS3,4,5)
- PhpDesigner 8
- Notepad++
- ...

4

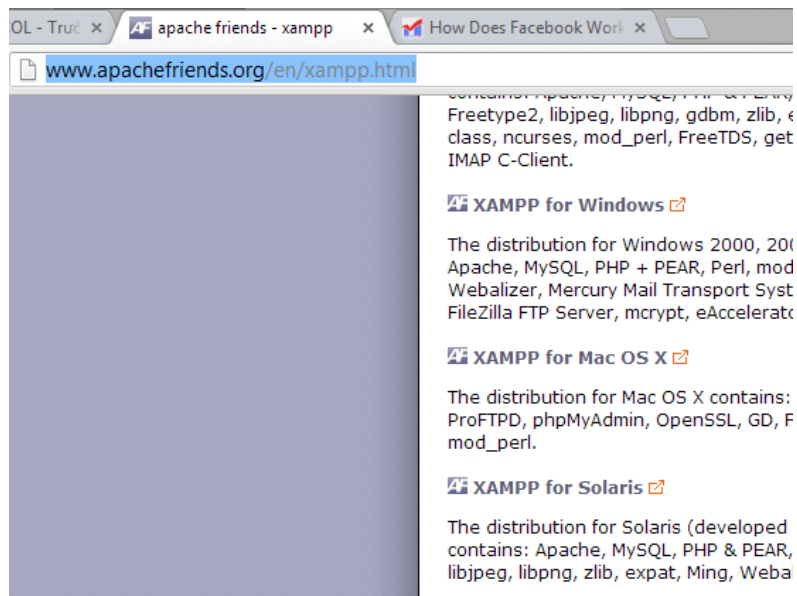
Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

- Tải Xampp : <http://www.apachefriends.org/en/xampp.html>



Page 5

5

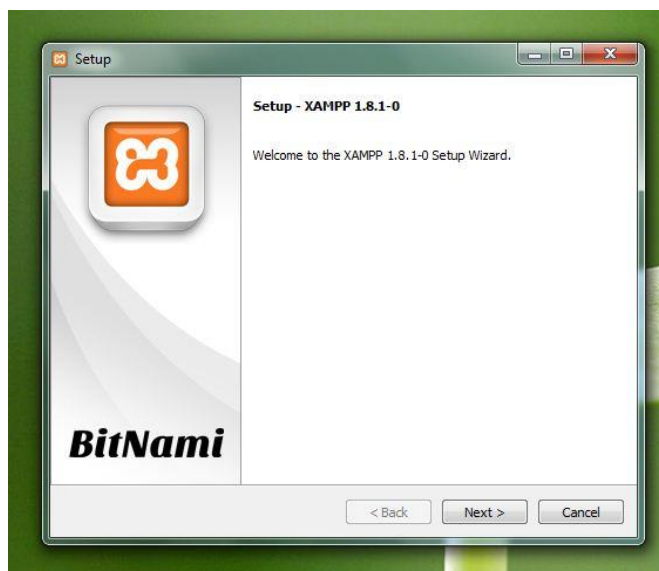
Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

- Tiến hành cài đặt



Page 6

6

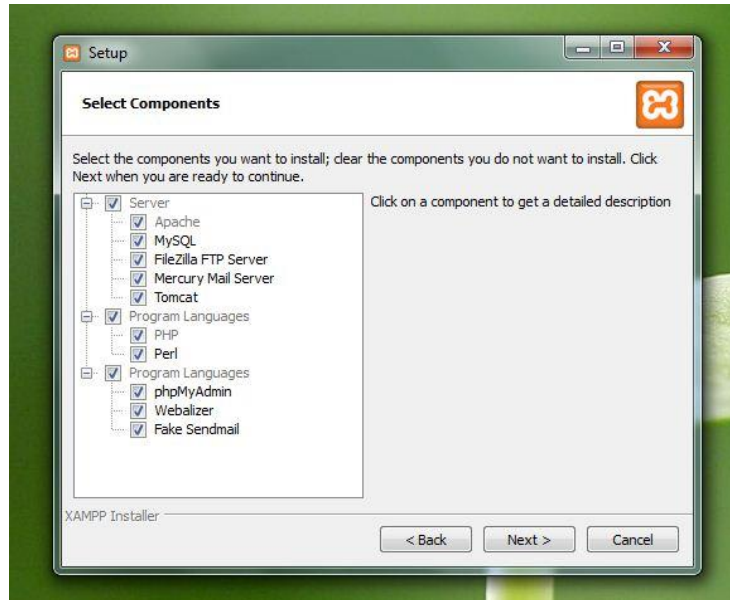
Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

- Tiến hành cài đặt



Page 7

7

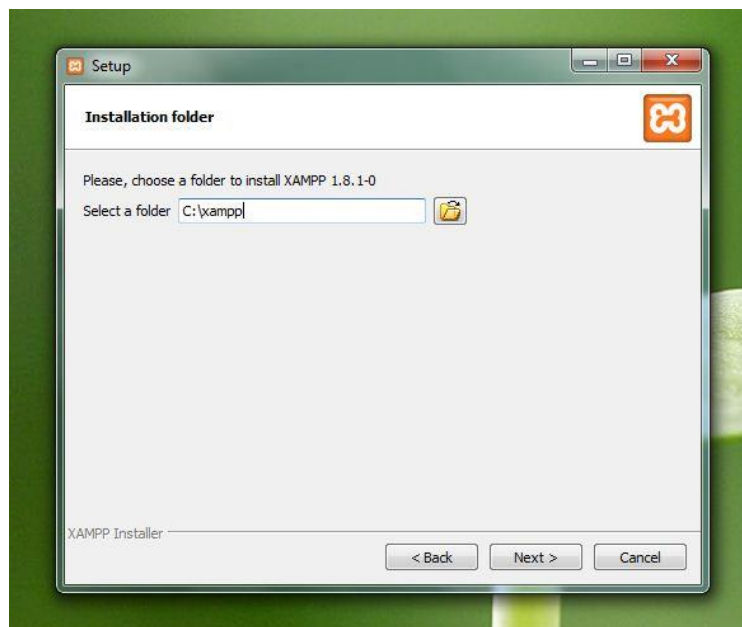
Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

- Tiến hành cài đặt



Page 8

8

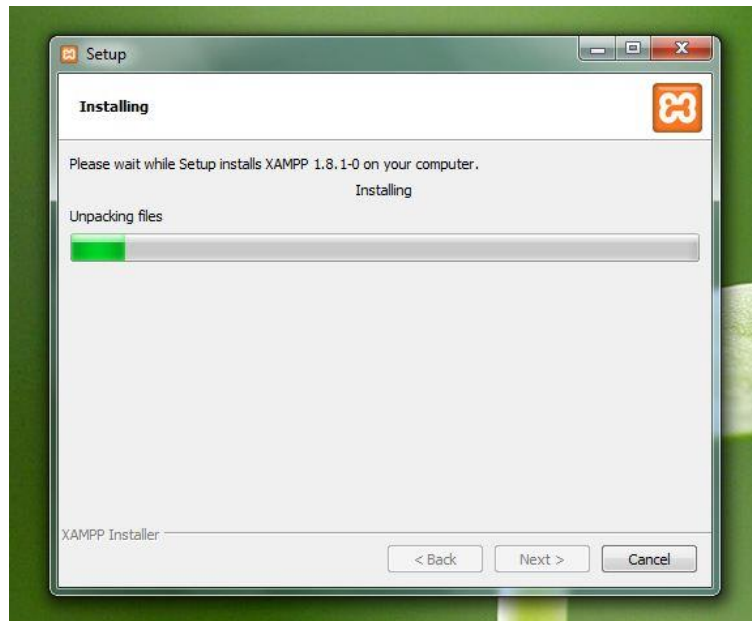
Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

- Tiến hành cài đặt



Page 9

9

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

- Tiến hành cài đặt



Page 10

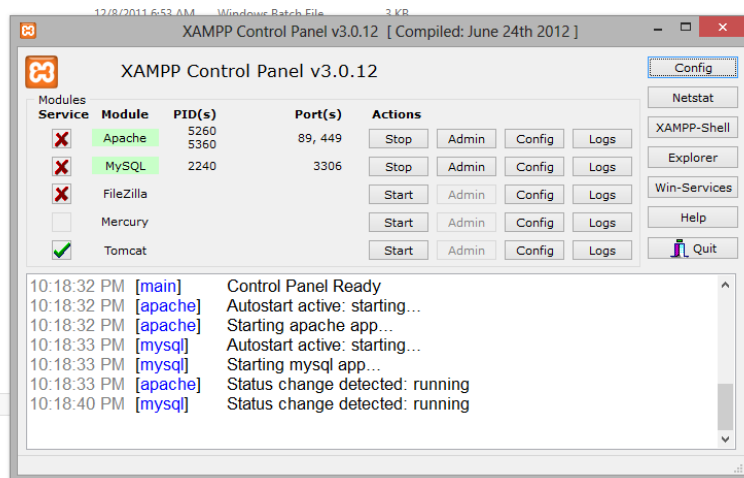
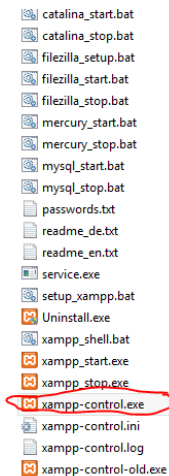
10

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp



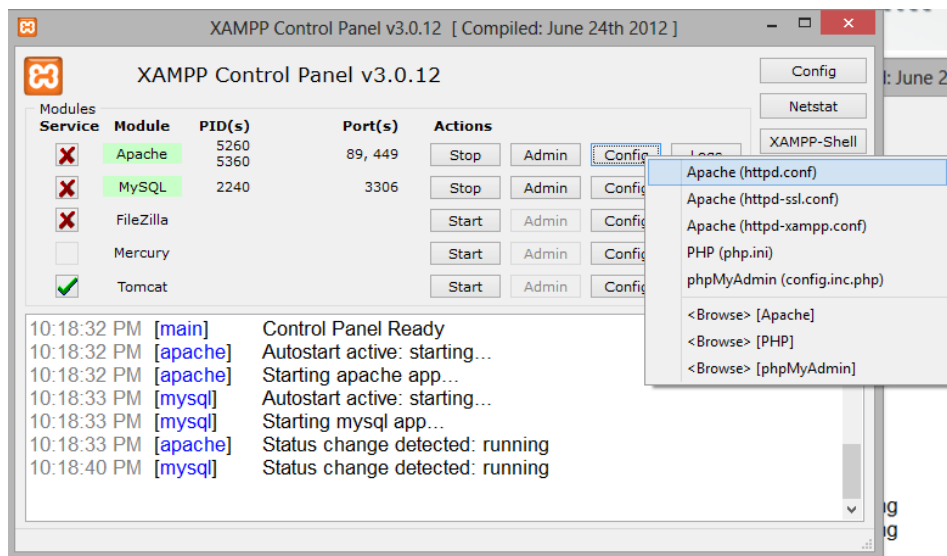
Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

• Cấu hình XAMPP



Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

• Cấu hình XAMPP

```

httpd.conf - Notepad
File Edit Format View Help
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 0.0.0.0:80
#Listen [::]:80
Listen 89

# Default charset UTF8
# AddDefaultCharset utf-8

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by `httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#

```

Page 13

13

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

• Cấu hình XAMPP

```

httpd.conf - Notepad
File Edit Format View Help
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName localhost:89

#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:/xampp/htdocs"

#
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# features.
#

```

Page 14

14

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

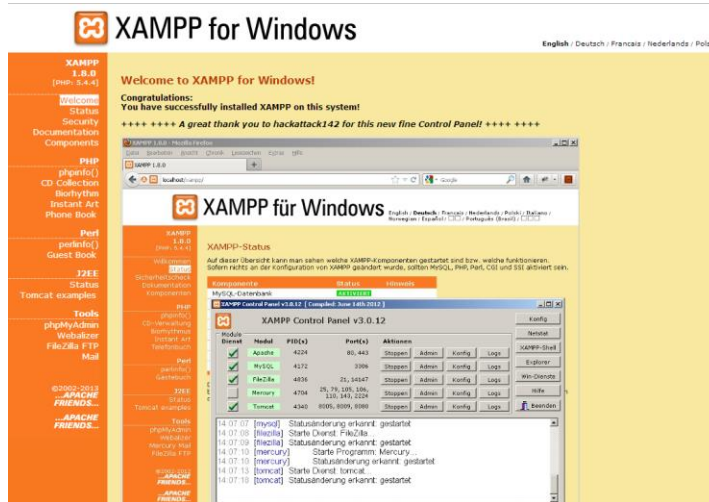
1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

- Sau khi cấu hình thành công chạy trên trình duyệt với địa chỉ sau :

HTTP://LOCALHOST:CỔNG



Chương 1 : Tổng quan về lập trình website

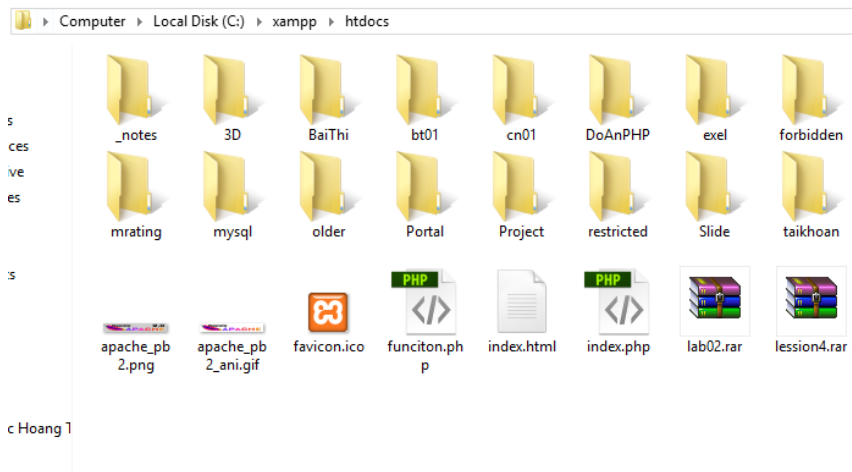
1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

- Nơi lưu trữ code trên Server:



Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

- Năm 1991, Tim Berners Lee trong quá trình nghiên cứu Internet đã hình dung ra một khái niệm về phương cách truyền các tập tin qua lại giữa kết nối UUCP và FTP.

