

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN



BÀI GIẢNG MÔN HỌC

CÔNG NGHỆ WEB

TÀI LIỆU THAM KHẢO

Sách, bài giảng, giáo trình chính:

- [1] Mai Văn Hà, *Bài giảng Công Nghệ Web*, tài liệu lưu hành nội bộ, 2017
- [2] Internet & word wide web *How to program Deteil & Nieto*, NXB Prentice Hall, 2000.

Sách, tài liệu tham khảo:

- [1] Ried Robert.H, *Architecis of the Web*
- [2] Nguyễn Phương Lan, Hoàng Đức Hải, *Lập trình ứng dụng Web với JSP/Servlet*, Nhà xuất bản Lao động – Xã hội, 2003
- [3] Ebook “Thương mại Việt Nam hội nhập thế giới”

NỘI DUNG CHƯƠNG TRÌNH

Chương 1 : Internet và Web

Giới thiệu cơ bản về Internet và Web, giúp người học có một tầm nhìn về mạng Internet, các trang web đóng vai trò gì trên mạng Internet, nắm bắt được hệ thống web gồm những thành phần nào.

Chương 2 : Ngôn ngữ HTML & CSS

Trình bày nội dung cơ bản của ngôn ngữ HTML & CSS

Chương 3: JavaScript

Trình bày cách sử dụng các kỹ thuật lập trình đơn giản để điều khiển các sự kiện trên trang web.

Chương 4: Công nghệ Web động

ASP, PHP, JSP, SERVLET, (AJAX), ...

Chương 5: Cài đặt và cấu hình Web Server

Cài đặt và cấu hình Web Server để quản lý và vận hành Website

CHƯƠNG 1 : INTERNET VÀ WEB

I. KHÁI NIỆM VỀ MẠNG MÁY TÍNH

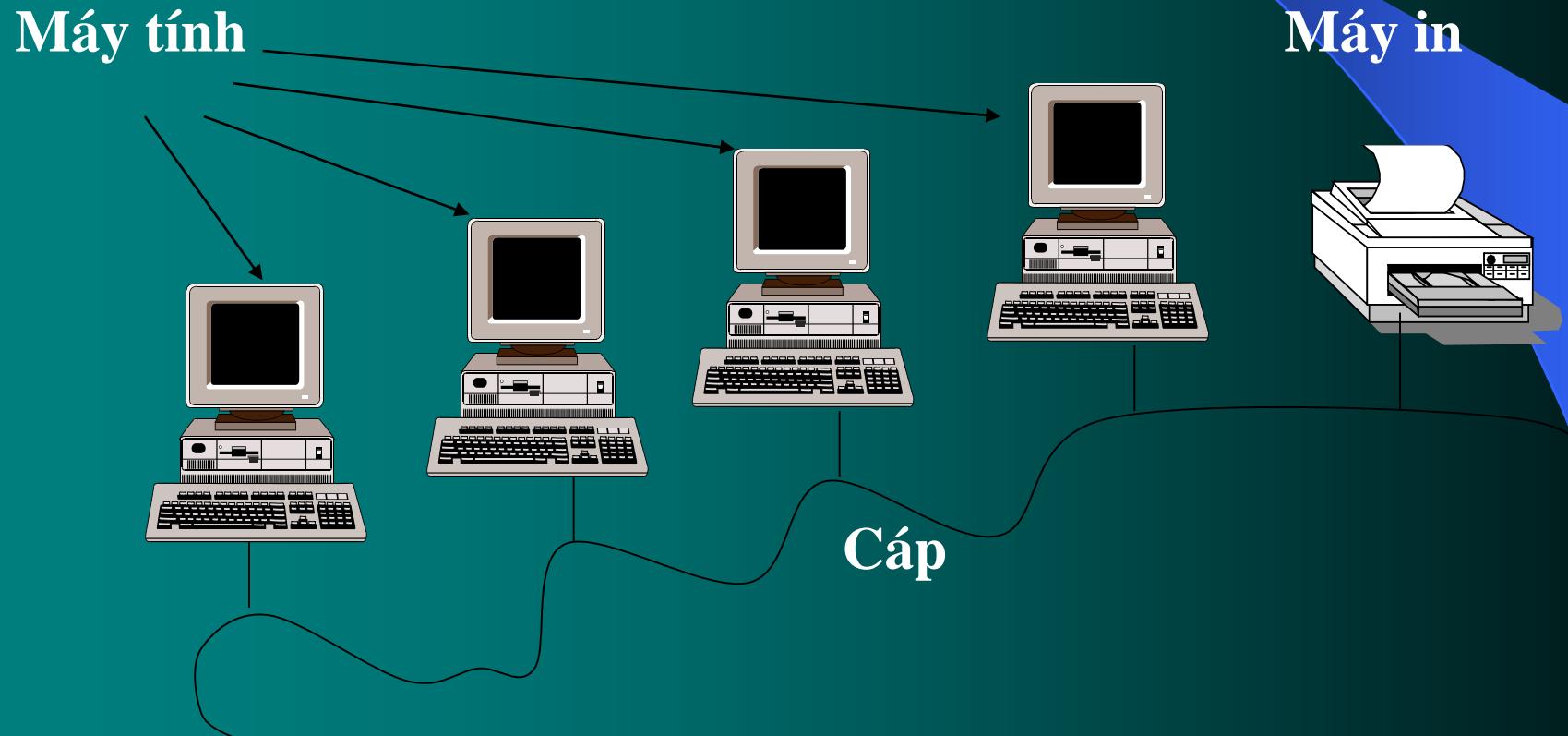
☞ *Mạng máy tính (network) là bao gồm các máy tính được nối với nhau sao cho chúng có thể dùng chung dữ liệu và các thiết bị.*

☞ *Các máy tính trong mạng có thể dùng chung các loại tài nguyên sau:*

- *Dữ liệu*
- *Thông điệp*
- *Hình ảnh*
- *Máy fax*
- *Modem*
- *Máy in*
- *Các tài nguyên phần cứng khác*

CHƯƠNG 1 : INTERNET VÀ WEB

Hệ thống mạng máy tính đơn giản



CHƯƠNG 1 : INTERNET VÀ WEB

II. CÁC LOẠI MẠNG MÁY TÍNH

1. MẠNG NỘI BỘ (LOCAL AREA NETWORK - LAN)

Mạng nội bộ (LAN) là một mạng của các máy tính được nối với nhau trong một phạm vi hẹp như trong một tòa nhà, một công sở nhờ một số loại cáp dẫn, không dùng thuê bao điện thoại.

2. MẠNG DIỆN RỘNG (WIDE AREA NETWORK - WAN)

Mạng diện rộng (WAN) là một mạng của nhiều mạng nội bộ nối với nhau thông qua đường dây điện thoại hoặc nhờ các công nghệ khác như hệ thống điện tử viễn thông hoặc vệ tinh.

CHƯƠNG 1 : INTERNET VÀ WEB

II. CÁC LOẠI MẠNG MÁY TÍNH

3. MẠNG INTERNET LÀ GÌ?



CHƯƠNG 1 : INTERNET VÀ WEB

II. CÁC LOẠI MẠNG MÁY TÍNH

3. MẠNG INTERNET LÀ GÌ?

Internet là liên mạng toàn cầu lớn nhất bao trùm lên tất cả các mạng khác phân bố trên phạm vi toàn thế giới.

+ Giúp các máy tính kết nối và trao đổi thông tin với nhau mà không cần biết vị trí địa lý của chúng

+ Cho phép truy cập lên các tài nguyên lưu trữ ở xa.

+ Cho phép trao đổi thư điện tử (Email), trò chuyện (Chat), gọi điện thoại, gửi tin nhắn,..

+ Các tư liệu thông tin được tổ chức theo từng trang (trang Web). Các trang này được liên kết với nhau nhờ các liên kết hay các chỉ mục.

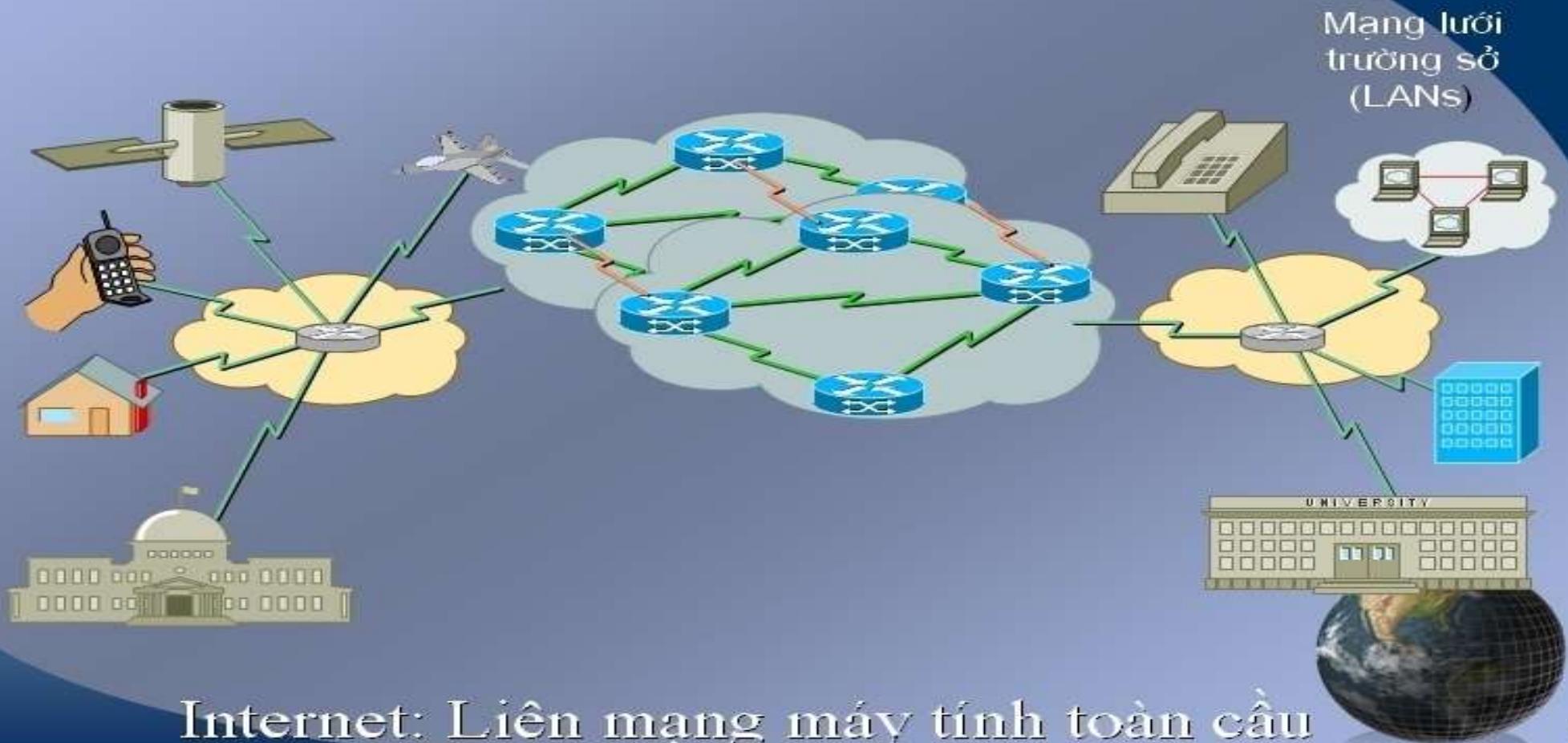
CHƯƠNG 1 : INTERNET VÀ WEB

Lịch sử phát triển Internet

- Tiền thân của Internet là ARPANET, 1960
- Năm 1983, giao thức TCP/IP chính thức được coi như một chuẩn cho việc kết nối máy tính
- Năm 1984, ARPANET đã được chia ra thành hai phần: phần thứ nhất vẫn được gọi là ARPANET, dành cho việc nghiên cứu và phát triển; phần thứ hai được gọi là MILNET, là mạng dùng cho các mục đích quân sự.
- WWW ra đời, 1989
- Ngày nay chúng ta thấy Internet xuất hiện trong mọi lĩnh vực : thương mại, chính trị, quân sự, nghiên cứu, giáo dục, văn hoá, xã hội ...

CHƯƠNG 1 : INTERNET VÀ WEB

Mô hình mạng Internet



INTERNET

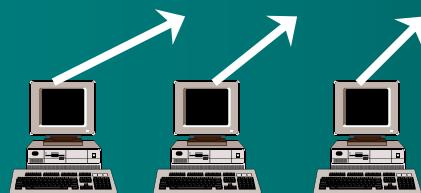
IAP: Internet Access Provider

Nhà cung cấp khả năng
truy cập Internet

Máy chủ



Người dùng
truy cập
trực tiếp



Người dùng truy cập từ xa
(điện thoại, thuê bao)

ISP

IAP

ISP

Nhà cung cấp dịch
vụ Internet

ISP

Máy chủ



Máy chủ



Mạng người dùng công cộng

ISP: Internet Service Provider

CHƯƠNG 1 : INTERNET VÀ WEB

III. CÁC KHÁI NIỆM

1. GIAO THỨC TCP/IP

TCP/IP là một giao thức để kết nối và truyền dữ liệu giữa các máy tính với nhau.

+ Mục đích chính của IP là cung cấp khả năng kết nối các mạng con thành liên mạng để truyền dữ liệu. IP là một giao thức kiểu không liên kết.

+ TCP là một giao thức kiểu "có liên kết", nghĩa là phải thiết lập liên kết giữa một cặp thực thể TCP trước khi chúng trao đổi dữ liệu với nhau.

CHƯƠNG 1 : INTERNET VÀ WEB

III. CÁC KHÁI NIỆM

2. ĐỊA CHỈ IP :

Địa chỉ IP dùng để các máy tính liên lạc được với nhau và phân biệt được nhau.

+ IP tĩnh và IP động

Cấu trúc của địa chỉ IP này gồm một chuỗi số 32 bit và được chia thành 4 nhóm, cách nhau bởi dấu chấm (.). Mỗi nhóm gồm 3 chữ số có giá trị từ 000 đến 255.

Ví dụ : địa chỉ IP của một máy tính : 192.168.2.3

CHƯƠNG 1 : INTERNET VÀ WEB

III. CÁC KHÁI NIỆM

3. Tên miền (Domain Name)

Địa chỉ IP là những dãy số, do đó người dùng Internet rất khó nhớ. Vì vậy người ta dùng khái niệm tên miền (Domain Name) để ánh xạ tới địa chỉ IP. Tuy nhiên tên miền chỉ là tên dành cho người dùng dễ nhớ, cốt lõi của địa chỉ Internet vẫn là IP.

Tên miền của một máy chủ gồm các chuỗi ký tự phân cách nhau bằng dấu chấm (.).

Khuôn dạng tên miền: **xxx.xxx.xxx.xxx**

- Nhóm ký tự cuối cùng đại diện cho tên nước.
- Nhóm kế tiếp đại diện cho loại cơ quan
- Nhóm kế tiếp đại diện cho tên cơ quan
- Nhóm đầu tiên đại diện cho tên máy tính

CHƯƠNG 1 : INTERNET VÀ WEB

III. CÁC KHÁI NIỆM

3. Tên miền (Domain Name)

Ví dụ:

<i>Địa chỉ IP</i>	<i>Địa chỉ theo tên miền</i>
113.115.12.37	www.pacific.net.sg
203.162.16.235	home.vnn.vn
203.162.7.48	fpt.com.vn
110.32.26.74	www.yahoo.com

CHƯƠNG 1 : INTERNET VÀ WEB

III. CÁC KHÁI NIỆM

4. Máy chủ (Server)

- ☞ Là một máy tính được nối vào mạng và có khả năng cung cấp thông tin cho máy tính khác trong mạng.
- ☞ Là máy trong mạng mà các máy tính khác trong mạng có thể truy cập được.

5. Máy khách (Client)

- ☞ Là một máy tính được nối vào mạng và có khả năng truy cập vào các máy khác để yêu cầu đáp ứng thông tin. (Máy của những người sử dụng bình thường).

CHƯƠNG 1 : INTERNET VÀ WEB

IV. CÁC DỊCH VỤ CỦA INTERNET

1. Dịch vụ thư điện tử (E-mail) :

- Electronic mail (E-mail) : Là một dịch vụ của Internet giúp cho việc trao đổi thông điệp giữa những người dùng hay nhóm người dùng trên mạng
- Dựa trên giao thức chuẩn Internet: Simple Mail Transfer Protocol (SMTP)
- Khả năng gửi tới nhiều người cùng một thời điểm
- Nhanh chóng chuyển giao được tài liệu
- Chi phí thấp

CHƯƠNG 1 : INTERNET VÀ WEB

IV. CÁC DỊCH VỤ CỦA INTERNET

1. Dịch vụ thư điện tử (E-mail) :

Các mô hình hoạt động E-Mail :

- Mô hình thông điệp trực tiếp
- Mô Hình hộp thư lưu
- Mô hình Internet

CHƯƠNG 1 : INTERNET VÀ WEB

IV. CÁC DỊCH VỤ CỦA INTERNET

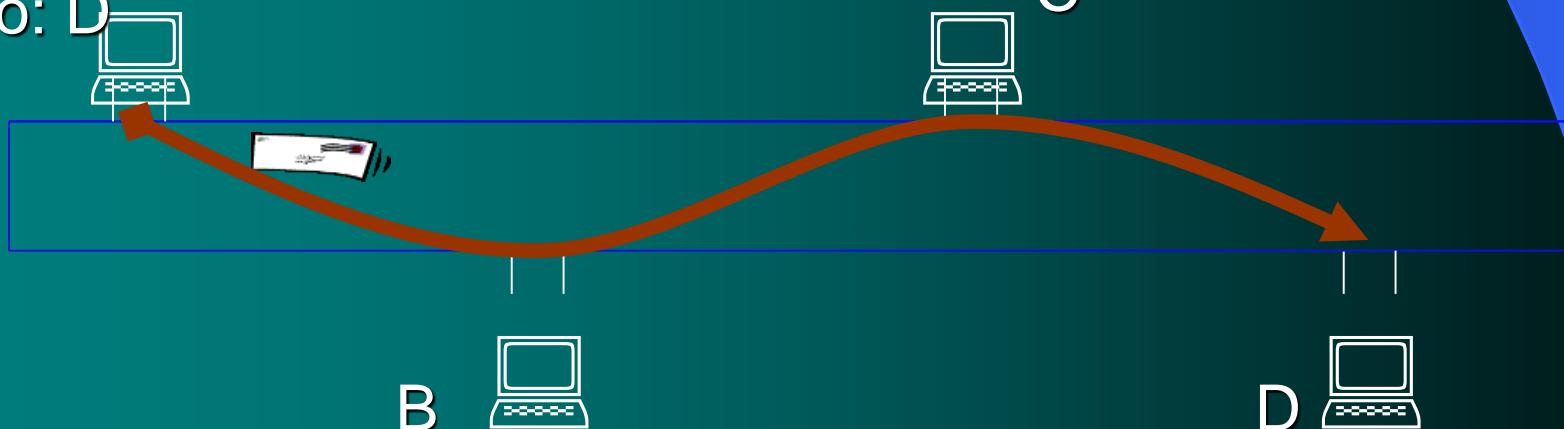
1. Dịch vụ thư điện tử (E-mail) :

Mô hình thông điệp trực tiếp

Các thông điệp được gửi trực tiếp ngay lập tức tới máy đang hoạt động trong mạng nội bộ

from: A

to: D



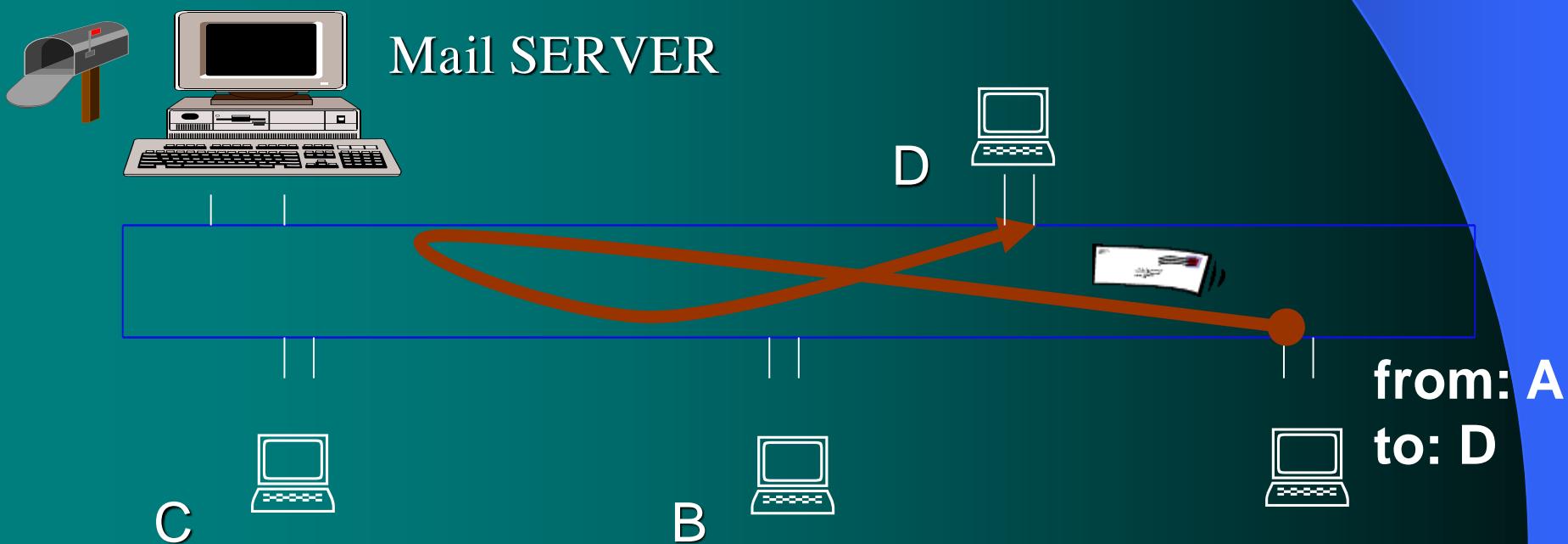
CHƯƠNG 1 : INTERNET VÀ WEB

IV. CÁC DỊCH VỤ CỦA INTERNET

1. Dịch vụ thư điện tử (E-mail):

Mô hình hộp thư lưu

Thông điệp được gửi gián tiếp tới một máy phục vụ đang hoạt động trong mạng nội bộ



CHƯƠNG 1 : INTERNET VÀ WEB

Các thành phần cơ bản

- Mail server: Chương trình phục vụ thư
- Mail Client: Chương trình cho người sử dụng
- Cách thức giao nhận thư

CHƯƠNG 1 : INTERNET VÀ WEB

IV. CÁC DỊCH VỤ CỦA INTERNET

2. World Wide Web (www)

- World Wide Web (WWW) hoặc Web là một dịch vụ của Internet
- Là dịch vụ cung cấp thông tin một cách toàn diện nhất cho người dùng.
- Web chứa thông tin bao gồm văn bản, hình ảnh, âm thanh và thậm chí cả video được kết hợp với nhau.
- Web là kho thông tin khổng lồ: phong phú về nội dung, đa dạng về hình thức, thường xuyên được cập nhật, đổi mới và phát triển không ngừng

CHƯƠNG 1 : INTERNET VÀ WEB

IV. CÁC DỊCH VỤ CỦA INTERNET

3. Chat

- Là hình thức hội thoại trực tiếp trên Internet, là một dịch vụ của Internet
- Dịch vụ này cho phép hai hay nhiều người có thể cùng trao đổi thông tin trực tuyến với nhau thông qua bàn phím và màn hình.

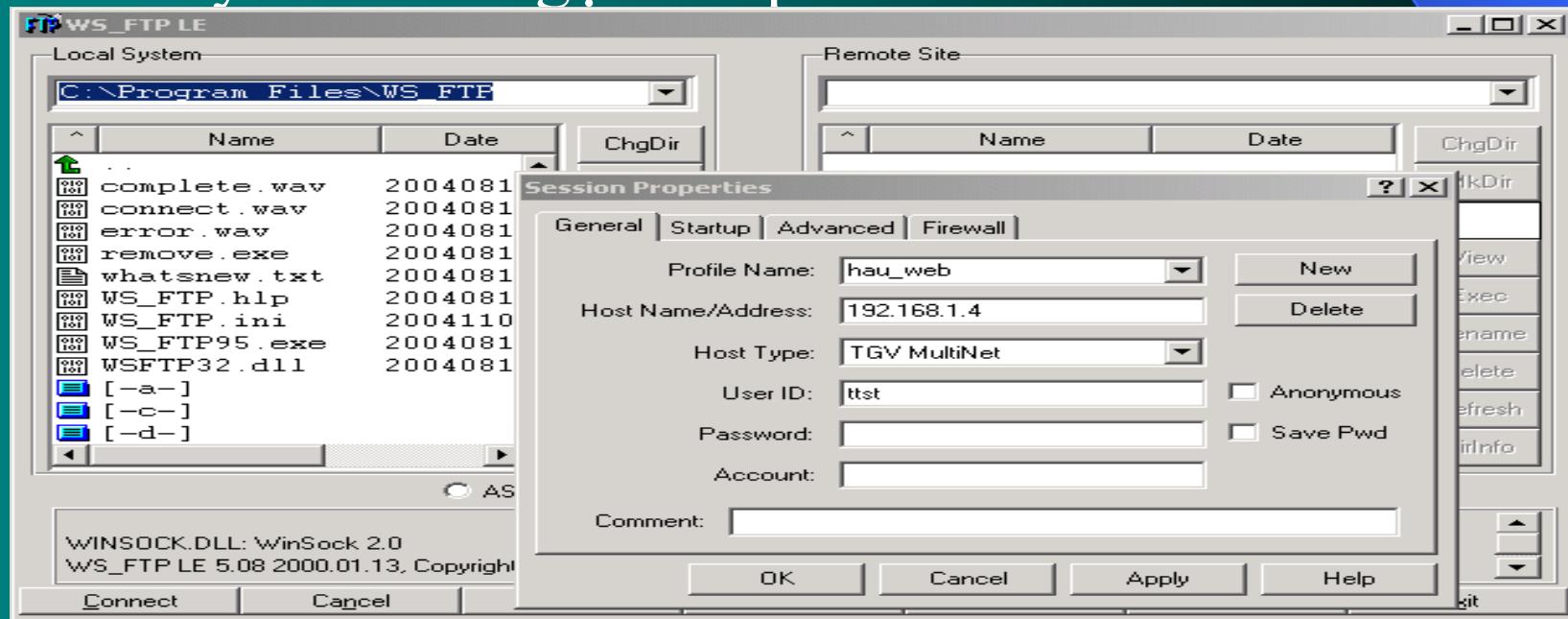
CHƯƠNG 1 : INTERNET VÀ WEB

IV. CÁC DỊCH VỤ CỦA INTERNET

4. FTP (File Transfer Protocol)

Dịch vụ này cho phép truyền các tập tin từ máy tính này đến máy tính khác trong mạng theo giao thức FTP.

Lấy file từ máy tính ở xa về người ta gọi là Download, gửi file đến máy tính ở xa gọi là Upload



CHƯƠNG 1 : INTERNET VÀ WEB

IV. CÁC DỊCH VỤ CỦA INTERNET

5. Dịch vụ Telnet

- * Cho phép người sử dụng đăng nhập vào máy tính chủ từ xa thông qua một máy khác trên mạng
- * Hệ thống ở xa sẽ yêu cầu xác nhận tên đăng ký và mật khẩu để đăng nhập
- * Sau khi kết nối thì những gì ta nhập từ bàn phím của người sử dụng đều được gửi đến máy tính từ xa
- * *Lệnh: Telnet <tên host> hoặc <địa chỉ IP>*
- * *Trong đó: <tên host> là tên máy chủ muốn truy cập. <địa chỉ IP> là địa chỉ của máy chủ muốn truy cập.*

CHƯƠNG 1 : INTERNET VÀ WEB

IV. CÁC DỊCH VỤ CỦA INTERNET

6. Các dịch vụ khác :

- * **Gopher** – Tra cứu thông tin theo thực đơn
- * **Dịch vụ WAIS** (Wide Area Information Server)

Tìm kiếm thông tin theo diện rộng

- * **Mailing list**: Danh sách trao đổi trên Email

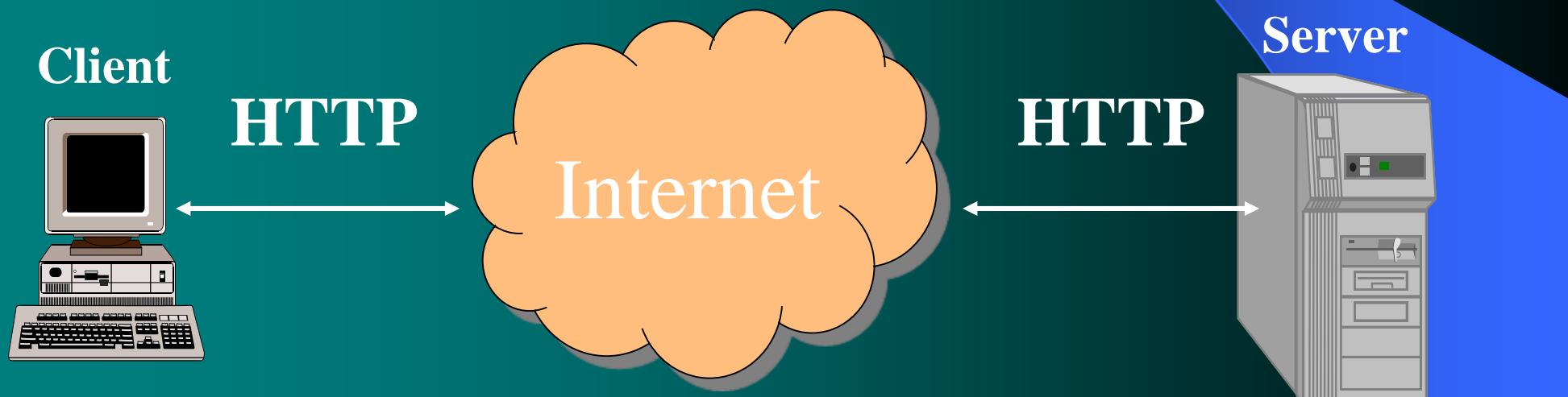
Các dạng mailing list :

- *Mailing list không có nhà quản trị*
- *Mailing list có nhà quản trị (thông dụng)*
- *Mailing list một chiều*

CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WORLD WIDE WEB (WWW)

1. Mô hình chung :



Client gửi yêu cầu thông tin lên Internet

Server nhận yêu cầu, xử lý và gửi kết quả về cho Client

CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

2. Các khái niệm :

a. Địa chỉ URL :

URL (Uniform Resource Locator). Là địa chỉ đầy đủ chỉ đến một tài nguyên thông tin trên Internet.

Ví dụ:

<http://www.hssv.vnn.vn:80/dethidapan/index.html>

Trong đó:

http:// Gọi là tên dịch vụ (có thể gọi là phần giao thức).

www.hssv.vnn.vn/ Là phần tên máy chủ.

dethidapan/index.html Đường dẫn đến tài nguyên.

CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

2. Các khái niệm :

a. Địa chỉ URL :

Lợi ích của URL là cung cấp một phương pháp đơn giản, đồng bộ cho việc định danh các tài nguyên có sẵn trên Internet, thông qua rất nhiều giao thức khác nhau như là HTTP, FTP, Telnet,...

Bảng dưới đây mô tả một số khuôn dạng URL chuẩn đang được dùng rất phổ biến:

CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

2. Các khái niệm :

a. Địa chỉ URL :

Dịch vụ	Khuôn dạng URL	Ngầm định
HTTP	<u>Http://Host:Port/Path</u>	Port: 80
FTP	<u>Ftp://User.Password@Host:Port/Path</u>	User: Anonymous Password: địa chỉ e-mail Port: 21
GOPHER	<u>Gopher://Host:Port/Path</u>	Port: 70
MAILTO	<u>Mailto:Address@Host</u>	Không
TELNET	<u>Telnet://User:Password@Host:Port</u>	Port: 23

CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

2. Các khái niệm :

b. Giao thức HTTP :

Web Browser và Web Server giao tiếp với nhau thông qua một giao thức được gọi là HTTP. Sự kết nối bằng Http qua 4 giai đoạn:

- + **Tạo kết nối** : Web Browser giao tiếp với Web Server nhờ địa chỉ Internet và số cổng (ngầm định là 80) được đặc tả trong URL
- + **Thực hiện yêu cầu** : Web Browser gửi thông tin tới Web Server để yêu cầu phục vụ. Việc gửi và nhận thông tin ở đây theo phương thức POST, GET
- + **Phản hồi** : Web Server gửi một phản hồi về Web Browser nhằm đáp ứng yêu cầu của Web Browser.
- + **Kết thúc kết nối** : Khi kết thúc quá trình trao đổi giữa Web Browser và Web Server thì sự kết nối chấm dứt.

CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

2. Các khái niệm :

c. Web Server

- ☞ Là một phần mềm được sử dụng trên máy chủ (máy phục vụ) để lắng nghe các yêu cầu từ các máy khách **Client (Web Client)** và đáp ứng yêu cầu cho **máy Client**.
- ☞ Có rất nhiều Web Server chạy trên nhiều platform : Web Server của NSCA trên UNIX, Web Server dùng trên môi trường Windows, Web Server của Oracle được thiết kế để tận dụng khả năng truy cập vào cơ sở dữ liệu hùng mạnh của Oracle.

CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

2. Các khái niệm :

d. Web Browser

- ☞ Là một phần mềm được sử dụng trên máy khách (Client), dùng để gửi các yêu cầu lên **Web Server** và nhận các kết quả đáp ứng từ Server đồng thời hiển thị lên với khuôn dạng thích hợp.
- ☞ Các trình duyệt : Nestcape Navigator, Mosaic NCSA, Internet Explore

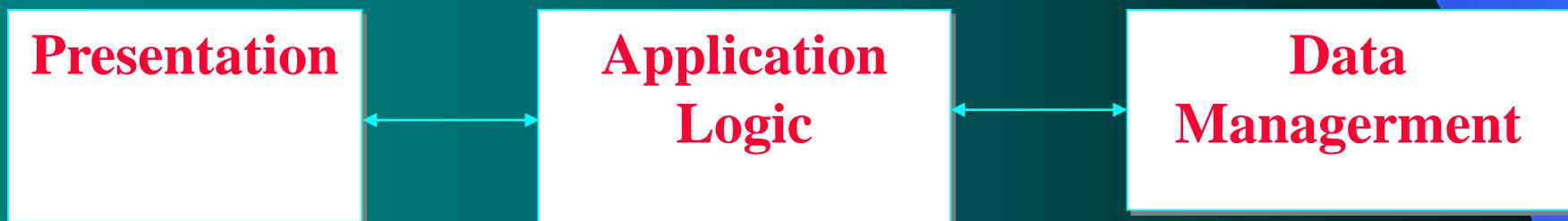
CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

3. Mô hình xử lý Client/Server :

Như chúng ta đã biết, mọi vấn đề được thực hiện trên máy tính đều phải thông qua việc viết chương trình.

Một chương trình có thể chia làm 3 lớp như sau:



CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

3. Mô hình xử lý Client/Server :

+ Lớp Presentation (lớp trình bày): Lớp này quản lý giao diện người sử dụng, cho phép người sử dụng xem các thông tin, báo biểu của chương trình, hoặc cho phép người sử dụng nhập các thông tin đầu vào.

+ Lớp Application Logic (lớp logic ứng dụng): Lớp này được xem như là trái tim của ứng dụng. Nó bao gồm các module cần thiết để hoàn thiện các chức năng của ứng dụng.

+ Lớp Data Management (lớp quản lý dữ liệu): lớp này bao gồm tất cả các dữ liệu được sử dụng bởi ứng dụng, cũng như các quy tắc hợp lệ khác nhau được yêu cầu để đảm bảo tính toàn vẹn dữ liệu.

CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

3. Mô hình xử lý Client/Server :

Một mô hình xử lý Client/Server phải có 3 thành phần cơ bản :

Front-end Client

Back-end Server

Mạng máy tính (Network)

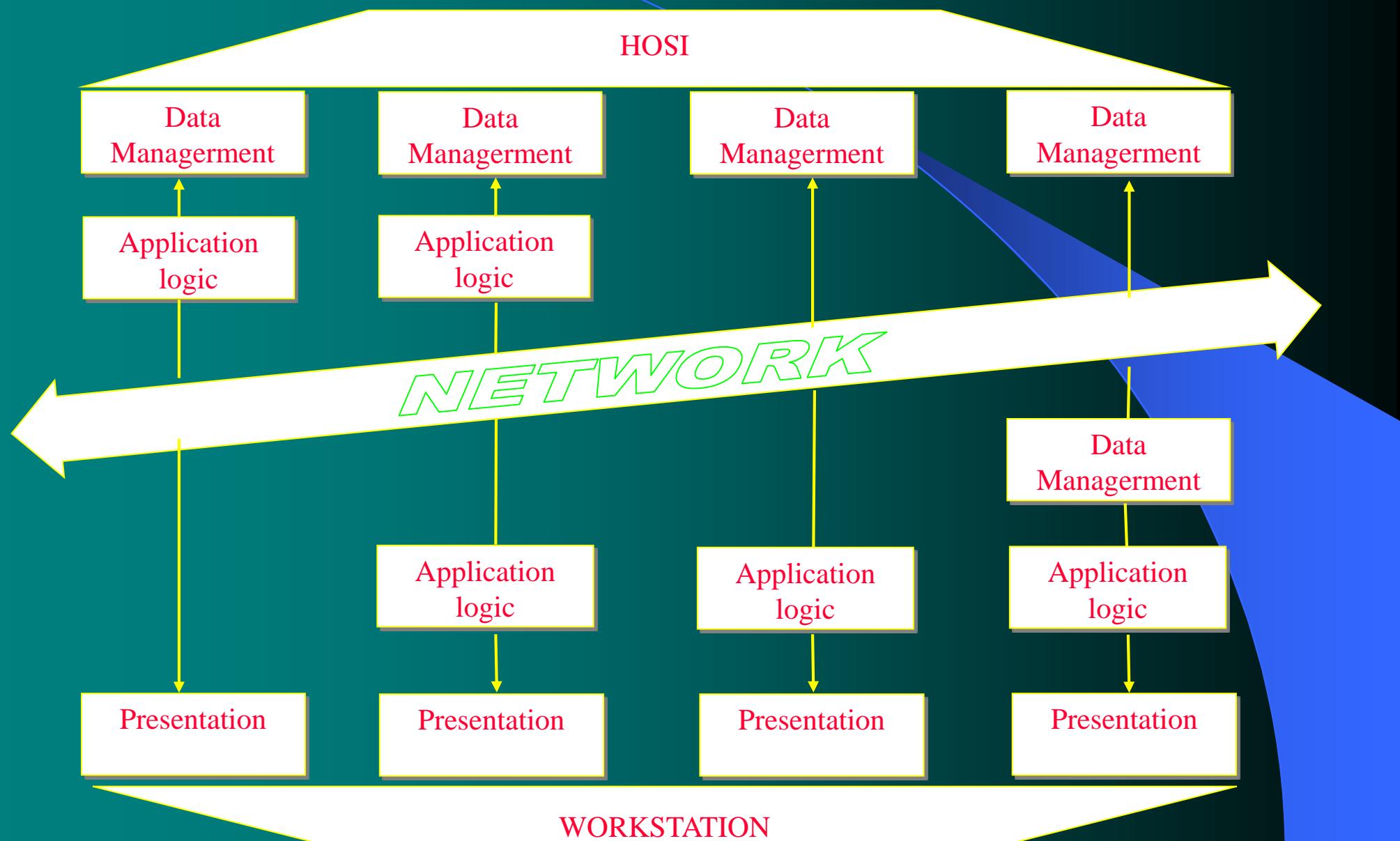
Một chương trình **Front-end Client** chạy trên trạm làm việc (Workstation), mà ở đó người sử dụng giao tiếp với ứng dụng để yêu cầu cung cấp dịch vụ, như truy vấn dữ liệu.