Khái niệm này sau đó được phát triển mạnh mẽ và trở thành World Wide Web (WWW).

- Các file truyền trên WWW hoặc WEB lúc ban đầu là những văn bản dạng TEXT phải được đánh dấu định dạng bởi một ngôn ngữ có tên Hyper Text Markup Language (HTML).

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

- World Wide Web, được gọi tắt là Web - là mạng lưới nguồn thông tin cho phép mọi người khai thác thông tin qua một số công cụ hoặc là chương trình hoạt động dưới các giao thức mạng.
- Web có khả năng liên kết với những trang Web khác, thông qua các “siêu liên kết” (Hyperlink), siêu liên kết về bản chất là địa chỉ trở tới nguồn thông tin (trang Web) nằm đâu đó trên Internet.

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

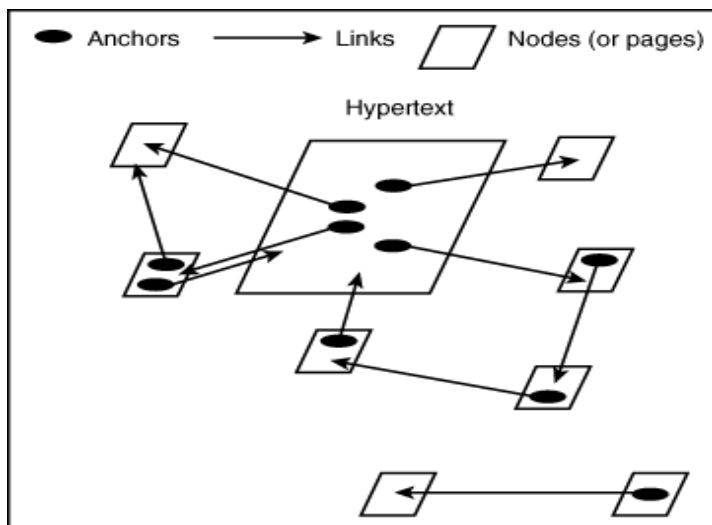
1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

- Đặc điểm và Nguyên lý hoạt động : WWW được xây dựng dựa trên một kỹ thuật có tên gọi là hypertext (siêu văn bản).



Page 19

19

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

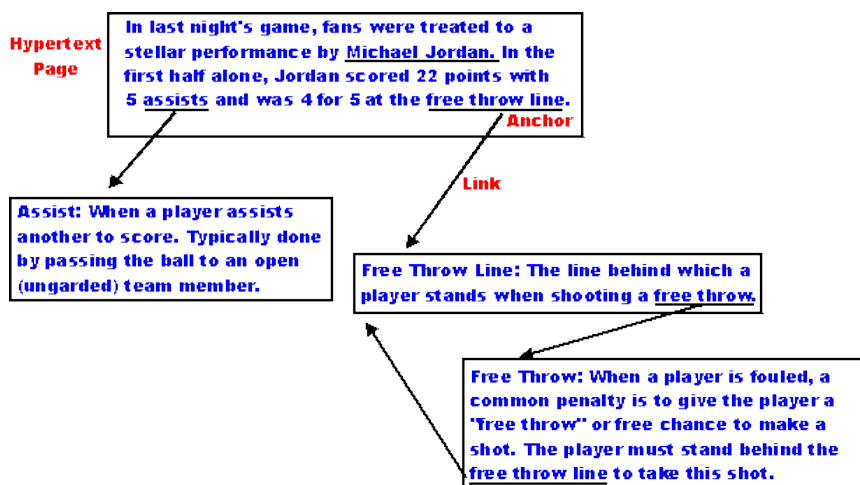
1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

- Mô hình thực tế :



Page 20

20

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

- Trên cùng một trang thông tin có thể cần có nhiều kiểu dữ liệu khác nhau như TEXT, ảnh hay âm thanh. Dẫn đến sự xuất hiện ngôn ngữ HTML (HyperText Markup Language).
- Người dùng sử dụng một phần mềm Web Browser để xem thông tin trên các máy chủ WWW. Tại server phải có một phần mềm Web server.

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

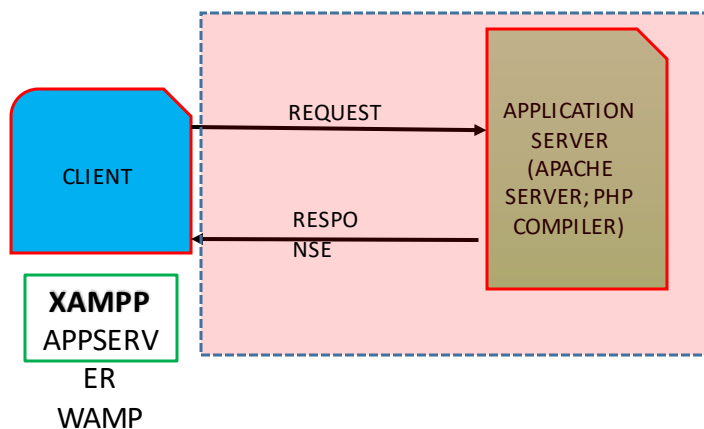
1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

- Mô hình web tĩnh :

➤ Web tĩnh



Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

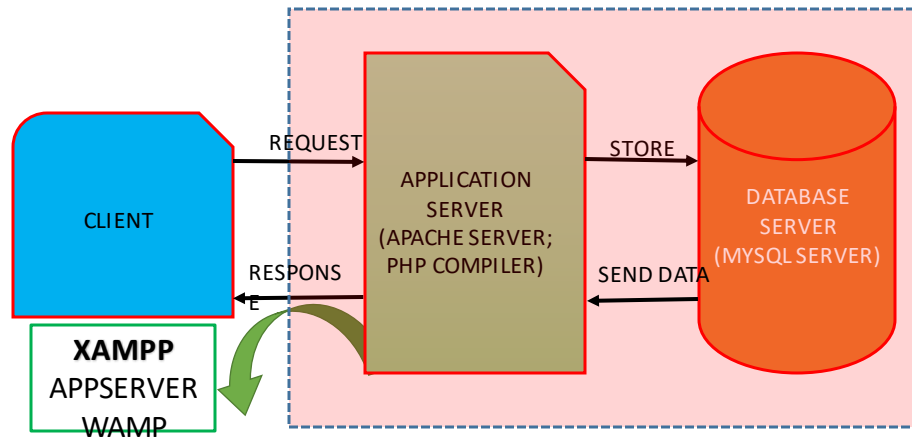
1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

- Mô hình web động:



Page 23

23

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

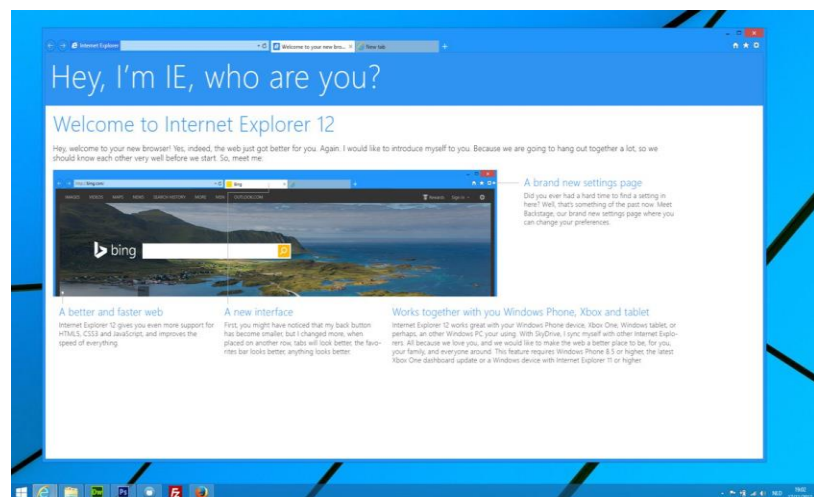
1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

- Internet Explorer



Page 24

24

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

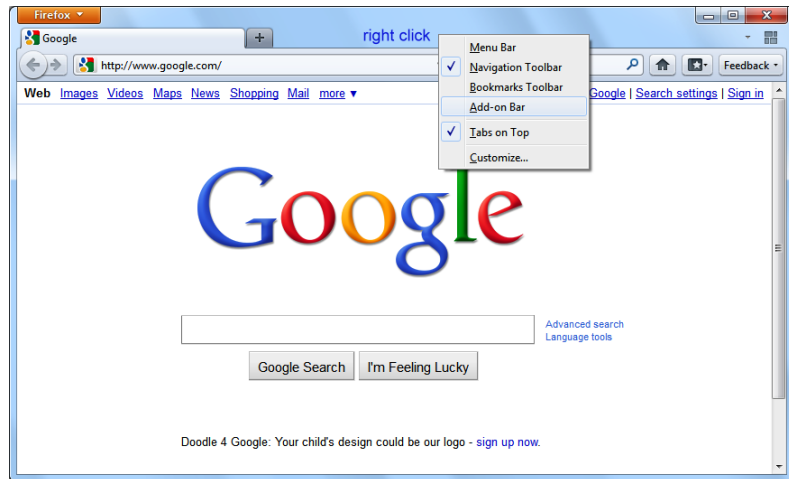
1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

• Mozilla FireFox



Page 25

25

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

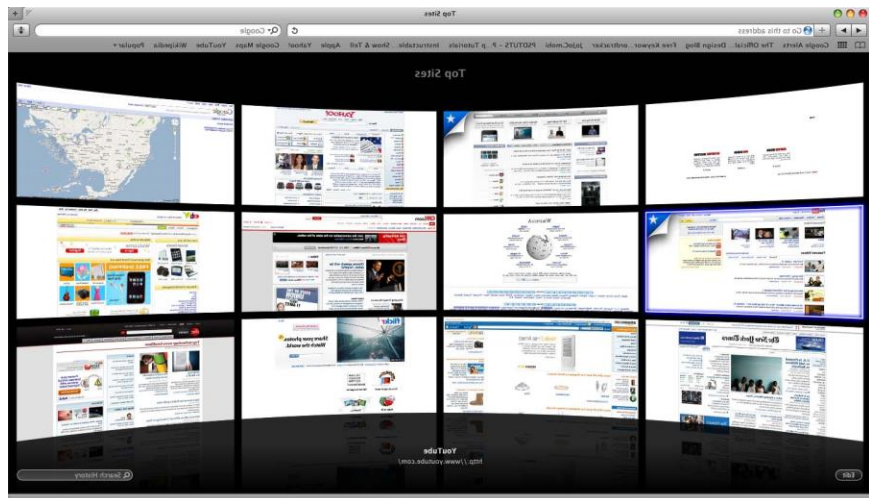
1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

• Safari



Page 26

26

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

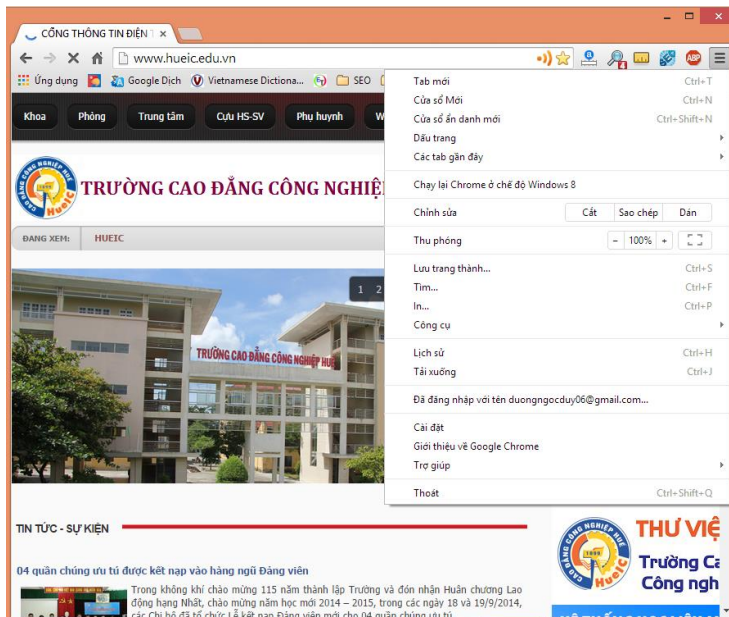
1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

• Google Chrome



Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

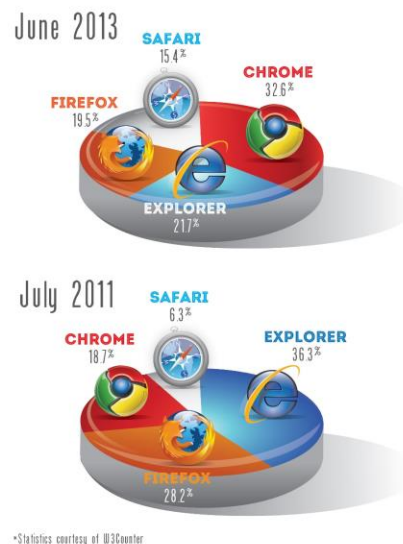
1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

• So sánh trình duyệt



WEB BROWSER USE



Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

1.2.4. Thuật Ngữ Web

- Webpage - Trang web
- Website
- Các dạng trang Web
 - Website giới thiệu
 - Website lưu trữ thông (hay là thư viện điện tử)
 - Website truyền dữ liệu
 - Website thương mại

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

1.2.4. Thuật Ngữ Web

- Webpage - Trang web
- Website
- Các dạng trang Web
 - Website giới thiệu
 - Website lưu trữ thông (hay là thư viện điện tử)
 - Website truyền dữ liệu
 - Website thương mại

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

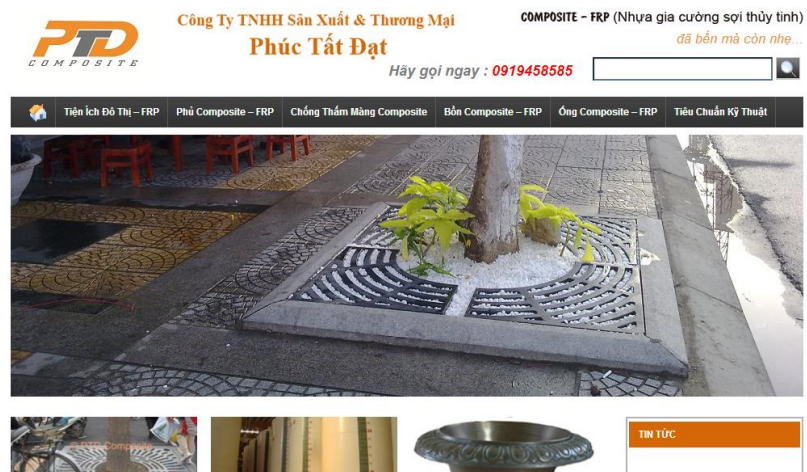
1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

1.2.4. Thuật Ngữ Web

• Website giới thiệu



Page 31

31

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

1.2.4. Thuật Ngữ Web

• Website lưu trữ thông tin



Page 32

32

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

1.2.4. Thuật Ngữ Web

• Website thương mại

The screenshot shows the Lazada.vn homepage with a dark blue header. The main navigation bar includes a search bar, a shopping cart icon, and a balance of 50,000 VND. Below the header, there are several promotional banners and product listings. A large banner for 'ĐUA VOUCHER MUA HÀNG XIN' is prominent, along with a 'GIÁ TỐT MỖI NGÀY' banner. The page also features a sidebar with category icons and a right sidebar with promotional text and a 'ƯU ĐÃI ĐẶC BIỆT' banner.

Page 33

33

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

1.2.4. Thuật Ngữ Web

1.2.5. Các loại trang Web

- Homepage - Trang chủ
- Trang chuyên đề
- Trang đơn vị
- Trang phản hồi – Feedback

Page 34

34

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

1.2.4. Thuật Ngữ Web

1.2.5. Các loại trang Web

1.2.6. Các thành phần thường có trong trang web

- Hình ảnh
- Banner : Biểu ngữ
- Logo
- Search Form
- Navigator
- Header : phần đầu trang web
- Footer : phần chân trang web
- Frame
- Multimedia

Chương 1 : Tổng quan về lập trình website

1.1 Cấu hình và cài đặt

1.1.1 Các phần mềm cần thiết

1.1.2 Cài đặt Xampp

1.1.3 Chạy ứng dụng web

1.2 Kiến trúc WWW

1.2.1 Nguồn gốc WWW

1.2.2 Khái niệm về WWW

1.2.3. Trình Duyệt

1.2.4. Thuật Ngữ Web

1.2.5. Các loại trang Web

1.2.6. Các thành phần thường có trong trang web

1.2.7. Các kỹ thuật dùng cho Web



Bài giảng lý thuyết:

LẬP TRÌNH PHP&MYSQL

Faculty of IT

Email: khoacntt@hueic.edu.vn

Lập trình PHP&MYSQL

Chương 1. Tổng quan về lập trình website

Chương 2. Ngôn ngữ lập trình PHP

Chương 3. Hệ quản trị CSDL MySQL

Chương 4. Lập trình website động với PHP&MySQL

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.5 Hàm

2.6 Kiểu dữ liệu trong mảng

MỤC TIÊU

1. Nắm được kiến thức cơ bản về PHP : biến, hằng, các lệnh cơ bản, hàm, kiểu dữ liệu mảng.
2. Viết được hàm cơ bản.

- PHP - Hypertext Preprocessor - là một ngôn ngữ lập trình kịch bản được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở.
- Nó rất thích hợp với web và có thể dễ dàng nhúng vào trang HTML.
- Do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

CÁC PHIÊN BẢN CỦA PHP

PHP 5.

- Hỗ trợ lập trình hướng đối tượng (OOP).
- Xử lý XML tốt hơn.
- Hỗ trợ giao thức máy khách mới.
- Hỗ trợ dịch vụ web yếu (kỹ thuật caching).

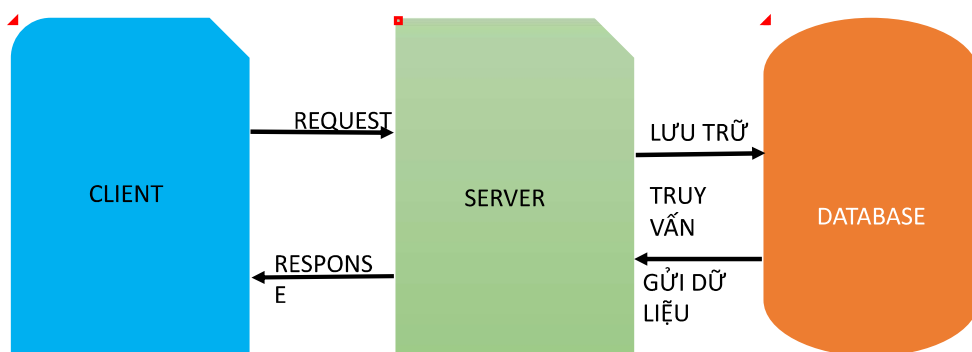
PHP 6. Hiện nay phiên bản tiếp theo của PHP đang được phát triển.

- Hỗ trợ namespace;
- Hỗ trợ Unicode hoàn thiện hơn;
- Sử dụng PDO làm API chuẩn cho việc truy cập cơ sở dữ liệu, các API cũ sẽ bị đưa ra thành thư viện PECL...

APACHE SERVER

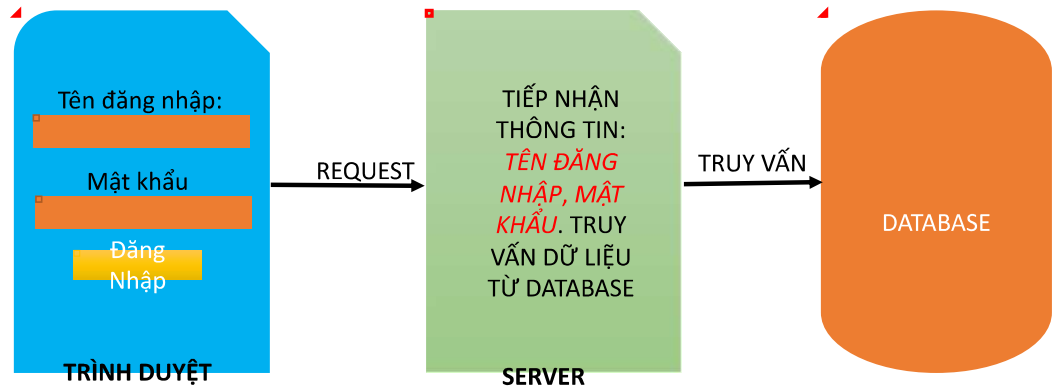
- Apache là chương trình máy chủ đối thoại qua giao thức HTTP.
- Apache chạy trên các hệ điều hành tương tự như Unix, Microsoft Windows, Linux, MacOS...
- Apache đã không ngừng phát triển và trở thành một phần mềm có sức cạnh tranh mạnh so với các chương trình máy chủ khác về mặt hiệu suất và tính năng.
- Từ năm 1996, Apache trở thành một chương trình máy chủ HTTP thông dụng nhất

Lập trình web bằng PHP



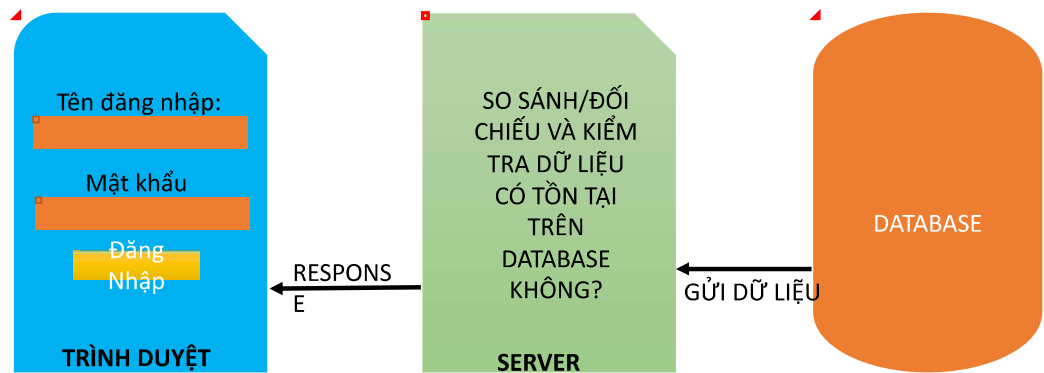
2.1 Giới thiệu

- Ví dụ quá trình đăng nhập



2.1 Giới thiệu

- Ví dụ. quá trình đăng nhập



Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

• Ví dụ về PHP :

```
1 <html>
2   <head>
3     <meta http-equiv="content-type" charset="utf-8" />
4   </head>
5   <body>
6     <?php
7       echo "Xin chào! Tôi là người mới học PHP.";
8       echo (<br/>);
9       print "<br/>";
10      print("Xin chào! Tôi cũng là người mới học PHP.");
11    <?>
12  </body>
13 </html>
```

1. Bật xampp control
2. Tạo một thư mục có tên baigiangPHP trong htdocs.
3. Tạo một tập tin index.php trong thư mục baigiangPHP như trên
4. Mở trình duyệt gõ: <http://localhost/lop13CDTH11>

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

• Một số quy ước :

1. Php là ngôn ngữ dạng kịch bản, nhúng vào một trang html.
2. Để viết mã php, ta sử dụng cặp dấu
3. <?php ?>
4. Php là ngôn ngữ phân biệt hoa/thường với biến và không phân biệt hoa/thường với hàm. Để thống nhất, ta quy ước là có phân biệt.
5. Để tạo một dòng ghi chú, ta có thể sử dụng /* ... */ hoặc //
6. Một lệnh trong PHP phải kết thúc bằng dấu chấm phẩy ;
7. Để in ra một xâu kí tự, một biến, hằng... ta có thể dùng lệnh print hoặc echo. Các lệnh này có thể viết theo hai cú pháp khác nhau.

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

- Biến dùng để lưu trữ giá trị. Giá trị của biến có thể được thay đổi.
- Khi muốn thay đổi giá trị cho biến, ta sử dụng toán tử gán.
- PHP không yêu cầu mạnh về kiểu dữ liệu của biến, không yêu cầu một biến phải khai báo mới được phép sử dụng.
- Một biến trong PHP luôn bắt đầu bằng dấu \$.
- Tên của một biến có sự phân biệt chữ hoa/thường.
- Ví dụ :

```
<?php $a = 10 ?>
```

```
<?php $A = "chao ban"; ?>
```

```
<?php echo "Gia tri cua a :".$a."<br> ?>
```

```
<?php echo "Gia tri cua A:". $A."<br> ?>
```

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

- Hằng dùng để lưu trữ giá trị. Giá trị của hằng không thể thay đổi.
- PHP không yêu cầu mạnh về kiểu dữ liệu của hằng.
- Một hằng trong PHP được định nghĩa bằng lệnh define: define('tên_hằng', giá_trị);
- Tên của một hằng có sự phân biệt chữ hoa/thường. Thông thường, hằng được đặt tên bằng chữ hoa.
- Ví dụ :

```
<?php  
define('MAX', 12);  
define("MYNAME", "PHP");  
echo MAX."<br/>";  
echo MYNAME."<br/>";  
?>
```

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

• Toán tử số học

Toán tử +

thực hiện phép toán cộng

Toán tử -

thực hiện phép toán trừ

Toán tử *

thực hiện phép toán nhân

Toán tử /

thực hiện phép toán chia

Toán tử %

thực hiện phép toán chia lấy phần dư

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng


2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

• Toán tử số học

```
<?php
$a = 5;
$b = 2;
echo "a+b=" . ($a+$b) . "<br/>";
echo "a-b=" . ($a-$b) . "<br/>";
echo "a*b=" . ($a*$b) . "<br/>";
echo "a/b=" . ($a/$b) . "<br/>";
echo "a%b=" . ($a%$b) . "<br/>";
?>
```



a+b=7
a-b=3
a*b=10
a/b=2.5
a%b=1

Trong php sử dụng toán tử chấm . để nối hai hay nhiều chuỗi, biến lại với nhau.