Chương trình **Back-end Server** chạy trên máy chủ (Host) tiếp nhận thông tin và cung cấp dịch vụ được yêu cầu, như phản hồi truy vấn.

Mạng máy tính có chức năng truyền tải thông tin.

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

3. Mô hình xử lý Client/Server : 4 mô hình xử lý phổ biến



CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

3. Mô hình xử lý Client/Server :

+ **Mô hình đầu tiên là mô hình trình bày từ xa :**

mô hình này máy chủ thực hiện truy cập dữ liệu, xử lý tính toán. Kết quả tính toán được trả về và trình bày trên máy trạm.

+ **Mô hình thứ hai là mô hình xử lý phân tán :**

mô hình này chia xẻ năng lượng tính toán trên máy chủ cho máy trạm. Máy trạm ngoài chức năng quản lý giao diện còn được phân bổ một số chức năng xử lý thích hợp.

CHƯƠNG 1 : INTERNET VÀ WEB

V. KIẾN TRÚC CƠ BẢN CỦA WEB (WWW)

3. Mô hình xử lý Client/Server :

+ **Mô hình thứ 3 là mô hình quản lý dữ liệu từ xa** : mô hình này tận dụng khả năng tính toán của máy trạm. Máy chủ chỉ có chức năng quản lý dữ liệu, còn máy trạm có nhiệm vụ xử lý tính toán và quản lý giao diện người sử dụng.

+ **Mô hình cuối cùng là mô hình quản lý dữ liệu phân tán**: đặc điểm của mô hình này là toàn bộ năng lượng tính toán tập trung trên máy trạm. Tuy nhiên một phần dữ liệu được quản lý bởi máy chủ.

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

I. GIỚI THIỆU

1. Giới thiệu về HTML

☞ Khi chúng ta thiết kế một trang Web dễ dàng dưới dạng các hình ảnh trực quan, nhưng cốt lõi bên trong, cấu trúc của một trang Web được qui định bởi ngôn ngữ siêu liên kết HTML.

☞ HTML là chữ viết tắt của Hypertext Markup Language (ngôn ngữ đánh dấu siêu văn bản) là hàng loạt các thẻ (Tag) với *các qui ước* để tạo ra các trang Web và các trang Web được hiển thị bởi những trình duyệt Web (Web browser).

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

I. GIỚI THIỆU

1. Giới thiệu về HTML

- ☞ HTML là một ngôn ngữ mô tả tài liệu, được hình thành từ một dạng ngôn ngữ mô tả tài liệu tổng quát có tên là SGML (Standard Generalized Markup Language), đây là ngôn ngữ do hãng IBM đề xướng từ những năm 1960.
- ☞ Thực chất SGML là một ngôn ngữ máy tính nhằm mô tả tài liệu, cho phép ta thiết lập mối quan hệ giữa các thành phần khác nhau trong cùng một tài liệu, hoặc giữa nhiều tài liệu thuộc các dạng dữ liệu khác nhau

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

I. GIỚI THIỆU

2. Các khái niệm

a. Khái niệm TAG

Một Tag của HTML là dấu hiệu báo cho trình duyệt Web biết rằng nên làm điều gì đó ngoài việc chỉ thuận tuý thể hiện văn bản lên màn hình. Các Tag trong HTML thường được gọi là các thẻ.

Ví dụ: Để hiển thị một dòng văn bản in đậm, trong tài liệu HTML ta chèn vào lệnh sau:

** dòng văn bản được in đậm **

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

I. GIỚI THIỆU

2. Các khái niệm

b. Thuộc tính của Tag

Mỗi Tag trong HTML đều có một vài thuộc tính (hay còn gọi là tham số) dùng để bổ sung hoặc thay đổi cách thể hiện thông thường của Tag đó.

Ví dụ: Để hiển thị hình ảnh trên trang web ta dùng tag như sau:

```
<IMG SRC="bground.gif" WIDTH="100" HEIGHT="150">
```

Trong đó:

- + Chỉ thị IMG báo cho trình duyệt biết bắt đầu hiển thị hình ảnh
- + Thuộc tính SRC cho biết hình ảnh cần hiển thị
- + Thuộc tính WIDTH có giá trị 100, HEIGHT có giá trị 150 báo cho trình duyệt biết độ rộng và độ cao của hình ảnh cần hiển thị là 100 và 150

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

I. GIỚI THIỆU

2. Các khái niệm

c. Tag chứa và Tag rỗng

- + Tag chứa: Một tag được gọi là tag chứa nếu nó đi cùng với một tag khác dùng để báo hiệu tag đó mất hiệu lực, hoặc bao gồm trong nó nhiều tag khác để tăng thêm tính năng của nó. Tag **** là một ví dụ về tag chứa, chúng gồm hai thành phần, phần đầu tag (****) và phần cuối tag (****).
- + Tag rỗng: tag rỗng là tag chỉ có một thành phần chứa tên tag và không đi kèm với một tag khác. Thông thường loại tag này chỉ có hiệu lực tại vị trí nó xuất hiện. Ví dụ nh tag **
** có nhiệm vụ ngắt dòng, khi trình duyệt nhận thấy tag này trong tài liệu thì nó tiếp tục hiển thị thông tin sang một dòng mới.

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

I. GIỚI THIỆU

3. Cấu trúc một trang Web ở dạng HTML

Một tài liệu HTML thông thường gồm 2 phần chính: phần đầu tài liệu (Head) và phần thân tài liệu (Body)

Dưới đây là cấu trúc cơ bản của một tài liệu HTML:

<HTML>	<-- tag định nghĩa 1 tài liệu HTML
<HEAD>	<-- tag định nghĩa phần đầu của tài liệu
-----	<-- các tag khác thuộc về phần đầu tài liệu
</HEAD>	<-- tag báo hiệu kết thúc phần đầu tài liệu
<BODY>	<-- tag báo hiệu bắt đầu phần thân tài liệu
-----	<-- nội dung và các tag phần thân tài liệu
</BODY>	<-- tag báo hiệu kết thúc phần thân tài liệu
</HTML>	<-- tag kết thúc 1 tài liệu HTML

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

1. Các tag trong phần đầu tài liệu

a. Tag <!DOCTYPE>

Tag này dùng ghi thông tin về version HTML áp dụng trong tài liệu Web. Thường đây là dòng đầu tiên trong file HTML.

- Ví dụ:
- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Draft//EN">
- hay <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML Level 1//EN">

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

1. Các tag trong phần đầu tài liệu

b. Tag <TITLE>

Đây là loại tag chứa, dùng để thiết lập tiêu đề của tài liệu

Ví dụ:

```
<Title>Ví dụ cho tiêu đề tạo web</Title>
```

c. Tag <BASE>

Thành phần này dùng để chỉ định địa chỉ cơ bản cho các mối nối liên kết. Coi như đây là địa chỉ gốc để tìm các địa chỉ khác.

Ví dụ:

```
<BASE HREF="http://www.pc3.com/Scripts/">
```

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

1. Các tag trong phần đầu tài liệu
 - d. Tag <META>

Tag này cho phép ta tự do đặt nhiều thông tin vào phần đầu của tài liệu web như : địa chỉ trả lời thư, các bản thông báo, thời gian làm tươi trang web, thời gian quá hạn thông tin...

Các thuộc tính của tag:

- + Content: gán giá trị cho tên thuộc tính
- + HTTP-EQUIV: kết nối các siêu thông tin vào phần header của đáp ứng Server (HTTP)
- + Name: tên của thành phần siêu thông tin, được xem giống như là HTTP-EQUIV nếu không được xác định.

Ví dụ: <META HTTP-EQUIV="Refresh" Content="10">

Tag này báo cho trình duyệt biết rằng cứ 10 giây thì làm tươi (Refresh) trang web một lần.

CHƯƠNG 2 : NGÔN NGỮ HTML (HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

1. Các tag trong phần đầu tài liệu

e. Tag <SCRIPT>

Đây là tag chứa, dùng để chứa các hàm (hoặc các thủ tục) được viết bằng JavaScript (hoặc VBScript). Tag này có thuộc tính Language để chỉ ra ngôn ngữ được sử dụng là ngôn ngữ gì.

Ví dụ:

```
<SCRIPT LANGUAGE="JavaScript">
    function Vidu(){
        //các lệnh của hàm nằm ở đây.
    }
</SCRIPT>
```

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

a. Tag BODY

Đây là loại tag chứa, dùng để bao lấy các tag khác trong phần thân của tài liệu HTML.

Cách sử dụng:

```
<BODY>  
<!--các tag khác nằm ở đây-->  
</BODY>
```

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

Thuộc tính	Chức năng
Bgcolor="#RRBBGG"	Xác định màu nền (Background) cho trang web , tên màu nền phải bắt đầu bằng ký tự #, tiếp sau đó là 6 chữ cái, mỗi chữ cái là 1 số Hecxa (từ 0 tới F) dùng để xác định tổ hợp màu cho nó.
BACKGROUND="Image"	Xác định ảnh nền cho trang web
COLOR="#RRBBGG"	Xác định màu của văn bản trong phần thân tài liệu , tên màu phải bắt đầu bằng ký tự #, tiếp sau đó là 6 chữ cái, mỗi chữ cái là 1 số Hecxa (từ 0 tới F) dùng để xác định tổ hợp màu cho nó.
Topmargin="n"	Canh lề trên cho trang web (giá trị n được tính bằng Pixel)

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

b. Tag HEADING

Đây là loại tag chứa dùng để làm nổi bật một tiêu đề hay một mục nào đó trong trang web.

Có 6 Heading tương ứng với 6 mức độ khác nhau, với độ ưu tiên giảm dần từ 1 đến 6.

Trong HTML ta sử dụng tag <Hn> để thể hiện Heading, trong đó n là một số đi từ 1-> 6.

CHƯƠNG 2 : NGÔN NGỮ HTML (HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

b. Tag HEADING

Cách sử dụng	Thể hiện trên Browser
<code><H1>Tiêu đề 1</H1></code>	Tiêu đề 1
<code><H2>Tiêu đề 2</H2></code>	Tiêu đề 2
<code><H3>Tiêu đề 3</H3></code>	Tiêu đề 3
<code><H4>Tiêu đề 4</H4></code>	Tiêu đề 4
<code><H5>Tiêu đề 5</H5></code>	Tiêu đề 5
<code><H6>Tiêu đề 6</H6></code>	Tiêu đề 6

CHƯƠNG 2 : NGÔN NGỮ HTML (HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

c. Tag Table

+ Để mô tả bảng ta dùng Tag `<TABLE>` và `</TABLE>`. Mỗi hàng trong bảng được xác định bằng cặp tag `<TR>` và `</TR>`, mỗi ô của bảng được xác định bằng cặp tag `<TD>` và `</TD>`.

+ Muốn xác định tiêu đề của bảng ta sử dụng tag `<CAPTION>` và `</CAPTION>`

+ Ngoài ra có thể sử dụng các thuộc tính `HEIGHT`, `WIDTH`, `BORDER` để thay đổi độ cao, độ rộng, và đường viền của bảng

CHƯƠNG 2 : NGÔN NGỮ HTML (HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

c. Tag Table

Ví dụ:

```
<TABLE BORDER=1>
  <CAPTION>Ví dụ về cách tạo bảng </CAPTION>
  <TR>
    <TH>STT</TH>
    <TH>Nội dung</TH>
  </TR>
  <TR>
    <TD>1</TD>
    <TD>Nội dung ô thứ 2 hàng 2</TD>
  </TR>
  <TR>
    <TD>2</TD>
    <TD>Nội dung ô thứ 2 hàng 3</TD>
  </TR>
  <TR>
    <TD>3</TD>
    <TD>Nội dung ô thứ 2 hàng 4</TD>
  </TR>
</TABLE>
```

Ví dụ về cách tạo bảng

STT	Nội dung
1	Nội dung ô thứ 2 hàng 2
2	Nội dung ô thứ 2 hàng 3
3	Nội dung ô thứ 2 hàng 4

CHƯƠNG 2 : NGÔN NGỮ HTML (HyperText Markup Language)

III. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

d. Tag Form

+ Biểu mẫu cho phép ta tập hợp các kiểu dữ liệu từ trang web và gửi lên Server xử lý thông qua các trường nhập liệu như Text box, các nút điều khiển như Check box, Radio button, và Command button,...

Tất cả các trường này đều phải nằm giữa cặp tag **<FORM>** và **</FORM>**

Một Form dữ liệu có các thuộc tính sau:

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

III. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

d. Tag Form

Thuộc tính	Chức năng
NAME="form_name"	Tên của Form
ACTION="URL_execute"	Chỉ định tên của một file trên Server, mà cho phép xử lý dữ liệu từ Form này gửi lên.
METHOD="POST/GET"	Xác định phương thức mà Browser sẽ gửi dữ liệu lên Server. Giá trị POST dùng để gửi dữ liệu, GET dùng để lấy dữ liệu từ Server.

CHƯƠNG 2 : NGÔN NGỮ HTML (HyperText Markup Language)

III. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

d. Tag Form

Trong 1 Form có thể bao gồm nhiều trường dữ liệu, mỗi trường dữ liệu đều được mô tả bằng một từ khoá INPUT và có 3 thuộc tính : **TYPE**, **NAME**, và **VALUE**.

+ Type : dùng để xác định kiểu trường dữ liệu (text box, check box,...),

+ Name : dùng để xác định tên của trường

+ Value : dùng để xác định giá trị của trường đó.

Ví dụ: <INPUT TYPE="Text" Name="T1" Value="V1">

CHƯƠNG 2 : NGÔN NGỮ HTML (HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

d. Tag Form

Text Box	<INPUT TYPE="Text" >	Cho phép nhập dữ liệu chỉ trên một hàng
Password box	<INPUT TYPE="Password" >	Dùng để nhập mật khẩu, khi sử dụng trường này dữ liệu nhập vào sẽ được hiển thị bằng các dấu *
Hidden box	<INPUT TYPE="Hidden" >	Dùng để xác định một số giá trị bổ sung cho việc xử lý của chương trình mà không cần hiển thị chúng.
Check box	<INPUT TYPE="Checkbox" >	Lựa chọn nhiều giá trị một lúc

CHƯƠNG 2 : NGÔN NGỮ HTML (HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

d. Tag Form

Radio button	<INPUT TYPE="Radio" >	Lựa chọn một trong nhiều giá trị
Text window	<TEXTAREA>...</TEXTAREA>	Cho phép nhập dữ liệu trên nhiều hàng
File	<INPUT TYPE="File" >	Mở cửa sổ liệt kê các file và th mục trên máy tính để chọn file
Select box	<SELECT Name="select_name"> <OPTION VALUE="nvalue"> Đề mục chọn lựa </OPTION> </SELECT>	Cho phép chọn 1 đề mục thích hợp trong danh sách đổ xuống của các đề mục
Button	<INPUT TYPE="Submit/Reset/Button">	Gửi số liệu lên Server (nếu Type="Submit"), hoặc trả về giá trị ngầm định của các trường trên form (nếu Type="Reset")

CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

d. Tag Form

VÍ DỤ VỀ TẠO FORM

Text box:

Text Area:

File box:

Browse...

Radio button

Check box

Submit

Reset

CHƯƠNG 2 : NGÔN NGỮ HTML (HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

e. Tag Frame

Mục đích của Frame là nhằm chia cửa sổ trình duyệt ra làm hai hay nhiều vùng nhỏ hơn, để khi đó có thể tải các tài liệu HTML khác nhau vào trong mỗi vùng giúp người sử dụng có thể theo dõi nhiều trang web đồng thời.

Mỗi vùng nhỏ gọi là một Frame, tập hợp các vùng này gọi là Frameset.

Để tạo các Frame sử dụng cặp tag `<FRAMESET>` và `</FRAMESET>`, cặp tag này có hai thuộc tính cơ bản là ROWS và COLS dùng để mô tả số hàng và số cột cần chia thành Frame.

CHƯƠNG 2 : NGÔN NGỮ HTML (HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

2. Các Tag trong phần thân tài liệu

e. Tag Frame

Ví dụ: <FRAMESET COLS="150,*">

```
<FRAME NAME="menu" SRC="Menu.htm">
<FRAME NAME="main" SRC="Main.htm">
</FRAMESET>
```

+ Đoạn mã trên chia cửa sổ ra làm hai cột, cột thứ nhất rộng 150 pixel, cột thứ hai là khoảng trống còn lại của cửa sổ.

+ Trong cặp tag <FRAMESET> còn chứa các tag <FRAME> dùng để xác định các trang web sẽ được hiển thị trong mỗi Frame khi Frameset được nạp lên lần đầu tiên.

+ Mỗi Frame có thuộc tính NAME dùng đặt tên cho Frame đó, đồng thời làm đích để các trang web khác nhảy đến frame đó

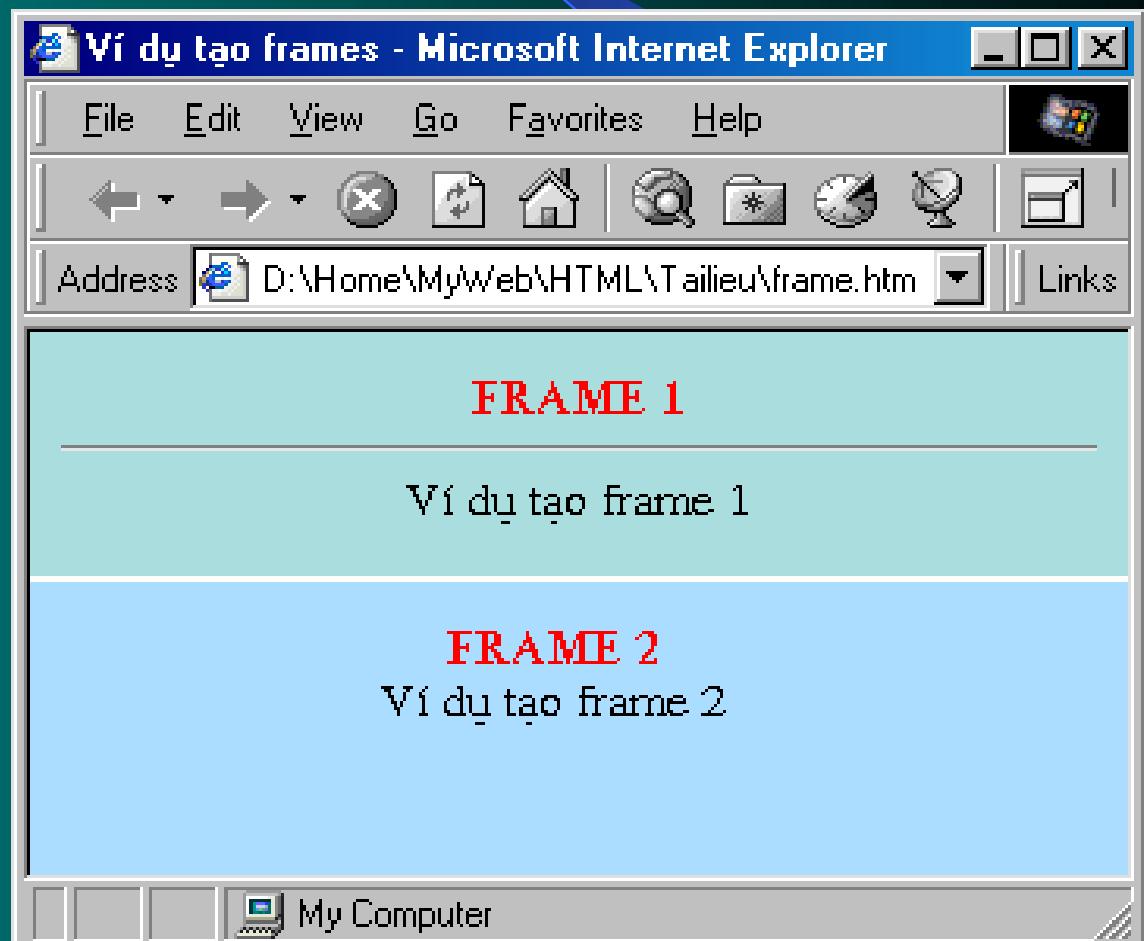
CHƯƠNG 2 : NGÔN NGỮ HTML

(HyperText Markup Language)

II. CÁC TAG CƠ BẢN CỦA HTML

e. Tag Frame

Kết quả hiển thị như sau:



Chương 3: JavaScript

I. GIỚI THIỆU

- + Script có thể xem như một công cụ nhằm hỗ trợ người lập trình tạo ra các ứng dụng nhanh chóng, đơn giản không mang tính chuyên nghiệp như một ngôn ngữ lập trình thật sự.
- + JavaScript cũng là một loại Script cho phép phát triển các ứng dụng trên Internet ở cả phía Client và Server bằng cách viết các chương trình với cú pháp hoàn toàn giống Java.

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

- + JavaScript là một ngôn ngữ script, dùng để phát triển các ứng dụng Internet ở Client và Server.
- + JavaScript có thể viết kèm trong file HTML để xây dựng một trang Web chật chẽ, phong phú về nội dung và hình thức như : kiểm tra tính tương thích và thông báo lỗi trong quá trình nhập liệu...
- + Cách viết JavaScript trong một trang HTML như sau:

```
<Script Language="JavaScript">  
    //Các hàm của JavaScript đặt ở đây  
</Script>
```

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

1. Một số định nghĩa

1.1. Các kiểu dữ liệu :

Trong JavaScript có các kiểu dữ liệu :

- **Integer:** Gồm các số ở dạng cơ số 10, cơ số 16, cơ số 8, cơ số 2
 - Ví dụ: 42, 0xFFFF, -345...
- **Số thực :** Ví dụ: 3.1415, -3.1E12, 2e-12 ... lưu ý E, e đều có ý nghĩa như nhau.
- **Boolean:** Có giá true hoặc false
- **Chuỗi (String):** Một chuỗi là một dãy các ký tự nằm giữa hai cặp dấu ("") hoặc (""). Ví dụ: "blah",'blah','1234', "dong mot\n dong hai".

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

1. Một số định nghĩa

1.2. Biến

+ Biến là một đại lượng dùng để lưu trữ dữ liệu mà giá trị của nó có thể thay đổi trong quá trình thực hiện chương trình.

+ Để biểu diễn biến người ta sử dụng tên biến, khi khai báo một biến thì tên biến được cấp phát một địa chỉ cụ thể trong bộ nhớ máy tính.

+ Tên biến là chuỗi gồm các ký tự, chữ số, và dấu gạch dưới (_)

+ Tên biến trong ngôn ngữ JavaScript có phân biệt chữ hoa và chữ thường.

+ Để định nghĩa một biến bằng cách dùng từ khoá **var**

Ví dụ: `var x=12`

Trong JavaScript có thể dùng biến mà không cần khai báo.

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

1. Một số định nghĩa

1.3. Các phép toán

- + Các phép toán **số học**
- + Các phép toán so sánh
- + Các phép toán thao tác trên bit
- + Các phép toán logic

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

1. Một số định nghĩa

1.3. Các phép toán

+ Các phép toán số học

Phép toán viết tắt	Ý nghĩa
$x += y$	$x = x + y$
$x -= y$	$x = x - y$
$x *= y$	$x = x * y$
$x /= y$	$x = x / y$
$x \% = y$	$x = x \% y$
$x <<= y$	$x = x << y$
$x >>= y$	$x = x >> y$
$x >>>= y$	$x = x >>> y$
$x &= y$	$x = x \& y$
$x ^= y$	$x = x ^ y$
$x = y$	$x = x y$

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

1. Một số định nghĩa

1.3. Các phép toán

+ Các phép toán so sánh

Phép toán	Kiểu trả về	ý nghĩa
==	Boolean	Có giá trị True nếu x bằng y, ngược lại có giá trị False
!=	Boolean	Có giá trị True nếu x khác y và ngược lại cho giá trị False
>	Boolean	Có giá trị True nếu x lớn hơn y, ngược lại cho giá trị False
<	Boolean	Có giá trị True nếu x nhỏ hơn y, ngược lại cho giá trị False
>=	Boolean	Có giá trị True nếu x lớn hơn hoặc bằng y, ngược lại cho giá trị False
<=	Boolean	Có giá trị True nếu x nhỏ hơn hoặc bằng y, ngược lại cho giá trị False

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

1. Một số định nghĩa

1.3. Các phép toán

+ Các phép toán thao tác trên bit

Phép toán	ý nghĩa
$a \& b$	a and b
$a b$	a or b
$a ^ b$	a xor b
$\sim a$	Phủ định tất cả các bit trong a (0 thành 1 và 1 thành 0).
$a << b$	dịch a sang trái b bit.
$a >> b$	dịch a sang phải b bit.

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

1. Một số định nghĩa

1.3. Các phép toán

+ Các phép toán logic

Cho `expr`, `expr1`, `expr2` là các biểu thức logic.

<code>Expr1 && Expr2</code>	Trả về giá trị <code>true</code> nếu cả hai biểu thức <code>expr1</code> và <code>expr2</code> đều có giá trị là <code>true</code> , ngược lại có giá trị <code>false</code> .
<code>Expr1 Expr2</code>	Trả về giá trị <code>true</code> nếu một trong hai biểu thức <code>expr1</code> hoặc <code>expr2</code> có giá trị là <code>true</code> , ngược lại có giá trị <code>false</code> .
<code>! Expr</code>	Trả về <code>true</code> nếu <code>expr</code> là <code>false</code> , ngược lại có giá trị <code>false</code> (phép phủ định)

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

1. Một số định nghĩa

1.3. Các phép toán:

Ưu tiên các phép toán
trong JavaScript:

Thứ tự	Các phép toán
1	=, +=, -=, *=, /=, %=, <<=, >>=, >>>=, &=, ^=, =
2	?:
3	
4	&&
5	
6	^
7	&
8	==, !=
9	<, <=, >, >=
10	<<, >>, >>>
11	+, -
12	*, /, %
13	!, -, ++, --
14	(), []

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

2. Cấu trúc các lệnh điều khiển trong JavaScript

- + Lệnh if...else
- + Lệnh switch .. case
- + Lệnh while
- + Lệnh for
- + Lệnh do..while
- + Lệnh break và continue

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

3. Các hàm trong JavaScript:

Tên hàm	Chức năng
<code>stringName.length()</code>	Trả về chiều dài một chuỗi, <code>stringName</code> : tên của một chuỗi cho trước
<code>stringName.indexOf(stringsearch,[start])</code>	Trả về vị trí đầu tiên của chuỗi <code>stringsearch</code> trong chuỗi <code>stringName</code> , kể từ vị trí <code>start</code> .
<code>stringName.lastIndexOf(stringsearch,[start])</code>	Trả về vị trí cuối cùng của chuỗi <code>stringsearch</code> trong chuỗi <code>stringName</code> kể từ vị trí <code>start</code>
<code>stringName.substring(indexA,indexB)</code>	Trả về chuỗi con của chuỗi <code>stringName</code> , từ vị trí <code>indexA</code> đến vị trí <code>indexB</code>
<code>stringName.toUpperCase()</code>	Biến đổi tất cả các ký tự thường thành các ký tự hoa trong chuỗi <code>stringName</code>

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

3. Các hàm trong JavaScript:

<code>stringName.toLowerCase()</code>	Biến đổi tất cả các ký tự hoa thành các ký tự thường trong chuỗi <code>stringName</code>
<code>stringName1.concat(stringName2)</code>	Nối chuỗi <code>stringName2</code> vào sau chuỗi <code>stringName1</code>
<code>isNaN(value)</code>	Cho giá trị <code>true</code> nếu <code>value</code> không phải là một số, ngược lại có giá trị <code>false</code>
<code>Math.PI</code>	Hằng số <code>PI=3.14159</code>
<code>Math.max(number1, number2)</code>	Trả về giá trị lớn nhất giữa hai số <code>number1</code> và <code>number2</code>
<code>Math.min(number1, number2)</code>	Trả về giá trị nhỏ nhất giữa hai số <code>number1</code> và <code>number2</code>
<code>Math.abs(number)</code>	Lấy giá trị tuyệt đối của <code>number</code>

Chương 3: JavaScript

II. Ngôn ngữ JavaScript

3. Các hàm trong JavaScript:

Math.random()	Trả về một số thực ngẫu nhiên nằm giữa 0 và 1
Math.sqrt(number)	Trả về là giá trị bình phương số <i>number</i>
Math.sin(number)	Lấy sin của <i>number</i>
Math.cos(number)	Lấy cos của <i>number</i>
Math.tan(number)	Lấy tan của <i>number</i>
Math.acos(number)	Lấy arccos của <i>number</i>
Math.asin(number)	Lấy arcsin của <i>number</i>
Math.atan(number)	Lấy arctan của <i>number</i>

Chương 3: JavaScript

III. Xử lý các sự kiện trên trang Web

1. Sự kiện và đối tượng

+ Định nghĩa sự kiện: sự kiện là một hành động xảy ra khi người sử dụng làm một việc gì đó trên trang web.

Ví dụ: Kích (Click) một nút, thay đổi (Change) dữ liệu trong trường Textbox, di chuyển con trỏ chuột (Mouse) trên một siêu liên kết (Hyperlink),...

+ Và như vậy bạn phải tiến hành viết các lệnh để khi các sự kiện này xảy ra thì định hướng người sử dụng đáp ứng đúng nhu cầu của ứng dụng như : thông báo nhập sai dữ liệu, dữ liệu chưa đầy đủ,....

Chương 3: JavaScript

III. Xử lý các sự kiện trên trang Web

1. Sự kiện và đối tượng

- + Việc viết các lệnh để đáp ứng các sự kiện này gọi là điều khiển sự kiện. Trong JavaScript thì cách sử dụng các điều khiển nhau: **trước tiên là từ "on" và theo sau là tên sự kiện**
- Ví dụ : Khi kích một nút thì sự kiện Click xảy ra, để đáp ứng sự kiện này bạn dùng điều khiển onClick.
- + Bảng dưới đây liệt kê các sự kiện và các điều khiển sự kiện tương ứng cho tất cả các sự kiện trên web như sau :

Sự kiện	Điều khiển	Xảy ra khi	Áp dụng cho
blur	onBlur	Người sử dụng di chuyển focus rời khỏi phần tử trên form	Textbox, Textarea, Select list
click	onClick	Người sử dụng kích vào một phần tử trên form hoặc kích vào 1 liên kết	Button, button, button, button, button, Link
change	onChange	Người sử dụng thay đổi giá trị của phần tử	Textbox, Textarea, Select list
focus	onFocus	Người sử dụng di chuyển focus đến phần tử trên form	Textbox, Textarea, Select list
select	onSelect	Người sử dụng chọn dữ liệu trong các phần tử	Textbox, Textarea
load	onLoad	Nạp trang web vào trình duyệt	Document
unload	onUnload	Thoát khỏi một trang web	Document
mouseover	onMouseOver	Người sử dụng di chuyển con trỏ chuột trên một liên kết	Link
submit	onSubmit	Người sử dụng cho thi hành ứng dụng	Submit button

Chương 3: JavaScript

III. Xử lý các sự kiện trên trang Web

Ví dụ: Đây là một trang HTML hoàn chỉnh khi kích vào nút "Click here" sẽ xuất hiện câu thông báo "Hello..."

- <Script language="JavaScript">
- function thongbao()
- {
- alert("Hello...");
- }
- </Script>

Chương 3: JavaScript

III. Xử lý các sự kiện trên trang Web

Ví dụ: Đây là một trang HTML hoàn chỉnh khi kích vào nút "Click here" sẽ xuất hiện câu thông báo "Hello..."

- <BODY>
- <Form name="form1">
- <input type="button" value="Click here" onClick="thongbao()">
- </Form>
- </BODY>

Chương 3: JavaScript

III. Xử lý các sự kiện trên trang Web

1. Sự kiện và đối tượng

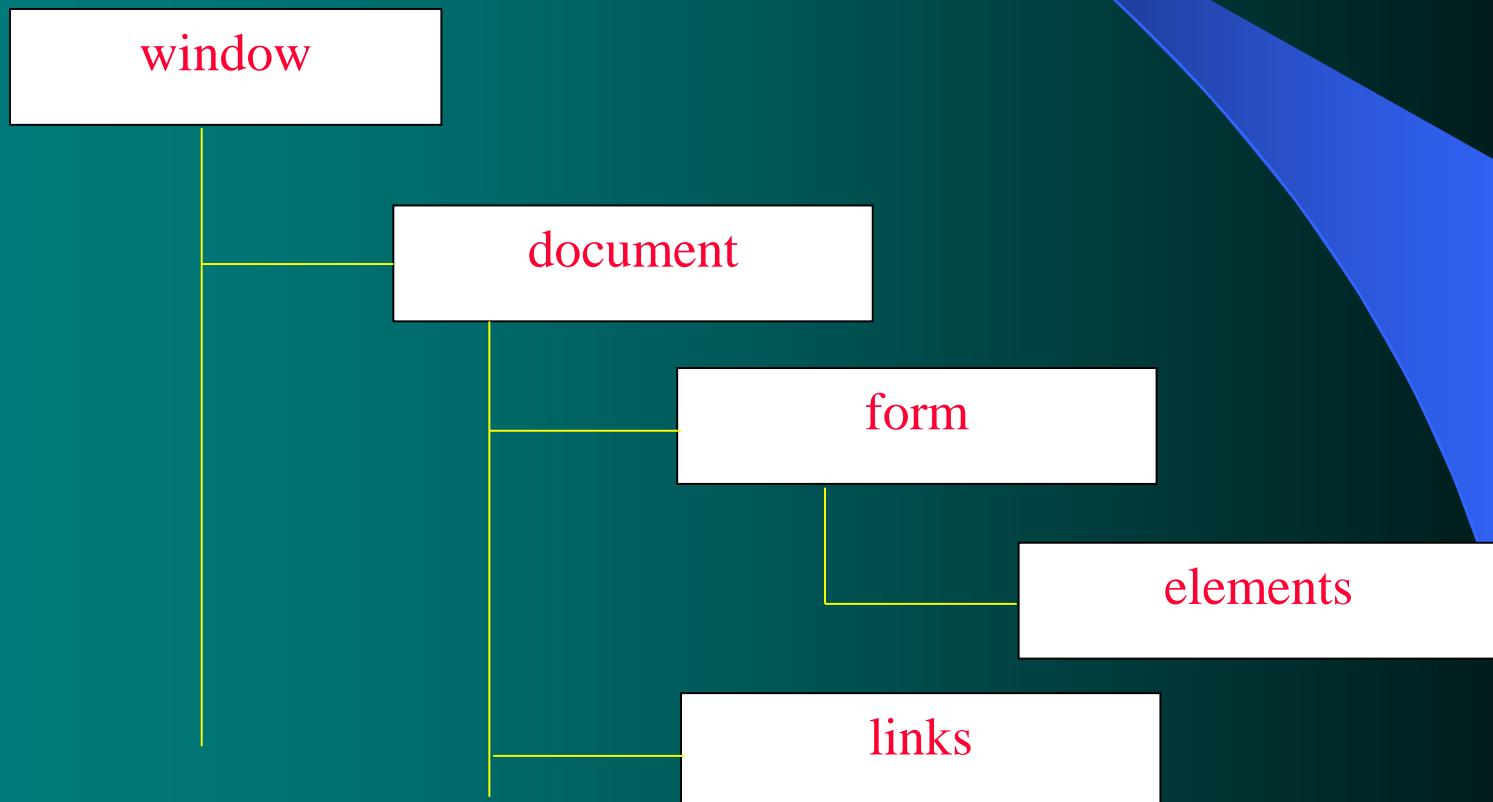
Để đáp ứng đúng sự kiện cho các đối tượng cần điều khiển, bạn có thể hình dung ra các đối tượng trên một trang web như sau:

- + Đối tượng lớn nhất của trình duyệt là window
- + Trong window chứa các tài liệu HTML (mỗi tài liệu là một đối tượng document)
 - + Trong document có thể chứa nhiều Form (có đối tượng là form), và các liên kết (có đối tượng là links)
 - + Trong Form có thể chứa nhiều phần tử (có đối tượng là elements)

Chương 3: JavaScript

III. Xử lý các sự kiện trên trang Web

1. Sự kiện và đối tượng



Chương 3: JavaScript

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

+ Để tham chiếu đến các thuộc tính (hay các phong thức) của một đối tượng nào đó bạn cần đặc tả tên của đối tượng đó cùng với các đối tượng cấp cao hơn nó.

Ví dụ: Để gán giá trị chuỗi "Hello" cho trường textbox (có tên là txtbox) trong một form (có tên là form1) bạn sử dụng lệnh sau:

```
document.form1.txtbox.value="Hello"
```

+ Trong đó : **value** là thuộc tính của đối tượng **txtbox**, các thành phần **document**, **form1** là các đối tượng cha của nó.