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

• Toán tử gán kết hợp

Toán tử +=

thực hiện phép cộng dồn: $a+=b$ tương đương $a=a+b$

Toán tử -=

thực hiện phép cộng dồn: $a-=b$ tương đương $a=a-b$

Toán tử *=

thực hiện phép cộng dồn: $a*=b$ tương đương $a=a*b$

Toán tử /=

thực hiện phép cộng dồn: $a/=b$ tương đương $a=a/b$

Toán tử %=

thực hiện phép cộng dồn: $a%=b$ tương đương $a=a\%b$

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

• Toán tử gán kết hợp

```
<?php
    $a = 5;
    $b = 2;
    $a += $b;
    echo "a+b=" . ($a) . "<br/>";
    $a -= $b;
    echo "a-b=" . ($a) . "<br/>";
    $a *= $b;
    echo "a*b=" . ($a) . "<br/>";
    $a /= $b;
    echo "a/b=" . ($a) . "<br/>";
    $a %= $b;
    echo "a%b=" . ($a) . "<br/>";
?>
```

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

Toán Tử so sánh

Toán Tử >

$A > b$ trả về giá trị đúng nếu a lớn hơn b , và sai nếu a bé hơn hoặc bằng b

Toán tử <

$A < b$ trả về giá trị đúng nếu a bé hơn b , và sai nếu a lớn hơn hoặc bằng b

Toán tử >=

$A \geq b$ trả về giá trị đúng nếu a lớn hơn hoặc bằng b , và sai nếu a bé hơn b

Toán tử <=

$A \leq b$ trả về giá trị đúng nếu a bé hơn hoặc bằng b , và sai nếu a lớn hơn b

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

Toán tử so sánh

Toán tử ==

$A == b$ trả về giá trị đúng nếu a bằng b , và sai nếu a khác b

Toán tử ===

$A === b$ tương tự $==$, nhưng so sánh cả kiểu dữ liệu. là phép so sánh đồng nhất

Toán tử !=

$A != b$ trả về giá trị đúng nếu a khác b , và sai nếu a bằng b

Toán tử !==

tương tự toán tử $!=$ nhưng so sánh cả kiểu dữ liệu.

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

Các toán tử logic

Toán tử &&

Trả về giá trị true nếu cả hai cùng true. Trả về giá trị false trong t/h còn lại.

Toán tử ||

Trả về giá trị false nếu cả hai cùng false. Trả về giá trị true trong t/h còn lại.

Toán tử !

Phủ định của false là true. Phủ định của true là false.

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

Các toán tử khác

TOÁN TỬ ++

Tăng giá trị của biến lên 1

`x++;`

`++x;`

Toán tử --

Giảm giá trị của biến xuống 1.

`x--;`

`--x;`

Toán tử =

Gán giá trị cho biến.

Toán tử điều kiện

Trả về giá trị là a cho x nếu $a > b$ là đúng; và ngược lại.

`var x = (a > b)?a:b;`

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

Biểu thức: là tập hợp các toán tử và toán hạng (biến, hằng) được sắp xếp theo một trật tự có ý nghĩa. Biểu thức được chia làm các loại cơ bản sau:

- Biểu thức số học: trả về giá trị là số. Nó là dạng biểu thức chứa các toán tử số học.
- Biểu thức logic: trả về giá trị true/false. Nó là dạng biểu thức chứa các toán tử logic và so sánh.

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

So sánh dấu “ và ‘

```
1 <html>
2   <head>
3     <meta http-equiv="content-type" charset="utf-8" />
4   </head>
5   <body>
6     <?php
7       $a = 5;
8       echo 'Giá trị của $a<br/>';
9       echo "Giá trị của $a<br/>";
10    ?>
11  </body>
12 </html>
```

- Dấu nháy đơn và dấu nháy đôi đều sử dụng cho chuỗi ký tự.
- Khi đưa tên biến vào trong dấu nháy đơn, thì tên biến sẽ được in ra cùng với các ký tự khác trong chuỗi.
- Khi đưa tên biến vào trong dấu nháy đôi, thì giá trị của biến sẽ được in ra cùng với các ký tự khác trong chuỗi.

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

• Cú pháp lệnh IF:

```
if(□iều_kiện){  
    Lệnh 1;  
}else{  
    Lệnh 2;  
}
```

- Chức năng: nếu điều kiện đúng, thì Lệnh 1 sẽ được thực hiện; nếu điều kiện sai, thì Lệnh 2 sẽ được thực hiện.

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

• Ví dụ lệnh IF:

```
1 <html>  
2     <head>  
3         <meta http-equiv="content-type" charset="utf-8" />  
4     </head>  
5     <body>  
6         <?php  
7             $a = 5;  
8             $b = 3;  
9             if($a>$b)  
10                echo "Khoảng chênh lệch giữa $a và $b là: " . ($a-$b);  
11             else  
12                echo "Khoảng chênh lệch giữa $a và $b là: " . ($b-$a);  
13         ?>  
14     </body>  
15 </html>
```


Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

- Cú pháp lệnh Switch:

```
switch(bt){  
    case h1: Lệnh 1; break;  
    case h2: Lệnh 2; break;  
    ...  
    case hn: Lệnh n; break;  
    default: Lệnh n+1; [break;]  
}
```

- Chức năng: nếu bt nhận giá trị là h nào (h1, h2..., hn) thì Lệnh tương ứng (h1, h2..., hn) sẽ được thực hiện. Nếu bt không nhận bất kì giá trị h nào, thì lệnh ở default sẽ được thực hiện (Lệnh n+1).

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

- Ví dụ lệnh Switch:

```
1 <html>  
2 <head>  
3 <meta http-equiv="content-type" charset="utf-8" />  
4 </head>  
5 <body>  
6 <?php  
7 $thang = 3;  
8 switch($thang){  
9     case 1: case 2: case 3:  
10         echo "Quý I"; break;  
11     case 4: case 5: case 6:  
12         echo "Quý II"; break;  
13     case 7: case 8: case 9:  
14         echo "Quý III"; break;  
15     case 10: case 11: case 12:  
16         echo "Quý IV"; break;  
17     default:  
18         echo "Không phải là tháng";  
19 }  
20 ?>  
21 </body>  
22 </html>
```

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Câu lệnh for và foreach

- Cú pháp lệnh For:

```
for(bt1; bt2; bt3){  
    Lệnh;  
}
```

- Chức năng: Lệnh được thực hiện cho đến khi bt2 không còn đúng. bt1 là biểu thức khởi gán, bt2 là biểu thức điều kiện; bt3 là biểu thức tăng/giảm.

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Câu lệnh for và foreach

- Cú pháp lệnh For:

```
for(bt1; bt2; bt3){  
    Lệnh;  
}
```

- Chức năng: Lệnh được thực hiện cho đến khi bt2 không còn đúng. bt1 là biểu thức khởi gán, bt2 là biểu thức điều kiện; bt3 là biểu thức tăng/giảm.

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Câu lệnh for và foreach

• Ví dụ lệnh For:

```
1 <html>
2   <head>
3     <meta http-equiv="content-type" charset="utf-8" />
4   </head>
5   <body>
6     <?php
7       $n=10;
8       for($i=0; $i<$n; $i++)
9         echo "<input type='button' value='Nút ".$i."'
10      onclick='alert(".$i.");' >";
11     ?>
12   </body>
</html>
```

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Câu lệnh for và foreach

• Cú pháp lệnh foreach :

```
foreach ($array as $key => $value){
    // Các dòng lệnh
}
```

- Chức năng : Vòng lặp foreach trong php dùng để lặp các phần tử trong mảng, chính vì thế nó được sử dụng rất nhiều khi chúng ta làm dự án với PHP, hầu như ai cũng thích nó vì sử dụng đơn giản và hiệu quả.

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Câu lệnh for và foreach

- Ví dụ foreach :

```
// Danh sách các năm  
$nam = array(  
    1990,  
    1991,  
    1992,  
    1993,  
    1994,  
    1995  
);
```

```
//Dùng foreach xuất ra các năm trong $nam  
foreach ($nam as $key => $value){  
    echo $value;  
}
```

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Lệnh for và foreach

2.4.3 Lệnh while và do...while

- Cú pháp lệnh while :

```
while(điều_kiện){  
    Lệnh;  
}
```

- Chức năng: Kiểm tra điều kiện, nếu đúng thì thực hiện Lệnh. Sau đó, tiếp tục quay trở lại kiểm tra điều kiện, nếu điều kiện còn đúng thì Lệnh còn thực hiện. Lệnh chỉ ngừng khi điều kiện sai.

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Lệnh for và foreach

2.4.3 Lệnh while và do...while

• Ví dụ lệnh while :

```
1 <html>
2   <head>
3     <meta http-equiv="content-type" charset="utf-8" />
4   </head>
5   <body>
6     <?php
7       function makeRGB(){
8         return "rgb(".rand(0, 255).",".rand(0, 255).",".rand(0, 255).")";
9       }
10      $n=10;
11      $i=0;
12      while($i<$n){
13        echo "<div style='background-color:".makeRGB()."';></div>";
14        $i++;
15      }
16    ?>
17    <style>
18      div{width:50px; height:50px; float:left;}
19    </style>
20  </body>
21 </html>
```

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Lệnh for và foreach

2.4.3 Lệnh while và do...while

• Cú pháp lệnh do...while :

```
do{
    Lệnh;
}while(điều_kiện);
```

- Chức năng: Thực hiện Lệnh, sau đó kiểm tra điều kiện. Nếu điều kiện còn đúng, thì tiếp tục thực hiện lệnh. Lệnh chỉ ngừng, khi điều kiện sai.

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Lệnh for và foreach

2.4.3 Lệnh while và do...while

2.5 Hàm

2.5.1 Khai báo

- Cú pháp:

```
function Tên_Hàm(tham_số){  
    //nội dung  
    //return giá trị nếu cần  
}
```

Trong đó,

function: từ khóa bắt buộc để khai báo hàm.

return: nếu hàm trả về giá trị, thì cần có return giá trị.

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Lệnh for và foreach

2.4.3 Lệnh while và do...while

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

- Ví dụ :

```
1 <html>  
2 <head>  
3 <meta http-equiv="content-type" charset="utf-8" />  
4 </head>  
5 <body>  
6 <?php  
7     function Tong($a, $b){  
8         return $a+$b;  
9     }  
10    function InTB($a, $b){  
11        echo "Tổng của $a và $b là: ".Tong($a, $b);  
12    }  
13    InTB(4, 5);  
14    ?>  
15 </body>  
16 </html>
```

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Lệnh for và foreach

2.4.3 Lệnh while và do...while

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

- Hàm tự định nghĩa với các tham số:

Cú pháp:

```
function function_name($gt1,$gt2)
{
//Hành động
}
```

Ví dụ : <?php

```
function indulieu($text)
{
echo "$text<br>";
}
indulieu("welcome");
indulieu("who are you ?");
?>
```

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Lệnh for và foreach

2.4.3 Lệnh while và do...while

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

- Hàm tự định nghĩa với giá trị trả về

Cú pháp :

```
function function_name(Có hoặc không có đối số)
{
//Thực thi
return giatri;
}
```

Ví dụ :

```
<?php
function tinhtong($a,$b)
{
$total=$a+$b;
return $total;
}
echo tinhtong(19,31); ?>
```

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Lệnh for và foreach

2.4.3 Lệnh while và do...while

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

- Trong PHP, một biến không cần phải khai báo nhưng muốn sử dụng thì cần khởi tạo giá trị cho nó.
- Để biết được một biến đã được khởi tạo giá trị hay chưa, ta sử dụng hàm `isset`:

isset(tên_biến)

- Hàm `isset` sẽ trả về giá trị `true`, nếu biến đó đã khởi tạo giá trị; ngược lại, nếu chưa khởi tạo giá trị, thì `isset` sẽ trả về giá trị `false`.

```
1 <html>
2   <head>
3     <meta http-equiv="content-type" charset="utf-8" />
4   </head>
5   <body>
6     <?php
7       echo (isset($a))?$a:"Biến a chưa khởi tạo !";
8       echo "<br/>---<br/>";
9       $a = 12;
10      echo (isset($a))?$a:"Biến a chưa khởi tạo !";
11    <?>
12  </body>
13 </html>
```

Page 41

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Lệnh for và foreach

2.4.3 Lệnh while và do...while

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

- Nếu sử dụng một biến chưa khai báo giá trị thì sẽ có thông báo lỗi :