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.1 Đối tượng window

- * Các phương thức:

Phương thức	Mục đích
<code>alert("message")</code>	Đưa ra hộp thoại thông báo với nội dung là <i>message</i>
<code>confirm("message")</code>	Đưa ra hộp thoại yêu cầu người sử dụng xác nhận lại công việc cần hành động (chọn OK hay Cancel). Phương thức này trả về giá trị true (khi chọn OK) hoặc false (khi chọn Cancel)
<code>prompt("message",variable)</code>	Đưa ra hộp thoại cho phép người sử dụng nhập dữ liệu đầu vào, dữ liệu sau khi nhập được gán cho biến <i>variable</i> . <i>message</i> là dòng thông báo.
<code>close()</code>	Đóng cửa sổ trình duyệt
<code>open("URL","name")</code>	Mở trang web có địa chỉ là <i>URL</i> trong cửa sổ có tên là <i>name</i>

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.1 Đối tượng window

- * Các thuộc tính:

Thuộc tính	Mục đích
status	Chỉ định thông tin nằm trên thanh trạng thái ở cuối cửa sổ

- Để khi nạp trang web lên mà thanh trạng thái có dòng thông báo "*Chào các bạn...*" thì bạn dùng tag <BODY> như sau:

```
<BODY onLoad="window.status='Chào các bạn...'">
```

- * Các điều khiển sự kiện:

Đối tượng window có hai điều khiển sự kiện là **onLoad** và **onUnload**

Chương 3: JavaScript

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng
 - 2.2. Đối tượng document
 - 2.3. Đối tượng form
 - 2.4. Đối tượng links
 - 2.5. Đối tượng text, textarea, và password
 - 2.6. Đối tượng select (giống nh listbox)
 - 2.7. Đối tượng button, submit, reset
 - 2.8. Đối tượng checkbox, radio

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.2 Đối tượng document

- * Các phương thức:

Phương thức	Mục đích
write(<i>expression</i>) hoặc writeln(<i>expression</i>)	Viết biểu thức <i>expression</i> ra trang web trên cửa của trình duyệt Nếu có nhiều biểu thức cần viết thì các biểu thức này cách nhau dấu phẩy (,) writeln khác write ở chỗ là sau khi viết xong thì xuống hàng tiếp theo.

Ví dụ: Để viết một biểu thức $5+6=11$ ra trình duyệt bạn sử dụng cú pháp như sau:

- document.write("5 + 6 =",5+6)

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.2 Đối tượng document

* Các thuộc tính:

Thuộc tính	Mục đích
bgcolor="#rrggbb"	Xác định màu nền cho tài liệu
fgcolor="#rrggbb"	Xác định màu chữ cho tài liệu
linkColor="#rrggbb"	Xác định màu cho các liên kết khi chưa có sự viếng thăm liên kết đó
vlinkColor="#rrggbb"	Xác định màu cho các liên kết khi đã có sự viếng thăm liên kết đó
alinkColor="#rrggbb"	Xác định màu cho liên kết khi nó được kích hoạt

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.2 Đối tượng document

- Ví dụ : Thay đổi màu nền của trang web khi người sử dụng nhập vào tổ hợp màu (hay tên màu) trong ô textbox và kích nút "bgColor"

```
<html>
<head>
<script language="JavaScript">
function bgrColor()
{
    document.bgColor=document.form1.txtcolor.value
}
</script>
</head>
```

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.2 Đối tượng document

- Ví dụ : Thay đổi màu nền của trang web khi người sử dụng nhập vào tổ hợp màu (hay tên màu) trong ô textbox và kích nút "bgColor"

```
<body>
<form name="form1">
    <input type="text" name="txtcolor">
    <input type="button" value="bgColor" onClick="bgrColor()">
</form>
</body>
</html>
```

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.3 Đối tượng form

* Các phương thức

Phương thức	Mục đích
submit()	Cho thực thi ứng dụng (bắt đầu gửi số liệu lên Server xử lý)

* Các thuộc tính

Thuộc tính	Mục đích
Action	Đặc tả URL của Server mà dữ liệu từ form sẽ được gửi đến
Method	Đặc tả phương thức trao đổi thông tin giữa Client và Server (POST hay GET)
Name	Xác định tên của form
Target	Xác định tên của cửa sổ (hay của frame) mà khi dữ liệu phản hồi về từ Server sẽ được hiển thị trên cửa sổ đó

* Các điều khiển sự kiện : Chỉ có điều khiển sự kiện onSubmit

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.4 Đối tượng links

* Các thuộc tính

Thuộc tính	Mục đích
Target	Xác định tên của cửa sổ (hay cửa frame) mà khi kích vào liên kết thì trang được liên kết sẽ nạp vào cửa sổ đó.

* Các điều khiển sự kiện : Bao gồm onClick và onMouseOver

Ví dụ sau cho xuất hiện câu thông báo "*Click here*" ở thanh trạng thái khi bạn di chuyển con trỏ chuột trên dòng văn bản liên kết "*onMouseOver Sample*"

```
<a href="" onMouseOver="window.status='Click here'">onMouseOver Sample </a>
```

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.5 Đối tượng text, textarea, password

* Các phương thức

Phương thức	Mục đích
blur()	Di chuyển focus rời khỏi đối tượng
Focus()	Di chuyển focus tới đối tượng
Select()	Đánh khói dữ liệu đã được nhập cho đối tượng

* Các thuộc tính

Thuộc tính	Mục đích
defaultValue	Xác định giá trị ngầm định cho đối tượng
Name	Xác định tên của đối tượng
Value	Xác định giá trị hiện hành của đối tượng

* Các điều khiển sự kiện : Bao gồm onBlur, onChange, onFocus, và onSelect

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.5 Đối tượng text, textarea, password

Ví dụ: Tạo ra một form trong đó có một textbox dùng để nhập liệu. Sau khi nhập số liệu và dùng phím Tab để di chuyển focus khỏi ô nhập liệu thì tất cả văn bản trong ô đó sẽ chuyển sang chữ hoa, và khi nạp trang web lên thì focus (hay con trỏ) được đặt tại ô nhập liệu

```
<html>
<head>
<script language="JavaScript">
    function ToUpcase()
    {
        st=document.form1.txtbox.value
        document.form1.txtbox.value=st.toUpperCase()
    }
</script>
</head>
```

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.5 Đối tượng text, textarea, password

Ví dụ: Tạo ra một form trong đó có một textbox dùng để nhập liệu. Sau khi nhập số liệu và dùng phím Tab để di chuyển focus khỏi ô nhập liệu thì tất cả văn bản trong ô đó sẽ chuyển sang chữ hoa, và khi nạp trang web lên thì focus (hay con trỏ) được đặt tại ô nhập liệu

```
<body onLoad="document.form1.txtbox.focus()">
  <form name="form1">
    <input type="text" name="txtbox" onBlur="ToUppercase()">
  </form>
</body>
</html>
```

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.6 Đối tượng select (giống như listbox)

* Các thuộc tính

Thuộc tính	Mục đích
Length	Xác định số phần tử trong danh sách
Name	Xác định tên đối tượng
SelectedIndex	Trả về (hoặc thiết lập) chỉ mục của phần tử trong danh sách được chọn (nghĩa là phần tử thứ mấy được chọn, bắt đầu từ 0)

* Các điều khiển sự kiện : Bao gồm onBlur, onChange, và onFocus

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.6 Đối tượng select (giống như listbox)

Ví dụ: Tạo ra một form chứa 1 select box (gồm 4 phần tử từ 0 đến 3), 1 textbox để nhập liệu, và 1 button. Khi người sử dụng nhập một số vào ô textbox và kích nút button thì phần tử trong select box có chỉ mục bằng số được nhập sẽ được chọn

```
<html>
<head>
<script language="JavaScript">
    function SelectID()
    {
        s=document.form1.txtbox.value
        document.form1.lstbox.selectedIndex=s
        document.form1.txtbox.focus()
    }
</script>
</head>
```

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.6 Đối tượng select (giống như listbox)

Ví dụ: Tạo ra một form chứa 1 select box (gồm 4 phần tử từ 0 đến 3), 1 textbox để nhập liệu, và 1 button. Khi người sử dụng nhập một số vào ô textbox và kích nút button thì phần tử trong select box có chỉ mục bằng số được nhập sẽ được chọn

```
<body onLoad="document.form1.txtbox.focus()">
<form name="form1">
    <select name="lstbox">
        <option>Index 0</option>
        <option>Index 1</option>
        <option>Index 2</option>
        <option>Index 3</option>
    </select>
    <input type="text" name="txtbox" size=5>
    <input type="button" value="Click" onClick="SelectID()">
</form>
</body>
</html>
```

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.7 Đối tượng button, submit, reset

* Các thuộc tính

Thuộc tính	Mục đích
Name	Xác định tên đối tượng
Value	Xác định giá trị của đối tượng

* Các điều khiển sự kiện : Chỉ có onClick

III. Xử lý các sự kiện trên trang Web

2. Các phương thức và thuộc tính của các đối tượng

2.8 Đối tượng checkbox, radio

* Các thuộc tính

Thuộc tính	Mục đích
Checked	Xác định trạng thái được chọn hay không (true hay false) của đối tượng hay một mảng đối tượng
defaultChecked	Chỉ định giá trị ngầm định cho đối tượng được chọn
name	Xác định tên của đối tượng
value	Xác định giá trị của đối tượng

* Các điều khiển sự kiện : Chỉ có onClick

Active Server Pages

I. Tổng quan Về ASP

1. Giới thiệu

ASP là công cụ hỗ trợ cho việc tạo ra các trang web hùng mạnh (các trang này thường được gọi là các trang ASP, và có phần mở rộng là *.asp).

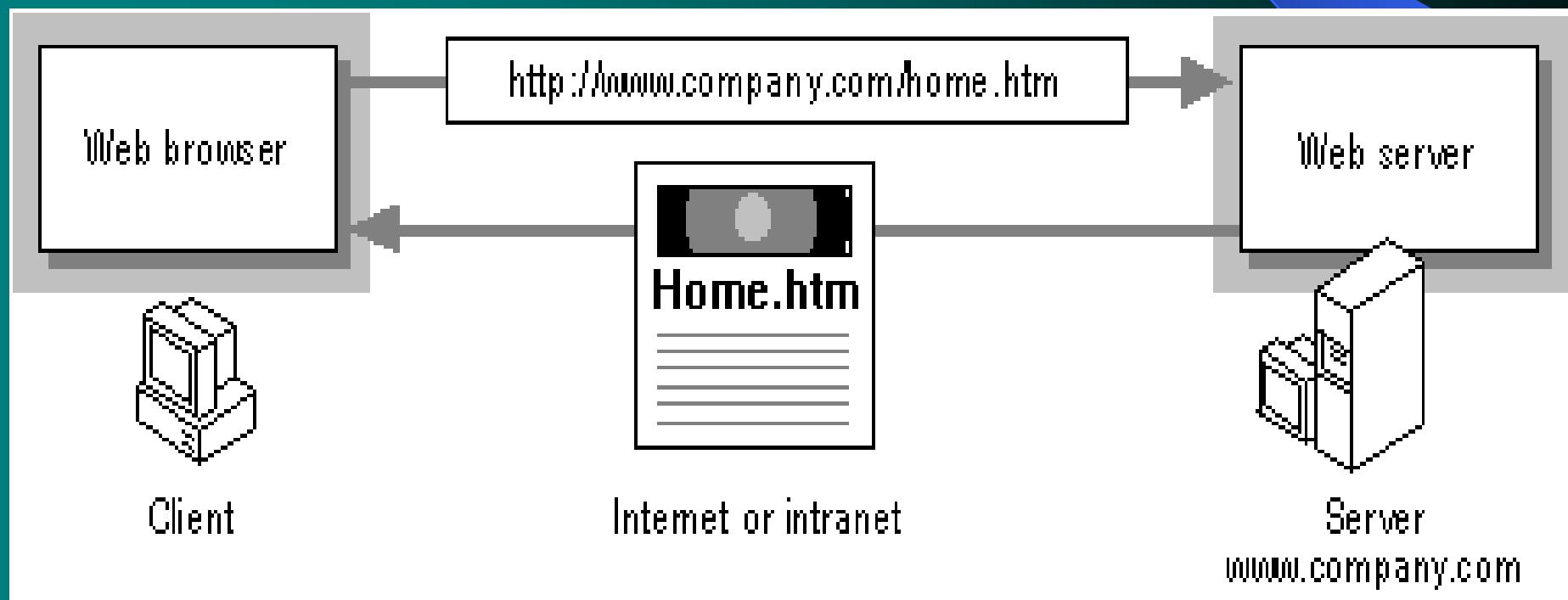
Trang ASP có thể xem là một trang HTML mở rộng, vì ngoài tính năng đầy đủ của một tài liệu HTML, trong nó còn có thể tích hợp các câu lệnh của VBScript (hoặc JavaScript) để truy cập tài nguyên hệ thống, xử lý tính toán số liệu, sử dụng các câu lệnh của ngôn ngữ SQL kết hợp với các đối tượng được cung cấp sẵn trong ASP để truy cập cơ sở dữ liệu như Access, SQL Server, Oracle,...

Active Server Pages

I. Tổng quan Về ASP

2. ASP làm việc như thế nào ?

Trang Web tĩnh :

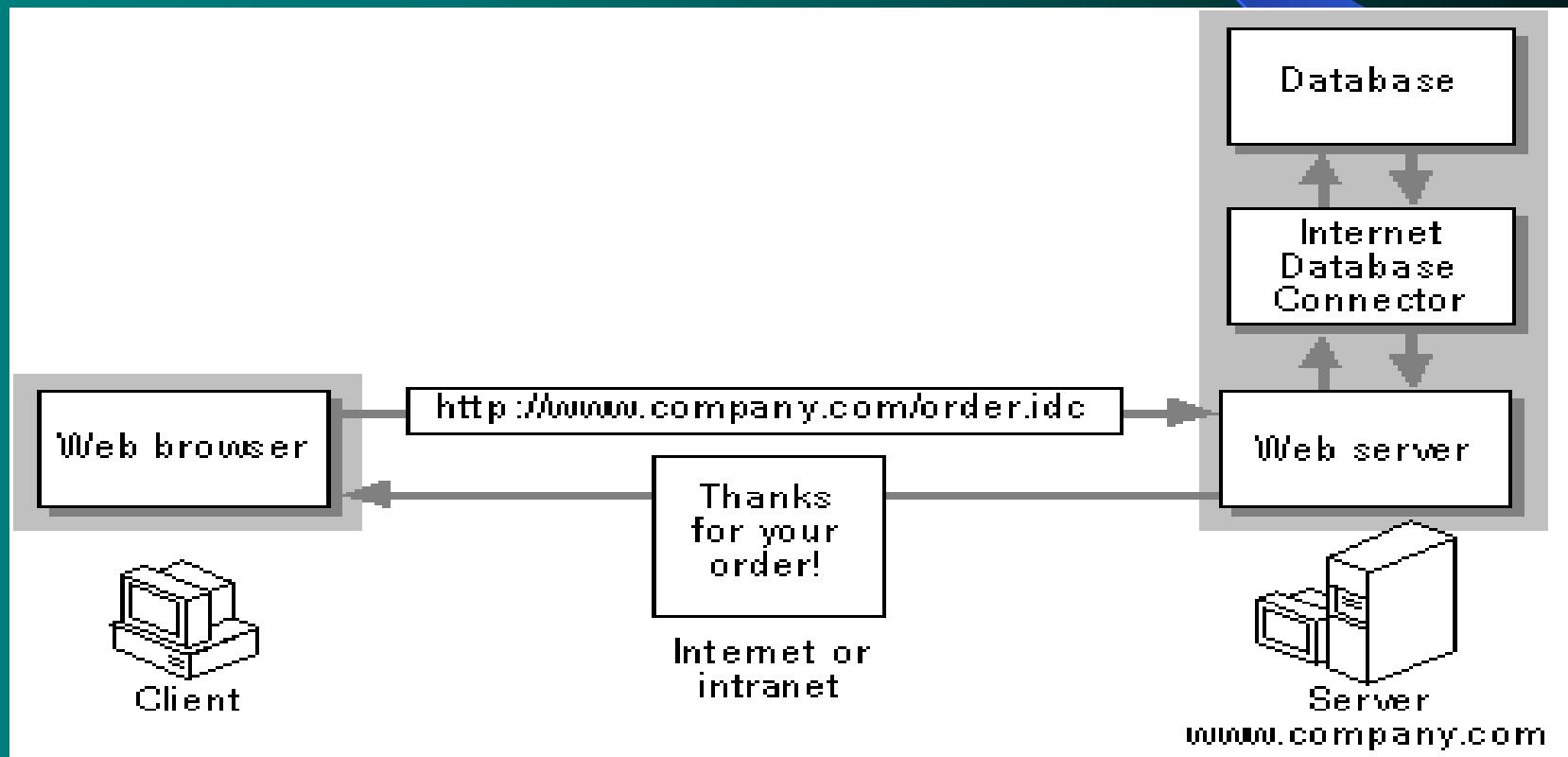


Active Server Pages

I. Tổng quan Về ASP

2. ASP làm việc như thế nào ?

Trang web động (Dynamic):

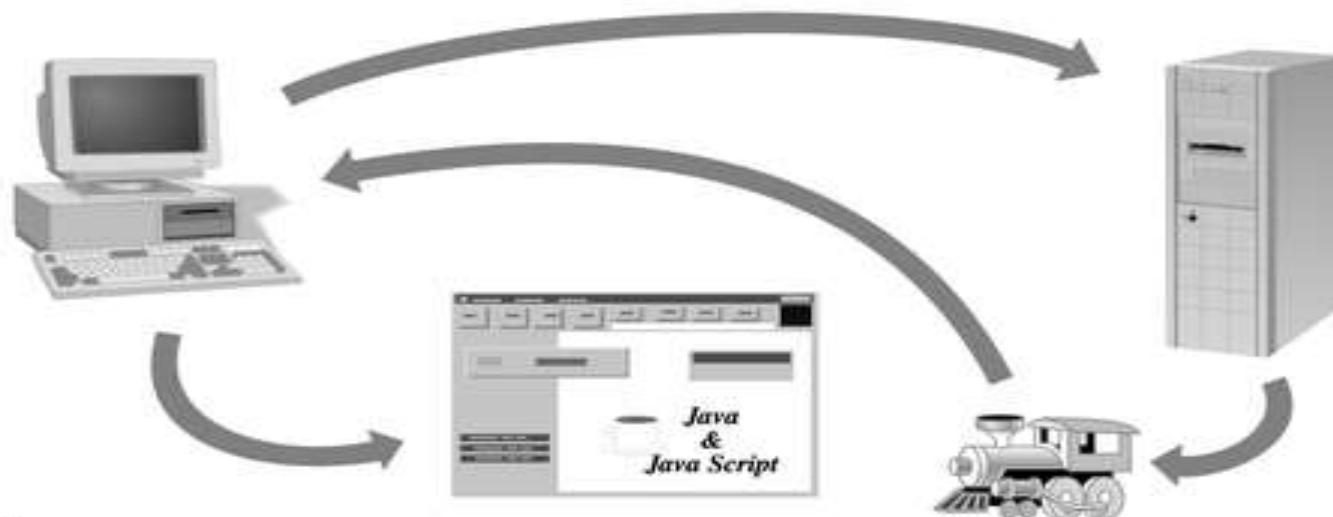


Active Server Pages

I. Tổng quan Về ASP

2. ASP làm việc như thế nào ?

SERVER-SIDE SCRIPTING MODEL



The browser requests a page.

The server sends the script to the engine.

The script engine executes the script.

The server sends the page to the browser.

The browser renders the page.

Active Server Pages

I. Tổng quan Về ASP

3. Cách xây dựng một trang ASP

Các file ASP có phần mở rộng là .ASP

Khi chèn thêm các lệnh này bạn cần chú ý:

- + Các lệnh chèn thêm vào phải nằm giữa khoá `<% ... %>`
- + Các lệnh thêm vào có thể đặt bất kỳ tại vị trí nào trong file HTML
- + Các câu ghi chú trong ASP (không kể các ghi chú của HTML) đều được bắt đầu bằng dấu nháy (`)
- + Các lệnh không phân biệt chữ hoa, chữ thường.

Active Server Pages

I. Tổng quan Về ASP

3. Cách xây dựng một trang ASP

Ví dụ: Một file asp đơn giản là cộng hai số và trả kết quả về cho trình duyệt.

<%

'bắt đầu cho phép viết các lệnh ASP

Dim a,b,c

'khai báo 3 biến a,b,c

a=10

'gán 10 cho biến a

b=20

'gán 20 cho biến b

c=a+b

'tính tổng c=a+b

'kết thúc viết lệnh cho ASP

%>

Active Server Pages

I. Tổng quan Về ASP

3. Cách xây dựng một trang ASP

<!--bắt đầu phần tài liệu HTML-->

- <HTML>
- <HEAD>
- <Title>Ví dụ về ASP</Title>
- <HEAD>
- <BODY>
- <!-- in kết quả ra trang web-->
- Tổng của a và b là <%=c%>
- </BODY>
- </HTML>

Active Server Pages

I. Tổng quan Về ASP

3. Cách xây dựng một trang ASP

- Với ví dụ ta chú ý đến lệnh `<%=c%>`, lệnh này cho phép viết kết quả sau khi tính toán ra trang web
- Như vậy để viết một kết quả ra trang web bạn thực hiện theo cú pháp:

`<%=bieu_thuc%>`

- Trong đó *bieu_thuc* có thể là một biến, một biểu thức, hay một hàm cần in kết quả.

Active Server Pages

II. Lập trình với ASP

1. Khai báo biến

- + Không bắt buộc nhưng nên khai báo để kiểm soát và bắt lỗi.
- + Nên sử dụng `<% Option Explicit %>` ở ngay đầu mỗi tệp ASP.
- + Cú pháp : `Dim biến 1, biến 2...`
- + Để khai báo mảng:
`Dim a(10) : chỉ số chạy từ 1->10`

- + Khi khai báo nên sử dụng các tiền tố:

Active Server Pages

II. Lập trình với ASP

1. Khai báo biến.

Dạng dữ liệu	Tiền tố
Boolean	Bln
Byte	Byt
Double	Dbl
Integer	Int
Long	Lng
Object	Obj
String	Str
ADO command	Cmd
ADO connection	Cnn
.....	

Active Server Pages

II. Lập trình với ASP

CONST tênhàng= giá trị

a. Cấu trúc IF..THEN..ELSE..IF

Cú pháp:

- <% If <Điều kiện 1> then
• <Các câu lệnh>
• Else
• If <Điều kiện 2> then
• <Các câu lệnh>
• End if
• End if
• %>

Active Server Pages

II. Lập trình với ASP

3. Các cấu trúc điều khiển

Ngoài ra có thể sử dụng : IF.. THEN. . . . ELSEIF. . . .
END IF

- <%
- If <Điều kiện 1> then
 - <Các câu lệnh 1>
- ElseIf <Điều kiện 2> then
 - <Các câu lệnh 2>
- Else
 - <Các câu lệnh 3>
- End if
- %>

Active Server Pages

II. Lập trình với ASP

3. Các cấu trúc điều khiển

b. Cấu trúc Select case...end select

Cú pháp: <% Select Case <tenbien>

Case <gia tri 1>

<Nhóm lệnh 1>

Case <gia tri 2>

<Nhóm lệnh 2>

Case <gia tri n>

<Nhóm lệnh n>

Case Else

<Nhóm lệnh n+1>

End Select

%>

Active Server Pages

II. Lập trình với ASP

3. Các cấu trúc điều khiển

c. Cấu trúc lặp tuần tự For...Next.

Dùng để lặp với số lần đã biết, tuy nhiên ta có thể ngắt bằng lệnh EXIT FOR.

Cú pháp:

<%

FOR <tenbien>=<giatriidau> TO <Giatribien> STEP <Buoc nhay>

NEXT

%>

Active Server Pages

II. Lập trình với ASP

3. Các cấu trúc điều khiển

d. Cấu trúc lặp khác

* DO WHILE....LOOP :

Cú pháp:

<%

Do while <Điều kiện>

<Các câu lệnh>

Loop

%>

Active Server Pages

II. Lập trình với ASP

3. Các cấu trúc điều khiển

d. Cấu trúc lặp khác

* WHILE....wend

Cú pháp:

<%

While <Điều kiện>

<Các câu lệnh>

Wend

%>

Active Server Pages

II. Lập trình với ASP

3. Các cấu trúc điều khiển

d. Cấu trúc lặp khác

* DO....Loop until

Cú pháp:

<%

DO

<Các câu lệnh>

Loop until <Điều kiện>

%>

Active Server Pages

II. Lập trình với ASP

4. Một số hàm hữu dụng trong ASP

a. Các hàm xử lý văn bản

- **TRIM(xâu as string)**: Bỏ khoảng trắng hai đầu kí tự
- **LEFT(Xâu as string, n as interger)**: Lấy bên trái xâu n kí tự.
- **RIGHT(Xâu as string, n as interger)**: Lấy bên phải xâu n kí tự.
- **LCASE(Xâu as string)** : Chuyển xâu về chữ thường
- **UCASE(Xâu as string)** : Chuyển xâu về chữ hoa
- **MID(xâu as string, n1, n2)**: Lấy n2 kí tự trong xâu bắt đầu từ vị trí n1.
- **CSTR(Biến)**: Hàm chuyển đổi biến thành kiểu string

Active Server Pages

II. Lập trình với ASP

4. Một số hàm hữu dụng trong ASP

a. Các hàm xử lý văn bản

Hàm JOIN/SPLIT(Xâu as string, kí tự ngăn cách): Sẽ Nối/Cắt xâu thành một/nhiều đoạn bằng cách xác định kí tự ngăn cách ở trên và cho các đoạn đó lần lượt vào một mảng.

Ví dụ:

```
<%
```

```
  x="Hà nội;Hải phòng; TPHCM"
```

```
  y=split(x,";")
```

```
  Response.write y(0)
```

```
  'y(0)="Hà nội"
```

```
%>
```

Active Server Pages

II. Lập trình với ASP

4. Một số hàm hữu dụng trong ASP

b. Các hàm xử lý số

- **SQR(n)**: Căn bậc hai của n
- **INT(n)** : Lấy phần nguyên n
- **MOD** : Phép chia d
- **Toán tử ** : Chia nguyên
- **Round(số, n)** : Làm tròn số với n chữ số thập phân
- **RND()**: Trả về số ngẫu nhiên bất kì trong khoảng [0,1]

c. Các hàm về ngày tháng, thời gian.

Active Server Pages

III. Các thành phần trong ASP

Trong ASP đã cung cấp sẵn 5 thành phần (Component) cho phép tạo các trang web động một cách nhanh chóng.

+ **Database Access**: đây là thành phần cho phép truy cập các loại cơ sở dữ liệu

+ **Ad Rotator**: cho phép hiển thị các hình ảnh quảng cáo một cách tự động theo thời gian quy định.

+ **Browser Capabilities**: xác định các khả năng, đặc điểm và phiên bản (version) của các trình duyệt khi truy cập Website.

+ **Content Linking**: tạo ra các bảng chỉ mục nội dung (giống nh mục lục) của các trang web.

+ **File Access**: cung cấp các chức năng truy cập tới hệ thống file và th mục trên Server.

Active Server Pages

III. Các thành phần trong ASP

1. Quá trình truy cập Cơ sở dữ liệu

DataBase Access sử dụng một đối tượng gọi là ADO (ActiveX Data Object) và kết hợp khả năng của ODBC (được giải thích ở phần sau trong chương này) để lấy thông tin trong một cơ sở dữ liệu. Việc truy cập cơ sở dữ liệu có thể tiến hành qua các bước sau:

- + Đăng ký một tên đại diện cho nguồn dữ liệu cần truy cập với ODBC (sẽ giải thích ở phần sau)
- + Tạo một nối kết tới nguồn dữ liệu cần truy cập
- + Mở Cơ sở dữ liệu để truy cập
- + Sử dụng ngôn ngữ SQL để truy vấn dữ liệu
- + Xử lý dữ liệu
- + Đóng nối kết.

Active Server Pages

III. Các thành phần trong ASP

1. Quá trình truy cập Cơ sở dữ liệu

- Một loạt các tiến trình trên có thể mô tả qua ví dụ sau:

Ví dụ này sử dụng file Vidu.asp để truy cập Cơ sở dữ liệu Access (với file có tên VIDU.MDB)

Giả sử rằng trong file này có 1 bảng (Table) với tên là KHACHHANG gồm các trường *STT*, *Hoten*, *Diachi* (các trường này trong Access được đánh số từ 0 tới 2).

Sau khi đăng ký file VIDU.MDB trong ODBC với tên VIDU (tên này là bất kỳ, không cần trùng với tên file .MDB, và việc đăng ký chỉ cần tiến hành một lần, các lần sau chỉ việc truy cập dữ liệu mà không cần đăng ký nữa)

Trong file Vidu.asp ta viết các lệnh sau:

Active Server Pages

III. Các thành phần trong ASP

1. Quá trình truy cập Cơ sở dữ liệu

- <%

'Tạo ra một đối tượng conn cho phép nối tới nguồn dữ liệu

- set conn=Server.CreateObject("ADODB.Connection")

'Sử dụng phương thức OPEN của đối tượng CONN để mở CSDL cần truy cập

- conn.open "VIDU"

'Biến sql chứa câu lệnh lấy tất cả dữ liệu từ bảng KHACHHANG

- sql="SELECT * FROM KHACHHANG"

'Sử dụng phương thức Execute của đối tượng Conn để thực hiện câu lệnh trong biến sql

'Kết quả sau khi truy vấn được trả về trong đối tượng RS

- Set RS = Conn.Execute(sql)

- %>

Active Server Pages

- <HTML>
- <Body>
- <%

'Sử dụng vòng lặp để kiểm tra RS có còn chứa dữ liệu hay không

- Do While NOT RS.EOF
- %>

STT: <%=RS(0)%>, Họ tên: <%=RS(1)%>, Địa chỉ:
<%=RS(2)%>

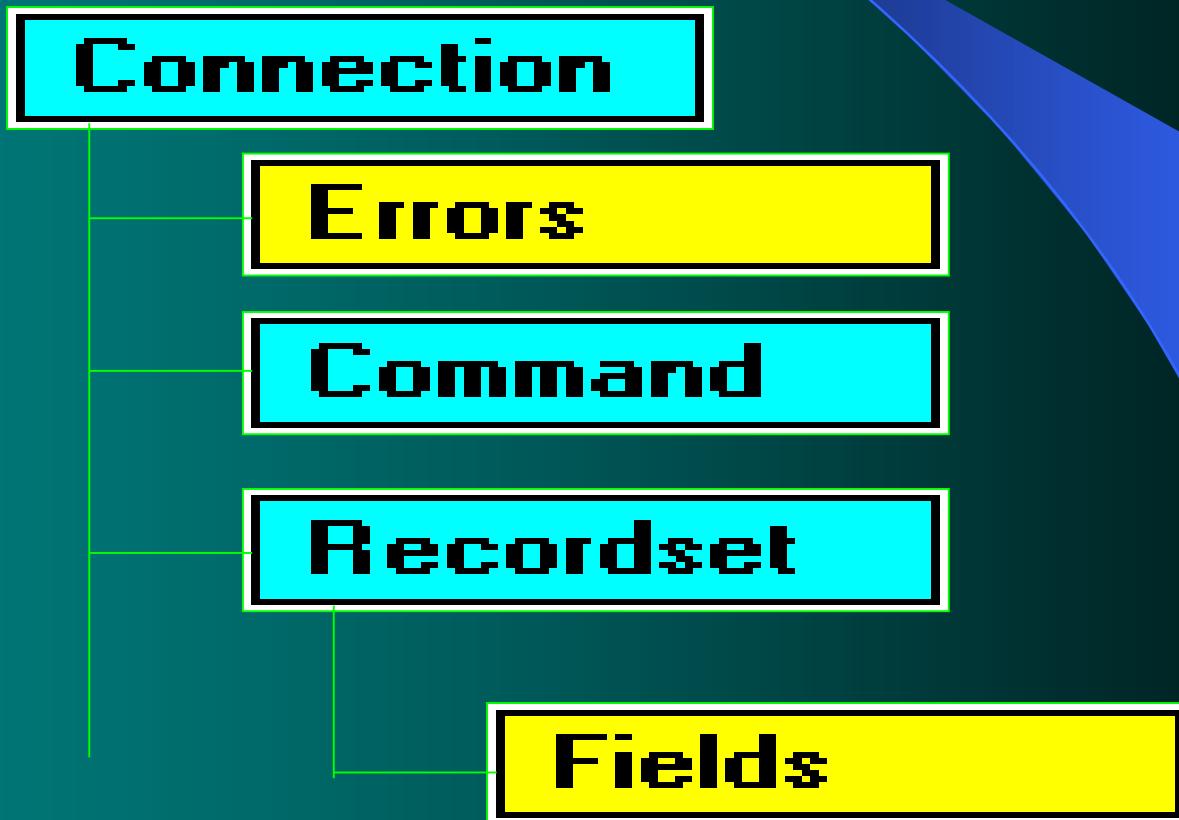
- <% RS.MoveNext
tiếp theo
Loop
Conn CLOSE
>
</BODY>
</HTML>
- 'di chuyển đến Record
'quay lại vòng lặp
'đóng nối kết

Active Server Pages

III. Các thành phần trong ASP

2. Các đối tượng xử lý dữ liệu

Đối tượng ADO cho phép tạo ra các đối tượng con khác :



Active Server Pages

III. Các thành phần trong ASP

2. Các đối tượng xử lý dữ liệu

+ **Đối tượng Connection** : mô tả sự nối kết tới nguồn dữ liệu. Đối tượng này cho phép tạo ra 3 đối tượng con là Errors, Command, và RecordSet

+ **Đối tượng Errors** : chứa chi tiết các lỗi về truy cập dữ liệu

+ **Đối tượng Command** : định nghĩa các lệnh để truy cập dữ liệu

+ **Đối tượng RecordSet** : chứa tất cả các record dữ liệu được lấy về từ các bảng (table) trong cơ sở dữ liệu, đối tượng này cho phép tạo ra một Collection là Fields

+ **Collection Fields** : dùng để chứa tất cả các trường nằm trong một Recordset

Active Server Pages

III. Các thành phần trong ASP

2. Các đối tượng xử lý dữ liệu

2.1. Các phương thức, thuộc tính của đối tượng Connection

Các Thuộc tính:

- ConnectionTimeout: Xác định khoảng thời gian cho phép nối kết tới nguồn dữ liệu
- CommandTimeout: Xác định khoảng thời gian cho phép truy vấn dữ liệu

Active Server Pages

III. Các thành phần trong ASP

2.1. Các phương thức, thuộc tính của đối tượng Connection

Các phương thức:

Connection: mở một nối kết tới nguồn dữ liệu

Cú pháp: *connection.Open ConnectionString*

Trong đó *ConnectionString* là một chuỗi cung cấp thông tin cho việc nối kết

Execute: thực hiện một câu lệnh được đặc tả

Cú pháp: Set *recordset* = *connection.Execute(Command)*

- Trong đó: *Command* là câu lệnh cần thực hiện
- *Recordset* là đối tượng Recordset
- *Connection*: là đối tượng Connection

Close: đóng một đối tượng đang mở và các đối tượng khác liên quan

Cú pháp: *object.Close*

Active Server Pages

III. Các thành phần trong ASP

2. Các đối tượng xử lý dữ liệu

2.2. Các phương thức, thuộc tính của Recordset

- Các Thuộc tính:

+ **BOF** : thuộc tính này trả về giá trị TRUE nếu vị trí của Record hiện hành trước record đầu tiên trong đối tượng Recordset, ngược lại trả về giá trị FALSE

+ **EOF** : trả về giá trị True nếu vị trí của Record hiện hành nằm sau record cuối cùng, ngược lại trả về giá trị False

Active Server Pages

III. Các thành phần trong ASP

2.2. Các phương thức, thuộc tính của Recordset

Các phương thức:

- Close: đóng đối tượng đang mở và các đối tượng khác liên quan

Cú pháp: *object.Close*

- MoveFirst: Di chuyển tới record đầu tiên trong 1 Recordset

Cú pháp: *recordset.MoveFirst*

- MoveLast: Di chuyển tới record cuối cùng trong 1 Recordset

Cú pháp: *recordset.MoveLast*

- MoveNext: Di chuyển tới record tiếp theo trong 1 Recordset

Cú pháp: *recordset.MoveNext*

- MovePrevious: Di chuyển tới record trước Record hiện hành trong 1 Recordset

Cú pháp: *recordset.MovePrevious*

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

ASP cung cấp sẵn 5 đối tượng cho phép bạn xử lý thông tin trao đổi giữa Client và Server.

Đối tượng	Chức năng
Request	Lấy thông tin từ Client gửi lên Server
Response	Gửi thông tin từ Server đến Client
Server	Cung cấp cách thức truy cập tới các phong thức và các thuộc tính mà được cung cấp sẵn như các ứng dụng trên Server
Session	Lưu trữ thông tin về từng phiên làm việc của từng Client
Application	Chia sẻ thông tin của một ứng dụng cho tất cả các Client

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.1. Đối tượng Request

Cú pháp chung: **Request[.Collection](variable)**

Trong đó *Collection* có thể là một trong số các giá trị sau:

- Form: dùng để lấy tất cả giá trị của các trường trong Form HTML, khi Form đó sử dụng method="post"

Cú pháp dùng cho Form:

Request.Form(*parameter*)[(*index*)]

- + *parameter* : tên của một trường cần nhận giá trị
- + *Index* là giá trị tùy chọn nó cho phép bạn truy cập đến những giá trị khác nhau của *parameter*.

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.1. Đối tượng Request

- **QueryString:** dùng để lấy tất cả giá trị của các biến nằm sau dấu hỏi (?) trong một URL (gọi là Query String), hoặc khi một form dùng Method="GET"

`Request.QueryString(variable)[(index)]`

- + *variable* là tên của biến trong chuỗi query string cần nhận
 - + *Index* là tham số tùy chọn cho phép nhận một hoặc nhiều giá trị của biến

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.1. Đối tượng Request

Ví dụ: Giả sử có một URL nh sau:

- /Scripts/Sample.asp?hoten=NguyenvanA&tuoi=20

Tất cả những gì nằm sau dấu hỏi (?) được gọi là Query String. Trong Query String chia ra làm nhiều phần, mỗi phần cách nhau bởi dấu &. Trong mỗi phần cũng bao gồm hai thành phần cách nhau bởi dấu bằng (=), phần trước dấu (=) là tên biến, phần sau dấu bằng là giá trị của biến.