Notice: Undefined variable: a in C:\xampp\htdocs\baigiangPHP\index.php on line 7

```
1 <html>
2   <head>
3     <meta http-equiv="content-type" charset="utf-8" />
4   </head>
5   <body>
6     <?php
7       echo $a;
8     <?>
9   </body>
10 </html>
```

Page 42

Chương 2. Ngôn ngữ lập trình PHP

2.1 Giới thiệu

2.2 Biến và hằng

2.2.1 Biến

2.2.2 Hằng

2.3 Toán tử và Biểu thức

2.4 Các lệnh cơ bản

2.4.1 Lệnh IF và SWITCH

2.4.2 Lệnh for và foreach

2.4.3 Lệnh while và do...while

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

- Nếu khi không cần sử dụng đến một biến nào đó, ta có thể giải phóng vùng bộ nhớ được cấp phát cho biến đó để tránh lãng phí, khi đó, ta có thể dùng hàm unset:

`unset(tên_biến);`

- Hàm unset sẽ giải phóng vùng bộ nhớ được cấp cho biến được chỉ định.

```
1 <html>
2 <head>
3 <meta http-equiv="content-type" charset="utf-8" />
4 </head>
5 <body>
6 <?php
7     $a = 12;
8     $b = 5;
9     echo $a."<br/>";
10    echo $b."<br/>";
11    unset($a);
12    unset($b);
13    echo $a."<br/>";
14    echo $b."<br/>";
15    ?>
16 </body>
17 </html>
```

localhost:89/baigiangPHP/index.php

12
5

Notice: Undefined variable: a in C:\xampp\htdocs\baigiangPHP\index.php on line 13

Notice: Undefined variable: b in C:\xampp\htdocs\baigiangPHP\index.php on line 14

Page 43

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

- PHP cung cấp nhiều hàm cho phép triệu gọi lại file. Như hàm `include("URL đến file")`, `require("URL Đến file")`.
- Ngoài hai cú pháp trên còn có `include_once()`, `require_once()`. Hai hàm này cũng có trách nhiệm gọi lại hàm. Những chúng sẽ chỉ gọi lại duy nhất 1 lần mà thôi.
- Lệnh include và require: dùng để chèn một nội dung từ một file php vào một file php khác.

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

```
include("menu.php");
```

```
include "menu.php";
```

```
require("menu.php");
```

```
require "menu.php";
```

về cơ bản, chúng tương đối giống nhau.

nhưng khác nhau trong trường hợp file được chèn vào không tồn tại, hoặc sai đường dẫn thì:

- Lệnh include: đưa ra cảnh báo, bỏ qua phần mã trong file chèn vào và tiếp tục chạy các lệnh tiếp theo (cảnh báo warning).
- Lệnh require: đưa ra một thông báo lỗi, dừng toàn bộ việc thực thi mã lệnh trên trang (lỗi compile_error).

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

- Ví dụ sử dụng lệnh include

1) Menu.php

```
1 <html>
2   <head>
3     <meta http-equiv="content-type" charset="utf-8" />
4   </head>
5   <body>
6     <?php include "menu.php"?>
7     <?php
8       echo "Hello, PHP!";
9     ?>
10  </body>
11 </html>
```

2) Index.php

```
1 <html>
2   <head>
3     <meta http-equiv="content-type" charset="utf-8" />
4   </head>
5   <body>
6     <?php include "menu.php"?>
7     <?php
8       echo "Hello, PHP!";
9     ?>
10  </body>
11 </html>
```

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

- Ví dụ sử dụng lệnh require

1) Module.Php

```
1 <?php
2     function Tong($a, $b){
3         return $a+$b;
4     }
5 >?>
```

2) Index.Php

```
1 <html>
2     <head>
3         <meta http-equiv="content-type" charset="utf-8" />
4     </head>
5     <body>
6         <?php require "module.php";?>
7         <?php
8             echo Tong(4, 5)."<br/>";
9             echo "Hello, PHP!";
10        ?>
11    </body>
12 </html>
```

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

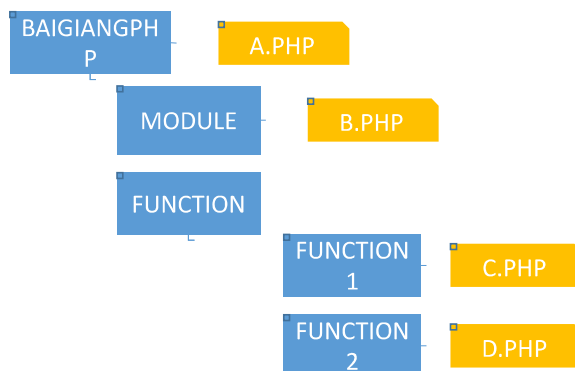
2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

- B.PHP
- include “../FUNCTION/FUNCTION1/C.PHP”;
- include “../FUNCTION/FUNCTION1/D.PHP”;
- A.PHP
- include “/MODULE/B.PHP”;



Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

- Khi sử dụng include/require, nếu một tập tin nào đó được chèn vào hai lần, thì sẽ tạo ra một lỗi. Để tránh tình huống này, php cho ta hai hàm tương ứng:
- Include_once
- Require_once
- Hai hàm này chỉ chèn tập tin vào một lần duy nhất (dù có gọi nhiều lần).
- Include_once("abc.Php");
- Include_once("abc.Php");
- Require_once("xyz.Php");
- Require_once("xyz.Php");

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

2.6 Kiểu dữ liệu mảng

2.6.1 Phân loại và khai báo mảng

• KHỞI TẠO MẢNG.

```
$mảng = array();
```

```
$mảng = array($gt1, $gt2,...);
```

```
<?php
//mảng rỗng
$a = array();

//mảng có dữ liệu thuần nhất
$b = array("Nam", "Bình", "Xuân");

//mảng có dữ liệu hỗn tạp
$c = array(1, "Nam", date_create());

//mảng chứa mảng
$d = array(array("Nam", 21), array("Bình", 20), array("Xuân", 19));
?>
```

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

2.6 Kiểu dữ liệu mảng

2.6.1 Phân loại và khai báo mảng

2.6.2 Làm việc với mảng

- Duyệt mảng : Để duyệt phần tử của mảng bằng vòng lặp for, ta cần sử dụng hàm count() để biết kích thước (số lượng phần tử) của mảng:

count(\$mảng)

```
<?php
$a = array("Nam", "Bình", "Xuân");
for($i=0; $i<count($a); $i++)
    echo "a[$i] = ".$a[$i]."<br/>";

echo "----<br/>";

$b = array(12, 23, 10, 5);
$sum = 0;
for($i=0; $i<count($b); $i++){
    $sum+=$b[$i];
    echo "b[$i] = ".$b[$i]."<br/>";
}
echo "Tổng là: $sum";
?>
```

a[0] = Nam
a[1] = Bình
a[2] = Xuân

b[0] = 12
b[1] = 23
b[2] = 10
b[3] = 5
Tổng là: 50

Page 51

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

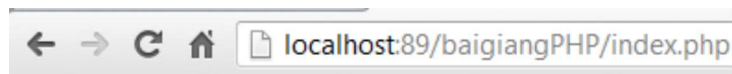
2.6 Kiểu dữ liệu mảng

2.6.1 Phân loại và khai báo mảng

2.6.2 Làm việc với mảng

- Duyệt mảng bằng for

```
<?php
$sv = array(array("Nam", 21), array("Tuấn", 20), array("Bình", 20));
for($i=0; $i<count($sv); $i++){
    echo "<b>Sinh viên</b>: ";
    for($j=0; $j<count($sv[$i]); $j++){
        echo $sv[$i][$j]." ";
    }
    echo "<br/>";
}
?>
```



Sinh viên: Nam 21

Sinh viên: Tuấn 20

Sinh viên: Bình 20

Page 52

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

2.6 Kiểu dữ liệu mảng

2.6.1 Phân loại và khai báo mảng

2.6.2 Làm việc với mảng

- Duyệt mảng bằng foreach: Foreach sẽ truy cập vào từng phần tử để tách giá trị của mảng, không yêu cầu truy cập thông qua chỉ số.

foreach(\$mảng as \$key=>\$giá_trị){...thao tác với \$giá_trị...}

```
<?php
$a = array("Nam", "Bình", "Xuân");
foreach($a as $key=>$value)
    echo "a[$key]=$value <br/>";

echo "----<br/>";

$b = array(12, 23, 10, 5);
$sum = 0;
foreach($b as $key=>$value){
    $sum += $value;
    echo "b[$key]=$value <br/>";
}
echo "Tổng: $sum";
?>
```



```
a[0]=Nam
a[1]=Bình
a[2]=Xuân
---
b[0]=12
b[1]=23
b[2]=10
b[3]=5
Tổng: 50
```

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

2.6 Kiểu dữ liệu mảng

2.6.1 Phân loại và khai báo mảng

2.6.2 Làm việc với mảng

- Duyệt mảng bằng foreach: Foreach sẽ truy cập vào từng phần tử để tách giá trị của mảng, không yêu cầu truy cập thông qua chỉ số.

foreach(\$mảng as \$giá_trị) {...thao tác với \$giá_trị...}

```
<?php
$sv = array(array("Nam", 21), array("Tuấn", 20), array("Bình", 20));
foreach($sv as $value){
    echo "<b>Sinh viên</b>:";
    foreach($value as $v){
        echo $v." ";
    }
    echo "<br/>";
}
?>
```



```
Sinh viên:Nam 21
Sinh viên:Tuấn 20
Sinh viên:Bình 20
```

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

2.6 Kiểu dữ liệu mảng

2.6.1 Phân loại và khai báo mảng

2.6.2 Làm việc với mảng

- Duyệt mảng bằng foreach: Foreach sẽ truy cập vào từng phần tử để tách giá trị của mảng, không yêu cầu truy cập thông qua chỉ số.

foreach(\$mảng as \$key=>\$giá_trị){...thao tác với \$giá_trị...}

```
<?php
$sv = array(array("Nam", 21), array("Tuấn", 20), array("Bình", 20));
foreach($sv as $value){
    echo "<b>Sinh viên</b>:";
    foreach($value as $v){
        echo $v." ";
    }
    echo "<br/>";
}
?>
```



Sinh viên:Nam 21

Sinh viên:Tuấn 20

Sinh viên:Bình 20

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bản hàm include và require

2.6 Kiểu dữ liệu mảng

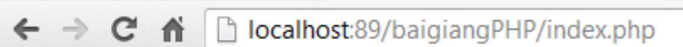
2.6.1 Phân loại và khai báo mảng

2.6.2 Làm việc với mảng

- Mảng nâng cao : trong php, mảng không chỉ cho phép sử dụng chỉ số tuần tự 0, 1, 2... để đánh chỉ số mà ta có thể sử dụng một kiểu dữ liệu phức tạp hơn (không phải là số thứ tự) mà người ta gọi là khóa \$key. mảng có cấu trúc dạng này còn gọi là dạng cấu trúc dữ liệu bảng.

\$mảng = array(\$key1=>\$gt1, \$key2=>\$gt2,...);

```
<?php
$con = array("cả"=>"Tùng", "ba"=>"Hòa", "tu"=>"Lan", "út"=>"Minh");
echo "<b>Anh cả</b>: ".$con["cả"]."<br/>";
echo "<b>Em út</b>: ".$con["út"]."<br/>";
?>
```



Anh cả: Tùng

Em út: Minh

Chương 2. Ngôn ngữ lập trình PHP

2.5 Hàm

2.5.1 Khai báo

2.5.2 sử dụng hàm

2.5.3 Các thao tác trên biến bằng isset và unset

2.5.4 chia tách các module bằng hàm include và require

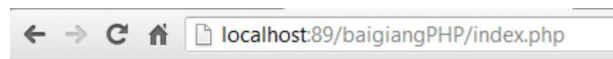
2.6 Kiểu dữ liệu mảng

2.6.1 Phân loại và khai báo mảng

2.6.2 Làm việc với mảng

- Mảng có khóa : với cấu trúc mảng dạng này, ta chỉ có thể sử dụng cấu trúc foreach để duyệt từng phần tử của mảng

```
<?php
$con = array("cả"=>"Tùng", "ba"=>"Hòa", "tu"=>"Lan", "út"=>"Minh");
foreach($con as $key=>$value){
    echo "Anh $key : ".$value."<br/>";
}
?>
```



```
Anh cả :Tùng
Anh ba :Hòa
Anh tư :Lan
Anh út :Minh
```


Bài giảng lý thuyết:

LẬP TRÌNH PHP&MYSQL

Faculty of IT

Email: khoacntt@hueic.edu.vn

Lập trình PHP&MYSQL

Chương 1. Tổng quan về lập trình website

Chương 2. Ngôn ngữ lập trình PHP

Chương 3. Hệ quản trị CSDL MySQL

Chương 4. Lập trình website động với PHP&MySQL

Chương 3. Hệ quản trị CSDL MySQL

3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

MỤC TIÊU

1. Hiểu được hệ quản trị CSDL MySQL
2. Hiểu và thành thạo các thao tác cơ bản trong CSDL MySQL
3. Thao tác thành thạo trên hệ quản trị CSDL PhpMyadmin

- PhpMyAdmin là một phần mềm được xây dựng trên nền tảng PHP và MySQL được sử dụng để quản lý cơ sở dữ liệu MySQL.
- PhpMyAdmin là phần mềm mã nguồn mở.
- PhpMyAdmin là phần mềm web based, cho nên nó độc lập với nền tảng.
- PhpMyAdmin hỗ trợ nhiều ngôn ngữ trong giao diện quản lý (nhưng hiện tại, chưa hỗ trợ tiếng Việt).
- PhpMyAdmin hỗ trợ 2 bộ engine để làm việc với MyISAM và InnoDB của MySQL là mysql và mysqli.
- Để truy cập vào PhpMyAdmin, ta nhập địa chỉ:
- <http://localhost/phpmyadmin/>

- MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở được phát triển bởi Sun Microsystems, nay được mua lại bởi Oracle.
- MySQL là hệ quản trị cơ sở dữ liệu phổ biến nhất.
- MySQL được sử dụng bởi rất nhiều công ty và tổ chức lớn như Facebook, Wikipedia, Wordpress...
- MySQL có nhiều bộ engine để quản lý, trong đó, thông dụng nhất là MyISAM và InnoDB. Trong đó, InnoDB mới hỗ trợ cơ sở dữ liệu quan hệ.
- Bộ engine MyISAM đơn giản, tốc độ thực thi truy vấn cao, phù hợp với người mới bắt đầu học về hệ quản trị CSDL.
- Các truy vấn trong MySQL vẫn tuân thủ theo cú pháp chuẩn của ngôn ngữ SQL.

- MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở được phát triển bởi Sun Microsystem, nay được mua lại bởi Oracle.
- MySQL là hệ quản trị cơ sở dữ liệu phổ biến nhất.
- MySQL được sử dụng bởi rất nhiều công ty và tổ chức lớn như Facebook, Wikipedia, Wordpress...
- MySQL có nhiều bộ engine để quản lý, trong đó, thông dụng nhất là MyISAM và InnoDB. Trong đó, InnoDB mới hỗ trợ cơ sở dữ liệu quan hệ.
- Bộ engine MyISAM đơn giản, tốc độ thực thi truy vấn cao, phù hợp với người mới bắt đầu học về hệ quản trị CSDL.
- Các truy vấn trong MySQL vẫn tuân thủ theo cú pháp chuẩn của ngôn ngữ SQL.

Tạo table :

```
CREATE TABLE <TenTable> (  
  TenField1 KiểuDuLieu (FieldSize) | NULL | NOT NULL |  
  AUTO_INCREMENT,  
  TenField2 KiểuDuLieu (FieldSize) | NULL | NOT NULL ,  
  ...  
  PRIMARY KEY (TenField) ,  
  UNIQUE (TenField),  
  FOREIGN KEY (TenField) REFERENCE TenTable  
);
```

Kiểu dữ liệu:

- Kiểu số nguyên: INT, INTEGER
- Số thực: FLOAT, DOUBLE
- Chuỗi: CHAR, VARCHAR
- Logic: BOOLEAN
- Ngày tháng: DATE, TIME, DATETIME
 - Sau Primary Key, UNIQUE có dấu phẩy.
 - UNIQUE là field có các giá trị không trùng nhau.
 - Foreign key nhiều record thì có dấu phẩy phía sau

- Ví dụ: Tạo table trong database QuanLyTinTuc

```
CREATE TABLE LOAISP(  
idLoai INT NOT NULL AUTO_INCREMENT,  
Ten VARCHAR(255) NOT NULL,  
ThuTu INT NULL,  
PRIMARY KEY (idLoai)  
);
```

Thêm dữ liệu vào table:

INSERT INTO <TenBang> VALUES (giaTri1, giatri2,...)

Các giá trị sẽ điền các giá trị vào các field theo đúng thứ tự xuất hiện trong bảng. Field làm khoá chính để chuỗi trống.

Ví dụ :

```
INSERT INTO LoaiSP VALUES ('', 'NoKia',2);
```

Thêm dữ liệu vào table:

**INSERT INTO <TenBang> (TenField1, TenField2,...)
VALUES (giaTri1, giatri2,...)**

- *Thứ tự các field có thể không nhất thiết giống như thứ tự các field trong bảng.*
- *Có thể không cần ghi mọi field.*
- *Thứ tự dữ liệu phải tương ứng với các field liệt kê phía trước.*

Ví dụ Thêm 1 Loại SP mới:

```
INSERT INTO LoaiSP (thutu, ten) VALUES ( 3, "Samsung");
```

Xem hết dữ liệu trong 1 table:

SELECT * <TenBang> ;

Ví dụ:

SELECT * FROM LoaiSP;

Chỉnh sửa dữ liệu:

UPDATE <tentable>

SET Tenfield1=giatri1, TenField2=giatri2

WHERE <ĐiềuKiện>

- Phần where có thể bỏ nhưng nguy hiểm

Ví dụ :

UPDATE LoaiSP SET ten='Motorola' where idLoai=1

Xoá dữ liệu:

DELETE FROM <TenBang> WHERE <ĐiềuKiện>

Phần where có thể bỏ nhưng nguy hiểm

Ví dụ:

DELETE FROM LoaiSP where idLoai=1

Truy vấn dữ liệu:

Cú pháp chung:

SELECT * | <TenField1,TenField2,...>

From <TenBang1, TenBang2,...>

WHERE <Điều kiện kết bảng> | <Điều kiện lọc>

ORDER BY <TênField1 ASC|DESC, TenField2 ASC|DESC ,...> // Điều kiện sắp xếp

GROUP BY <TenField1, TenField2,...> // Điều kiện phân nhóm

HAVING <Điều kiện lọc theo nhóm>

Truy vấn dữ liệu:

- Cú pháp chung:

SELECT * | <TenField1,TenField2,...>

From <TenBang1, TenBang2,...>

WHERE <Điều kiện kết bảng> | <Điều kiện lọc>

ORDER BY <TênField1 ASC|DESC, TenField2 ASC|DESC ,...>

GROUP BY <TenField1, TenField2,...> // Điều kiện phân nhóm

HAVING <Điều kiện lọc theo nhóm>

- Phần Select, From là bắt buộc. Phần where order by, group by, having là không bắt buộc
 - WHERE điều kiện lọc dữ liệu và điều kiện kết bảng
 - ORDER BY : sắp xếp dữ liệu theo field1, field 2.
 - Order by phải nằm sau where

Truy vấn dữ liệu:

- Phần Select, From là bắt buộc. Phần where order by, group by, having là không bắt buộc
 - WHERE điều kiện lọc dữ liệu và điều kiện kết bảng
 - ORDER BY : sắp xếp dữ liệu theo field1, field 2.
 - Order by phải nằm sau where

Câu lệnh Select đơn giản:

- SELECT idLT, Ten, AnHien FROM Loaitin ;
- SELECT idLT, Ten FROM LoaiTin WHERE AnHien=1;
- SELECT idTin, Ngay, TieuDe FROM Tin WHERE month(Ngay) =3;

Câu lệnh Select đơn giản:

- `SELECT idTin, Ngay, TieuDe FROM Tin WHERE Ngay=date('2008-4-3');`
- `SELECT idTin, Ngay, TieuDe FROM Tin WHERE Ngay=curdate();`
- ==> Lấy các tin trong ngày hôm nay (Bạn phải chỉnh đồng hồ của máy lại thành 1 ngày có tin trong database thì mới thấy tin)
- `SELECT Ngay, TieuDe FROM Tin WHERE month(Ngay)=month(curdate());`
- ==> Lấy các tin trong tháng hiện tại (Bạn phải chỉnh đồng hồ của máy lại thành tháng có tin trong database thì mới thấy tin)

Câu lệnh select có sắp xếp dữ liệu:

- `SELECT idLT, Ten, ThuTu FROM Loaitin ORDER BY Ten ASC;`
- `SELECT * FROM TheLoai WHERE AnHien=1 ORDER BY ThuTu ASC;`
- `SELECT idTin, Ngay, TieuDe FROM Tin ORDER BY Ngay DESC;`
- ==> Lấy mọi tin, tin mới hiện trước
- `SELECT idTin, Ngay, TieuDe, idLT FROM Tin WHERE idLT=1 ORDER BY Ngay DESC;`

Câu lệnh select có sắp xếp dữ liệu:

- `SELECT idTin, Ngay,TieuDe FROM Tin ORDER BY Ngay DESC limit 0,10;`
- `SELECT idTin, SoLanXem,TieuDe FROM Tin ORDER BY SoLanXem DESC limit 0,10;`
- `SELECT SoLanXem,TieuDe FROM Tin WHERE idLT=9 ORDER BY SoLanXem DESC limit 0,10;`

Câu lệnh select có phân nhóm:

Ví dụ:

`SELECT idUser, count(idTin) as SoTin FROM tin GROUP BY idUser;`

`SELECT idSK, count(idTin) as SoTin FROM tin GROUP BY idSK ;`

`SELECT idUser, count(idTin) as SoTin FROM tin GROUP BY idUser HAVING count(idTin) >=20`

Câu lệnh select tìm kiếm

- Dùng phép toán like kết hợp với ký tự % , _ (ký tự đại diện)

Ví dụ:

```
SELECT idTin, Ngay, TieuDe FROM Tin where TieuDe like 'C%'
```

==>

```
SELECT Username, HoTen, Email FROM Users where Username like 't_'
```

Câu lệnh select có kết bảng

- Là các câu lệnh lấy dữ liệu từ nhiều bảng. Bạn phải ghi điều kiện kết bảng trong phần where

Ví dụ 1:

```
SELECT idTin, Tin.idSK, Mota , TieuDe FROM Tin, SuKien WHERE Tin.idSK= SuKien.idSK ORDER BY idSK ASC
```

Ví dụ 2:

```
SELECT idTin, User.Username, Loaitin.Ten, TieuDe FROM Tin, User, LoaiTin WHERE Tin.idUser= User.idUser AND Tin.idLT=LoaiTin.idLT Order By Ngay Desc Limit 0,10
```

Câu lệnh select lồng nhau:

Là câu lệnh sql có nhiều câu “select” lồng nhau

Ví dụ:

1. `SELECT idtin, Ngay, TieuDe FROM tin WHERE Ngay >= All (SELECT Ngay FROM Tin)`
2. `SELECT idLT, Ten FROM loaitin WHERE idLT in (SELECT idLT FROM Tin group by idLT)`

Select ngẫu nhiên

`select idBC, MoTa from BinhChon order by rand() limit 0,1;`

`select idTin, TieuDe from tin order by rand() limit 0,3;`

Chương 3. Hệ quản trị CSDL MySQL

3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

- Tạo mới cơ sở dữ liệu:
- Chọn thẻ Database > Create Database.
- Nhập tên database vào textbox. Khi đặt tên cho database, nên đặt tên có tiền tố là db_
- Chọn bảng mã hỗ trợ tương ứng với Collation. Để hỗ trợ tiếng việt, nên chọn bảng mã là utf8_unicode_ci.
- Bấm vào nút create.
- Cơ sở dữ liệu mới tạo sẽ xuất hiện ở cột phía bên trái. Bấm vào cơ sở dữ liệu mới tạo để sử dụng nó và thực hiện các thao tác tiếp theo: thêm, sửa, xóa... dữ liệu. Chú ý quan sát kỹ thanh điều hướng:

Chương 3. Hệ quản trị CSDL MySQL

3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

- Một số thao tác trên cơ sở dữ liệu đã tạo:
- Xóa: chọn cơ sở dữ liệu cần xóa, chọn drop.
- Chỉnh sửa, chọn cơ sở dữ liệu cần chỉnh sửa. Chọn thẻ structure và hiệu chỉnh. Sau đó, chọn save.
- Nhập dữ liệu từ một tập tin sql vào trong cơ sở dữ liệu: chọn cơ sở dữ liệu cần nhập dữ liệu từ file sql vào. Chọn thẻ Import, chọn file sql và nhấp Go.
- Xuất dữ liệu từ một cơ sở dữ liệu cho trước ra file sql: chọn cơ sở dữ liệu cần xuất dữ liệu (để sao lưu dự phòng). Chọn thẻ Export, chọn SQL, chọn Go và lưu lại file sql.

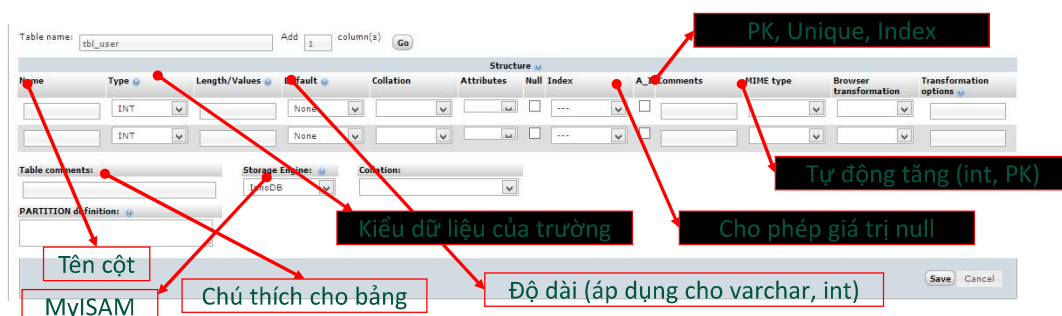
Chương 3. Hệ quản trị CSDL MySQL

3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

- Tạo mới bảng dữ liệu - khi đã chọn cơ sở dữ liệu cần làm việc, ta có thể tiến hành tạo mới bảng dữ liệu theo các thao tác:
- Chọn mục create table.
- Nhập tên cho bảng dữ liệu ở mục Name và nhập số cột tương ứng với các trường trong bảng dữ liệu ở mục Column.
- Nhấp nút save để lưu lại.