Nh vậy trong file Sample.asp muốn lấy hoten và tuoi bạn sử dụng lệnh nh sau:

```
<%
```

```
a=Request.QueryString("hoten")
```

'biến a=NguyenvanA

```
b=Request.QueryString("tuoi")
```

'biến b=20

```
%>
```

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.1. Đối tượng Request

- ServerVariables: Lấy giá trị của các biến môi trường (các biến do ASP cấp sẵn)

Cú pháp: Request.ServerVariables (*variable*)

Trong đó *variable* là tên của biến môi trường cần nhận.

Variable có các giá trị sau:

- + REQUEST_METHOD: lấy giá trị của cách thức mà Client yêu cầu Server.

Ví dụ:

```
<FORM ACTION = "/scripts/submit.asp" METHOD = "post">  
Request.ServerVariables("REQUEST_METHOD")="post".
```

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.1. Đối tượng Request

- + **PATH_INFO**: Xác định đường dẫn ảo (virtual path) của file đang thi hành
- + **SCRIPT_NAME**: Xác định đường dẫn ảo của script đang thi hành.
- + **LOGON_USER**: Lấy Username của người logon vào mạng
- + **REMOTE_ADDR** : Lấy địa chỉ IP của máy trạm yêu cầu thông tin
- + **SERVER_NAME**: Lấy tên hay địa chỉ IP của Server
- + **REMOTE_HOST**: Lấy địa chỉ IP của Server.

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.2. Đối tượng RESPONSE

Đối tượng này dùng để gửi dữ liệu về Client

Cú pháp:

`Response.property|method`

Trong đó:

properties gồm các giá trị sau:

- + **Buffer**: Xác định có nên sử dụng vùng đệm cho các trang thông tin hay không. Khi nó có giá trị True thì Server sẽ không trả lời bất kỳ một yêu cầu nào đến client cho đến khi trang hiện tại đã xử lý xong, hoặc cho đến khi gọi phương thức Response.Flush hay Response.End. Nên đặt Response.Buffer ở dòng đầu tiên trong file (.ASP).
- Cú pháp: `Response.Buffer=True/False`
- + **Expires**: Xác định thời gian mà trang thông tin nằm trong vùng đệm hết tác dụng
- Cú pháp: `Response.Expires [= number]`
 - Trong đó *number* là số phút mà trang hết thời hạn

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.2. Đối tượng RESPONSE

Method gồm các giá trị sau:

+ Write: dùng để in một biểu thức ra màn hình.

Cú pháp: Response.Write *Bieu_thuc*

+ Redirect : dùng để nhảy đến một URL khác.

Cú pháp: Response.Redirect *URL*

Với *URL* là một địa chỉ mới mà bạn cần chuyển đến.

+ Clear: Xoá vùng đệm của các trang HTML.

Phương thức này sẽ gặp lỗi nếu nh trước đó không gán Response.Buffer = True.

Cú pháp: Response.Clear

+ End: dừng việc xử lý file .asp và trả về kết quả hiện hành

Cú pháp: Response.End

+ Flush: gửi ngay lập tức dữ liệu trong bộ đệm ra màn hình.

Phương thức này sẽ gặp lỗi nếu như trước đó không gán Response.Buffer = TRUE.

Cú pháp: Response.Flush

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.3. Đối tượng SESSION

- Đối tượng Session dùng để lu trữ thông tin cho mỗi phiên làm việc (của mỗi User), khi User này kết nối vào WebSite của bạn.
- Những thông tin được lu trữ trong Session sẽ không bị mất đi khi User này nhảy từ trang này qua trang khác tại Web Site của bạn.
- Server sẽ tự động tạo ra đối tượng SESSION khi một trang web từ một ứng dụng được yêu cầu bởi người sử dụng, Server sẽ phá huỷ đối tượng SESSION khi kết thúc phiên làm việc, hoặc có yêu cầu phá huỷ.

Cú pháp: Session.*property|method*

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.3. Đối tượng SESSION

* *Properties* có các giá trị sau:

- + **SessionID**: Trả về định danh của Session cho từng User, định danh này là duy nhất cho mỗi User (giống như mã số để quản lý User)

Cú pháp: Session.SessionID

- + **TimeOut**: Thiết lập thời gian hết hạn cho đối tượng Session trong từng ứng dụng, được tính bằng phút
- Nếu trong khoảng thời gian này mà User không Refresh hoặc yêu cầu thông tin thì Session kết thúc.

Cú pháp: Session.Timeout [= nMinutes]

- Với *nMinutes* là số phút cần thiết lập (ngầm định là 20 phút)

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.3. Đối tượng SESSION

* *Method* có các giá trị sau:

- + **Abandon**: dùng để phá huỷ đối tượng Session và giải phóng khỏi bộ nhớ.

Cú pháp: **Session.Abandon**

Trong đối tượng SESSION còn có hai sự kiện là **Session_OnEnd** và **Session_OnStart**.

Sự kiện **Session_onStart** được đưa ra khi Server tạo ra Session mới, sự kiện **Session_onEnd** được đưa ra Session bị phá huỷ (bằng phương thức **Abandon** hay thuộc tính **Timeout**).

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.3. Đối tượng SESSION

- *Cú pháp của hai sự kiện này được viết như sau:*
- **<SCRIPT LANGUAGE=ScriptLanguage RUNAT=Server>**
- **Sub Session_OnStart**
- *. . . các lệnh cho khởi tạo phiên làm việc đặt ở đây*
- **End Sub**
- **</SCRIPT>**
- **<SCRIPT LANGUAGE=ScriptLanguage RUNAT=Server>**
- **Sub Session_OnEnd**
- *. . . các lệnh cho kết thúc phiên làm việc ở đây*
- **End Sub**
- **</SCRIPT>**
- *Hai sự kiện này được soạn trong một file có tên là Global.asa, và file này phải được đặt ở thư mục gốc của ứng dụng đang thi hành.*

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.4. Đối tượng Application

Dùng để chia sẻ thông tin giữa các user đang dùng chung một ứng dụng nào đó.

Do một biến Application có thể dùng chung cho tất cả các user, nên tại một thời điểm nếu bạn muốn thay đổi giá trị của biến Application thì sử dụng phong thức

- + Lock để khoá sự thay đổi ở các User khác
- + Sau đó sử dụng phong thức Unlock để trả lại trạng thái ban đầu của nó.

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.4. Đối tượng Application

Cú pháp:

- + **Application(var)**: dùng để chứa giá trị của biến Application. Với *var* là tên biến
- + **Application.Lock**: ngăn cản không cho client sửa đổi giá trị của biến Application
- + **Application.Unlock**: cho phép sử đổi giá trị của biến Application

Ví dụ: Để gán giá trị 25 cho biến *num* sử dụng cú pháp sau

Application("num") = 25

Biến *num* sẽ được truy xuất bởi tất cả các Client.

Hoặc để viết giá trị của biến *num* ra màn hình sử dụng cú pháp :

<% Response.write Application("num") %>

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.4. Đối tượng Application

+ Cũng giống như đối tượng Session, đối tượng Application có hai sự kiện là **Application_OnStart**, **Application_OnEnd**, sự kiện thứ nhất được đưa ra khi bắt đầu thực thi ứng dụng và sự kiện thứ hai cho kết thúc một ứng dụng.

+ Cả hai sự kiện này được soạn thảo trong file Global.asa, và file này phải được đặt ở thư mục gốc mà ứng dụng hiện tại đang hoạt động. Trong file Global.asa sự kiện **Application_OnStart** được ưu tiên tìm thấy và sử dụng trước sự kiện **Session_OnStart**. Sự kiện **Application_OnEnd** có ưu tiên sau **Session_OnEnd**.

Active Server Pages

III. Các thành phần trong ASP

3. Các đối tượng khác trong ASP

3.4. Đối tượng Application

Cú pháp của hai sự kiện như sau:

```
<SCRIPT LANGUAGE=ScriptLanguage RUNAT=Server>
  Sub Application_OnStart
    . . . các lệnh khi bắt đầu thực hiện ứng dụng
  End Sub
</SCRIPT>
<SCRIPT LANGUAGE=ScriptLanguage RUNAT=Server>
  Sub Application_OnEnd
    . . . các lệnh khi kết thúc ứng dụng
  End Sub
</SCRIPT>
```

Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

1. Giới thiệu

Như các bạn đã biết là mỗi loại cơ sở dữ liệu đều cung cấp một **phương pháp riêng để truy cập thông tin trong cơ sở dữ liệu** đó, điều này đã gây không ít khó khăn cho việc truy cập thông tin cơ sở dữ liệu trên mạng Internet.

Do đó trong nhiều năm, các tập đoàn trên thế giới đã đầu tư rất nhiều tiền của vào việc xây dựng một phương pháp lập trình nhằm cung cấp một giao diện lập trình ứng dụng duy nhất mà có thể **truy cập được nhiều loại cơ sở dữ liệu** của **nhiều hệ quản trị Cơ sở dữ liệu khác nhau**.

Để giải quyết vấn đề này, tập đoàn Microsoft đã ra một chuẩn gọi là **Open DataBase Connectivity** (ODBC: sự nối kết Cơ sở dữ liệu mở).

Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

1. Giới thiệu



- **Drivers Manager (Trình quản lý các trình điều khiển):** đây là một bộ phận thi hành như một kẽ trung gian giữa ứng dụng và trình điều khiển (Driver). Drivers Manager lấy thông tin từ ứng dụng và kích hoạt từng trình điều khiển thích hợp cho việc truy cập từng loại cơ sở dữ liệu.
- **Driver (Trình điều khiển):** cung cấp các trình điều khiển để truy cập cơ sở dữ liệu. Trình điều khiển này chịu sự điều khiển của các câu truy vấn SQL và trả kết quả về cho ứng dụng.
- **DataSource (Nguồn dữ liệu):** đây là nguồn dữ liệu cần truy cập để lấy thông tin.

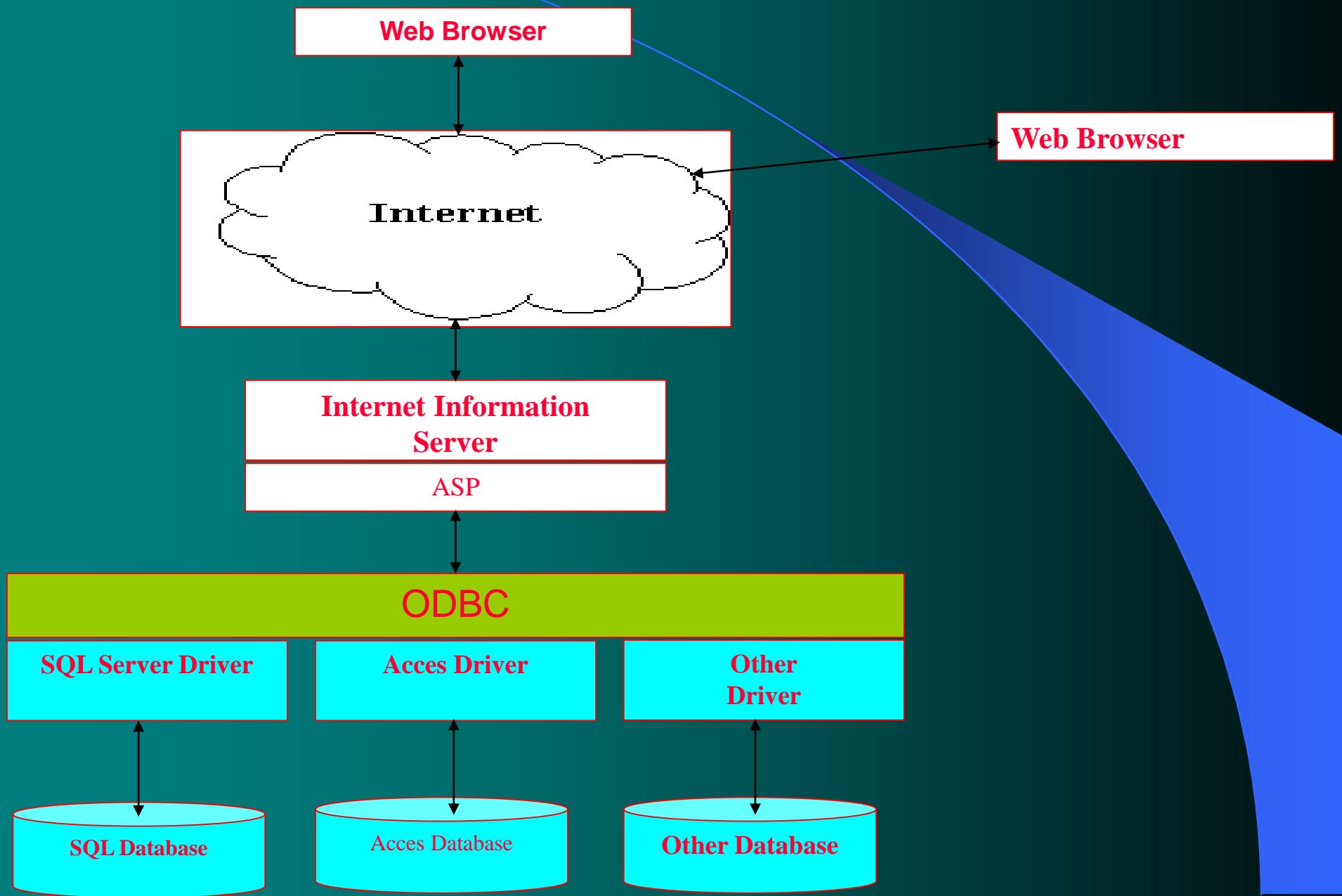
Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

1. Giới thiệu

Như vậy việc sử dụng ASP kết hợp với ODBC để truy cập thông tin trong các cơ sở dữ liệu qua mạng diện rộng có thể được mô tả ngắn gọn qua sơ đồ sau:

Active Server Pages



Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

1. Giới thiệu

- + Từ Web Browser, người sử dụng yêu cầu truy vấn dữ liệu bằng cách gửi các câu lệnh SQL đến Web Server (là IIS), IIS nhận thấy việc truy cập cơ sở dữ liệu cần đến ASP thì sử dụng ASP.
- + ASP sẽ gửi các thông tin cần thiết đến ODBC và ODBC sẽ lựa chọn trình điều khiển thích hợp để truy cập cơ sở dữ liệu.
- + Sau khi xử lý xong, kết quả trả ngược lại cho người sử dụng cũng ở trên Web Browser.

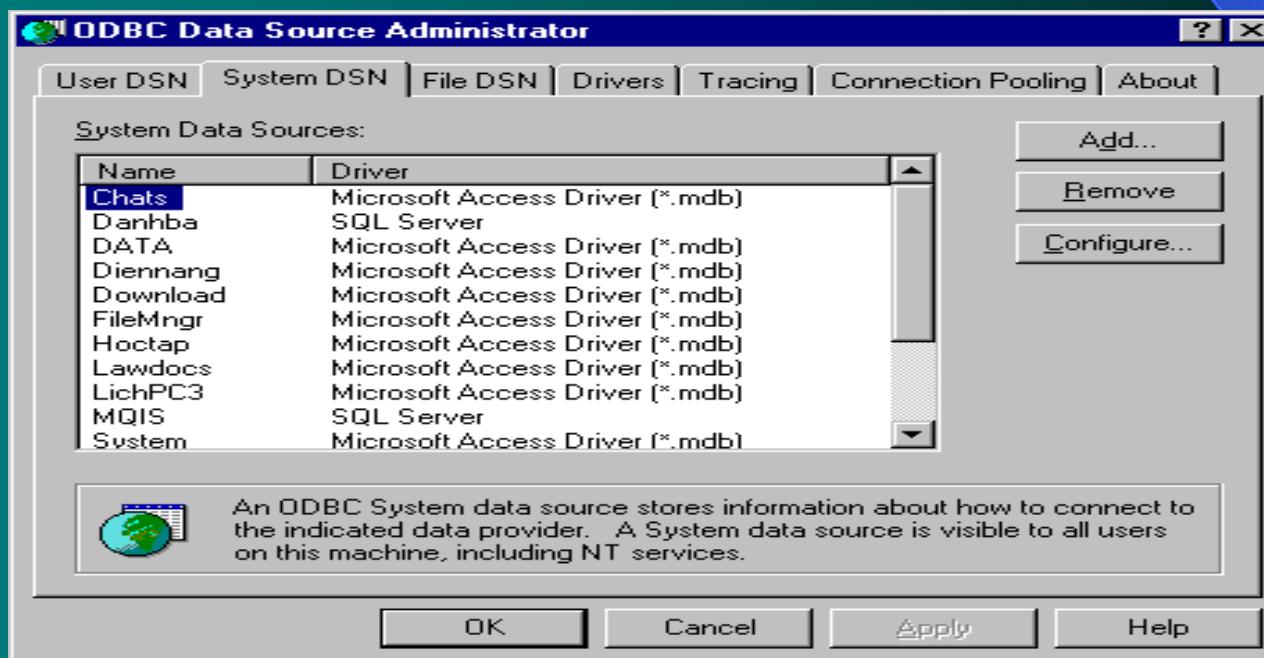
Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

2. Đăng ký nguồn dữ liệu với ODBC

Để đăng ký dữ liệu với ODBC tiến hành theo các bước sau:

- Từ menu Start-> chọn Settings -> chọn Control Panel
- Kích đồi biểu tượng ODBC (32 bit), xuất hiện màn hình như sau:



Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

2. Đăng ký nguồn dữ liệu với ODBC

Trong phần System Data Source liệt kê tên các cơ sở dữ liệu đã được đăng ký. Trong phần này gồm có hai phần:

+ Name: tên đã đăng ký (chú ý là tên này không phải là tên của cơ sở dữ liệu, mà đây chỉ là tên đại diện cho cơ sở dữ liệu)

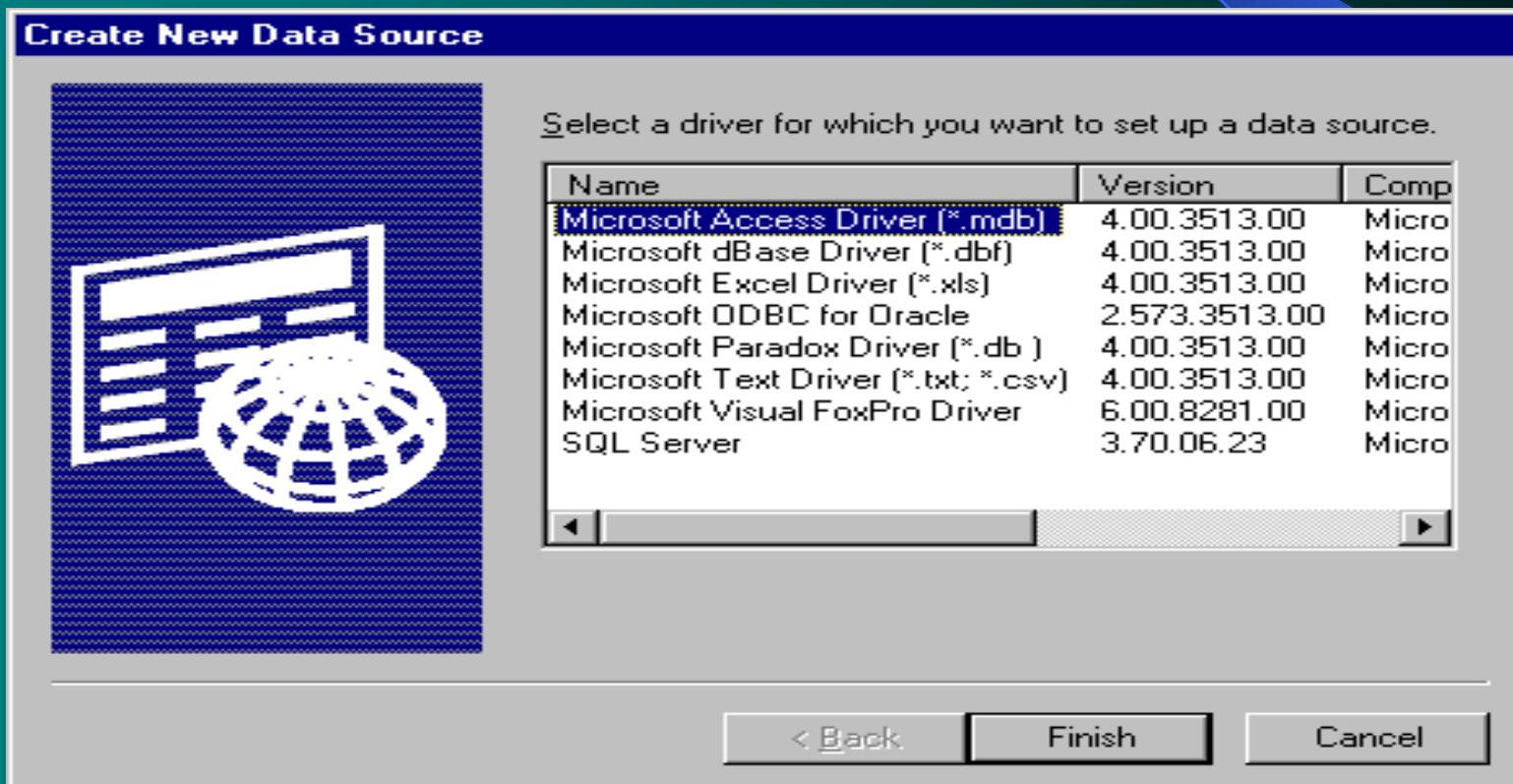
+ Driver: Trình điều khiển để truy cập từng loại cơ sở dữ liệu cụ thể (như Access, SQL Server, ...)

Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

2. Đăng ký nguồn dữ liệu với ODBC

c. Để đăng ký nguồn dữ liệu mới, kích nút Add, xuất hiện cửa sổ sau:



Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

2. Đăng ký nguồn dữ liệu với ODBC

Trong cửa sổ này cho phép bạn chọn lựa trình điều khiển của từng loại cơ sở dữ liệu mà bạn sử dụng.

- + Nếu Cơ sở dữ liệu là SQL thì chọn SQL Server
- + Nếu Cơ sở dữ liệu là Access thì chọn Microsoft Access Driver,...

Giả sử chọn Microsoft Access Driver, và kích nút Finish

Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

2. Đăng ký nguồn dữ liệu với ODBC

d. Sau khi chọn Driver và kích nút Finish, xuất hiện cửa sổ như bên dưới:

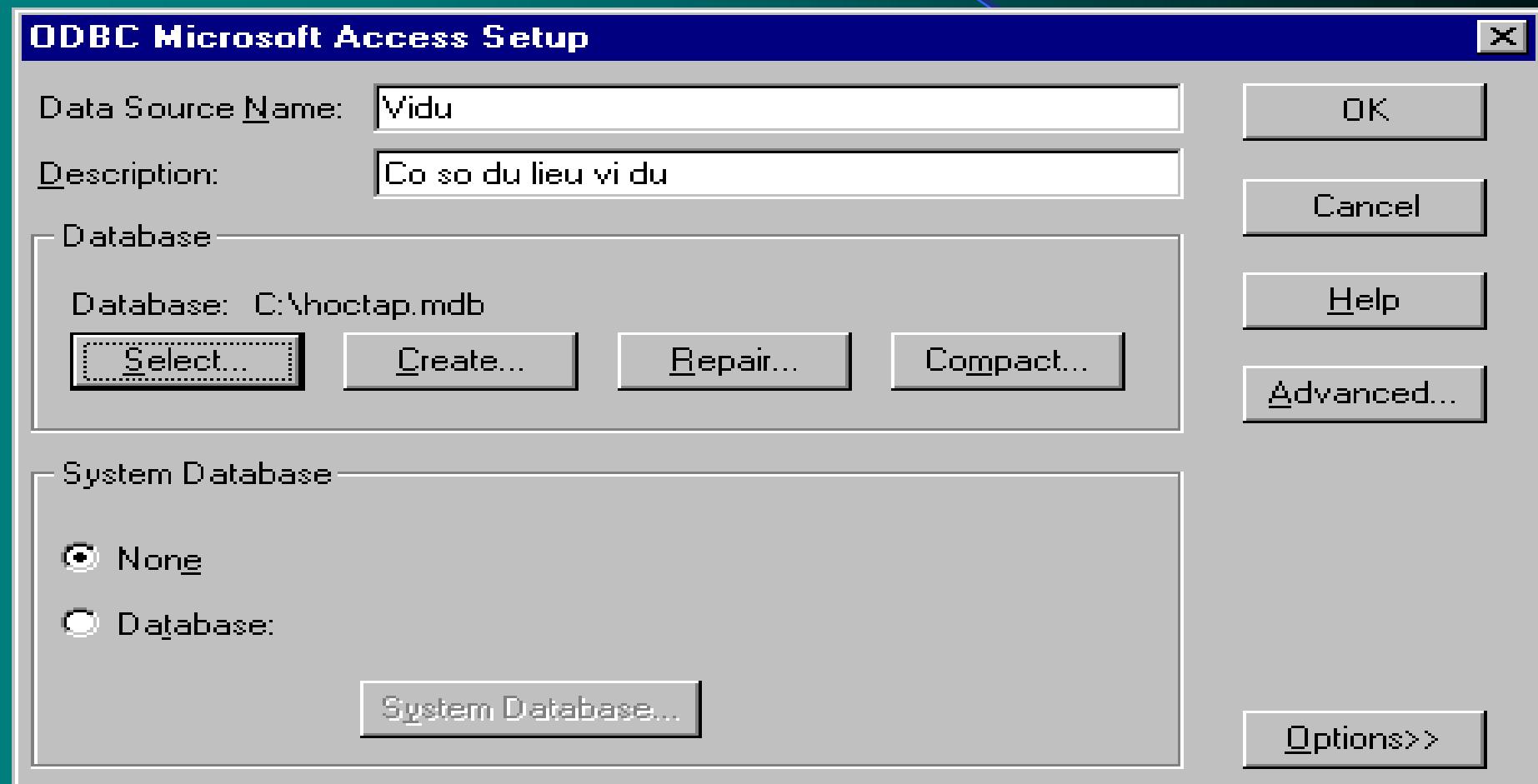
- Data Source Name: nhập vào tên đại diện cho cơ sở dữ liệu (tên này sẽ được dùng để kết nối đến cơ sở dữ liệu trong ASP, ví dụ câu lệnh trong ASP: conn.open "Vidu")
- Description: nhập vào phần mô tả cho cơ sở dữ liệu (phần này có thể bỏ trống)
- Database: tên của cơ sở dữ liệu cần truy cập, để chọn file Cơ sở dữ liệu bạn kích nút Select...

Sau khi chọn Cơ sở dữ liệu bạn kích nút OK, và tiếp tục kích OK để kết thúc việc đăng ký cơ sở dữ liệu với ODBC.

Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

2. Đăng ký nguồn dữ liệu với ODBC



Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

3. Giới thiệu về IIS - Internet Information Server.

a. IIS là gì ?

Microsoft Internet Information Server là một ứng dụng server chuyển giao thông tin bằng việc sử dụng giao thức chuyển đổi siêu văn bản HTTP

b. IIS có thể làm được gì ?

- Xuất bản một Home page lên Internet.
- Tạo các giao dịch thương mại điện tử trên Internet(Quá trình giao dịch, đặt hàng...)
- Cho phép người dùng từ xa có thể truy xuất Cơ sở dữ liệu (DataBase Remote Access)

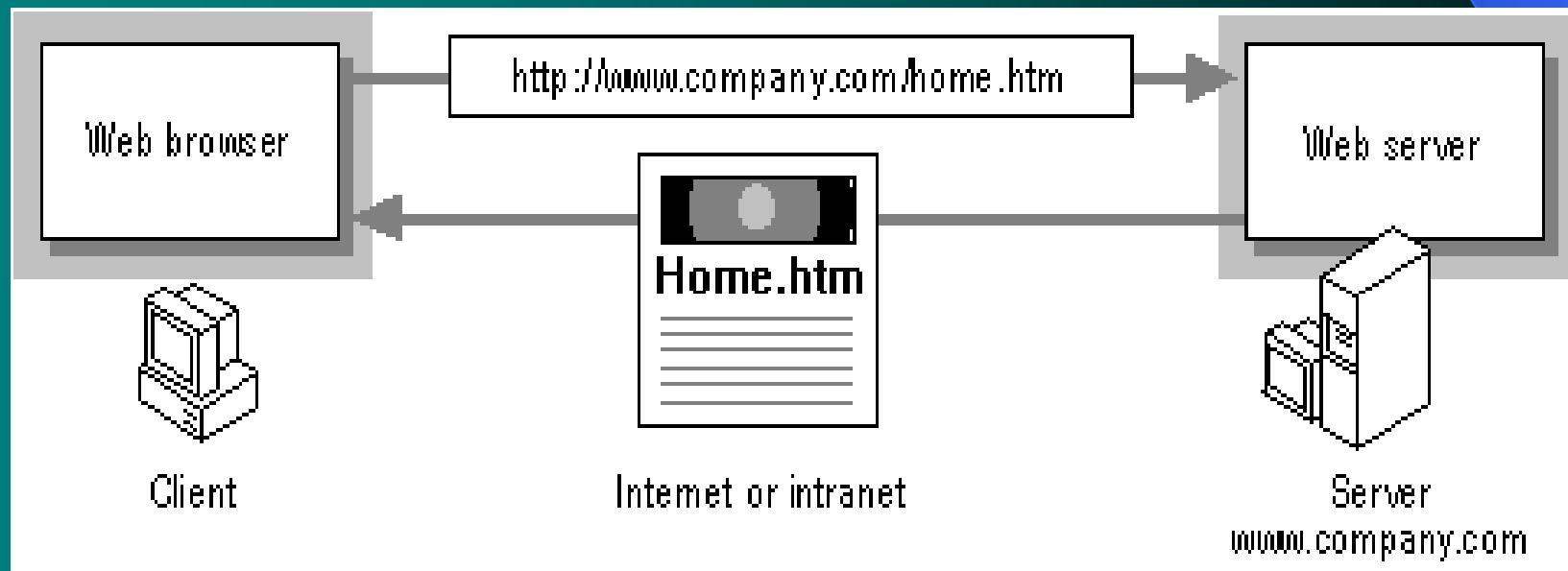
Active Server Pages

IV. Đăng ký nguồn dữ liệu với ODBC

3. Giới thiệu về IIS - Internet Information Server.

c. ISS hoạt động như thế nào ?

Web về cơ bản thực sự là một hệ thống các yêu cầu (Request) và các đáp ứng (Response). IIS phản hồi lại các yêu cầu đòi thông tin của Web Browser. IIS lắng nghe các yêu cầu đó từ phía Users trên một mạng sử dụng WWW.



jspEC Project

Java Server Pages

Nội Dung

- * Giới thiệu
- * Cơ chế hoạt động của JSP
- * Cách xây dựng Web bằng JSP
- * Các cú pháp cơ bản của JSP
- * Nhúng và chuyển hướng giữa các trang
- * Các đối tượng mặc định của JSP
- * Phương thức truy xuất cơ sở dữ liệu

Java Server Pages

I. Giới Thiệu

JSP là cạnh tranh của Sun đối với công nghệ ASP của Microsoft.

ASP cũng là các trang Web thực hiện cơ chế diễn dịch thay các lệnh Java là các lệnh Visual Basic. ASP không có cơ chế diễn dịch ra mã nhị phân như JSP.

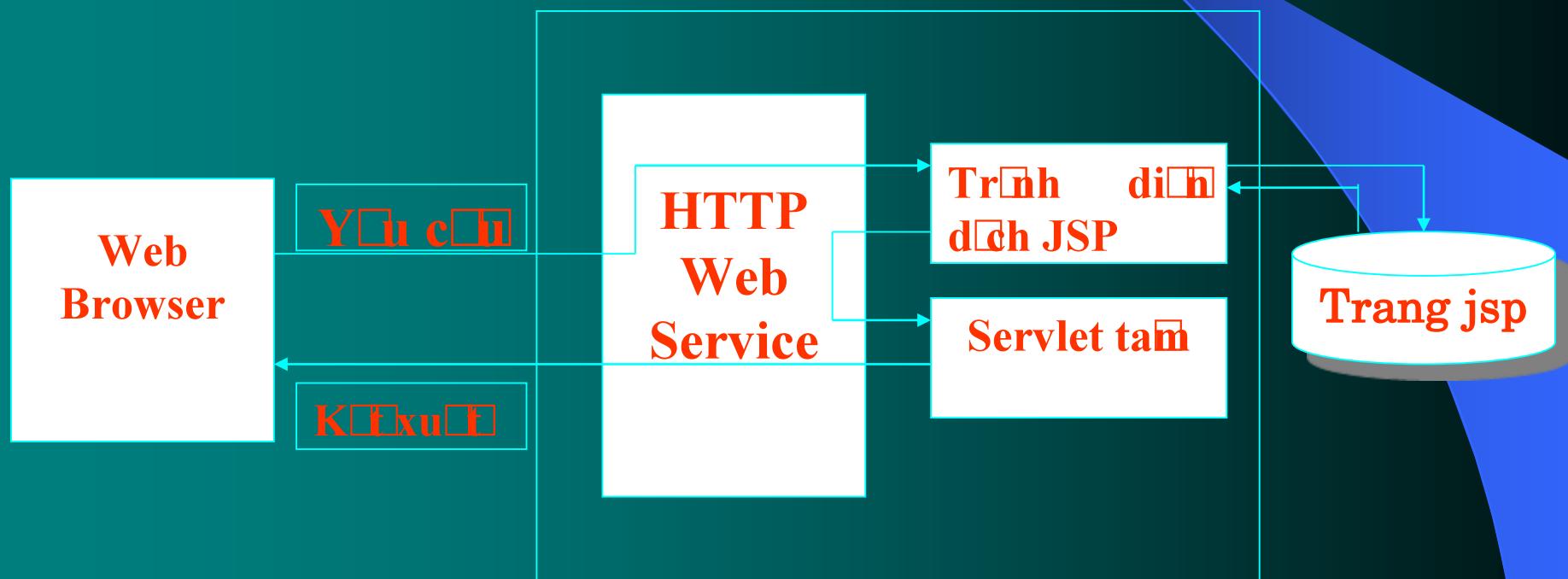
JSP (Java Server Page) là cách đơn giản để người dùng tiếp cận được hóng lập trình Web phía máy chủ hiệu quả và nhanh hơn.

Nếu như đối với Servlet bạn phải viết mã Java và biên dịch bằng tay trước khi đưa vào sử dụng với trình chủ Web Server thì JSP không cần điều này

Java Server Pages

II. Cơ Chế Hoạt Động Của JSP

Cơ chế hoạt động của trang JSP được minh họa như hình sau :



Java Web Server

Java Server Pages

II. Cơ Chế Hoạt Động Của JSP :

- + Khi bạn yêu cầu một trang JSP, trình chủ sẽ đọc trang JSP từ đĩa cứng
- + Bộ diễn dịch JSP sẽ diễn dịch mã lệnh Java chứa trong trang JSP thành một servlet.
- + Sau đó trình chủ Java Web Server sẽ triệu gọi servlet trả kết xuất thuần HTML về cho trình khách

Java Server Pages

III. Cách Xây Dựng Web Bằng JSP

Một trang JSP với phần mở rộng là .jsp như là một trang HTML, có thể chèn thêm các câu lệnh Java bằng cặp

`<%. . . %>`

hoặc: `<jsp:scriptlet>... </jsp:scriptlet>`.

Giữa cặp dấu này là các câu lệnh của ngôn ngữ Java nằm trong phần thân (`<Body> . . . </Body>`) của tài liệu, hoặc phần đầu tiên trước cả tag `<HTML>` của tài liệu.

Java Server Pages

III. Cách Xây Dựng Web Bằng JSP

Mô tả cấu trúc của một trang JSP

```
<%----- Các câu lệnh Java ở đây -----%>  
< HTML>----- Các thành phần định nghĩa khác -----  
<BODY>  
<%----- Các câu lệnh Java ở đây -----%>  
----- Các định dạng khác của HTML -----  
< /BODY>  
</HTML>
```

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

Tương tự thẻ HTML, thẻ lệnh JSP cũng bao gồm thẻ mở và thẻ đóng. Mỗi thẻ có các thuộc tính qui định cách sử dụng thẻ.

1. Thẻ `<jsp:scriptlet>` hay `<% %>`

Với thẻ này bạn có thể đặt các đoạn mã lệnh Java ở giữa cặp thẻ tương tự một chương trình Java thông thường. Bên trong mã Java nếu muốn kết xuất dữ liệu HTML bạn gọi phương thức `println()` của đối tượng `out`.

`<jsp:scriptlet>`

mã lệnh Java

`</jsp:scriptlet>`

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

1. Thẻ `<jsp:scriptlet>` hay `<% %>`

Khi gặp thẻ này, bộ diễn dịch JSP sẽ biết được lệnh cần thực hiện tiếp theo là mã lệnh Java. Bạn hoàn toàn sử dụng mọi cú pháp của Java ở đây, kể cả lệnh chú thích thông thường. Nếu muốn viết tắt thì có thể sử dụng cặp thẻ `<% %>` thay thế.

`<%`

mã lệnh Java

`%>`

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

2. Hiển thị kết xuất bằng cú pháp `<%=>`

Thay vì sử dụng cú pháp `<% %>` để diễn đạt một khối gồm nhiều lệnh, ta có thể sử dụng cú pháp `<%=>` chỉ để hiển thị kết xuất của một giá trị biến hay hàm nào đó.

Riêng biểu thức nằm trong thẻ `<%=>` không được tận cùng bằng dấu chấm phẩy (`;`) như các lệnh hay khai báo biến của ngôn ngữ Java.

Ví dụ:

Welcome `<%=>username%>`

username : là tên biến do người dùng đăng nhập.

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

3. Chèn chú thích vào trang JSP :

Cũng như Java, JSP cho phép bạn dùng cú pháp `//` để chú thích một dòng mã lệnh trong khi cú pháp `/* */` áp dụng cho nhiều dòng. Các dòng chú thích sẽ bỏ qua khi trình chủ diễn dịch trang JSP.

```
<%  
// Lấy biến dữ liệu mang tên username từ đối tượng session  
String username=session.getAttribute("username");  
%>
```

JSP còn cung cấp cho bạn cú pháp chú thích `<%-- --%>`. Tất cả các khối lệnh Java và HTML nằm giữa hai dấu chú thích này sẽ được trình biên dịch trang bỏ qua không quan tâm đến.

Ví dụ : `<%--`
`out.println("You will never see this line");`
`--%>`

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

4. Thẻ khai báo phương thức và biến bằng <%! %>

Ta có thể dùng thẻ <%! %> để khai báo một biến hoặc một phương thức dùng cho toàn trang jsp. Biến khai báo phải theo cú pháp của ngôn ngữ Java. Phương thức và biến sau đó có thể được gọi bất kỳ nơi đâu trong trang JSP.