Chương 3. Hệ quản trị CSDL MySQL

3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

- Một số thao tác khác trên bảng dữ liệu:
- Thêm mới dữ liệu: chọn bảng dữ liệu cần thêm mới, chọn thẻ Insert và nhập dữ liệu, sau đó chọn save.
- Chỉnh sửa cấu trúc của bảng: chọn bảng dữ liệu cần chỉnh sửa cấu trúc, chọn thẻ Structure, chỉnh sửa, sau đó chọn save.
- Thêm mới một/nhiều cột: chọn bảng dữ liệu cần thêm mới cột, chọn Structure, chọn Add column bên dưới, nhập số cột cần thêm và nhấp Go. Sau đó, nhập tên cột và chỉnh các thông số cần thiết. Cuối cùng bấm save.

Chương 3. Hệ quản trị CSDL MySQL

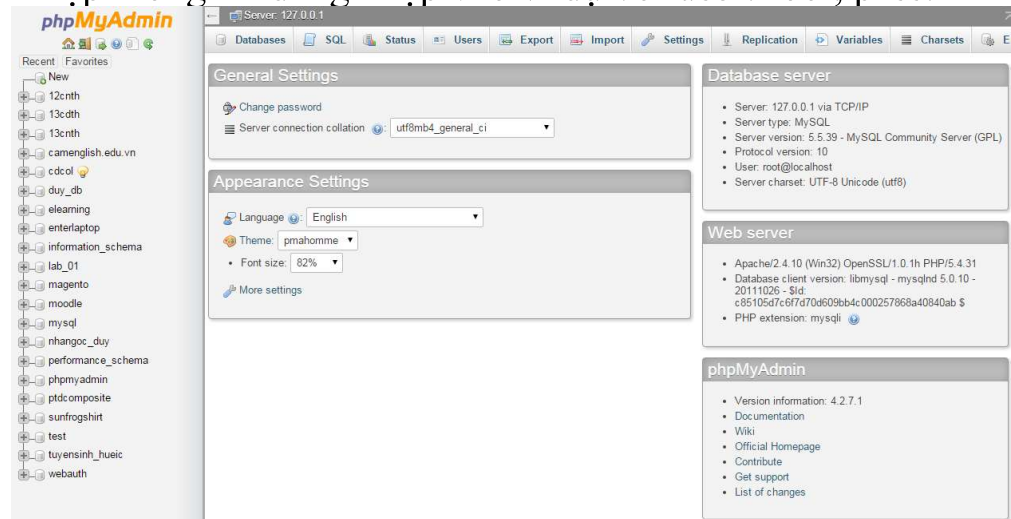
3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

- Truy cập vào PHPMYAdmin:

Để truy cập vào phpMyAdmin ở localhost, các bạn truy cập vào theo đường link: <http://localhost/phpmyadmin>. Sau đó nhập thông tin đăng nhập vào ví dụ với user: root, pass:



Chương 3. Hệ quản trị CSDL MySQL

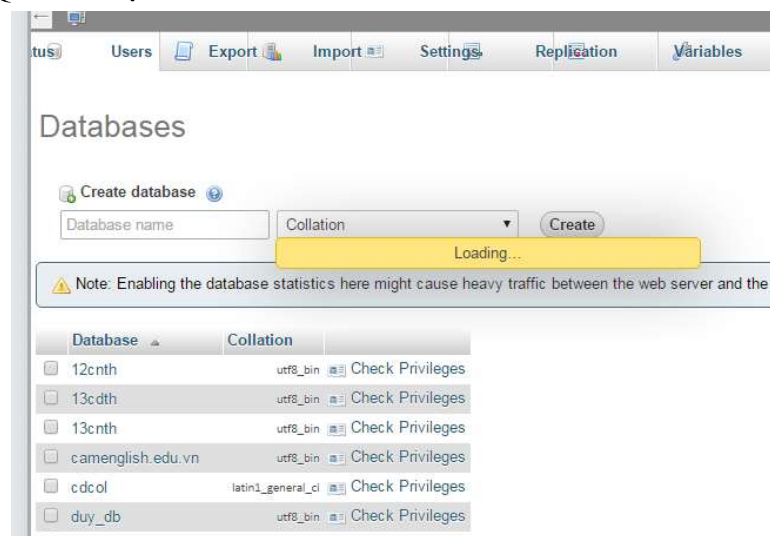
3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

- Quản lý cơ sở dữ liệu:

Để truy cập vào phần quản lý các CSDL của hệ thống, các bạn truy cập vào phần Databases tại đây các CSDL của hệ thống MySQL sẽ hiện ra như hình sau:



Chương 3. Hệ quản trị CSDL MySQL

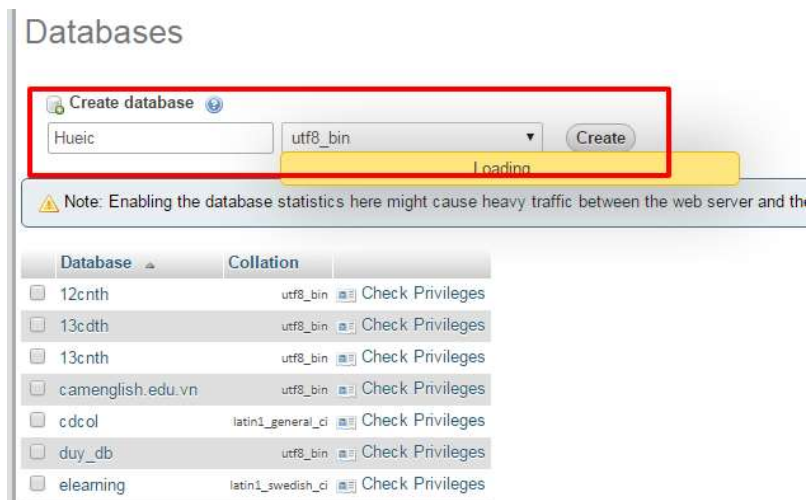
3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

• Tạo CSDL:

Để tạo một cơ sở dữ liệu, tại phần Create database, các bạn nhập tên database muốn khởi tạo, phần Collation chọn utf8_general_ci sau đó bấm nút Create là một cơ sở dữ liệu sẽ được tạo ra cho bạn.



Chương 3. Hệ quản trị CSDL MySQL

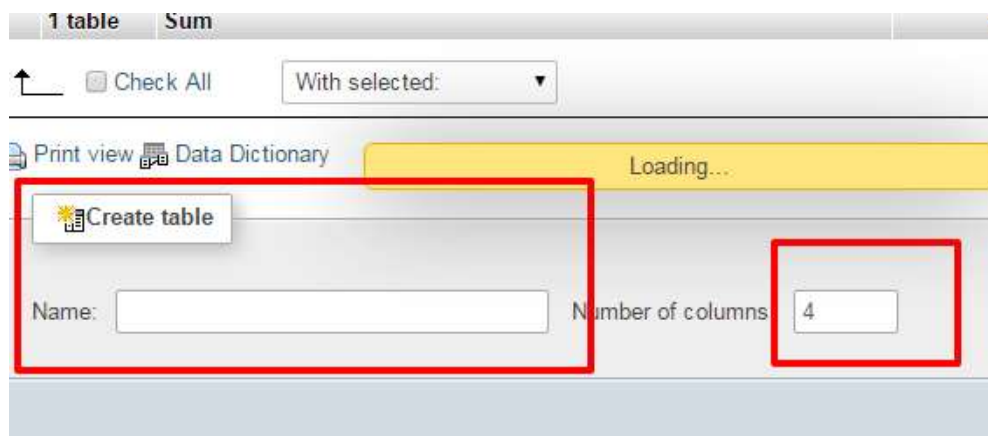
3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

• Quản lý table (bảng dữ liệu):

Sau khi tạo được xong CSDL, tên CSDL của bạn ở phần menu bên trái sẽ được hiện ra. Các bạn ấn vào tên CSDL ở phía tay trái này để có thể truy cập vào trang quản lý CSDL của mình vừa tạo ra như hình sau:



Chương 3. Hệ quản trị CSDL MySQL

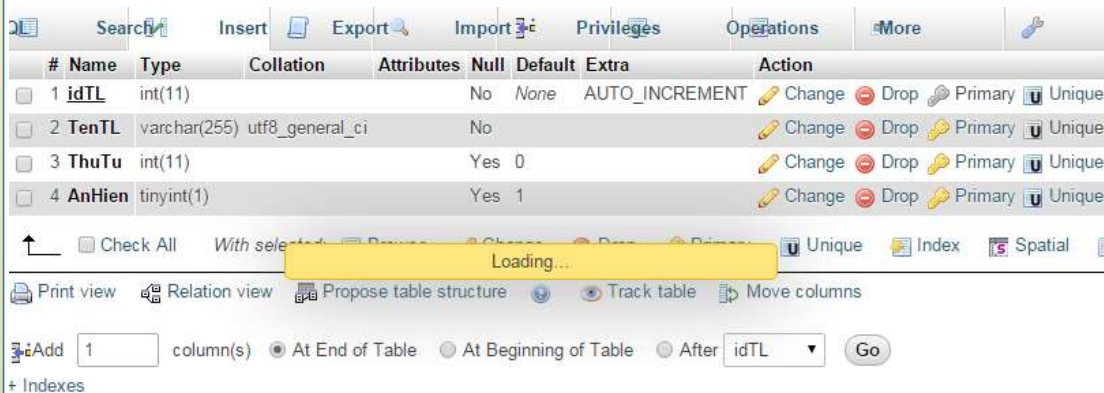
3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

• Tạo Bảng dữ liệu:

Tại mục “Create Table” các bạn nhập tên bảng muốn khởi tạo, số field (trường) sau đó bấm nút “Go“. Một cửa sổ nhập các thông tin về Table sẽ hiện ra để bạn nhập các thông tin và khởi tạo như hình dưới đây:



Chương 3. Hệ quản trị CSDL MySQL

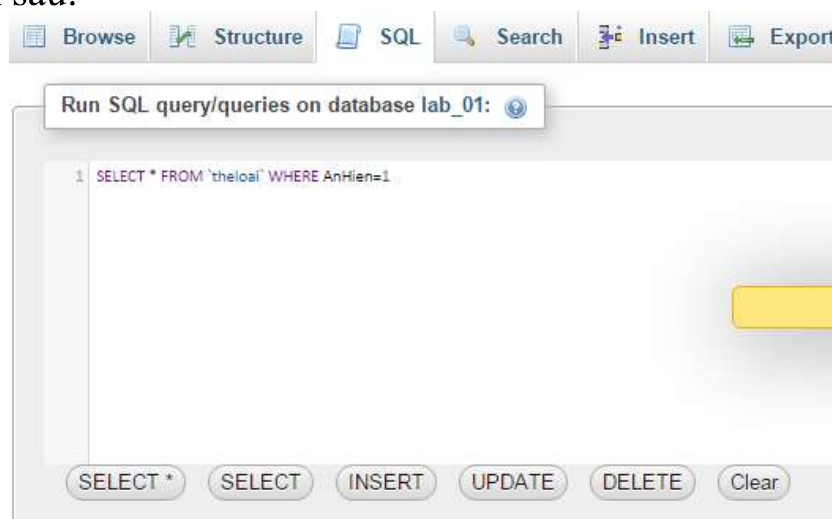
3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

• Thực hiện truy vấn:

Để thực hiện được các truy vấn SQL trong phpMyAdmin các bạn truy cập Tab “SQL” từ cửa sổ giao diện quản lý của một cơ sở dữ liệu. Cửa sổ thực hiện các lệnh SQL sẽ hiện ra như hình sau:



Chương 3. Hệ quản trị CSDL MySQL

3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

- Thực hiện truy vấn:

Chú ý :

- Các bạn gõ lệnh SQL vào cửa sổ nhập nội dung, sau đó bấm nút “**Go**” để hệ thống thực hiện chạy các truy vấn đó. Mỗi truy vấn được phân cách nhau bởi dấu “;” chấm phẩy.
- Trong quá trình lập trình viết Code, nếu muốn biết lỗi gì xảy ra với cú pháp SQL của mình, các bạn chỉ việc in câu lệnh SQL ra bằng cú pháp PHP, sau đó copy vào đây để kiểm tra lỗi của câu lệnh.

Chương 3. Hệ quản trị CSDL MySQL

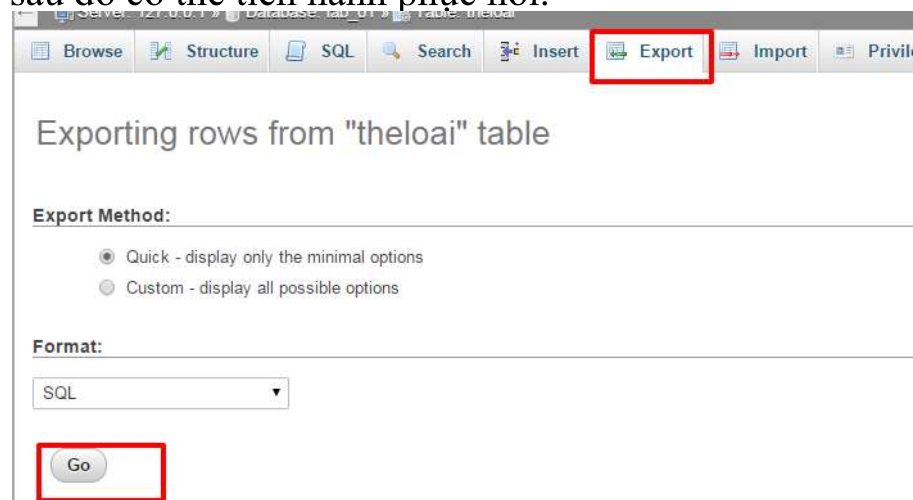
3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

- Sao lưu cơ sở dữ liệu:

Để sao lưu được cơ sở dữ liệu, các bạn truy cập thẻ “**Export**”. Tính năng này của phpMyAdmin có công dụng trích xuất toàn bộ CSDL của các bạn ra thành các cú pháp câu lệnh SQL để sau đó có thể tiến hành phục hồi.



Chương 3. Hệ quản trị CSDL MySQL

3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

- Phục hồi cơ sở dữ liệu

Để phục hồi được CSDL thông qua tính năng “Import“. Bạn cần sở hữu một tập tin *.sql được tải về thông qua quá trình Export. Sau khi truy cập thẻ “Import” các bạn sẽ có cửa sổ giao diện như hình sau:



Chương 3. Hệ quản trị CSDL MySQL

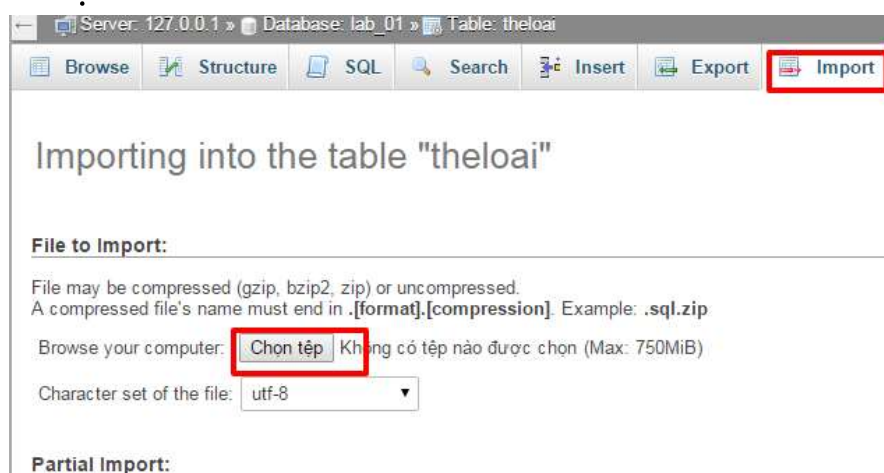
3.1 Giới thiệu

3.2 Các truy vấn

3.3 Quản trị dữ liệu bằng PhpMyadmin

- Phục hồi cơ sở dữ liệu

Để phục hồi được CSDL thông qua tính năng “Import“. Bạn cần sở hữu một tập tin *.sql được tải về thông qua quá trình Export. Sau khi truy cập thẻ “Import” các bạn sẽ có cửa sổ giao diện như hình sau:



Bài giảng lý thuyết:

LẬP TRÌNH PHP&MYSQL

Faculty of IT

Email: khoacntt@hueic.edu.vn

Lập trình PHP&MYSQL

Chương 1. Tổng quan về lập trình website

Chương 2. Ngôn ngữ lập trình PHP

Chương 3. Hệ quản trị CSDL MySQL

Chương 4. Lập trình website động với PHP&MySQL

Chương 4. Lập trình website động với PHP & MySQL

4.1 Kết nối CSDL

4.2 Các thao tác cơ bản với CSDL

4.3 Session, Cookie và các ứng dụng

4.4 Kỹ thuật phân trang

4.5 Một số lỗi bảo mật trong PHP thường gặp và biện pháp khắc phục

MỤC TIÊU

1. **Nắm được phương pháp kết nối CSDL và làm việc được với CSDL bằng PHP**
2. **Hiểu và thao tác thành thạo trên CSDL, Cookie, Session**
3. **Nắm được các ứng dụng cơ bản : kỹ thuật phân trang, làm việc với CSDL.**

- Các bước xây dựng kết nối tới CSDL My SQL

Bước 1: Thiết lập kết nối tới CSDL.

Bước 2: Lựa chọn CSDL.

Bước 3: Tiến hành các truy vấn SQL, xử lý các kết quả trả về nếu có

Bước 4: Đóng kết nối tới CSDL.

- **Thiết lập kết nối tới MySQL**

Để kết nối tới MySQL, ta sử dụng hàm `mysql_connect()`

Cú pháp:

Với PHP 5

```
mysql_connect (host , ten_truy_cap , mat_khau ) ;
```

Với PHP 7:

```
mysqli_connect (host , ten_truy_cap , mat_khau ) ;
```

Trong đó:

`host` là chuỗi chứa tên (hoặc địa chỉ IP) của máy chủ cài đặt MySQL.

`ten_truy_cập` là chuỗi chứa tên truy cập hợp lệ của CSDL cần kết nối

`mat_khẩu` là chuỗi chứa mật khẩu tương ứng với tên truy cập

Ví dụ:

```
<? mysqli_connect ("localhost", "root","root"); ?>
```

• Thiết lập kết nối tới MySQL

Ví dụ :

```
<?php
    $conn = mysqli_connect("localhost", "mysql_user",
    "mysql_password");
    if (!$conn)
    {
        echo "Không thể kết nối được tới CSDL MySQL";
    }
?>
```

• Lựa chọn CSDL

Để lựa chọn một CSDL nào đó mà người sử dụng có tên là "tên_truy_cập" có quyền sử dụng, ta dùng hàm `mysql_select_db`:

```
mysqli_select_db (thông tin kết nối
, tên CSDL);
```

Hàm này thường được dùng sau khi thiết lập kết nối bằng hàm `mysql_connect`. Hàm này trả về true nếu thành công, false nếu thất bại.

• **Lựa chọn CSDL**

Code:

```
<?php
    $conn = mysqli_connect("localhost", "mysql_user",
"mysql_password");
    if (!$conn)
    {
        echo "Không thể kết nối được tới MySQL";
    }
    if (!mysqli_select_db ($conn,"database_name"))
    {
        echo "Không thể lựa chọn được CSDL Forums";
    }
?>
```

• **Đóng kết nối tới CSDL:**

Để đóng kết nối tới CSDL, ta dùng hàm `mysqli_close`.

Hàm này có nhiệm vụ đóng kết nối tới CSDL có mã định danh được tạo ra bởi hàm

`mysqli_close(connection)`

Cú pháp:

`mysqli_close(mã_định_danh_kết_nối)`.

```
<?php
    $conn = mysqli_connect("localhost", "mysql_user",
"mysql_password");
    if (!$conn)
    {
        echo "Không thể kết nối được tới MySQL";
    }
    mysqli_close($conn);
?>
```


- Thực hiện các truy vấn thêm, xóa, sửa bản ghi thông qua câu lệnh SQL:

Để thực thi một câu lệnh SQL bất kỳ trong PHP tác động lên MySQL, ta dùng hàm :

mysqli_query (ketnoi,chuỗi_câu_lệnh_SQL)

- Ví dụ :

```
Mysqli_query ( $conn,"Select * From TheLoai");
```

- Thêm một bản ghi vào bảng:

Ví dụ chèn thêm một menu liên hệ vào bản và link tới file contact.php :

```
<?php
```

```
$sql = " INSERT INTO `menu` (`idMenu`, `idCapCha`, `TieuDe`, `LienKet`, `ThuTu`, `AnHien`) VALUES (NULL, '0', 'Liên Hệ', 'contact.php', '1', '1') ";
```

```
mysqli_query ($conn,$sql);
```

```
?>
```

- Xóa một bản ghi khỏi bảng

Trong trường hợp này, ta sử dụng câu lệnh SQL DELETE FROM:

VD: Xóa khỏi bảng nncms_menus với menu có idMenu =30

```
<?
```

```
$sql = " DELETE FROM `menu` WHERE `idMenu` = 30 ";
```

```
mysqli_query ($conn,$sql);
```

```
?>
```

- Sửa thông tin của bản ghi trong bảng

Trong trường hợp này, chúng ta sử dụng cú pháp UPDATE:

VD: Cập nhật cấp cha cho menu Liên hệ có idMenu =30 :

```
<?
```

```
$sql = " UPDATE `nncms_menus` SET `idCapCha` = '1' WHERE `idMenu` =30 ";
```

```
mysqli_query ($conn,$sql);
```

```
?>
```

- Sửa thông tin của bản ghi trong bảng

Trong trường hợp này, chúng ta sử dụng cú pháp UPDATE:

VD: Cập nhật cấp cha cho menu Liên hệ có idMenu =30 :

```
<?
```

```
$sql = " UPDATE `nncms_menus` SET `idCapCha` = '1' WHERE `idMenu` =30 ";
```

```
mysqli_query ($conn,$sql);
```

```
?>
```

- Sử dụng SQL để truy vấn dữ liệu trong PHP

Dùng hàm **mysqli_query()** để tiến hành các truy vấn sửa, thêm và xóa. Bài này tập trung vào việc ứng dụng PHP để lấy thông tin từ CSDL.

Cú pháp hàm **mysqli_query** như sau:

```
mysqli_query($conn, câu lệnh sql).
```

- Sử dụng SQL để truy vấn dữ liệu trong PHP

VD: Để lấy tất cả các bản ghi trong bảng "menu", ta sẽ gọi hàm `mysqli_query()` như sau:

```
<?
```

```
mysqli_query ($conn,"Select * from menu")
```

```
?>
```

Vấn đề là khi thực hiện truy vấn này, dữ liệu sẽ được trả về dưới dạng một bảng. Ta sẽ tiến hành lấy từng dòng của bảng đó ra dưới dạng một mảng (sử dụng hàm `mysqli_fetch_array()`), tiếp đó, việc xử lý các phần tử của mảng đó như thế nào là tùy ý các bạn.

- Sử dụng SQL để truy vấn dữ liệu trong PHP

```
<?php
```

```
    $result = mysqli_query($conn,"SELECT * FROM theloai");
```

```
    while ($row = mysqli_fetch_array($result, MYSQL_ASSOC))
```

```
    {
```

```
        printf ("ID: %s Ten TL: %s", $row["TenTL"]);
```

```
    }
```

```
    mysqli_free_result($result); // giải phóng truy vấn
```

```
?>
```

- Sử dụng SQL để truy vấn dữ liệu trong PHP

Sau khi đã có được câu truy vấn SQL, ta chỉ việc kết hợp với hàm `mysqli_query()` để trả kết quả về dưới dạng một biến nào đó.

Phần tiếp theo, chúng ta sẽ dùng một hàm có tên là `mysqli_fetch_array()` để lấy từng dòng dữ liệu ra để xử lý.

- Hàm `mysqli_fetch_array` có hai tham số:

- Tham số đầu tiên xác định biến nắm giữ kết quả trả về bởi hàm `mysqli_query()`

- Tham số thứ hai xác định kiểu định danh trường:

- Sử dụng SQL để truy vấn dữ liệu trong PHP

- + `MYSQL_ASSOC`: Đưa ra một mảng với chỉ số của mảng là tên trường

- + `MYSQL_NUM`: Đưa ra một mảng với chỉ số của mảng là số thứ tự của trường được liệt kê trong câu truy vấn SQL

- + `MYSQL_BOTH`: Đưa ra một mảng với chỉ số kết hợp giữa hai loại trên

```
<?php
$conn = mysqli_connect("localhost",
"mysql_user", "mysql_password");
mysqli_select_db("database", $conn);
$result = mysqli_query($conn,"SELECT * FROM
table1", $conn);
$num_rows = mysqli_num_rows($result);
echo "Có $num_rows bản ghi được tìm thấy\n";
?>
```

```
<?php
$conn = mysqli_connect("localhost",
"mysql_user", "mysql_password");
mysqli_select_db("database", $conn);
$result = mysqli_query($conn,"SELECT * FROM
table1", $conn);
$num_rows = mysqli_num_rows($result);
echo "Có $num_rows bản ghi được tìm thấy\n";
?>
```

- **Đếm số record trong recordset**

```
mysqli_num_rows($TenRecordset);
```

Ví dụ: `echo mysqli_num_rows($kq);`

- **Duyệt recordset :**

```
$kq= mysqli_fetch_assoc($TenRecordSet);
```

- Hàm trả về thông tin của record hiện hành (hoặc giá trị false nếu record rỗng), đồng thời di chuyển con trỏ sang record kế
- Muốn hiện dữ liệu của field nào thì ghi như sau `$kq['TenField']`

- **Nhảy đến 1 record trong recordset**

```
mysqli_data_seek($TenRecordset, ThuTuRecord);
```

Ví dụ: `mysqli_data_seek($rdLT, 0);`

- **Thông báo lỗi của MYSQL**

```
mysql_error();
```

Ví dụ: `$kq=mysqli_query("Delete form quangcao") or die(mysql_error());`