5. Thẻ chỉ dẫn biên dịch trang <%@ page%>

Thẻ <%@ page%> chỉ dẫn một số tính chất biên dịch áp dụng cho toàn trang jsp. Ta có thể sử dụng thẻ này để khai báo các thư viện import của Java, chỉ định tùy chọn trang jsp có cần giữ trên cache bộ nhớ của trình chủ để tăng tốc hay không.

Ví dụ:

```
<%@ page import="java.sql.*" %>
```

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

6. Nhúng Và Chuyển Hướng Giữa Các Trang

Trong quá trình xử lý và chuyển dữ liệu cho form bạn có thể nhúng và triệu gọi những trang JSP khác với trang hiện hành.

Có thể đưa một trang HTML hoặc JSP vào trang JSP hiện tại để hỗ trợ thêm thư viện hay tính năng bổ sung nào đó.

Có thể ghi nhận được dữ liệu submit từ trình khách, trang JSP nhận được dữ liệu có thể chuyển hướng hoặc triệu gọi đến trang JSP khác.

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

6.1. Nhúng file vào trang với chỉ thị include file :

Ta có thể nhúng một trang jsp hay trang html vào trang jsp hiện hành bằng thẻ `<%@ include file %>`.

Thẻ này tương tự như chỉ dẫn `#include` trong ngôn ngữ C. Cú pháp đầy đủ của thẻ này là :

`<%@ include file="URL or Filepath"%>`

6.2. Sử dụng thẻ `<jsp:include>`

Chỉ thị `<@ include %>` chỉ dùng để nhúng các mã nguồn tĩnh. Nếu muốn nhúng kết quả kết xuất từ các trang jsp, servlet hay html khác vào trang hiện hành bạn sử dụng thẻ `<jsp:include>` với cú pháp như sau :

`<jsp:include page="filename" flush="true" />`

Lưu ý : `<jsp: include>` đưa kết quả kết xuất của trang được nhúng vào trang hiện hành, trong khi `<%@ include %>` dùng để đưa mã nguồn của trang được nhúng vào trang hiện hành.

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

6.3 Chuyển tiếp đến trang khác với thẻ `<jsp:forward>`

Ta sử dụng thẻ `<jsp:forward>` để triệu gọi và chuyển dữ liệu cho trang jsp khác xử lý.

Ví dụ : Trang Page1.jsp chỉ hiển thị thông tin để người dùng nhập vào username và password

`<html>`

`<form method=post action=page1.jsp>`

username : `<input type=text name="username" >`

password : `<input type=password name="password" >`

`<input type=submit value="Login" >`

`</form>`

`</html>`

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

6.3 Chuyển tiếp đến trang khác với thẻ `<jsp:forward>`

Trang Page2.jsp kiểm tra nếu username khác rỗng sẽ chuyển tiếp quyền xử lý file Page3.jsp, ngược lại nếu username bằng rỗng sẽ đưa ra thông báo yêu cầu người dùng nhập vào.

```
<% if (!request.getParameter("username").equals(" ")) {  
    %>  
<jsp:forward page="Page3.jsp" />  
<%} else { out.println(" Please input username and password ");}  
%>  
<html>  
    Your username : <%=request.getParameter("username")%>  
<br>  
    Your password : <%=request.getParameter("password")%>;  
<br>  
</html>
```

Java Server Pages

IV. Các Cú Pháp Cơ Bản Của JSP

6.4. Chuyển hướng sang trang mới với sendRedirect()

Ngoài việc chuyển tiếp bạn có thể sử dụng phương thức sendRedirect() để chuyển hướng và triệu gọi trang JSP khác xử lý.

Sự khác nhau giữa chuyển tiếp (forward) và chuyển hướng (redirect) đó là lệnh sendRedirect() chỉ thực hiện triệu gọi trang mà không chuyển tham số của trang hiện hành cho trang chuyển hướng.

Ví dụ : Từ Page2.jsp bạn có thể chuyển hướng đến trang Page3.jsp như sau :

```
<%
```

```
response.sendRedirect("Page3.jsp");  
%>
```

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

Trình diễn dịch JSP cho phép ta sử dụng một số đối tượng đã khai báo trước. Điều này sẽ giúp ta viết mã lệnh trong trang jsp nhanh hơn servlet.

1. **Đối tượng out** : Xuất phát từ lớp `java.io.PrintWriter`. Bạn có thể sử dụng đối tượng này để định dạng kết xuất gửi về máy khách.

Ví dụ: `<% out.println("Result "+7*3); %>`

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

2. Đối tượng Request :

Xuất phát từ lớp HttpServletRequest, đối tượng này giúp bạn lấy về các tham số hay dữ liệu do trình khách chuyển lên.

Một trong những ứng dụng thông thường nhất của đối tượng request là nắm giữ các tham số. Bạn có thể thấy điều này bằng cách gọi phương thức getParametter() của request.

Phương thức này kế thừa từ lớp cha là javax.servlet.ServletRequest. Phương thức này nhận tên tham số và trả về giá trị chuỗi tương ứng với tên của tham số đó.

Ví dụ: Lấy về các thông tin đăng nhập từ file Bc_td1.jsp

```
String USERNAME=request.getParameter("username");
String PASSWORD=request.getParameter("password");
```

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

3. Đối tượng Session :

Xuất phát từ lớp HttpSession. Bạn sử dụng đối tượng session để theo dõi kết nối và lưu vết một phiên làm việc giữa trình khách và trình chủ.

Đối tượng session tham chiếu đến đối tượng javax.servlet.http.HttpSession. Đối tượng session được khởi tạo bằng cách gọi phương thức pageContext.getSession() để tạo ra trang servlet.

```
session = pageContext.getSession();
```

Khi cần một biến nào đó có giá trị toàn cục từ khi mở cho đến khi kết thúc trình duyệt đó, bạn nên nghĩ đến session. Đối tượng session sẽ thực hiện việc đó giúp bạn. Đối tượng session tạo biến cục bộ cho phép lưu một giá trị nào đó từ trang JSP này sang trang JSP khác.

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

3. Đối tượng Session :

Ví dụ : Để lưu thông tin về phiên làm việc vào session ta thực hiện như sau:

```
session.putValue("NGAY", NGAY);
```

```
session.putValue("THANG", THANG);
```

```
session.putValue("NAM", NAM);
```

Sau khi ghi thông tin vào session, khi cần lấy về giá trị trong session ta thực hiện như sau :

```
String NGAY=(String) session.getValue("NGAY");
```

```
String THANG=(String) session.getValue("THANG");
```

```
String NAM=(String) session.getValue("NAM");
```

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

4. Đối tượng Response :

Đối tượng response cho phép phản hồi thông tin xử lý từ trình chủ trở về trình duyệt. Thực ra đối tượng response thường gọi phương thức response.getWriter() để thực hiện kết xuất. Các phương thức mà đối tượng response thường được liệt kê như sau :

Liên kết trực tiếp đến trang Bc_td2.jsp

```
response.sendRedirect("Bc_td2.jsp");
```

Định kết xuất ra HTML hay text

```
response.setContentType("text/html");
```

Lấy dữ liệu trong vùng đệm nhờ OutputStream();

```
response.getOutputStream();
```

Java Server Pages

V. Các Đối Tượng Mặc Định Của Trang JSP

5. Đối tượng Exception :

Đối tượng exception chỉ tồn tại trong trang xử lý lỗi (errorPage). Nó dùng để tham chiếu đến nguyên nhân gây ra lỗi mà trang error có liên quan.

Java Server Pages

VI. Truy Xuất CSDL Bằng JSP :

1. JDBC và trình điều khiển truy xuất cơ sở dữ liệu

Công việc lập trình mà JSP thường thực hiện nhất đó là lưu trữ thông tin và truy xuất cơ sở dữ liệu (database) cung cấp thông tin cho trình khách.

Việc truy xuất cơ sở dữ liệu trong JSP dựa vào trình truy xuất JDBC mà Java hỗ trợ. JDBC là các trình đạo diễn cho phép bạn truy cập vào cơ sở dữ liệu của các hệ như Access, MS SQL Server, Oracle,... bằng ngôn ngữ truy vấn SQL. Có 4 kiểu JDBC sau:

- + *JDBC sử dụng cầu nối ODBC*
- + *JDBC kết nối trực tiếp với các trình điều khiển cơ sở dữ liệu.*
- + *JDBC kết nối thông qua các ứng dụng mạng trung gian.*
- + *JDBC kết nối thông qua các trình điều khiển đặc thù ở xa*

Java Server Pages

VI. Truy Xuất CSDL Bằng JSP :

1. JDBC và trình điều khiển truy xuất cơ sở dữ liệu

+ Kiểu 1: JDBC sử dụng cầu nối ODBC

Nếu bạn đã từng làm việc với các hệ cơ sở dữ liệu trên windows bạn sẽ biết đến cách kết nối ODBC (Open Database Connectivity).

ODBC là cách kết nối tổng quát nhất đến mọi hệ cơ sở dữ liệu thông qua trình quản lý ODBC của Windows. Mỗi hệ cơ sở dữ liệu cung cấp một trình điều khiển (driver) có khả năng đọc hiểu được cơ sở dữ liệu của chúng.

Trình điều khiển này sẽ được đăng ký với bộ quản lý ODBC. Java cung cấp cho bạn khả năng sử dụng JDBC để gọi ODBC và bằng cách này bạn có thể kiểm soát rất nhiều hệ quản trị cơ sở dữ liệu khác nhau.

Java Server Pages

VI. Truy Xuất CSDL Bằng JSP :

1. JDBC và trình điều khiển truy xuất cơ sở dữ liệu

+ Kiểu 2: JDBC kết nối trực tiếp với các trình điều khiển cơ sở dữ liệu.

Khi Java phát triển, các nhà cung cấp hỗ trợ các trình điều khiển JDBC kết nối trực tiếp với các trình điều khiển đặc thù của mỗi hệ cơ sở dữ liệu (native database driver) mà không cần phải qua trung gian ODBC của windows nữa. Các kết nối này nhanh và hiệu quả hơn kiểu 1 JDBC-ODBC

Java Server Pages

VI. Truy Xuất CSDL Bằng JSP :

1. JDBC và trình điều khiển truy xuất cơ sở dữ liệu

+ Kiểu 3: JDBC kết nối thông qua các ứng dụng mạng trung gian.

Ngày nay, các ứng dụng đều gắn liền với Internet. Theo mô hình lập trình phân tán, ứng dụng trung gian (middle-ware) trên máy chủ sẽ chịu trách nhiệm kết nối với cơ sở dữ liệu.

Nhiệm vụ của trình khách là sử dụng những trình điều khiển JDBC gọn nhẹ có khả năng giao tiếp với ứng dụng trung gian qua Internet để truy vấn dữ liệu.

Java Server Pages

VI. Truy Xuất CSDL Bằng JSP :

1. JDBC và trình điều khiển truy xuất cơ sở dữ liệu

+ Kiểu 4: JDBC kết nối thông qua các trình điều khiển đặc thù ở xa.

Mô hình kết nối theo kiểu 3 do phải thông qua ứng dụng trung gian (middle-ware) nên có thể ảnh hưởng đến tốc độ truy xuất cơ sở dữ liệu phía máy khách. Java đã ra kết nối JDBC kiểu 4 cho phép trình khách sử dụng các trình điều khiển gọn nhẹ kết nối trực tiếp vào trình điều khiển cơ sở dữ liệu đặc thù (thuần java) trên máy chủ ở xa thông qua mạng Internet.

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

2. Phương Thức Truy Xuất CSDL Bằng Jsp

2.1 Thực hiện một kết nối với CSDL :

Để kết nối với CSDL, JDBC đòi hỏi hai yếu tố là **trình điều khiển** (tương ứng với các kiểu kết nối) và **thông tin để kết nối** (địa chỉ máy chủ, tên tài khoản, mật khẩu,...).

+ *Nạp trình điều khiển*: Ta sử dụng trình điều khiển miễn phí do Sun cung cấp mang tên JdbcOdbcDriver. Ta khai báo tên lớp này như sau :

```
String drivername="sun.jdbc.odbc.JdbcOdbcDriver";
```

Tiếp đến ta gọi phương thức tĩnh forName của lớp Class để nạp và khởi tạo trình điều khiển :

```
Class.forName("drivername").newInstance();
```

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

2. Phương Thức Truy Xuất CSDL Bằng Jsp

2.1 Thực hiện một kết nối với CSDL :

+ *Tạo thông tin kết nối* : Lúc này ta đã hoàn toàn có khả năng sử dụng trình điều khiển JDBC-ODBC để truy xuất dữ liệu. Tuy nhiên trình điều khiển cần biết thêm những thông tin cụ thể như tài khoản đăng nhập, thông tin kết nối,...

Ta đặt những thông tin này trong chuỗi kết nối như sau :

```
String url="jdbc:odbc:tên CSDL";
String username="";
String password="";
```

ở đây chuỗi jdbc:odbc: là bắt buộc với trình điều khiển JDBC-ODBC. Tiếp sau chuỗi này là thông tin kết nối cụ thể đến CSDL do ODBC qui định.

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

2. Phương Thức Truy Xuất CSDL Bằng Jsp

2.1 Thực hiện một kết nối với CSDL :

+ *Tạo thông tin kết nối :*

Tiếp tục ta tạo một đối tượng Connection bằng cách gọi phương thức getConnection() của lớp DriverManager để yêu cầu trình điều khiển nạp bởi Class.forName() trước đó tiếp nhận thông tin và thực thi kết nối như sau:

```
Connection conn = null;  
conn = DriverManager.getConnection(url, "username", "password ");
```

url : chuỗi nêu lên đặc điểm của CSDL có dạng
jdbc:subprotocol:subname

subprotocol : giao thức con tương ứng với loại CSDL
subname : tên của CSDL

username : tên người dùng khi đăng nhập vào CSDL
password : mật khẩu người dùng khi đăng nhập CSDL

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

2. Phương Thức Truy Xuất CSDL Bằng Jsp

2.2 Thực hiện các lệnh :

Các chương trình JSP chỉ thực hiện được các lệnh trên CSDL thông qua đối tượng Statement. Đối tượng này được dùng để thực thi mọi câu lệnh sql.

Ta tạo đối tượng Statement từ kết nối *conn*

```
Statement stmt =null;  
stmt = conn.createStatement();
```

Các câu lệnh sql có thể được thực hiện tức thì thông qua đối tượng Statement, có thể là một câu lệnh biên dịch trước (đối tượng PreparedStatement) hay có thể là một lệnh gọi các thủ tục cài sẵn (store procedure) trong CSDL (đối tượng CallableStatement).

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

2. Phương Thức Truy Xuất CSDL Bằng Jsp

2.2 Thực hiện các lệnh :

rs=stmt.executeQuery(sql);

hoặc : rs=stmt.executeUpdate(sql);

Các câu lệnh sql có thể thực hiện thông qua :

 + phương thức executeQuery() -nếu kết quả là một đối tượng ResultSet

 + phương thức executeUpdate()-nếu kết quả là một số nguyên cho biết tổng số các dòng dữ liệu chịu tác dụng của câu lệnh vừa thực hiện.

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

3. Đoạn lệnh dùng JSP truy cập dữ liệu trên Server

// Nạp trình điều khiển JDBC

- String drivername="sun.jdbc.odbc.JdbcOdbcDriver";
- Class.forName(drivername).newInstance();

// Thông tin kết nối JDBC-ODBC

- String url="jdbc:odbc:Database_Name";
- String username="";
- String password="";

Java Server Pages

VI. Phương Thức Truy Xuất CSDL Bằng JSP :

3. Đoạn lệnh dùng JSP truy cập dữ liệu trên server

- Connection con=null;
// Thực hiện kết nối
- con=DriverManager.getConnection(url, username, password);
- Statement stmt=null;
- stmt=con.createStatement();
// Thực thi câu lệnh truy vấn và nhận kết quả trả về
- ResultSet rs=null;
// Tạo câu lệnh truy vấn
- String sql="select * from TABLE_NAME";
- rs=stmt.executeQuery(sql);

NGÔN NGỮ LẬP TRÌNH PHP

I. Giới thiệu về PHP

1. Giới thiệu

- + PHP (*Personal Home Page*) là một kịch bản trên Server (Server Script)
- + Là một môi trường lập trình cung cấp cho việc kết hợp HTML, ngôn ngữ kịch bản như Javascript và các thành phần được viết trong các ngôn ngữ nhằm tạo ra một ứng dụng Internet mạnh mẽ và hoàn chỉnh.
- + PHP là một phần mềm mã nguồn mở (Open Source)

NGÔN NGỮ LẬP TRÌNH PHP

I. Giới thiệu về PHP

2. PHP file là gì?

+ File được tạo với phần mở rộng .PHP. Trong file này chứa các thẻ HTML, các kịch bản Scripting như JavaScript.

+ Khi cần sửa đổi các file PHP ta chỉ cần ghi lại trên server thôi. Vào những lần sau khi trang PHP được gọi, các Script trong file PHP tự động biên dịch lại.

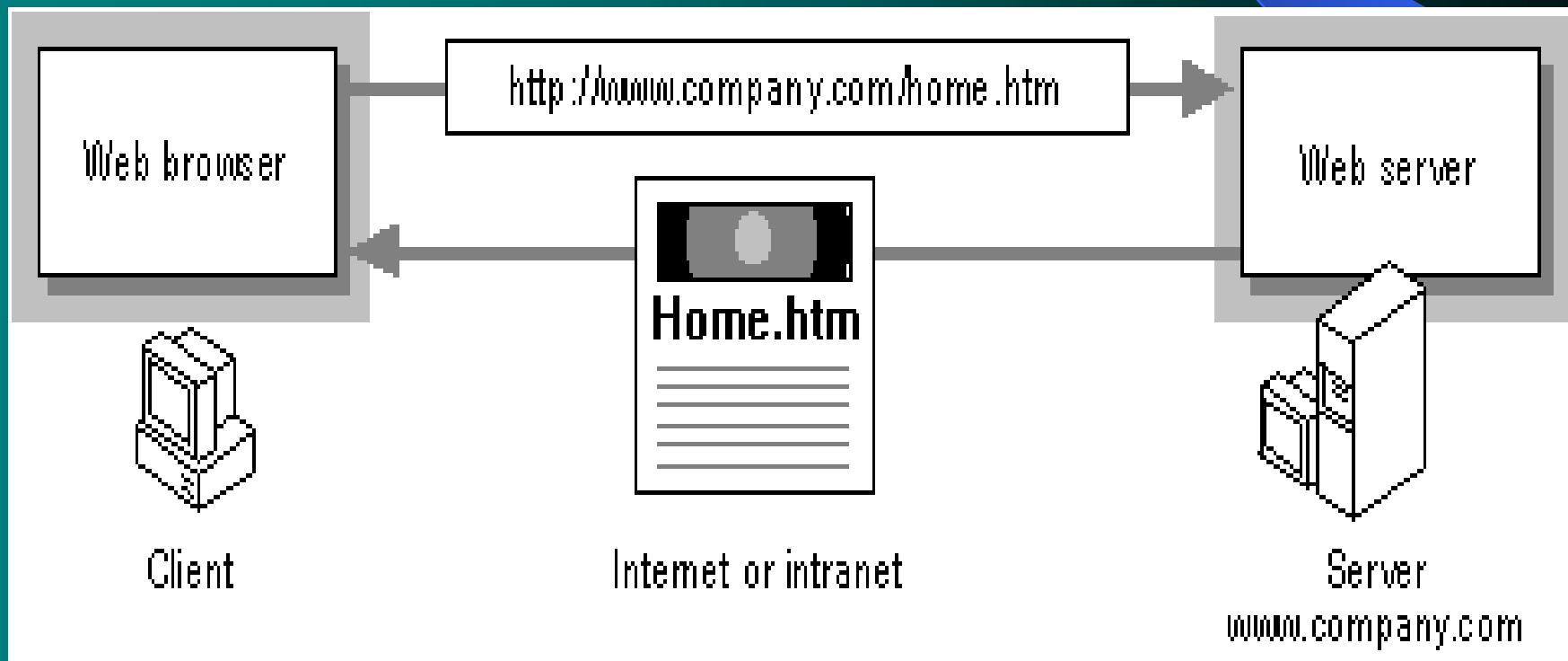
+ Cấu trúc mã gần giống ngôn ngữ lập trình C

NGÔN NGỮ LẬP TRÌNH PHP

I. Giới thiệu về PHP

3. Cách thức làm việc của trang PHP

+ Trang HTML tĩnh:

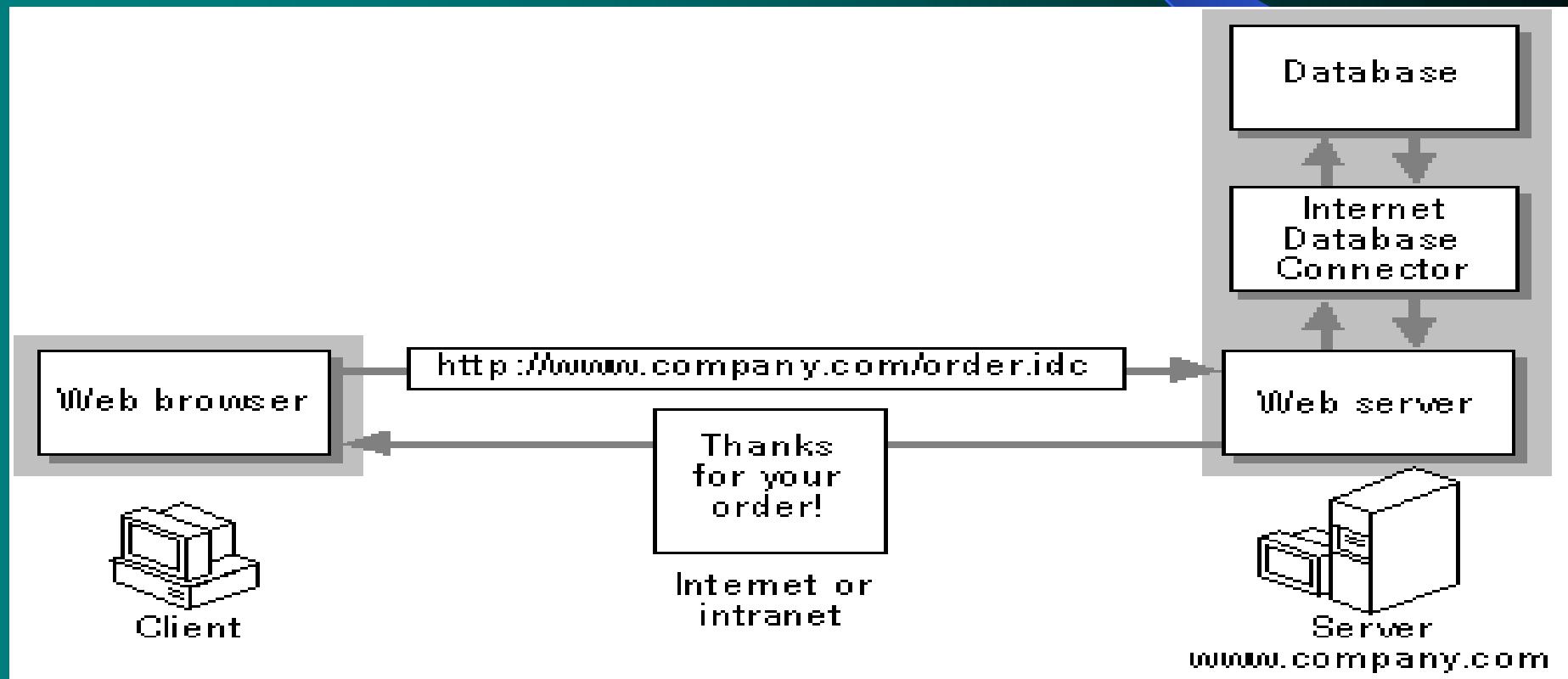


NGÔN NGỮ LẬP TRÌNH PHP

I. Giới thiệu về PHP

3. Cách thức làm việc của trang PHP

+ Trang PHP :

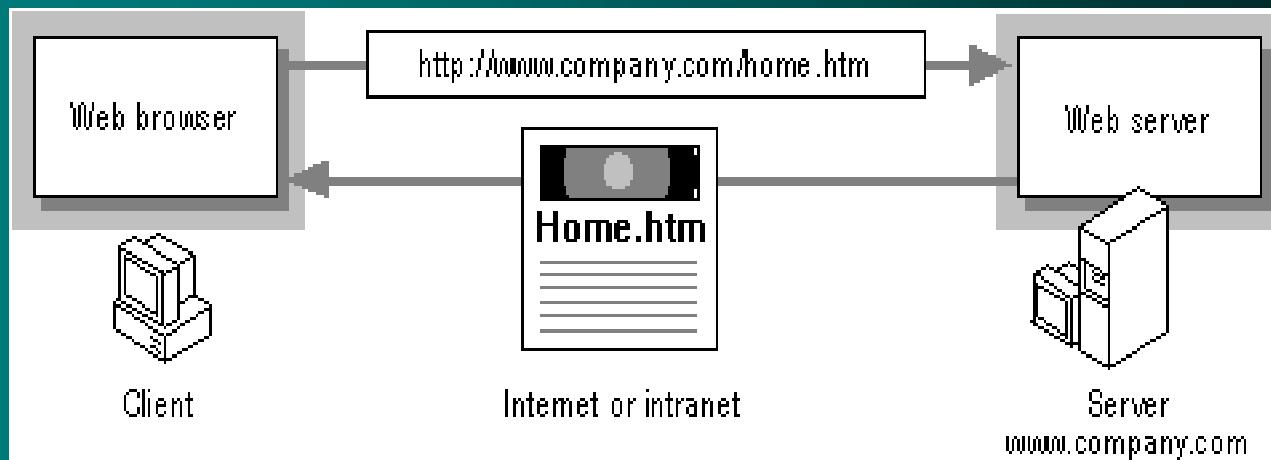


NGÔN NGỮ LẬP TRÌNH PHP

I. Giới thiệu về PHP

4. Web Server Apache thực thi trang PHP

- + Web, về cơ bản thực sự là một hệ thống các yêu cầu (Request) và các đáp ứng (Response).
- + Webserver phản hồi lại các yêu cầu thông tin của Web Browser. Nó lắng nghe các yêu cầu đó từ phía user trên một mạng sử dụng WWW.



NGÔN NGỮ LẬP TRÌNH PHP

II. Cách xây dựng Web bằng PHP

1. Nhúng PHP vào HTML

Có 4 cách để nhúng mã PHP vào trong HTML :

- + Dùng cặp thẻ <? ?>
- + Dùng <?php?>
- + Dùng <Script language = "php" >
.....
</script>
- + Dùng <% %>

NGÔN NGỮ LẬP TRÌNH PHP

II. Cách xây dựng Web bằng PHP

1. Nhúng PHP vào HTML

Ví dụ 1: Lưu file sau lên đĩa với tên vd.php

```
<html>
```

```
  <head>
```

```
    <title>Testing page</title>
```

```
  </head>
```

```
  <body><?php echo "Hello you!"; ?>
```

```
  </body>
```

```
</html>
```

NGÔN NGỮ LẬP TRÌNH PHP

II. Cách xây dựng Web bằng PHP

2. Một số chú ý khi bắt đầu lập trình PHP

a. Lệnh echo :

Lệnh này dùng để xuất chuỗi văn bản về cho trình duyệt.

Ví dụ: Câu lệnh *echo "Hello, world!"*;
trình duyệt sẽ nhận được chuỗi văn bản Hello, world!.

Câu lệnh *echo 1+2*; sẽ trả về cho trình duyệt chuỗi văn bản 3.

Và câu lệnh *echo 5+2, "Hello world!"*; sẽ trả về
trình duyệt chuỗi *7Hello world!*

NGÔN NGỮ LẬP TRÌNH PHP

II. Cách xây dựng Web bằng PHP

2. Một số chú ý khi bắt đầu lập trình PHP

b. Phân cách các lệnh bằng dấu chấm phẩy (;

Tương tự như các ngôn ngữ lập trình khác, một câu lệnh của PHP được kết thúc bằng dấu chấm phẩy (;

Ví dụ: `echo 1+2;` `echo "Hello, world!"`;

c. Chú thích trong chương trình

- Chú thích 1 dòng được bắt đầu bằng // hoặc #, và những gì được ghi từ đó về sau là chú thích.

- Chú thích nhiều dòng được bắt đầu bằng /* và kết thúc bằng */, những gì ở giữa là chú thích.

NGÔN NGỮ LẬP TRÌNH PHP

II. Cách xây dựng Web bằng PHP

2. Một số chú ý khi bắt đầu lập trình PHP

c. *Chú thích trong chương trình*

Ví dụ:

```
<?php
```

```
//Đây là chú thích 1 dòng
```

```
#Đây cũng là chú thích 1 dòng
```

```
/* Đây là chú thích nhiều dòng Đoạn chương trình  
sau sẽ in ra chuỗi abc123 */
```

```
echo "abc123";
```

```
?>
```

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

1. *Kiểu dữ liệu :*

PHP hỗ trợ 8 kiểu dữ liệu chính:

- + 4 kiểu dữ liệu vô hướng: boolean, integer, float (double), string.
- + 2 kiểu dữ liệu tổ hợp: array, object.
- + 2 kiểu dữ liệu đặc biệt : resource, NULL

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

1. *Kiểu dữ liệu :*

+ *Kiểu Boolean* : mang 1 trong 2 giá trị *TRUE* (đúng) hoặc *FALSE* (sai).

Ví dụ: <?php \$a = TRUE; \$b = FALSE;

//phép toán == kiểm tra xem 2 biểu thức

\$c = (7 == 2);

//vì 7 khác 2 nên \$c mang giá trị *FALSE*

\$d = ("abc" == "abc"); // \$d mang giá trị *TRUE* ?>

?>

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

1. *Kiểu dữ liệu :*

+ *Kiểu Integer :*

- *Mang các giá trị số nguyên ..., -2, -1, 0, 1, 2,...*
- *Trên hầu hết các hệ thống, kiểu số nguyên có kích thước 32 bit, mang giá trị từ -2147483647 cho đến 2147483648.*

Ví dụ:

```
<?php $a = 1234; $b = -123; $c = 0123;  
?>
```

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

1. *Kiểu dữ liệu :*

+ *Kiểu Float (Double) :* là kiểu số thực, có thể mang bất cứ giá trị số thực nào. Trên hầu hết các hệ thống, kiểu số thực có kích thước 64 bit.

Ví dụ: <?php

```
$a = 1.234;
```

```
$b = 1.2e3; //= 1.2*10^3 = 1200
```

```
$c = 7E-10; //= 7*(10^-10) = 0.0000000007
```

```
$d = -1.23;
```

```
?>
```

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

1. *Kiểu dữ liệu :*

+ *Kiểu String* : lưu giữ một chuỗi ký tự, mỗi ký tự có kích thước 1 byte. Nội dung string được đặt giữa hai dấu nháy, nháy đơn ('') hoặc nháy kép ("").

Ví dụ: <?php

\$a = 'Đây là chuỗi được đặt giữa dấu nháy đơn';

\$b = "Đây là chuỗi được đặt giữa dấu nháy kép";

\$c = 'Đây là chuỗi được đặt giữa dấu nháy đơn với "vài dấu nháy kép ở giữa"';

\$d = "chuỗi đặt giữa dấu nháy kép với 'vài dấu nháy đơn ''";

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

1. *Kiểu dữ liệu :*

+ Nếu muốn sử dụng dấu nháy đơn trong một chuỗi được bọc bởi dấu nháy đơn, hoặc sử dụng dấu nháy kép đặt giữa chuỗi được bọc bởi dấu nháy kép thì để thêm ký tự \ (ký tự escape) ở phía trước.

Ví dụ: <?php \$a = 'Dấu \'nháy đơn\' ở giữa chuỗi';
//\$a mang giá trị: Dấu 'nháy đơn' ở giữa chuỗi
\$b = "Dấu \"nháy kép\" ở giữa chuỗi";
//\$b mang giá trị: Dấu "nháy kép" ở giữa chuỗi
\$c = "Dùng ký tự \\ ở giữa câu \\ thì sao?";
//\$c mang giá trị: Dùng ký tự \ ở giữa câu \ thì sao?
?>

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

1. *Kiểu dữ liệu :*

- Ngoài ra, nếu để một biến vào giữa một chuỗi được bọc với dấu nháy kép, giá trị của biến sẽ được thay thế vào trong chuỗi.

Ví dụ: <?php \$a = 1;

\$b = 2;

\$c = 3;

\$d = "\$a \$b \$c";

//\$d sẽ mang giá trị là chuỗi "1 2 3"

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

1. *Kiểu dữ liệu :*

+ *Kiểu Array* : *Array là một mảng gồm nhiều phần tử. Array được tạo qua lệnh Array*

Ví dụ:

```
<?php $a = Array(1,2,3);  
      echo $a[0]; //in ra giá trị 1  
      echo $a[2]; //in ra giá trị 3  
      $a[1] = 5; //giờ đây $a = Array(1,5,3)  
?>
```

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

1. *Kiểu dữ liệu :*

- *Mảng còn có thể được tạo thành bởi các cặp (khoá, giá trị).*

Ví dụ:

```
<?php $a = Array(  
    "khoá 1" => "giá trị 1",  
    "khoá 2" => "giá trị 2",  
    "khoá 3" => "giá trị 3");  
echo $a["khoá 1"]; //in ra: giá trị 1  
?>
```

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

1. *Kiểu dữ liệu :*

+ *Kiểu NULL* : Đây là một giá trị đặc biệt, báo cho PHP biết rằng một biến nào đó chưa/không mang giá trị nào

Ví dụ: <?php \$a = 1; // \$a mang giá trị 1

\$a = NULL; // bây giờ \$a không mang giá trị

\$a = 2; // giờ đây \$a mang giá trị 2

// hàm unset sẽ làm cho 1 biến có giá trị là NULL

unset(\$a); // lúc này \$a lại là NULL

?>

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

2. Biến trong PHP :

Một biến trong PHP được bắt đầu bằng ký tự \$ và đi theo ngay sau đó là tên của biến.

+ Biến trong PHP phân biệt chữ hoa và chữ thường. \$Abc và \$abc là hai biến hoàn toàn khác nhau.

+ Tên biến chỉ được bao gồm các ký tự chữ cái (a..z hoặc A...Z), chữ số (0...9) và ký tự gạch dưới (_); nhưng tên biến không được bắt đầu bằng ký tự gạch dưới hoặc chữ số. Ví dụ: \$a: biến có tên là a \$abc123: biến có tên là abc123

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

2. Biến trong PHP :

Ví dụ : Các tên biến sau là không hợp lệ :

`$_abc` Không hợp lệ! bắt đầu bằng ký tự gạch dưới

`$1abc` Không hợp lệ! bắt đầu bằng chữ số

`$nguyễn` Không hợp lệ! tên biến có ký tự đặc biệt (ê)

* *Tầm vực (scope) của biến* : là ngữ cảnh mà ở trong đó biến được định nghĩa.

Ví dụ: `<?php $a = 1; //tầm vực của biến $a từ đây`

`include 'b.php'; //trải dài tới bên trong file b.php`

`//tới cuối file vẫn còn hợp lệ`

`?>`

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

3. Biểu thức :

+ Biểu thức là nền tảng quan trọng của PHP. Hầu như mọi thứ bạn ghi trong file php đều là biểu thức. Ta xét câu lệnh đơn giản sau:

$\$a = 5;$

$\$b = \$a;$

Tuy nhiên biểu thức trong PHP có thể phức tạp hơn

Ví dụ: $\$a = 1; \$b = 2; \$c = 3; \$d = \$a + \$b + \$c;$

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

4. Toán tử :

* *Toán tử kết hợp các giá trị hoặc biểu thức lại với nhau và tạo ra một giá trị mới.*

Ví dụ: Trong biểu thức $1+2$ thì “+” là toán tử kết hợp hai giá trị 1 và 2 lại với nhau tạo ra giá trị mới là 3.

* Các toán tử trong PHP được chia thành 3 nhóm:

- Các toán tử áp dụng trên một giá trị, ví dụ như toán tử ++ hoặc --

- Các toán tử kết hợp hai hoặc nhiều giá trị, ví dụ như toán tử +, -, *, /

- Toán tử ?: dùng để chọn 1 trong 2 giá trị

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

4. Toán tử :

Toán tử	Sử dụng	Tương tự
$+=$	$\$a+=\b	$\$a*=\$a+\$b$
$++$	$\$a++$	$\$a*=\$a+1$
$-=$	$\$a-=\b	$\$a*=\$a-\$b$
$--$	$\$a- -$	$\$a*=\$a-1$
$*=$	$\$a*=\b	$\$a*=\$a*\$b$
$/=$	$\$a/=\b	$\$a*=\$a/\$b$
$\%=$	$\$a\%=\b	$\$a*=\$a\%\$b$
$.=$	$\$a.=\b	$\$a*=\$a.\$b$

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

5. Các phép toán cơ bản :

a. Các phép toán số học :

+, -, *, /, % (module)

b. Các phép toán về chuỗi : Ta sử dụng dấu . để nối 2 chuỗi.

Ví dụ:

```
<?php $a= “Xin chao”; $b=“cac ban”;  
 $a=$a . ”tat ca”;  
 echo $a . $b;  
?>
```

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

5. Các phép toán cơ bản :

c. Các phép toán so sánh :

Phép toán so sánh dùng để so sánh hai giá trị, nếu sử dụng để so sánh hai biểu thức, kết quả trả về là True hay False tùy vào kết quả so sánh.

Toán tử	Sử dụng	Tương tự
<code>==</code>	<code>\$a==\$b</code>	\$a bằng \$b
<code>==</code>	<code>\$a===\$b</code>	\$a bằng và cùng kiểu \$b
<code>!=</code>	<code>\$a!=\$b</code>	\$a khác \$b
<code><></code>	<code>\$a<>\$b</code>	\$a khác \$b
<code><</code>	<code>\$a<\$b</code>	\$a nhỏ hơn \$b
<code><=</code>	<code>\$a<=\$b</code>	\$a nhỏ hơn hoặc bằng \$b
<code>></code>	<code>\$a>\$b</code>	\$a lớn hơn \$b
<code>>=</code>	<code>\$a>=\$b</code>	\$a lớn hơn hoặc bằng \$b

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

5. Các phép toán cơ bản :

d. Phép toán logic

! Phép toán phủ định (not)

&& Phép toán và (and)

|| Phép toán hoặc (or)

and Phép toán và (and)

or Phép toán hoặc (or).

e. Phép toán Error

Trong trường hợp biểu thức hay phép toán báo lỗi, nhưng ta không muốn trình duyệt hiển thị lỗi ta dùng phép toán @ trước biểu thức

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

5. Các phép toán cơ bản :

e. Phép toán Error

Ví dụ:

```
<?php  
$a=10;  
$b=0;  
$c=@($a/$b);  
echo “Gia tri c = “.$c;  
?>
```

NGÔN NGỮ LẬP TRÌNH PHP

III. Lập trình PHP

6. Biến hàm

Là các thư viện hàm trong PHP cho phép thao tác và kiểm tra biến với nhiều cách khác nhau.

Các hàm sau trả về true nếu đúng, ngược lại là sai

- `is_array()` : Kiểm tra biến có phải kiểu mảng không
- `is_double()` : Kiểm tra biến có phải kiểu double không
- `is_float()` : Kiểm tra biến có phải kiểu float không
- `is_integer()` : Kiểm tra biến có phải kiểu int không
- `is_string()` : Kiểm tra biến có phải kiểu chuỗi không
- `is_object()` : Kiểm tra biến có phải kiểu đối tượng không

CẤU TRÚC ĐIỀU KHIỂN

I. Cấu trúc lựa chọn :

1. Câu lệnh *if*

Cú pháp đơn giản nhất của câu lệnh if có dạng như sau:

if (biểu thức) câu lệnh;

Câu lệnh if trên được diễn giải như sau: nếu biểu thức trả về giá trị TRUE (hoặc tương đương với TRUE sau khi chuyển đổi) thì câu lệnh sẽ được thực thi; ngược lại (khi biểu thức trả về giá trị FALSE) thì bỏ qua không thực thi câu lệnh nữa.

Cú pháp nâng cao của câu lệnh if có dạng như sau:

```
if ( biểu thức ) câu lệnh 1;  
else    câu lệnh 2;
```

Câu lệnh if trên được diễn giải như sau: nếu **biểu thức** trả về giá trị TRUE thì **câu lệnh 1** sẽ được thi hành, ngược lại thì **câu lệnh 2** sẽ được thi hành.

Các lệnh if có thể được lồng vào nhau để tạo ra câu lệnh if phức tạp hơn:

```
if( biểu thức 1 )
    if( biểu thức 2 ) câu lệnh 1;
    else câu lệnh 2;
else câu lệnh 3;
```

Nếu biểu thức 1 trả về giá trị FALSE thì câu lệnh 3 sẽ được thực hiện, ngược lại xét tiếp biểu thức 2: nếu biểu thức 2 trả về giá trị TRUE thì thực hiện câu lệnh 1, ngược lại nếu biểu thức 2 trả về giá trị FALSE thì thực hiện câu lệnh 2.

```
if (biểu thức 1) câu lệnh 1;  
else if (biểu thức 2) câu lệnh 2;  
else if (biểu thức 3) câu lệnh 3;  
else câu lệnh 4;
```

Nếu **biểu thức 1** trả về TRUE thì thực hiện **câu lệnh 1** (các câu lệnh 2,3,4 không thực hiện), nếu **biểu thức 1** trả về FALSE và **biểu thức 2** trả về TRUE thì **câu lệnh 2** được thực hiện, nếu **biểu thức 1** trả về FALSE **biểu thức 2** trả về FALSE và **biểu thức 3** trả về TRUE thì **câu lệnh 3** được thực hiện. Nếu cả 3 biểu thức 1,2,3 đều trả về FALSE thì thực hiện **câu lệnh 4**.

Ngoài ra PHP còn cung cấp từ khoá **elseif**, chính là ghép giữa từ khoá **else** và **if**.

2. Câu lệnh switch

Cú pháp :

```
switch ($bt) {  
    case h1: các lệnh1;  
    break;  
    case h2: các lệnh 2;  
    break;  
    .....  
    case hn: các lệnh n;  
    break;  
    default : các lệnh n+1;  
}
```

2. Câu lệnh switch

```
<?php
switch ($x) {
case 1: echo "Number 1";
          break;
case 2: echo "Number 2";
          break;
case 3: echo "Number 3";
          break;
default : echo "No number between 1 and 3";
}
?>
```

2. Câu lệnh switch

Câu lệnh switch hoạt động như là một loạt câu lệnh if ghép lại với nhau. Ta hãy xem câu lệnh if sau:

```
if ( $a == "abc" )
    {echo "Giá trị của a là abc"; }
else if ( $a == "def" )
    {echo "Giá trị của a là def"; }
else if ( $a == "123" )
    { echo "Giá trị của a là 123"; }
else
    { echo "Giá trị khác"; }
```

Có thể được viết lại bằng câu lệnh switch như sau:

```
switch ( $a )
{ case "abc" : echo "Giá trị của a là abc";
  break;
  case "def": echo "Giá trị của a là def";
  break;
  case "123": echo "Giá trị của a là 123";
  break;
  default: echo "Giá trị khác";
}
//end switch
```

Cấu trúc điều khiển

II. Cấu trúc lệnh lặp:

1. Câu lệnh *while* :

while (biểu thức) câu lệnh;

Trong khi biểu thức còn trả về giá trị TRUE thì tiếp tục thực hiện câu lệnh, sau khi thực hiện câu lệnh thì kiểm tra lại biểu thức, nếu vẫn còn trả về giá trị TRUE thì lại tiếp tục thực hiện câu lệnh...cứ tiếp tục như vậy cho tới khi nào biểu thức trả về giá trị FALSE thì ngưng.

Ví dụ: In ra các số từ 1 tới 10 với câu lệnh while:

```
<?php $i = 1;  
while ( $i <= 10 )  
{ echo $i, "<br>";  
 $i++;  
}  
//end while  
?>
```

Ghi chú: Câu lệnh **\$i++** tương đương với **\$i = \$i+1;**, câu lệnh này sẽ tăng giá trị của \$i lên 1 qua mỗi lần lặp.

2. Câu lệnh *do-while*

Câu lệnh do-while cũng tương tự như câu lệnh while, chỉ khác một điểm là câu lệnh được thực hiện trước rồi biểu thức mới được kiểm tra sau, nếu biểu thức còn trả về giá trị TRUE thì tiếp tục thực hiện câu lệnh.

Cú pháp của câu lệnh do-while như sau:

do

{ *câu lệnh*;

}

while (biểu thức);

Ví dụ:

In ra các số từ 1 tới 10 với câu lệnh **do-while**:

```
<?php $i =0;  
do  
{  
    $i++;  
    echo $i, “<br>”;  
} while ( $i <10 );  
?>
```

3. Câu lệnh *for*

Câu lệnh for cũng dùng để tạo vòng lặp. Đây là một trong những câu lệnh phức tạp nhất của PHP, cú pháp của nó như sau:

***for (biểu thức 1; biểu thức 2; biểu thức 3)
câu lệnh;***

Đầu tiên biểu thức 1 được thực hiện, Tiếp theo biểu thức 2 được kiểm tra Nếu trả về TRUE thì câu lệnh được thực hiện và sau đó thực hiện biểu thức 3. Nếu trả về FALSE thì kết thúc câu lệnh for.Kiểm tra lại biểu thức 2 và lặp lại quá trình như trên.

Ví dụ:

In ra các số từ 1 tới 10 với câu lệnh for:

```
<?php for ( $i = 1; $i < 10; $i++ )  
    { echo $i, “<br>”; }  
//end for  
?>
```

4. Câu lệnh *foreach*

Câu lệnh foreach chỉ làm việc với array. Câu lệnh foreach có 2 dạng cú pháp như sau:

foreach (\$array as \$value) câu lệnh;

foreach (\$array as \$key => \$value) câu lệnh;

Ví dụ 1:

```
<?php
$a = array('a' => '1', 'b' => '2', 'c' => '3');
foreach ( $a as $value )
{
    echo $value, "\n";
}
//end foreach
?>
```

Chương trình trên sẽ in ra 3 số 1, 2 và 3.

Ví dụ 2:

```
<?php
    $a = array('a' => '1', 'b' => '2', 'c' => '3');
    foreach ( $a as $key => $value )
        { echo $key, "=", $value, "\n"; }
    //end foreach
?>
```

Chương trình trên sẽ in ra 3 chuỗi **a=1**, **b=2** và **c=3**.

5. Câu lệnh *break*

Câu lệnh break sẽ dừng việc thực thi của các vòng lặp for, foreach, while, do-while và switch. Ở phần trước ta đã thấy câu lệnh break được sử dụng trong câu lệnh switch.

Ví dụ: sử dụng câu lệnh break trong vòng lặp for:

```
for ( $i=1; $i<=10; $i++ )  
{ echo $i;  
if ( $i == 5 ) break;  
}
```

6. Câu lệnh **continue**

Câu lệnh **continue** áp dụng lên các vòng lặp, lệnh **continue** sẽ bỏ qua lần lặp hiện thời và tiếp tục thực hiện các lần lặp tiếp theo.

Ví dụ:

```
for ( $i=1; $i<=5; $i++ )  
{ if ( $i == 2 ) continue;  
echo $i; }
```

Khi \$i đạt giá trị 2, câu lệnh **echo \$i;** sẽ được bỏ qua không thi hành nữa do câu lệnh **continue**. Và như vậy, đoạn lệnh trên khi chạy sẽ in ra các giá trị 1,3,4,5 (không có giá trị 2).

HÀM

1. Định nghĩa :

Trong lập trình, có một số đoạn mã được dùng nhiều lần ở nhiều nơi khác nhau trong chương trình.

PHP cung cấp một giải pháp đó là hàm do người dùng định nghĩa.

Ta có thể đưa đoạn mã đó vào trong một hàm, và ở chỗ nào cần dùng đoạn mã đó ta chỉ cần gọi hàm

Khi cần sửa đổi, ta chỉ cần sửa đổi một chỗ duy nhất là nội dung của hàm chứ không cần phải sửa ở nhiều nơi trong chương trình.

Cú pháp để tạo hàm như sau:

```
function tênHàm($tham_số1, $tham_số2,...)  
{  
    các lệnh trong thân hàm ;  
    return $kết_quả_trả_về;  
}  
//end
```

Khi cần sử dụng hàm ở chỗ nào, ta chỉ cần dùng cú pháp :
tênHàm(các tham số cần thiết)**);**

Ví dụ:

```
<?php
    function testing()
        { echo "Testing function";
            echo 1;
            echo 2;
            echo 3;
        } //end testing
    testing(); testing(); testing();
?>
```

Tên hàm cũng như tên biến , nhưng được phép bắt đầu bằng ký tự gạch dưới .

Tham số của hàm

Hàm có thể nhận vào các tham số, ví dụ:

```
<?php  
function testing($a)  
{ echo "Tham số là $a"; }  
//end testing testing(123);  
testing("abc");  
?>
```

Ở ví dụ trên, lời gọi hàm testing(123); sẽ in ra dòng Tham số là 123 và lời gọi hàm testing("abc"); sẽ in ra dòng Tham số là abc.

Ta có thể gán giá trị mặc định cho tham số của hàm:

```
<?php
    function testing($a="mặc định")
    { echo "Tham số là ". $a;
    }
//end
testing();
?>
```

Khi tham số tương ứng của hàm không được truyền, tham số đó sẽ nhận giá trị mặc định. Đoạn chương trình ví dụ ở trên khi chạy sẽ in ra dòng Tham số là mặc định.

Giá trị trả về từ hàm

Hàm còn thể trả về 1 giá trị cho nơi gọi:

```
<?php
    function binh_phuong($a)
    {
        $ketqua = $a * $a;
        return $ketqua;
    } //end testing
    echo binh_phuong(2);
?>
```

Đoạn chương trình trên khi chạy sẽ in ra số 4. Câu lệnh return biểu thức; sẽ kết thúc hàm và trả về giá trị của biểu thức cho nơi gọi.

Chú ý: bên trong hàm, biến cục bộ sẽ được dùng thay vì biến toàn cục.

Ví dụ:

```
<?php  
    $a = 1; //biến toàn cục //hàm do tự tạo  
    function test()  
    { echo $a; }  
    //end test  
?>
```

Để truy cập tới các biến toàn cục ở bên trong một hàm do người dùng định nghĩa, ta có thể dùng một trong hai cách sau:

Cách 1:

```
<?php $a = 1; //biến toàn cục
//hàm do tự tạo
function test()
{
    //tùy khoá global báo cho php biết là bên test
    //bây giờ ta sẽ dùng biến toàn cục
    global $a;
    echo $a; //in ra giá trị: 1
} //end test
```

Cách 2:

```
<?php  
$a = 1; //biến toàn cục  
//hàm do tự tạo  
function test()  
{ echo $GLOBALS['a']; //in ra giá trị: 1  
} //end test  
?>
```

2. Các hàm thông dụng của PHP

a. Hàm xem thông tin **phpinfo()**

Các tham số:

INFO_GENERAL

INFO_CREDITS

INFO_CONFIGURATION

INFO_MODULES

INFO_ENVIRONMENT

INFO_VARIABLES

INFO_LICENSE

INFO_ALL

```
<html>
<body>
    <?php
        phpinfo();
    ?>
    <?php
        phpinfo(INFO_GENERAL);
    ?>
</body>
</html>
```

b. Hàm hiển thị thông tin Server

```
<?php
    echo "Hỗ trợ: " . $_SERVER["HTTP_REFERER"] .
    "<br />";
    echo "Hỗ trợ Browser: " .
    $_SERVER["HTTP_USER_AGENT"] . "<br />";
    echo "Địa chỉ IP: " . $_SERVER["REMOTE_ADDR"];
?>
```

c. Hàm Header()

Chuyển đến một trang nào đó

```
<?php  
header("Location: http://www.ud.edu.vn/");  
?>  
<html> <body>  
.....</body> </html>
```

Lưu ý: Khi dùng hàm này cần:

* Phải đảm bảo các lệnh trước đó không có tác dụng
xuất bát cứ dữ liệu nào ra trình duyệt thì nó mới có tác
dụng (*kể cả trước đó là các thẻ HTML*)

3. Truy cập biến từ form

Trong PHP khai báo để lấy giá trị biến FORM từ trang HTML tương đối đơn giản, tên của một đối tượng Form trong HTML cũng chính là tên biến dùng trong PHP.

VD:

```
<?php echo "ten cua ban la: ".$txtUser;  
echo "Mat khau cua ban la: ".$txtPass;  
?>
```

Trong đó txtUser và txtPass là tên đối tượng dạng text trong thẻ FORM.

Sử dụng mảng hằng

```
$HTTP_POST_VARS["tên biến form"];  
$HTTP_GET_VARS["tên biến form"];
```

VD:<?php

```
echo "ten cua ban la: ".  
$HTTP_POST_VARS ["txtUser"];  
?>
```

Trong đó txtUser là tên đối tượng
dạng text trong thẻ FORM.

4. Khai báo hằng trong PHP

+ Tất cả các hằng số trong PHP là chữ hoa

+ Cú pháp khai báo:

`define("TENHANG", giá trị gán hằng)`

Ví dụ:

`define("PI",3.14);`

5. Sử dụng lại mã chương trình trong PHP

PHP cung cấp 2 cách để khai báo sử dụng bất kỳ đoạn mã chương trình nào đã có sẵn, bằng cách sử dụng 2 khai báo

require() và *include()*.

a. Sử dụng khai báo *require()*:

Cú pháp: *require(string path_file);*

Ví dụ : Tạo 1 file require1.php có nội dung như sau

```
<?php
$a=2;
$b=3;
$c=$a+$b;
echo "a + b = ".$c."<br>";
?>
```

- tạo một file php có nội dung

```
<?php  
    require("require1.php");  
?>
```

Các lưu ý:

- File được gọi có phần mở rộng tùy ý.
- Các đoạn mã PHP trong File được gọi phải đặt trong cặp thẻ <?php ...?>

b. Sử dụng khai báo *include()*:

Cú pháp: *include(string path_file);*

Giống như trường hợp trên, nhưng *include()* và *require()* khác nhau là: Tất cả nội dung đọc nạp lên bằng *require()*

Webserver phải dịch tất cả nội dung bên trong.

Do đó trong những file nạp bằng *require()* không nên dùng các cấu trúc điều khiển.

CÁC HÀM XỬ LÍ CHUỖI TRONG PHP

Xử lý chuỗi trong ứng dụng Web là một phần chính của PHP, điểm mạnh của PHP là khả năng xử lý chuỗi.

1. Hàm xử lý khoảng trắng trên chuỗi:

+Hàm kiểm tra chuỗi rỗng:

Cú pháp: *isempty(string str);*

Nếu chuỗi str trống thì trả về true, ngược lại trả về false.

+Hàm xóa bỏ khoảng trắng bên trái và bên phải chuỗi

Cú pháp: *ltrim(string str);*

và: *chop(string str);*

Ví dụ:

```
$name=ltrim($name);
```

```
$pass=chop($pass);
```

2. Hàm thay đổi kiểu chữ của chuỗi:

Hàm	Điễn giải	Sử dụng	Kết quả
strupper	Chữ hoa	<code>strupper("toi")</code>	TOI
strlower	Chữ thường	<code>strlower ("TOI")</code>	toi
ucfirst	Chữ hoa ký tự đầu tiên	<code>ucfirst("toi va ban")</code>	Toi va ban
ucword	Chữ hoa ký tự đầu mỗi từ	<code>ucword("toi va ban")</code>	Toi Va Ban

3. Định dạng chuỗi để đă vào và lấy ra từ CSDL:

Để lu trữ một chuỗi vào CSDL mà chuỗi đó cho chứa các ký tự ' hay " hay \ thì phải thêm dấu \ vào trước khi lưu.

Ta sử dụng hàm *AddSlashes(string str)*.

Hàm này sẽ tự tìm các ký tự trên và thêm \ vào trước.

Ngược lại ta dùng hàm :

StripSlashes(string str) để loại bỏ \.

Ví dụ: \$noidung=AddSlashes(\$noidung);

\$noidung=StripSlashes(\$noidung);

4. Hàm kết hợp hay tách chuỗi:

+ Tách chuỗi thành 2 chuỗi con bằng một ký tự phân cách và đưa vào mảng.

Cú pháp:

array=expode(string phan_cach, string str);

Ví dụ : tách địa chỉ Email thành 2 phần:

\$manga=expode("@",\$email);

echo manga[1];

+ Hàm substr();

Cú pháp:

string substr(string str, int i [,int l]);

Hàm này trả về chuỗi con của chuỗi str, bắt đầu từ vị trí i với l ký tự. Tham số l là tùy chọn, nếu không có thì trả về đến hết chuỗi str

5. Hàm so sánh chuỗi:

Ngoài toán tử `==` dùng so sánh 2 chuỗi, chúng ta còn có thể dùng các hàm :

int strcmp(string str1, string str2);

Nếu `str1==str2` : `=0`

`str1>str2` : `>0`

`str1<str2` : `<0`

So sánh `str1` và `str2` có phân biệt chữ hoa, chữ thường, ngoài ra còn có các hàm

int strcasecmp(string str1, string str2);

int strnatcasecmp(string str1, string str2);

6. Hàm kiểm tra chiều dài chuỗi:

cú pháp:

int strlen(string str);

7. Hàm tìm kiếm và thay thế chuỗi

+ Hàm tìm chuỗi:

string strstr(string str1, string str2);

Hàm này tìm chuỗi str2 trong str1, nếu có trả về chuỗi mới là chuỗi bắt đầu bằng chuỗi str2 đến hết chuỗi str1.

Ví dụ:

```
<?php  
$a="xin chao cac ban than men";
```

```
echo strstr($a,"c")."<br>";
```

```
?>
```

+ Hàm tìm vị trí chuỗi con:

int strpos(string str1, string str2);

Hàm này trả về vị trí tìm thấy của chuỗi str2 trong str1, nếu không tìm thấy thì trả về -1, nếu có nhiều chuỗi giống nhau, thì chỉ trả về vị trí tìm thấy đầu tiên.

(Vị trí của chuỗi tính từ 0)

Ví dụ:

```
<?php
$a="xin chao cac ban than men";
$vitri= strpos($a,"chao");
echo "vi tri la: ".$vitri."<br>";
?>
```

+ Hàm thay thế chuỗi:

string str_replace(string str1, string str2, string str);

. Hàm này trả về 1 chuỗi bằng cách thay thế tất cả các chuỗi str1 trong str bằng chuỗi str2.

. Để không chế chỉ cho thay đổi vị trí ta dùng hàm:

string substr_replace(string str1, string str2, int start [,int n]);

Thay thế str2 vào trong str1, bắt đầu từ ký tự có vị trí là start và có chiều dài là n ký tự

Ví dụ:

```
<?php
$a="xin chao cac ban va tat ca cac ban than men";
$vitri= strpos($a,"chao");
$chuoi= str_replace("ban","em",$a);
$chuoi2=substr_replace($chuoi,"ban",13,3);
echo "chuoi : ".$chuoi."<br>";
echo "chuoi 2 ".$chuoi2."<br>";
?>
```

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU **MySQL**

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

I. Truy xuất cơ sở dữ liệu bằng PHP

1. Cơ sở dữ liệu MySQL:

- + Để có thể liên kết và truy xuất dữ liệu trong PHP, trước tiên ta cần tìm hiểu cách thao tác làm việc trên cơ sở dữ liệu trong MySQL.
- + MySQL là một hệ quản trị cơ sở dữ liệu vừa và nhỏ.
- + Khi làm việc với cơ sở dữ liệu MySQL, bạn *đăng ký kết nối, tạo cơ sở dữ liệu, quản lý người dùng, phân quyền sử dụng, thiết kế đối tượng Table của cơ sở dữ liệu và xử lý dữ liệu.*

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

I. Truy xuất cơ sở dữ liệu bằng PHP

1. Cơ sở dữ liệu MySQL:

- + Nếu lần đầu tiên tạo kết nối cơ sở dữ liệu, chúng ta cần phải tạo một Connection, cung cấp tên Server hay IP của máy chứa MySQL.
- + Trong trường hợp máy chứa cơ sở dữ liệu MySQL là máy đang sử dụng, bạn có thể sử dụng tên **localhost**. Username mặc định của cơ sở dữ liệu MySQL là **root** và Password là **root**.
- + Cuối cùng bạn bắt đầu tạo cơ sở dữ liệu và làm việc với cơ sở dữ liệu.

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

I. Truy xuất cơ sở dữ liệu bằng PHP

2. *Truy xuất cơ sở dữ liệu trong PHP:*

a. **Thiết lập kết nối cơ sở dữ liệu:** PHP cung cấp cho chúng ta hàm kết nối cơ sở dữ liệu có cú pháp:

```
int mysql_pconnect("servername","username","password")
```

Ngoài ra có thể sử dụng phát biểu như sau:

```
$link=mysql_connect('servername','username','password') or die('Could not connect to MySQL Database');
```

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

I. Truy xuất cơ sở dữ liệu bằng PHP

2. *Truy xuất cơ sở dữ liệu trong PHP:*

b. Mở một cơ sở dữ liệu :

- + Sau khi khai báo kết nối đến cơ sở dữ liệu MySQL thành công, bạn khai báo phát biểu để mở cơ sở dữ liệu.
- + Nếu chọn cách 1 thì ta khai báo:

- *mysql_select_db("tên CSDL");*

- + Nếu chọn cách 2 thi ta khai báo:

- *mysql_select_db("tên CSDL",\$link);*

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

I. Truy xuất cơ sở dữ liệu bằng PHP

2. Truy xuất cơ sở dữ liệu trong PHP:

Vi du: Trong CSDL MySQL có cơ sở dữ liệu có tên là **DULIEU**. Để thực hiện kết nối đến cơ sở dữ liệu này ta khai báo đoạn lệnh sau:

```
<?php
```

```
//Khai báo kết nối
```

```
$link = mysql_connect("localhost", "root", "") or die ("Khong the  
ket noi den CSDL MySQL");
```

```
//Lựa chọn cơ sở dữ liệu
```

```
mysql_select_db("DULIEU", $link);
```

```
?>
```

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

I. Truy xuất cơ sở dữ liệu bằng PHP

2. Truy xuất cơ sở dữ liệu trong PHP:

c. Đóng kết nối dữ liệu :

+ Khi không sử dụng kết nối cơ sở dữ liệu, bạn có thể khai báo phát biểu đóng kết nối như sau:

mysql_close(database_connection);

Ví dụ: Đóng kết nối cơ sở dữ liệu trên:

mysql_close(\$link);

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

I. Truy xuất cơ sở dữ liệu bằng PHP

2. *Truy xuất cơ sở dữ liệu trong PHP:*

d. Truy vấn cơ sở dữ liệu:

+ Để truy vấn cơ sở dữ liệu ta có thể dùng hàm `mysql_query()` với tham số bắt buộc là chuỗi SQL, chuỗi kết nối là tham số tùy chọn như một trong ba phát biểu sau:

`int mysql_query(string sql);`

`int mysql_query(string sql, [int db_connection]);`

`int mysql_query(string database, string sql, [int db_connection]);`

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

I. Truy xuất cơ sở dữ liệu bằng PHP

2. Truy xuất cơ sở dữ liệu trong PHP:

```
<?php
//Khai báo kết nối
$link = mysql_connect("localhost", "root", "") or die
("Khong the ket noi den CSDL MySQL");
//Lựa chọn cơ sở dữ liệu
mysql_select_db("DULIEU", $link);
$sql = "Select * from TblUser";
$result = mysql_query($sql, $link);
?>
```

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

I. Truy xuất cơ sở dữ liệu bằng PHP

2. Truy xuất cơ sở dữ liệu trong PHP:

+ Để biết được số lượng mẫu tin trả về ta có thể dùng hàm :

int mysql_num_rows();

+ Bạn có thể đọc từng mẫu tin bằng các hàm sau:

mysql_fetch_array();

mysql_fetch_assoc();

```
<?php
```

```
$link = mysql_connect("localhost", "user", "password");  
mysql_select_db("database", $link);
```

```
$result = mysql_query("SELECT * FROM table1", $link);  
$num_rows = mysql_num_rows($result);
```

```
echo "$num_rows Rows\n";
```

```
?>
```

```
$row = mysql_fetch_array($result);  
$row = mysql_fetch_assoc($result);
```

Kết quả trả về từ hai hàm này là một **array**. Hai hàm này trả về dòng kết quả hiện thời của câu lệnh select và chuyển con trỏ tới dòng tiếp theo hoặc giá trị FALSE nếu như không còn dòng nào để trả về nữa.

```
mysql_error($conn);
```

Trả về thông báo lỗi của MySQL nếu như một lệnh trước đó có lỗi.

~~Sự khác nhau của `mysql_fetch_array()` và `mysql_fetch_assoc()`~~

+ Hàm `mysql_fetch_array()` sẽ trả về một array mà phần tử thứ [0] sẽ tương ứng với cột đầu tiên của table, phần tử thứ [1] sẽ tương ứng với cột thứ hai của table,..

Ví dụ : `while ($row = mysql_fetch_array($result))`

```
{ echo "Username = ".$row[0]."<br>\n";
echo "Password = ".$row[1]."<br>\n"; }
```

+ Hàm `mysql_fetch_assoc()` sẽ trả về một array mà các phần tử sẽ được truy cập qua tên, với tên được lấy từ tên các cột của table

Ví dụ : `while ($row = mysql_fetch_assoc($result))`

```
{ echo "Username = ".$row['username']. "<br>\n";
echo "Password = ".$row['password']. "<br>\n"; }
```

THỰC THI CÂU LỆNH SELECT

```
<?php
$result = mysql_query("SELECT id,email
FROM people WHERE id = '42'");
if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
$row = mysql_fetch_row($result);

echo $row[0]; // 42
echo $row[1]; // the email value
?>
```

.....

.....

```
while ($row = mysql_fetch_assoc($result)) {  
    echo $row["userid"];  
    echo $row["fullname"];  
    echo $row["userstatus"];  
}  
mysql_free_result($result);
```

?>

THỰC THI CÂU LỆNH INSERT

```
<?php
$link = mysql_connect('localhost', 'user', 'password');
if (!$link)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db('mydb', $link);
$sql = "INSERT INTO thanhvien (id, user, pass, email) VALUES (1,
    'admin', '123')";
$result = mysql_query($sql, $link);
//chèn thêm 1 dòng vào table
if ( !$result ) die("Không thể thực hiện được câu lệnh
    SQL:".mysql_error($link));
echo "Số lượng row được chèn: ".mysql_affected_rows($link)."<br>\n";
//ta chỉ chèn 1 dòng nên hàm mysql_affected_rows sẽ trả về 1
?>
```

THỰC THI CÂU LỆNH UPDATE

```
<?php
$link = mysql_connect('localhost', 'user', 'password');
if (!$link)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db('mydb', $link);
$sql = "UPDATE thanhvien SET pass='abc' WHERE username='admin'";
$result = mysql_query($sql, $link);
//đổi password của admin
if ( !$result ) die("Không thể thực hiện được câu lệnh SQL:
".mysql_error($link));
echo "Số lượng row được thay đổi:
".mysql_affected_rows($link)."<br>\n";
?>
```

THỰC THI CÂU LỆNH DELETE

```
<?php
$link = mysql_connect('localhost', 'user', 'password');
if (!$link)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db('mydb',$link);
$sql = "DELETE * FROM thanhvien";
$result = mysql_query($sql, $link);
//xoá hết tất cả các account
if ( !$result ) die("Không thể thực hiện được câu lệnh SQL:
    ".mysql_error($link));
echo "Số lượng row được xoá: ".mysql_affected_rows($conn)."<br>\n";
?>
```

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

II. CÀI ĐẶT PHP, MYSQL, APACHE

1. Cài đặt Apache trên Window:

+ Kiểm tra sau khi cài đặt:

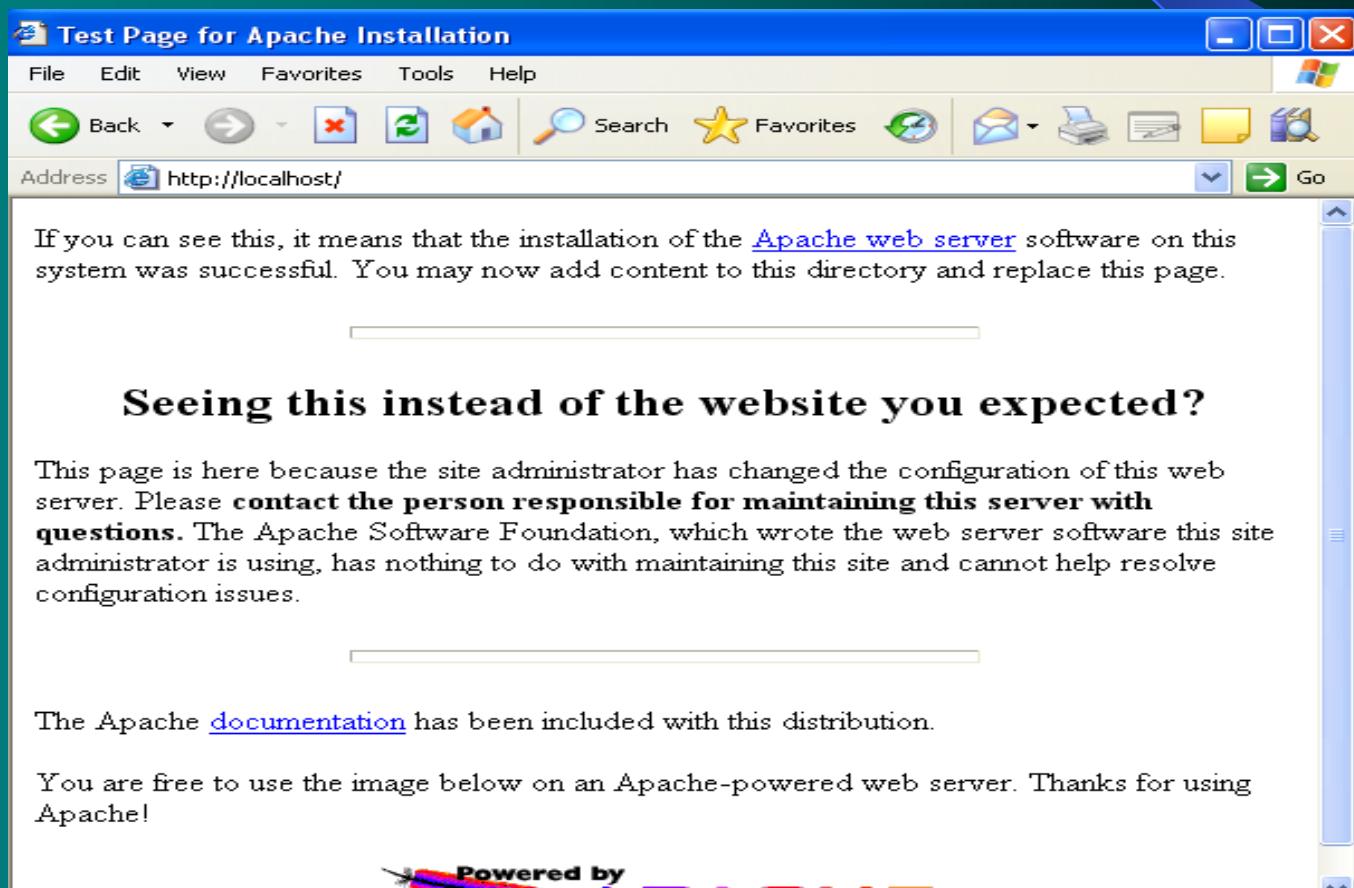
- Sau khi khởi động Apache nó sẽ lắng nghe trên Port 80.
- Để kiểm tra Apache server có hoạt động không bạn kết nối đến Apache web server và truy cập nhập địa chỉ <http://localhost/> hoặc <http://127.0.0.1>
- Nếu Apache Web server hoạt động được thì nó sẽ hiển thị trang nào chứa các liên kết đến tài liệu apache như hình sau:

HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

II. CÀI ĐẶT PHP, MYSQL, APACHE

1. Cài đặt Apache trên Window:

+ Kiểm tra sau khi cài đặt :

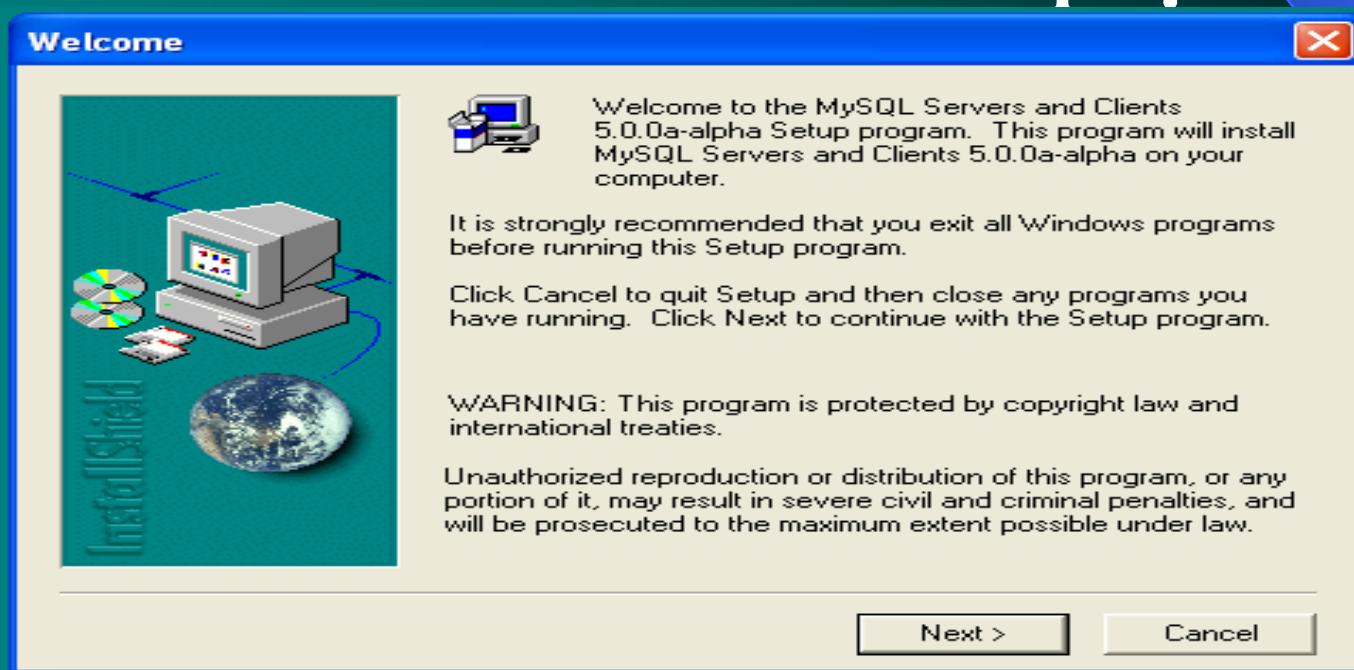


HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

II. CÀI ĐẶT PHP, MYSQL, APACHE

2. Cài đặt MySQL :

+ Nhấp đúp chuột lên tập tin cài đặt Setup.exe màn hình chào MySQL Server and Client xuất hiện như hình sau. Bấm Next để tiếp tục :

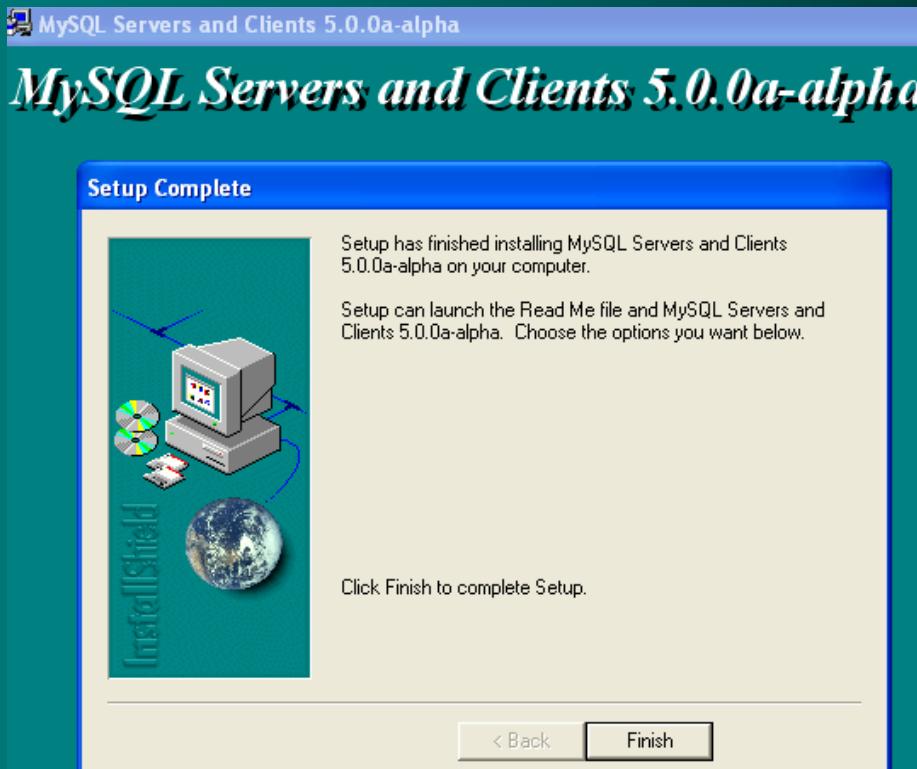


HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

II. CÀI ĐẶT PHP, MYSQL, APACHE

2. Cài đặt MySQL :

+ Đợi cho đến khi quá trình cài đặt kết thúc.
Bấm nút Finish để kết thúc quá trình cài đặt.

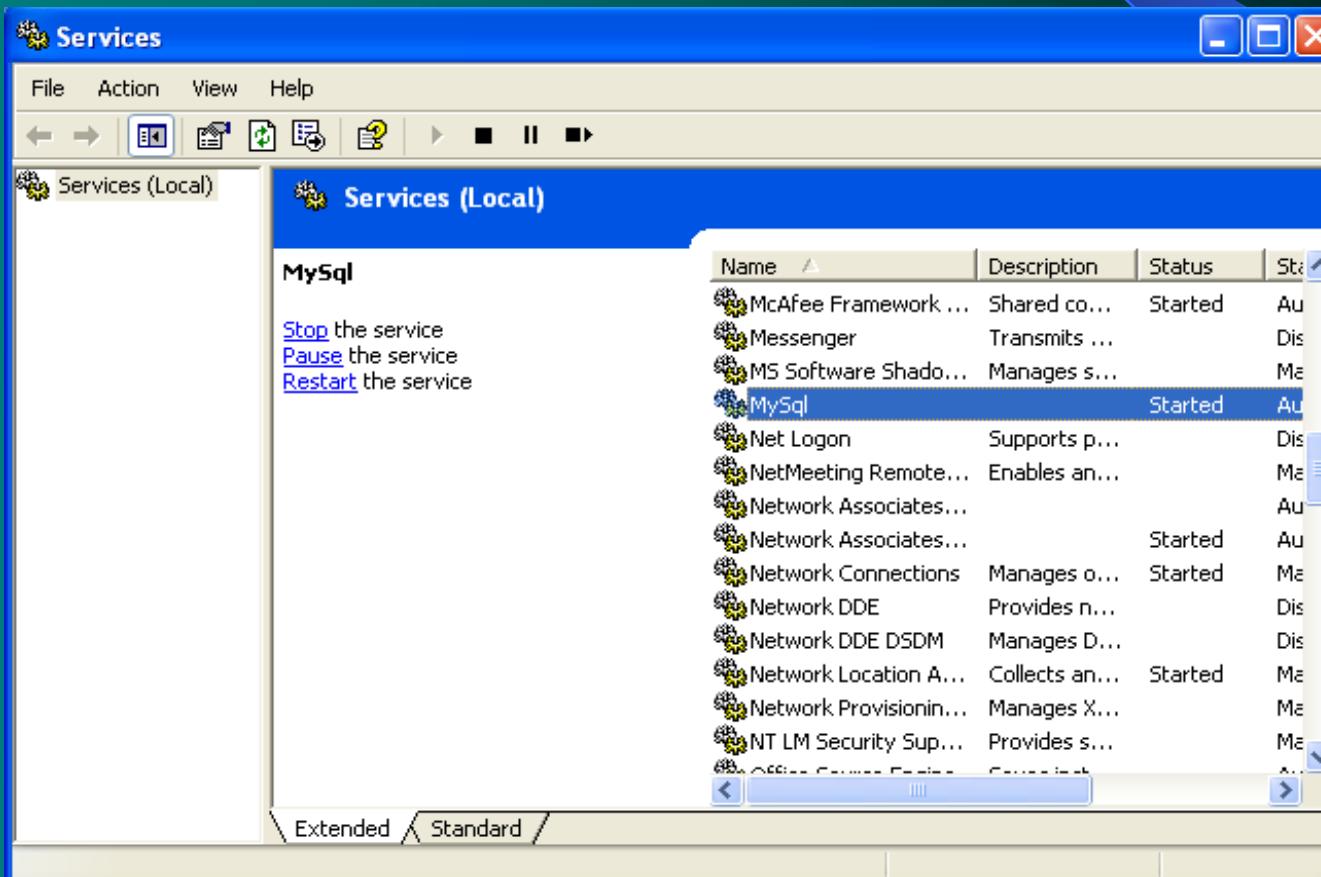


HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

II. CÀI ĐẶT PHP, MYSQL, APACHE

2. Cài đặt MySQL :

Khởi động MySQL: (Dùng Server)

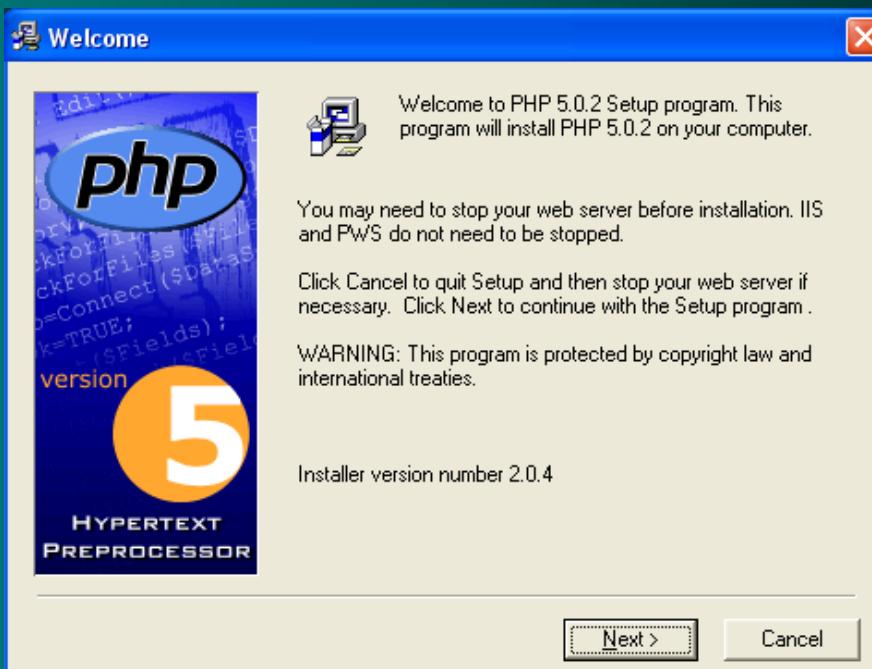


HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

II. CÀI ĐẶT PHP, MYSQL, APACHE

3. Cài đặt PHP:

+ Nhấp đôi chuột lên chương trình cài đặt. Màn hình chào chương trình cài đặt PHP xuất hiện. Bấm Next để tiếp tục

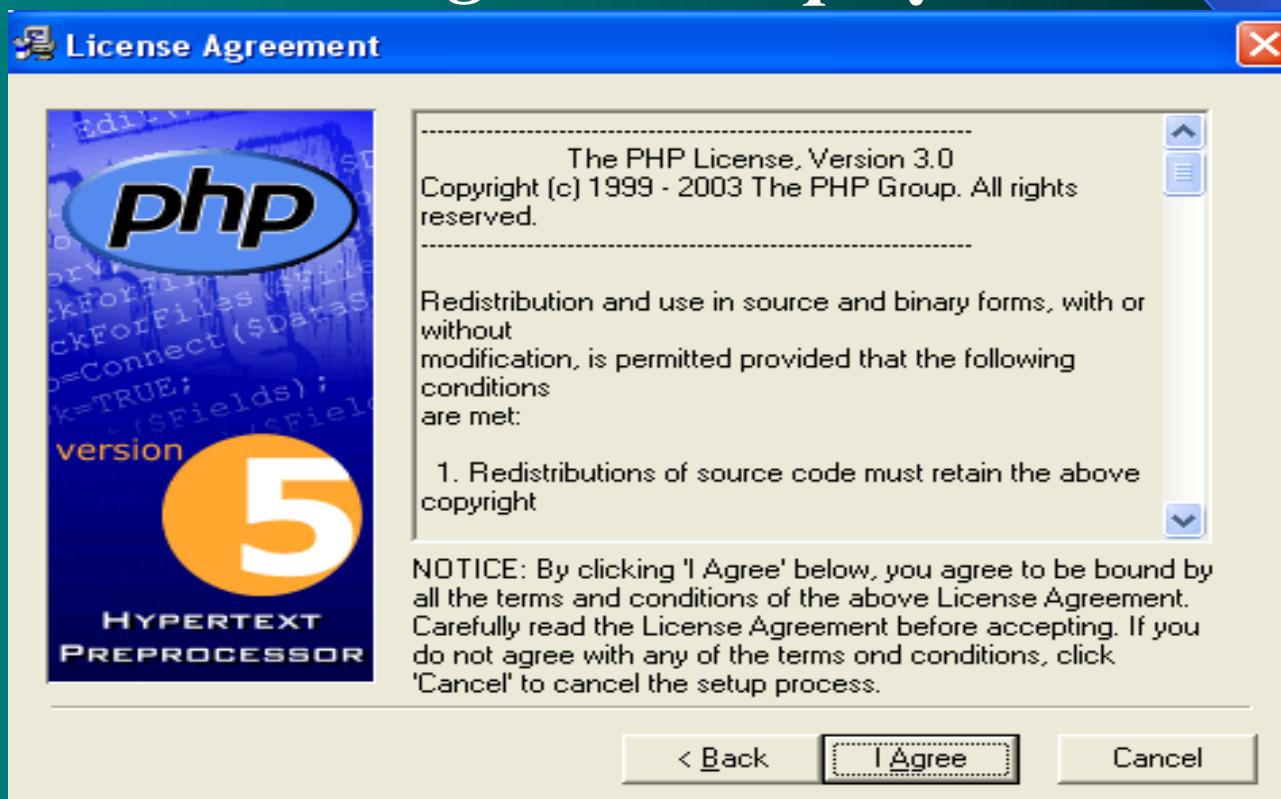


HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

II. CÀI ĐẶT PHP, MYSQL, APACHE

3. Cài đặt PHP:

+ *Màn hình thỏa thuận bản quyền xuất hiện như hình sau. Bấm I Agree để tiếp tục.*



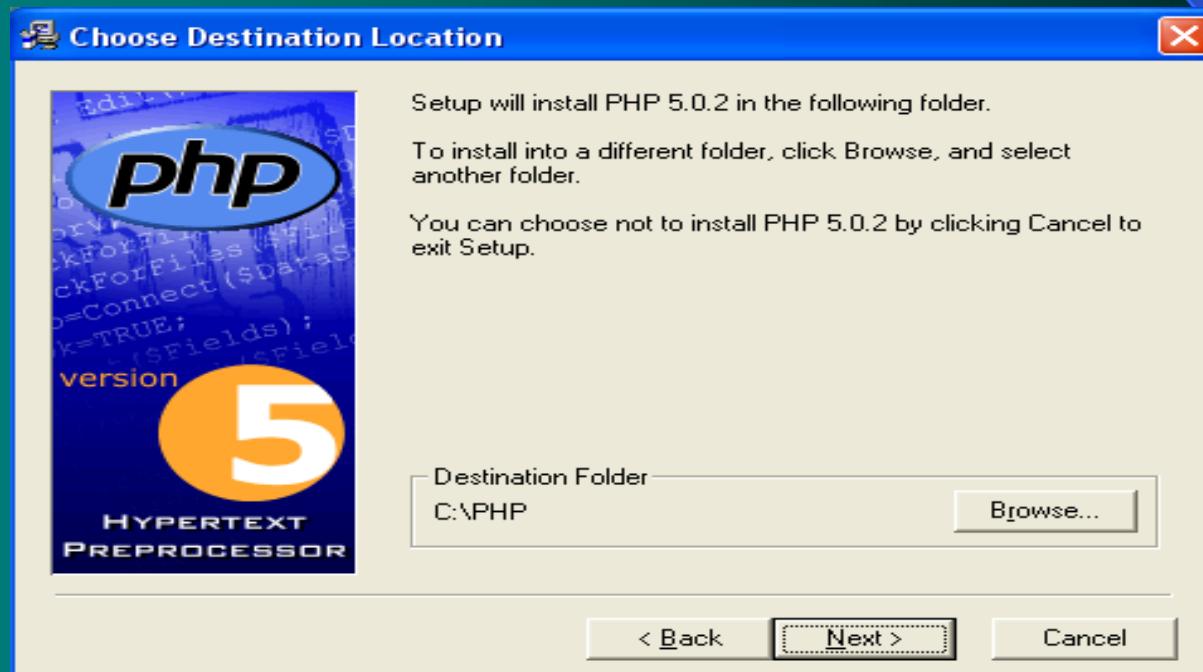
HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

II. CÀI ĐẶT PHP, MYSQL, APACHE

3. Cài đặt PHP:

+ Tiếp theo chọn kiểu cài đặt, hãy chọn Standard.

Bấm Next để tiếp tục

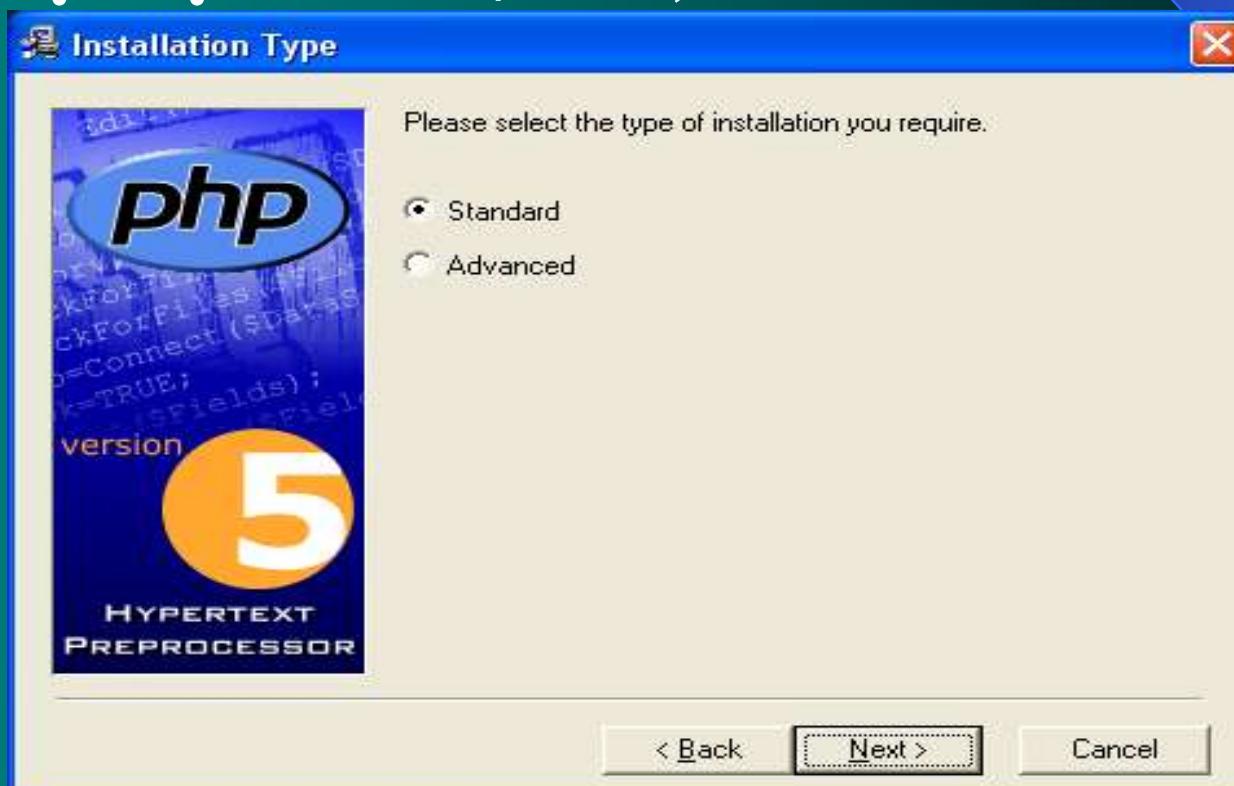


HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

II. CÀI ĐẶT PHP, MYSQL, APACHE

3. Cài đặt PHP:

+ Tiếp theo phép bạn chọn thư mục cài đặt PHP (mặc định là c:\PHP). Bấm Next để tiếp tục

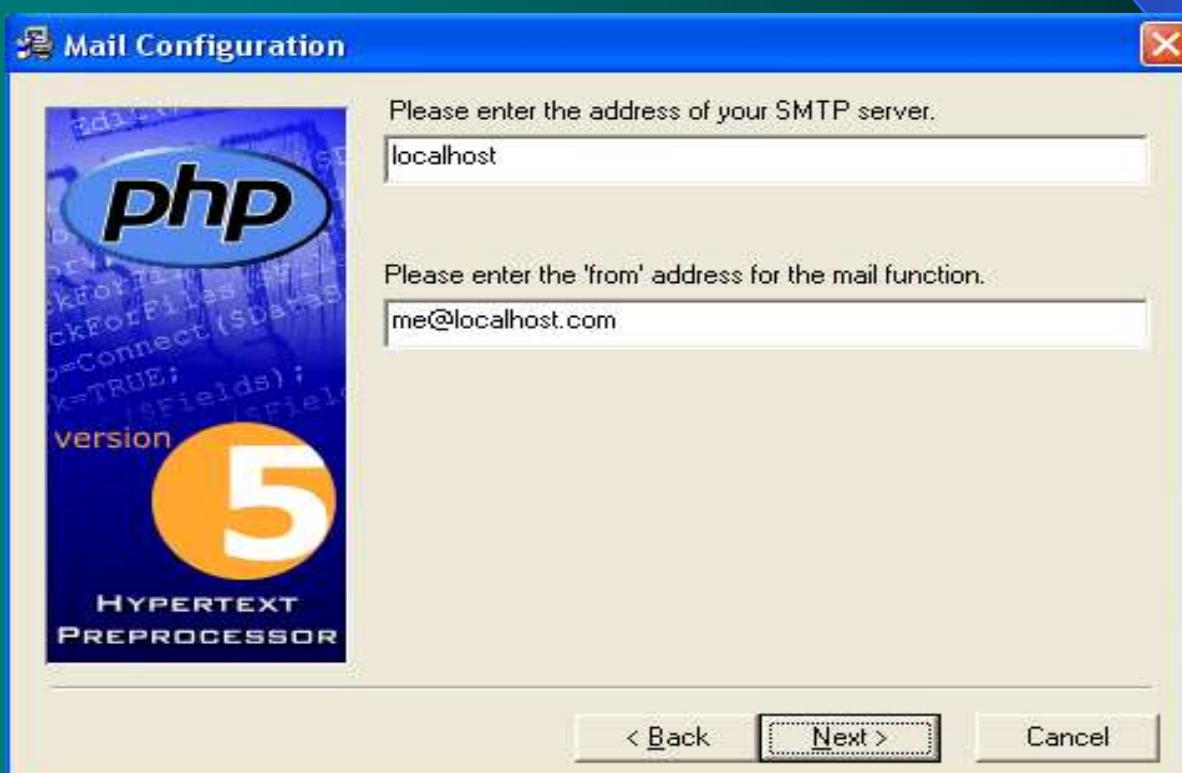


HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

II. CÀI ĐẶT PHP, MYSQL, APACHE

3. Cài đặt PHP:

+ *Màn hình thiết lập Mail xuất hiện như hình sau. Bấm Next để tiếp tục*



HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

II. CÀI ĐẶT PHP, MYSQL, APACHE

3. Cài đặt PHP:

- + Tiếp theo bạn chọn kiểu Web server bạn muốn cấu hình để chạy PHP, trong trường hợp này chúng ta dùng Apache.
- + Bấm Next để tiếp tục. Tiếp tục quá trình cài đặt.
- + Cuối cùng bạn nhận được thông báo cho biết quá trình cài đặt PHP đã kết thúc

CÔNG NGHỆ AJAX

NỘI DUNG

- 1. Giới thiệu*
- 2. Cơ chế và nguyên lý hoạt động*
- 3. Mô hình tương tác Asynchronous*
- 4. Đối tượng XMLHttpRequest*
- 5. Các công nghệ hỗ trợ liên quan*
- 6. Ajax và ứng dụng*

CÔNG NGHỆ AJAX

I. Khái quát chung :

1. Giới thiệu :

+ Ajax - Asynchronous JavaScript and XML (JavaScript và XML không đồng bộ) là công cụ cho phép tăng tốc độ ứng dụng Web bằng cách cắt nhỏ dữ liệu và chỉ hiển thị những gì cần thiết thay vì phải tải đi tải lại toàn bộ trang Web.

+ Ajax không phải là một công nghệ đơn lẻ mà nó là tập hợp của nhiều công nghệ.

AsynchronousJavascript + CSS + DOM + XMLHttpRequest

CÔNG NGHỆ AJAX

I. Khái quát chung :

1. Giới thiệu :

+ Ajax bao gồm:

- Thể hiện Web theo tiêu chuẩn XHTML và CSS.
- Nâng cao tính năng động và phản hồi bằng DOM (Document Object Model).
- Trao đổi và truy cập/tác động lên thông tin, dữ liệu bằng XSL và XSLT.
- Nhận thông tin không đồng bộ (Synchronous) dùng XMLHttpRequest.
- Và tất cả các kỹ thuật trên được liên kết lại với nhau bằng Javascript.

CÔNG NGHỆ AJAX

I. Khái quát chung :

2. Cơ chế và nguyên lý hoạt động :

* **Cơ chế :** Như vậy Ajax nó đóng vai trò như một lớp trung gian giữa giao diện trên trình duyệt và máy chủ xử lý thông tin. Có thể mô tả một cách cụ thể cách thức hoạt động của Ajax như sau :

- Ajax thực hiện tương tác với máy chủ bằng cách sử dụng đối tượng XMLHttpRequest, nhận kết quả trả về dưới dạng XML và phân tích kết quả bằng công nghệ DOM
- Tương tác giữa Ajax và giao diện người dùng được thực hiện thông qua các mã Javascript và XHTML+CSS.

CÔNG NGHỆ AJAX

I. Khái quát chung :

2. Cơ chế và nguyên lý hoạt động :

* **Cơ chế :** - Công nghệ Ajax thì sẽ cho phép tạo ra một Ajax Engine. Khi đó các yêu cầu gửi Request và nhận Response sẽ do Ajax Engine thực hiện.

- Web server gửi trả dữ liệu dưới dạng XML và Ajax Engine sẽ tiếp nhận, sau đó thực hiện phân tích chuyển hóa thành XHTML+CSS cho trình duyệt hiển thị.

- Các việc này được thực hiện trên Client nên giảm tải rất nhiều cho Server, đồng thời với người sử dụng thì sẽ thấy kết quả hiển thị ngay tức thì mà không cần nạp lại nguyên cả trang.

CÔNG NGHỆ AJAX

1. Khái quát chung :

2. Cơ chế và nguyên lý hoạt động :

**Nguyên lý hoạt động : Để minh họa cho cơ chế Ajax xin đưa ra hai mô hình tương tác như sau :*

+ Mô hình thứ nhất là mô hình tương tác giữa máy chủ và máy khách trong các ứng dụng Web truyền thống.

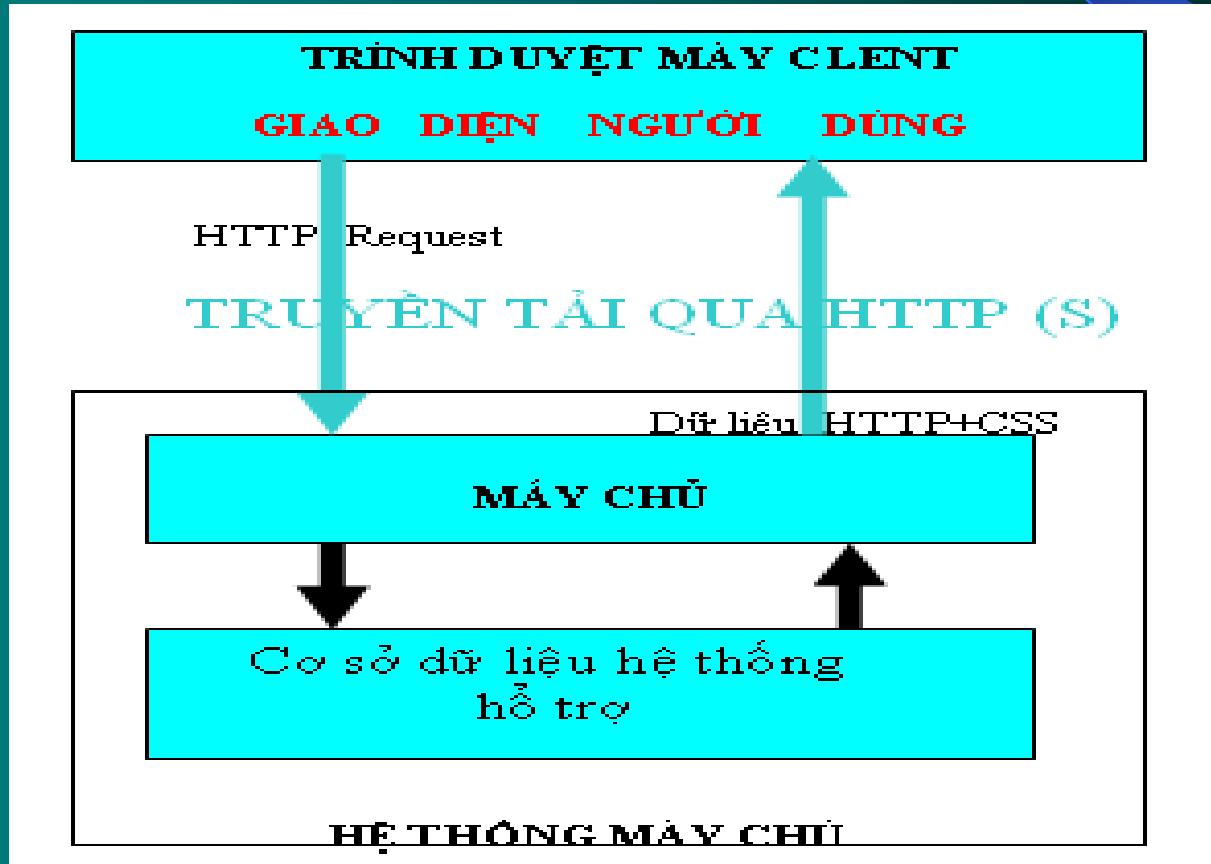
+ Mô hình thứ hai thể biểu diễn quá trình tương tác giữa máy chủ và máy khách trong các ứng dụng Web có sử dụng công nghệ Ajax.

CÔNG NGHỆ AJAX

I. Khái quát chung :

2. Cơ chế và nguyên lý hoạt động :

**Nguyên lý hoạt động :*

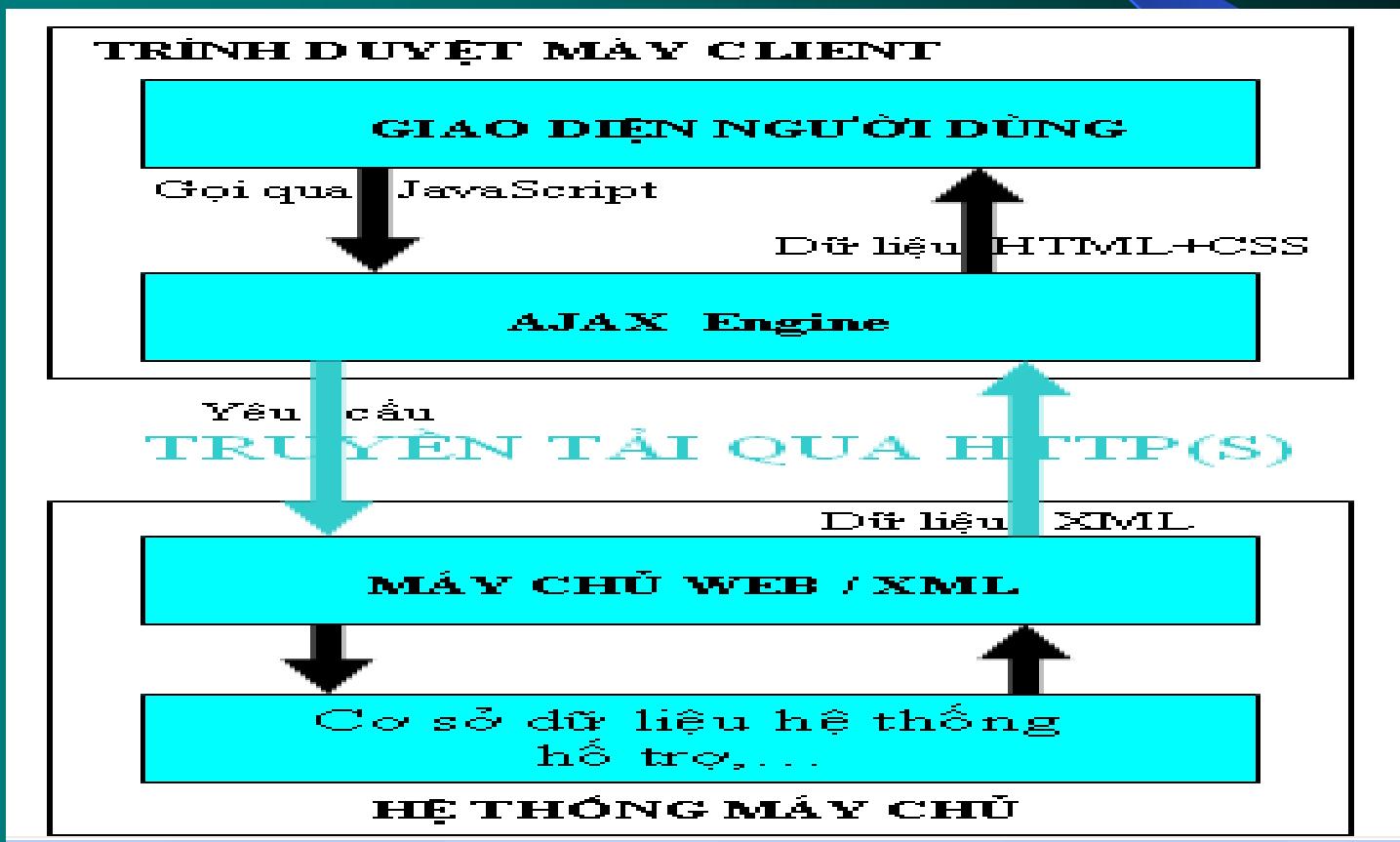


CÔNG NGHỆ AJAX

I. Khái quát chung :

2. Cơ chế và nguyên lý hoạt động :

*Nguyên lý hoạt động :



CÔNG NGHỆ AJAX

I. Khái quát chung :

2. Cơ chế và nguyên lý hoạt động :

*Nguyên lý hoạt động :

- + Một trong những điểm mấu chốt của công nghệ Ajax là không tương tác trực tiếp với máy chủ như truyền thống mà là qua một lớp trung gian của Ajax.
- + Chính điều đó sẽ tạo ra sự ngạc nhiên về cơ chế xử lý này vì sẽ phải qua thêm một lớp trung gian và như vậy thì sẽ làm chậm đi quá trình tương tác.

Tuy nhiên thực tế thì không như vậy, nguyên nhân là :

CÔNG NGHỆ AJAX

1. Khái quát chung :

2. Cơ chế và nguyên lý hoạt động :

**Nguyên lý hoạt động :*

- Không phải lúc nào cũng cần phải tương tác với máy chủ như trong trường hợp của Gmail, một khi giao diện của Gmail đã được tải về xong thì những tác vụ như chuyển từ thư mục này sang xem thư mục khác không nhất thiết phải đòi hỏi gọi lại máy chủ.

- Bằng cách chỉ nhận những thông tin cần thiết (ở dạng XML), dung lượng truyền tải giữa máy khách và máy chủ sẽ giảm đi rất nhiều. Điều này nghĩa là giao diện của ứng dụng chỉ cần tải về một lần (lần đầu)

CÔNG NGHỆ AJAX

I. Khái quát chung :

2. Cơ chế và nguyên lý hoạt động :

*Nguyên lý hoạt động :

- Ajax sẽ không phải thực hiện tải lại toàn bộ giao diện đó mỗi khi có tương tác với máy chủ. Thay vào đó Ajax sẽ gửi yêu cầu đến máy chủ và nhận kết quả từ máy chủ về những gì đã thay đổi sau khi máy chủ thực hiện yêu cầu đó.

- Ajax sau đó sẽ thông báo cho phần giao diện (qua các lời gọi Javascript) thực hiện các thay đổi tương ứng trên giao diện

CÔNG NGHỆ AJAX

1. Khái quát chung :

2. Cơ chế và nguyên lý hoạt động :

**Nguyên lý hoạt động :*

- Các tương tác giữa phần giao diện với Ajax là tương tác nội bộ trên trình duyệt. Việc này giúp cho việc thực hiện các thay đổi được diễn ra rất nhanh (gần như ngay tức thì).

- Vì thế khi duyệt một trang hỗ trợ Ajax, người sử dụng không bao giờ thấy một cửa sổ trống (blank) và biểu tượng đồng hồ cát – là những dấu hiệu cho thấy máy chủ đang thực hiện công việc.

CÔNG NGHỆ AJAX

I. Khái quát chung :

3. Mô hình tương tác Asynchronous

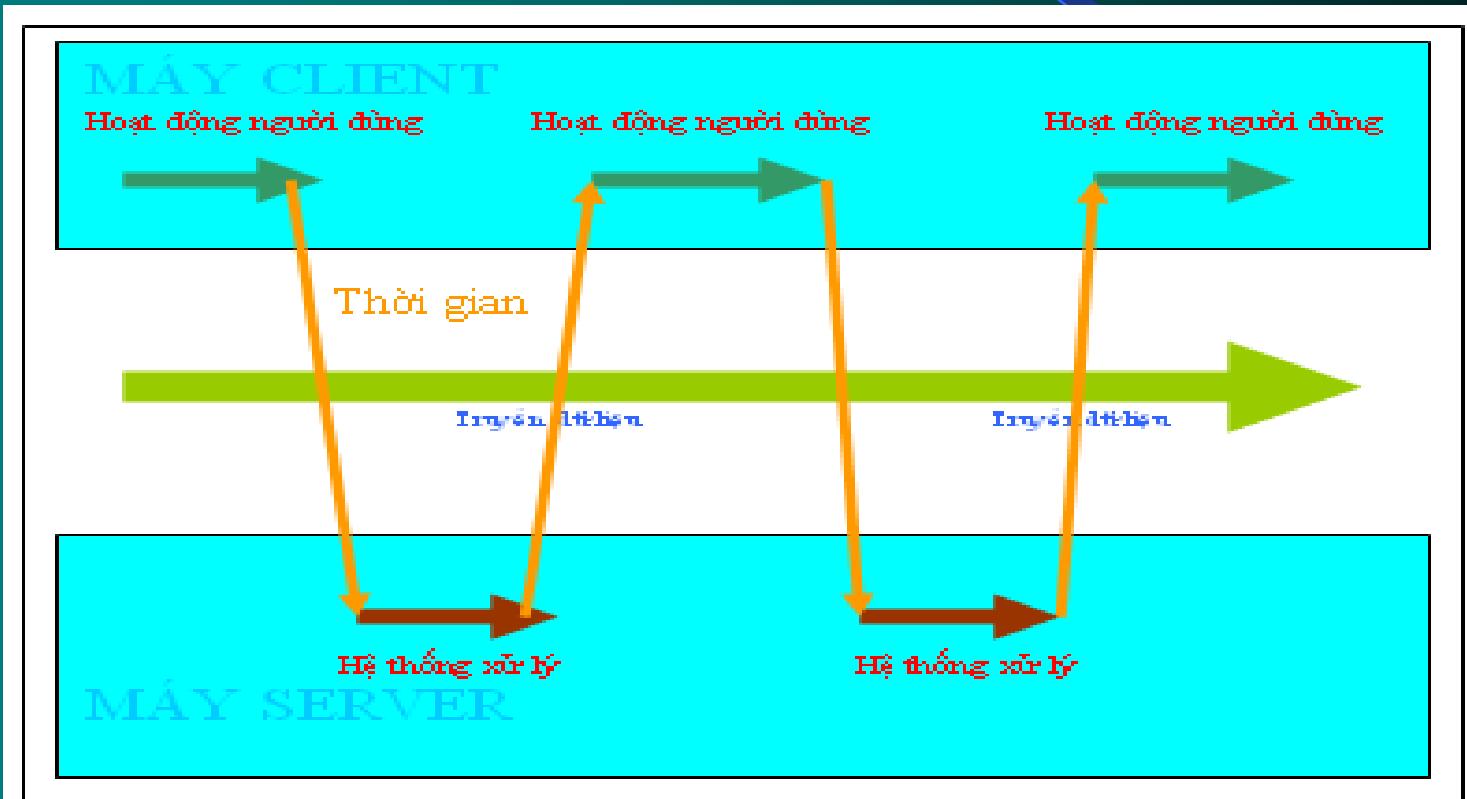
+ Sự cải tiến trong mô hình tương tác máy chủ - máy khách là một trong những điểm khác biệt của Ajax

+ Mô hình tương tác của các ứng dụng web truyền thống (Mô hình đồng bộ)

CÔNG NGHỆ AJAX

I. Khái quát chung :

3. Mô hình tương tác Asynchronous



Mô hình tương tác của các ứng dụng web truyền thống
(Mô hình đồng bộ)

CÔNG NGHỆ AJAX

I. Khái quát chung :

3. Mô hình tương tác Asynchronous

+ Theo mô hình trên có sự thiếu hợp lý ở chỗ: Khi người dùng có yêu cầu và gửi lên máy chủ thì chờ cho tới khi máy chủ xử lý xong và sau khi nhận được kết quả từ máy chủ trả về thì người dùng mới có thể tiếp tục thực hiện các thao tác tiếp theo.

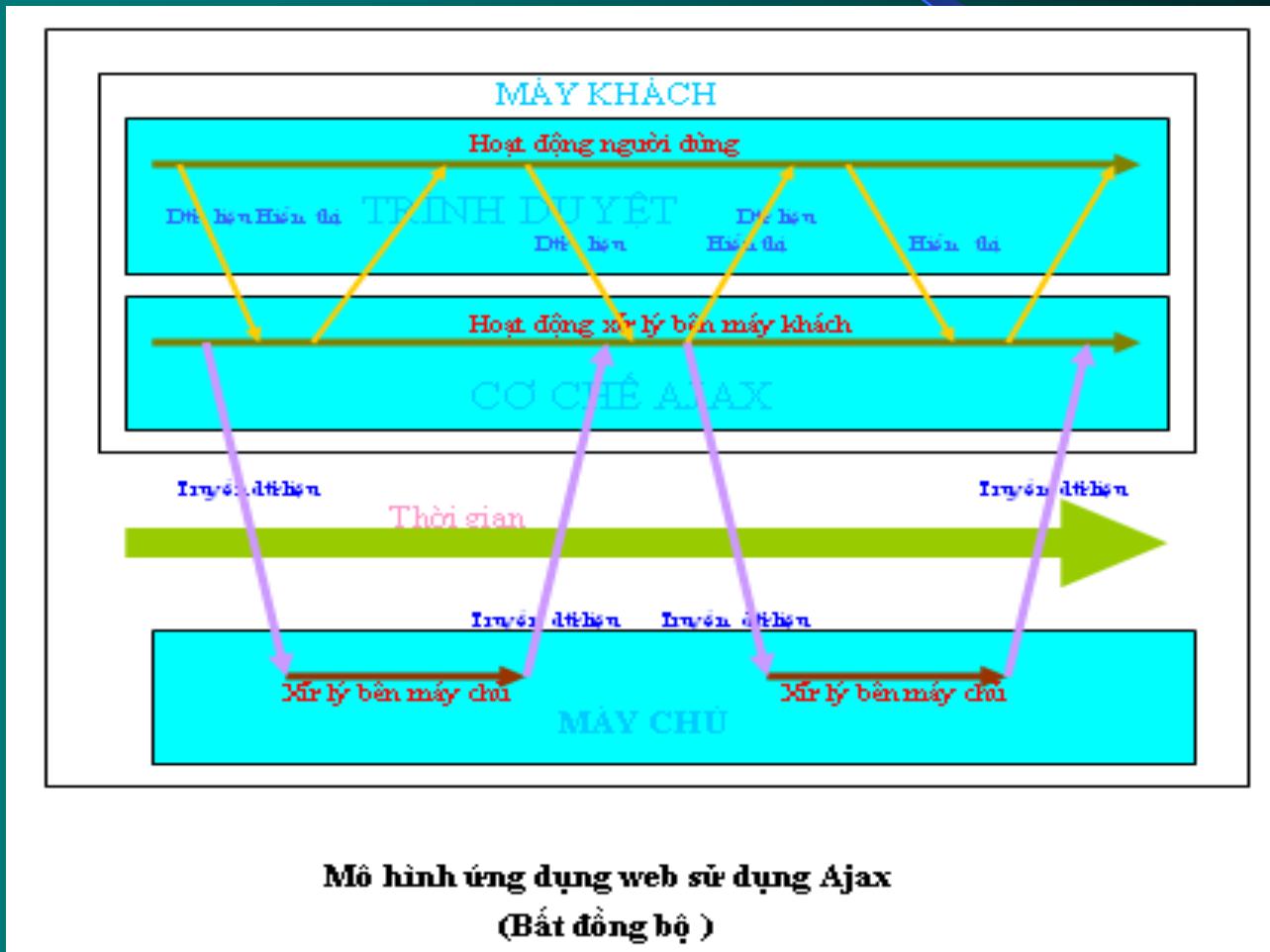
+ Như vậy thì đặt trong trường hợp nếu ta cần xử lý dữ liệu lớn thì không còn cách nào khác là phải chờ khá lâu.

+ Đây chính là nhược điểm của các công nghệ Web truyền thống và Ajax đã giải quyết được vấn đề này .

CÔNG NGHỆ AJAX

I. Khái quát chung :

3. Mô hình tương tác Asynchronous



CÔNG NGHỆ AJAX

I. Khái quát chung :

3. Mô hình tương tác Asynchronous

+ Để ý thấy rằng tương tác giữa giao diện máy khách của chúng ta với Ajax diễn ra mang tính chất ngẫu nhiên và rất thường xuyên

+ Tương tác giữa Ajax và máy chủ xảy ra với chu kỳ dài hơn nhiều. Chính vì vậy việc xử lý những yêu cầu thực hiện trên máy chủ sẽ không ảnh hưởng gì đến quá trình tương tác giữa người dùng với máy client. Do đó làm giảm đi khoảng thời gian đợi như ở các mô hình truyền thống.

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

1. Ngôn ngữ XML

+ XML (Extensible Markup Language) - là một bộ quy luật bao gồm các quy tắc để chia một tài liệu ra làm nhiều phần, rồi đánh dấu và ráp các phần khác nhau lại để dễ nhận diện chúng.

+ Tổ hợp Web toàn cầu W3C gọi XML là “một cú pháp thông dụng cho việc hiển thị cấu trúc trong dữ liệu”.

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

1. Ngôn ngữ XML

+ Mục tiêu chính của XML là tách biệt nội dung và dạng biểu diễn của văn bản (Baldwin).

+ Cùng một nội dung ta có thể biểu diễn dưới nhiều hình thức khác nhau : chẳng hạn in ra giấy hoặc đưa lên Web, và có thể áp dụng nhiều thuộc tính trên văn bản như in đậm, in nghiêng, in gạch dưới, dùng màu,...

+ Ngoài ra nhờ việc sử dụng các ký tự thông thường mà con người có thể hiểu được nội dung của văn bản ngay khi còn đang ở dạng thô, chưa được áp dụng bất kỳ dạng biểu diễn nào

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

1. Ngôn ngữ XML

Một trang XML cần phải theo đúng các quy luật sau đây :

- + Trang XML phải bắt đầu bằng câu tuyên bố XML
- + Mỗi bộ phận, gọi là “element” phải nằm giữa Tag Pair.
- + Nếu Tag nào không chứa gì ở giữa thì phải chấm dứt bằng ”>”

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

1. Ngôn ngữ XML

Một trang XML cần phải theo đúng các quy luật sau đây : + Một trang XML phải có một element độc nhất chứa tất cả các element khác. Đó là root của cây biểu diễn trang XML .

+ Các Tag Pair không được xen kẽ nhau, các Tag Pair phải đánh vần đúng y như nhau kể cả chữ hoa, chữ thường (<STUDENT> và </student> là bất hợp lệ) và các Attribute đều phải nằm giữa hai ngoặc kép (standalone=yes là bất hợp lệ, phải viết standalone="yes").

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

1. Ngôn ngữ XML

Ví dụ :

```
<?xml version="1.0" standalone="yes"?>
```

```
<gicungduoc>
```

Chào XML ! Mạnh giỏi ?

```
</gicungduoc>
```

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

2. *Đối tượng DOM (Document Object Model)*

+ Mô hình đối tượng tài liệu là một giao diện lập trình ứng dụng (API).

+ Thông thường DOM có dạng một cây cấu trúc dữ liệu và được dùng để truy xuất các tài liệu HTML và XML.

+ Mô hình DOM hoạt động độc lập với hệ điều hành và dựa theo kỹ thuật lập trình hướng đối tượng để mô tả tài liệu.

+ Trong Ajax mô hình DOM đóng vai trò phân tách dữ liệu truyền dưới dạng XML.

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

2. Đối tượng DOM

+ Để phân tích một văn bản XML bằng máy tính ta thường dùng một trong hai hoặc cả hai mô hình phân tích là SAX (Simple API for XML) và DOM.

+ Theo mô hình phân tích DOM, toàn bộ nội dung của một văn bản XML sẽ được lưu giữ vào bộ nhớ trong dưới dạng một cấu trúc cây

CÔNG NGHỆ AJAX

II. Các khái niệm liên quan Ajax :

3. Giới thiệu về XHTML

+ Trong Ajax XHTML cùng với CSS đóng vai trò hiển thị dữ liệu cho trang Web.

+ XHTML là một chuẩn kết hợp giữa HTML và XML và trở thành một định dạng riêng cho web. XHTML cho phép HTML được mở rộng bằng các thẻ sở hữu.

+ Ngoài ra XHTML cũng được mã hoá chặt chẽ hơn HTML và phải tuân thủ nhiều quy tắc cấu trúc hơn HTML

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

3. Giới thiệu về XHTML

+ Trong XHTML tất cả các thẻ phải được đóng, các thuộc tính phải được cho vào dấu ngoặc kép (""), và đặc biệt tất cả các thẻ và thuộc tính đều phải viết bằng chữ cái thường.

+ Mặc dù điều này có thể XHTML bị xem là cứng nhắc nhưng chính điều này lại là ưu điểm của nó: quy chuẩn và nguyên tắc.

+ XHTML là phiên bản kết hợp của HTML nên có những điểm khác cơ bản với HTML

CÔNG NGHỆ AJAX

II. Các khái niệm liên quan Ajax :

3. Giới thiệu về XHTML

- Các File XHTML cần phải well-formed có nghĩa là các thẻ `<htm>` `<head>` `<body>`...phải tuân thủ theo chuẩn .

- Các thẻ lồng nhau phải đúng trật tự .
- Các thẻ và các thuộc tính của các phần tử phải được viết bằng chữ thường.
- Các thuộc tính của các phần tử phải được đóng nháy kép: ví dụ `<table border="1" width="100%">`...

- Các thẻ được gọi là noempty cần phải được đóng bằng một thẻ đóng(`
` ...`</br>`)

CÔNG NGHỆ AJAX

II. Các khái niệm liên quan Ajax :

3. Giới thiệu về XHTML

- Không cho phép giản lược các thuộc tính.

Ví dụ : trong HTML có thể viết `<option selected>` nhưng trong XHTML phải viết đủ `<option selected="selected">` và không được đóng bằng thẻ `</option>`.

- Thuộc tính name được thay bằng thuộc tính id.

Ví dụ : `` được thay thế bằng ``

- Phải khai báo DOCTYPE trước thẻ `<html>`

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

3. Giới thiệu về XHTML

<!DOCTYPE Doctype goes here>

<html>

<head>

<title>Vi du ve XHTML</title>

</head>

<body>

This is XHTML page

</body>

</html>

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

4. CSS (*Cascading Style Sheets*)

+ Tập định kiểu theo tầng, được dùng để miêu tả cách trình bày các tài liệu viết bằng ngôn ngữ HTML, XHTML, XML, SVG, XUL...

+ Các thông số kỹ thuật của CSS do World Wide Web Consortium (W3C) quản lý. Trong Ajax CSS cùng với XHTML đóng vai trò hiển thị dữ liệu cho trang web .

CÔNG NGHỆ AJAX

II. Các khái niệm liên quan Ajax :

5. DHTML

+ Hay HTML động tạo một trang Web bằng cách kết hợp các thành phần ngôn ngữ đánh dấu HTML tinh, ngôn ngữ lệnh máy khách (như JavaScript) và ngôn ngữ định dạng CSS và DOM.

+ Do có khả năng phong phú, DHTML còn được sử dụng như một công cụ để xây dựng các trò chơi đơn giản trên trình duyệt .

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

6. Đối tượng XMLHttpRequest

+ Để hiểu được cơ chế hoạt động của Ajax thì chúng ta phải biết được XMLHttpRequest là gì ? Như nhiều chuyên gia đã nhận định một trong những điểm quan trọng nhất của Ajax là nằm ở XMLHttpRequest

+ Điều đó cho thấy rằng, vì XMLHttpRequest có vai trò quan trọng đặc biệt trong công nghệ như vậy nên một ứng dụng của Ajax có thực thi và mang hiệu quả hay không phụ thuộc vào việc trình duyệt có hỗ trợ đối tượng này. Nếu một trình duyệt không hỗ trợ XMLHttpRequest sẽ không thể diễn ra quá trình trao đổi giữa máy client và máy server

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

6. Đối tượng XMLHttpRequest

+ Thông thường trong một ứng dụng web, để có thể gửi một HTTP Request đến máy chủ bằng JavaScript bạn cần tạo một đối tượng của lớp cung cấp tính năng này. Trong Internet Explorer (IE) thì lớp này tồn tại dưới dạng một ActiveX có tên là XMLHttpRequest

+ Đối tượng XMLHttpRequest không phải là một chuẩn của W3C (World Wide Web Consortium). Đối tượng XMLHttpRequest được hỗ trợ ở IE 5.0+, Safari 1.2, Mozilla 1.0/Firefox, Opera 8.0 và Netcape 7+.

var httpReqquest = new XMLHttpRequest();

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

6. Đối tượng XMLHttpRequest

+ Do sự khác biệt này nên để các ứng dụng có thể chạy được trên các trình duyệt chúng ta phải khai báo:

```
if(window.XMLHttpRequest)
{
    //Mozilla,Safari,..
    httpRequest=new XMLHttpRequest();
}
else if(window.ActiveXObject)
{
    //IE..
    httpRequest=new
    ActiveXObject("Microsoft.XMLHTTP");
}
```

CÔNG NGHỆ AJAX

III. Các khái niệm liên quan Ajax :

6. Đối tượng XMLHttpRequest

+ Do ActiveX trên IE rất nguy hiểm cho nhiều người dùng nên trong nhiều trường hợp tính năng này bị cấm. Vì vậy bạn cần phải kiểm tra trình duyệt trước khi gọi đối tượng XMLHttpRequest. Quá trình kiểm tra này được thực hiện thông qua giá trị của window.ActiveXObject

Ví dụ : `if(window.ActiveXObject) {
 xmlhttp=new
 ActiveXObject("Microsoft.XMLHTTP");
}
else{...}`

Ngôn ngữ Truy vấn dữ liệu (SQL)

I. Giới thiệu

Ngôn ngữ SQL (Structured Query Language) được hãng máy tính IBM đưa ra vào năm 1970.

Là một ngôn ngữ có cấu trúc, có nghĩa là bằng tập hợp các dòng lệnh sẽ chỉ cho hệ quản trị CSDL biết là người sử dụng sẽ làm gì chứ không chỉ ra làm như thế nào.

Ngôn ngữ SQL cung cấp một công cụ truy vấn (query) nhằm cho phép người sử dụng đặt câu hỏi với dữ liệu đang chứa trong các bảng, dữ liệu được rút ra từ những truy vấn có thể có nguồn từ một hay nhiều bảng khác nhau.

Sau đó tham khảo kết quả này, xem xét, phân tích nó kỹ lưỡng để có những xử lý thông tin thiết thực hơn.

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

Trong mục này trình bày một số lệnh cơ bản dùng để thao tác trên cơ sở dữ liệu nh: chọn dữ liệu, thêm vào dữ liệu mới, xoá dữ liệu cũ, sửa đổi dữ liệu.

1. **Lệnh Select** : Lệnh này dùng để lấy dữ liệu trong một bảng thỏa mãn điều kiện nào đó.

Cú pháp:

```
Select [ *|fields] From tablename  
[Where expression]  
[Group by fields1]  
[Having expression1]  
[Order by {colname }[ASC | DESC]]
```

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

- ***: tất cả các trường đều được chọn.
- *colname*: tên cột.
- *tablename*: là tên của một Table hoặc nhiều Table mà các Record được chọn.
- *expression*: là biểu thức điều kiện được đặt ra mà các bản ghi được chọn từ *tablename* phải thoả mãn biểu thức này.
- *expression1*: một biểu thức điều kiện đi sau mệnh đề Having. Mệnh đề Having thông thường đi kèm với mệnh đề Group by, *expression1* là biểu thức không tác động vào toàn bảng được chỉ ra bởi mệnh đề From mà chỉ tác động lần lượt từng nhóm các bản ghi đã được chỉ ra bởi mệnh đề Group by.
- Order by: dùng để sắp các bản ghi tăng hay giảm

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

1. Lệnh Select

- Ví dụ: Giả sử có một Table có tên NHANSU như sau:

Maso	Hoten	Ngaysinh	Nghenghiep
1	Lê Văn Tuấn	10/8/1990	Học sinh
2	Nguyễn Văn Hà	7/8/1970	Kỹ s
3	Nguyễn Thị Hoa	30/07/1980	Sinh viên

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

1. Lệnh Select

- Nếu sử dụng lệnh: Select Hoten, Ngaysinh From nhansu
- Sẽ chọn tất cả các bản ghi nhưng chỉ giới hạn bởi hai trường Hoten và Ngaysinh. Cụ thể gồm các bản ghi sau:

Lê Văn Tuấn	10/8/1990
Nguyễn văn Hà	7/8/1970
Nguyễn Thị Hoa	30/07/1980

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

2. Lệnh Insert Into : Lệnh này dùng để chèn thêm một Record dữ liệu mới vào cho bảng.

- Cú pháp:

Insert Into *table*[(*field1* [, *field2* ,...])]

Values(*value1* [, *value2* ,...])

- Trong đó:

Table: tên của Table cần bổ sung bản ghi

field1, field2: tên của các trường cần bổ sung dữ liệu

value1, value2: giá trị cần bổ sung ứng với các trường *field1, field2...*

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

2. Lệnh Insert Into

- Ví dụ:
- Giả sử ta có được Table Nhansu nh trên, bây giờ bổ sung vào bản ghi mới:
Insert Into nhansu (Hoten, Ngaysinh, Nghenghiep)
Values("Nguyễn văn An","10/7/1995","Học sinh")
- Sau khi thực hiện lệnh này bảng Nhansu bây giờ như sau:

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

2. Lệnh Insert Into

Maso	Hoten	Ngaysinh	Nghenghiep
1	Lê Văn Tuấn	10/08/1990	Học sinh
2	Nguyễn văn Hà	07/08/1970	Kỹ s
3	Nguyễn Thị Hoa	30/07/1980	Sinh viên
4	Nguyễn văn An	10/07/1995	Học sinh

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

3. Lệnh Update : Lệnh này dùng để thay đổi dữ liệu cho các trường trong một bảng thỏa mãn điều kiện nào đó

- Cú pháp:

Update table Set Field1=value1[,...] Where expression

- Trong đó:

table: là tên của Table chứa bản ghi cần sửa đổi.

Field1: trường cần thay đổi dữ liệu

Value1: giá trị dữ liệu mới

expression : là biểu thức, mà các bản ghi được cập nhật phải thỏa mãn biểu thức này

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

3. Lệnh Update

- Ví dụ:
- Sửa đổi trường Nghenghiep có giá trị mới thành "CNVC" cho tất cả các record thoả mãn điều kiện là trường Ngaysinh="07/08/1970":

Update Nhansu Set Nghenghiep="CNVC" Where Ngaysinh="07/08/1970"

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

3. Lệnh Update

Table Nhansu bây giờ như sau:

Maso	Hoten	Ngaysinh	Nghenghiep
1	Lê Văn Tuấn	10/08/1990	Học sinh
2	Nguyễn văn Hà	07/08/1970	CNVC
3	Nguyễn Thị Hoa	30/07/1980	Sinh viên
4	Nguyễn văn An	10/07/1995	Hoc sinh

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

4. Lệnh Delete

Lệnh này dùng để xoá một hoặc nhiều record trong một bảng thoả mãn điều kiện nào đó.

- Cú pháp:

*Delete * From *table* Where *expression**

- Trong đó:

table: là tên Table chứa các bản ghi cần xoá.

**:* tất cả các trường của bản ghi đều bị xoá.

expression : là tất cả các bản ghi được xoá phải thoả mãn biểu thức này.

Ngôn ngữ Truy vấn dữ liệu (SQL)

II. Các câu lệnh SQL

4. Lệnh Delete

Ví dụ:

- Xoá tất cả các bản ghi trong Table Nhansu có trường Nghenghiep="Học sinh".
- Delete * From nhansu Where Nghenghiep="Học sinh"

Bây giờ table Nhansu trở thành:

Maso	Hoten	Ngaysinh	Nghenghiep
2	Nguyễn văn Hà	07/08/1970	CNVC
3	Nguyễn Thị Hoa	30/07/1980	Sinh viên

Ngôn ngữ Truy vấn dữ liệu (SQL)

Chú ý: Trong mệnh đề WHERE, các phép toán được sử dụng là:

Phép toán

=

\diamond

>

<

\geq

\leq

BETWEEN

LIKE

Mô tả

So sánh bằng

So sánh khác

Lớn hơn

Nhỏ hơn

Lớn hơn hoặc bằng

Nhỏ hơn hoặc bằng

Nằm giữa một khoảng

So sánh mẫu chuỗi

Lưu ý: Trong một số phiên bản của SQL, phép toán \diamond có thể được viết dưới dạng !=

Ngôn ngữ Truy vấn dữ liệu (SQL)

* Sử dụng dấu nháy

Lưu ý rằng ở ví dụ trên ta đã sử dụng hai dấu nháy đơn ('') bao quanh giá trị điều kiện 'Sandnes'.

SQL sử dụng dấu nháy đơn bao quanh các giá trị ở dạng chuỗi văn bản (text). Nhiều hệ CSDL còn cho phép sử dụng dấu nháy kép (""). Các giá trị ở dạng số không dùng dấu nháy để bao quanh.

+ Với dữ liệu dạng chuỗi văn bản:

Câu lệnh đúng:

```
SELECT * FROM Persons WHERE FirstName = 'Tove'
```

Câu lệnh sai:

```
SELECT * FROM Persons WHERE FirstName = Tove
```

+ VỚI DỮ LIỆU DẠNG SỐ:

Câu lệnh đúng:

```
SELECT * FROM Persons WHERE Year > 1965
```

Câu lệnh sai:

```
SELECT * FROM Persons WHERE Year > '1965'
```

* Phép toán điều kiện LIKE

Phép toán LIKE được dùng để tìm kiếm một chuỗi mẫu văn bản trên một cột.

Cú pháp :

Cú pháp của phép toán LIKE như sau:

**SELECT tên_cột FROM tên_bảng
WHERE tên_cột LIKE mẫu**

Một ký hiệu % có thể được sử dụng để định nghĩa các ký tự đại diện. % có thể được đặt trước và/hoặc sau mẫu.

+ Sử dụng LIKE

Câu lệnh SQL sau sẽ trả về danh sách những người có tên bắt đầu bằng chữ O:

```
SELECT * FROM SINHVIEN  
WHERE ten LIKE 'O%'
```

Câu lệnh SQL sau sẽ trả về danh sách những người có tên kết thúc bằng chữ a:

```
SELECT * FROM sinhvien  
WHERE ten LIKE '%a'
```

Câu lệnh SQL sau sẽ trả về danh sách những người có tên kết chứa chuỗi la:

```
SELECT * FROM sinhvien  
WHERE ten LIKE '%la%'
```

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

1. Giới thiệu về Jrun

- + Để nghiên cứu về JSP trước tiên phải có trình chủ Web Server "hiểu Java". Trình chủ đó phải có khả năng thông dịch và gọi thực thi được các đơn thể Java theo đặc tả của Sun.
- + Mặc dù hiện nay có rất nhiều trình chủ Web dành cho Java như Java Web Server, TomCat, IPLanet, Jrun,...Nhưng do Jrun viết bằng Java nên bạn có thể sử dụng và cài đặt Jrun cả trên môi trường Linux lẫn Unix.
- + Trình Jrun cung cấp khả năng triệu gọi mã Java thông qua Servlet. Jrun cung cấp cho ta hai dịch vụ Web hoạt động trên hai cổng:
 - Cổng 8100 là dịch vụ Web thông thường và cổng 8000 là dịch vụ Web phục vụ cho cấu hình trình chủ.
- Cũng như những Web server khác, tìm hiểu Jrun bao gồm các bước sau:
 - Cài đặt trình chủ Jrun.
 - Cấu hình kết nối với các Web server khác

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

2. Cài đặt Jrun

2.1. Cài đặt JDK

+ Trước khi cài đặt trình chủ Jrun Web Server ta phải cài đặt JDK. Mặc định JDK của Java sẽ được cài vào thư mục C:\Jdk. Khi Jrun yêu cầu chỉ định thư mục cài đặt JDK ta hãy chỉ đường dẫn đến thư mục đó.

2.2. Cài đặt Jrun Web Server

+ Yêu cầu về phần cứng và phần mềm :

- 32 MB Ram
- 20 MB dung lượng đĩa cứng
- Hệ điều hành : Windows9x hay NT/2000, Unix và Linux.
- Sử dụng trình duyệt Netscape hay Internet Explorer
- Môi trường máy ảo JRE (Java Runtime Environment)
- JDK

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

2. Cài đặt Jrun

2.3. Cấu trúc th mục của Jrun

Th mục

<i>Th mục</i>	<i>Nội dung</i>
/bin	Chứa các tập tin chsơng trình của Jrun
/connectors	Chứa các tập tin Web server kết nối
/docs	Chứa các trang tài liệu html hsóng dẫn sử dụng
/lib	Chứa các tập tin ths viện .jar
/lib/ext	Chứa các tập tin .jar gồm servlet.jar & ejb.jar
/logs	Chứa các tập tin log của Jrun
/samples	Chứa các trang jsp và servlet mâu
/servers	Chứa các ứng dụng Web của Jrun
/servers/lib	Chứa các tập tin .jar và các lớp đối tsợng .class
/servlets	Chứa các tập tin .class của servlet truy cập
/uninst	Thông tin về tháo bỏ cài đặt của Jrun

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3. Cửa sổ điều khiển (Jrun Management Console)

Jrun Management Console là một ứng dụng Web nhằm giúp cho thao tác trên giao diện Web để cấu hình Jrun.

Để dùng JMC nhs một người quản trị Web Server, ta có thể thực hiện theo các bước sau:

Nhập địa chỉ URL : <http://local:8000>

+ Chọn Start -> Program -> Jrun3.1 -> Jrun Admin Server

+ Chọn Start -> Program -> Jrun3.1 -> Jrun Management Console

Nếu thành công Jrun sẽ hiển thị màn hình đăng nhập (login) yêu cầu bạn cung cấp tài khoản người dùng (username) và mật khẩu (password) để đăng nhập.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3. Cửa sổ điều khiển (Jrun Management Console)



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3. Cửa sổ điều khiển (Jrun Management Console)

Sau khi đăng nhập, các thông tin quản trị sẽ

JRun Management Console - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Edit

Address: http://localhost:8000/index.jsp

JRUN Application Management Console and Server Administrator

(admin) welcome connector wizard password change manage JMC users serial number key search logout

Welcome to JRun

Developer Edition

JRun 3.1
Version 3.1.15506

Documentation

Release Notes
Product Documentation
Quick Start Tour
Example Applications

Online Resources

Product Registration
Support Forums
Knowledge Base
Developer Center
Check for Updates

About JRun Management Console

This console serves as an HTML tool that enables you to configure various aspects of JRun.

This window contains two panes. The left pane provides a "tree" view of the JRun Management Console, starting from the root level. The right pane displays the contents of the folder or object currently selected in the tree. An object may be a function (e.g., logout) or a group of properties (e.g., MIME Filters).

To preview the contents of a folder in the left pane, click the + sign preceding the folder. To close an open folder in the left pane, click the - sign. To select an object, click the object. The right pane then displays the appropriate data. To edit data displayed, either click the Edit button or click a property; then make your changes in the editor window that appears. In most cases, you will need to restart JRun after your changes.

Local intranet

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3. Cửa sổ điều khiển (Jrun Management Console)

Sau khi đăng nhập, các thông tin quản trị sẽ



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.1 Cấu hình quản trị Web Server

Thường bạn sử dụng trang quản trị để quản lý, khởi động, thay đổi tham số cho trình chủ Default Web Server phục vụ ứng dụng Web.

3.1.1 Cấu hình Jrun Default Web Server

Ta chỉ cần cấu hình theo các thông tin sau :

+ Web Server Address	*
+ Client IP filter	*
+ Web Server Port	8100
+ Idle threed timeout	300
+ Minimum threed count	1
+ Maximum active request	100
+ Maximum concurrent request	100
+ Jrun Web Server	28

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.1 Cấu hình quản trị Web Server

3.1.2 Tự tạo một Web Server

+ Ngoài Jrun Default Server, bạn có thể tự tạo một Web Server khác cho riêng mình

Lưu ý : trên một máy có thể có nhiều Web Server chạy cùng một lúc miễn sao chúng được cấu hình với số hiệu cổng khác nhau.

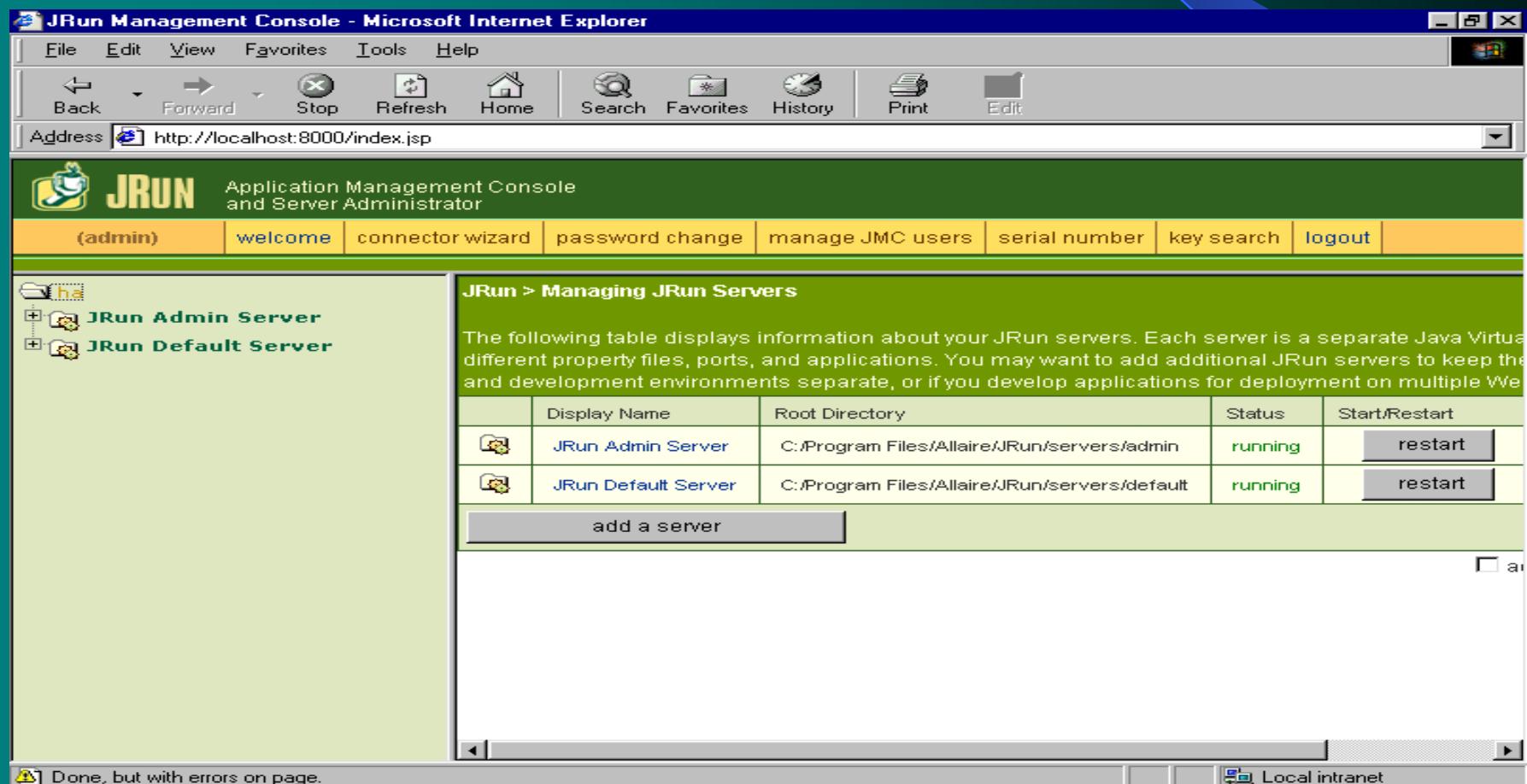
+ Để tạo Web Server khác, bạn chọn nhánh gốc bên khung trang trái. Nhấn nút *Add a server* bên khung trang phải và đặt tên cho Web Server cũng như các thông tin cấu hình khác mà Jrun yêu cầu

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.1 Cấu hình quản trị Web Server

3.1.2 Tự tạo một Web Server của riêng bạn



JRun Management Console - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Print Edit

Address http://localhost:8000/index.jsp

JRUN Application Management Console and Server Administrator

(admin) welcome connector wizard password change manage JMC users serial number key search logout

JRun > Managing JRun Servers

The following table displays information about your JRun servers. Each server is a separate Java Virtual Machine with its own configuration, property files, ports, and applications. You may want to add additional JRun servers to keep the development and production environments separate, or if you develop applications for deployment on multiple Web servers.

	Display Name	Root Directory	Status	Start/Restart
	JRun Admin Server	C:/Program Files/Allaire/JRun/servers/admin	running	restart
	JRun Default Server	C:/Program Files/Allaire/JRun/servers/default	running	restart

[add a server](#)

Done, but with errors on page. Local intranet

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.1 Tạo mới ứng dụng Web

Thông thường khi tạo một ứng dụng Web, ta cần phải cấu hình ứng dụng này trong Jrun Web Server. Để tạo một ứng dụng trên Web Server của Jrun, ta chỉ cần theo các bước sau:

- + Trước khi cấu hình Jrun, ta cần chú ý địa chỉ IP của máy chủ nơi Jrun đang chạy. Nếu không kết nối mạng, ta có thể sử dụng localhost hay 127.0.0.1 là địa chỉ IP cục bộ của mỗi máy.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.1 Tạo mới ứng dụng Web

- + Chọn Jrun Web Server (Admin Server hay Default Server)
- + Bấm vào link Create Application
- + Chọn Jrun Admin Server hay Jrun Default Server.
- + Nhập tên của ứng dụng. Ví dụ : DATN
- + Nhập vào địa chỉ gốc URL cho ứng dụng.
- + Chọn thư mục mà ta đặt các trang JSP của ứng dụng.

Ví dụ: C:/DATN/Jsp là thư mục chứa các trang JSP của ứng dụng.

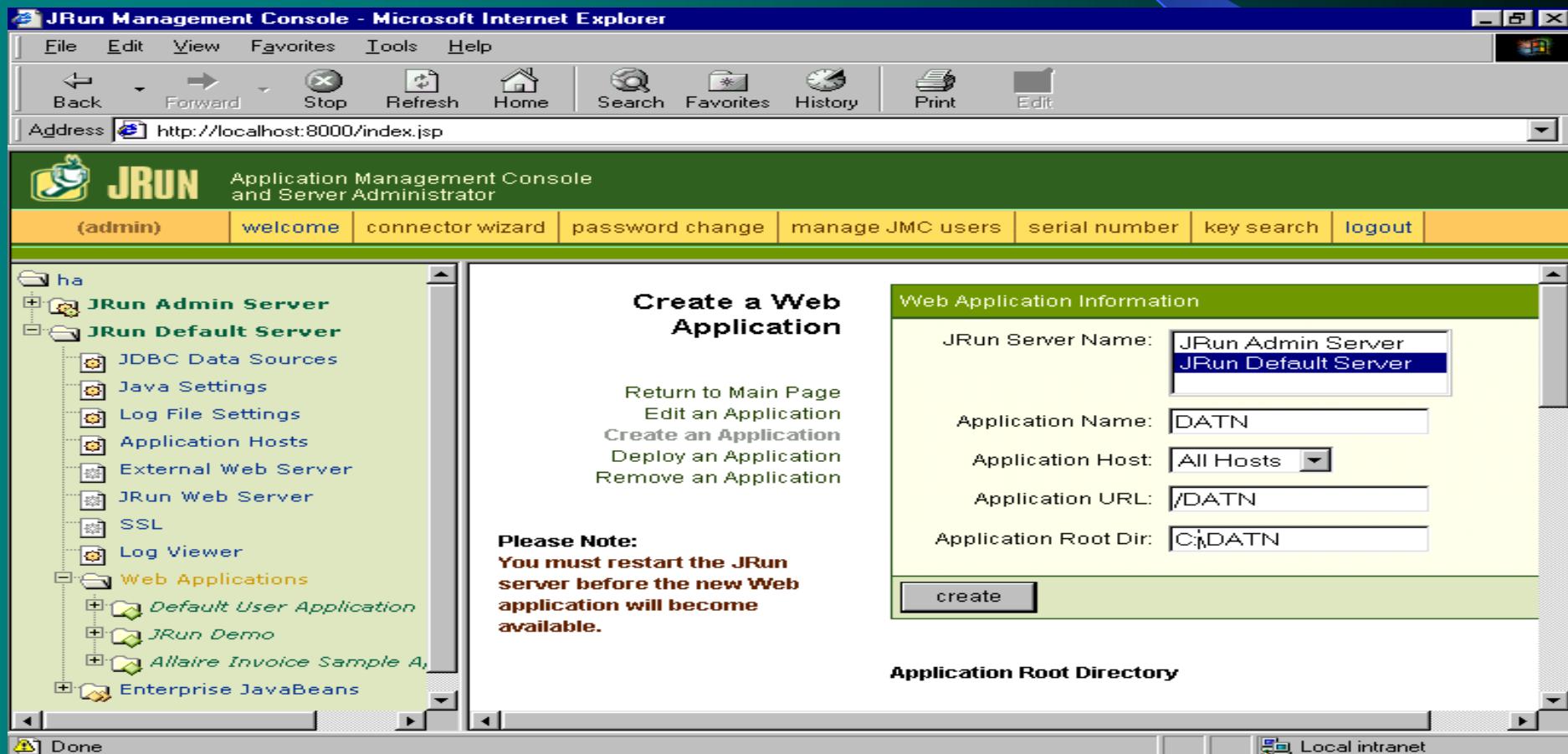
- + Sau khi điền thông tin ta nhấn nút Create.
- + Tiếp đến ta khởi động lại Web Server tương ứng với tên mà ta dùng để tạo ứng dụng(Jrun Default Server).
- + Cuối cùng ta thử trên màn hình Browser của trình duyệt bằng cách gõ vào địa chỉ URL nh sau : <http://localhost:8100/DATN/Jsp/index.jsp>

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.1 Tạo mới ứng dụng Web



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.2 Xóa một ứng dụng Web

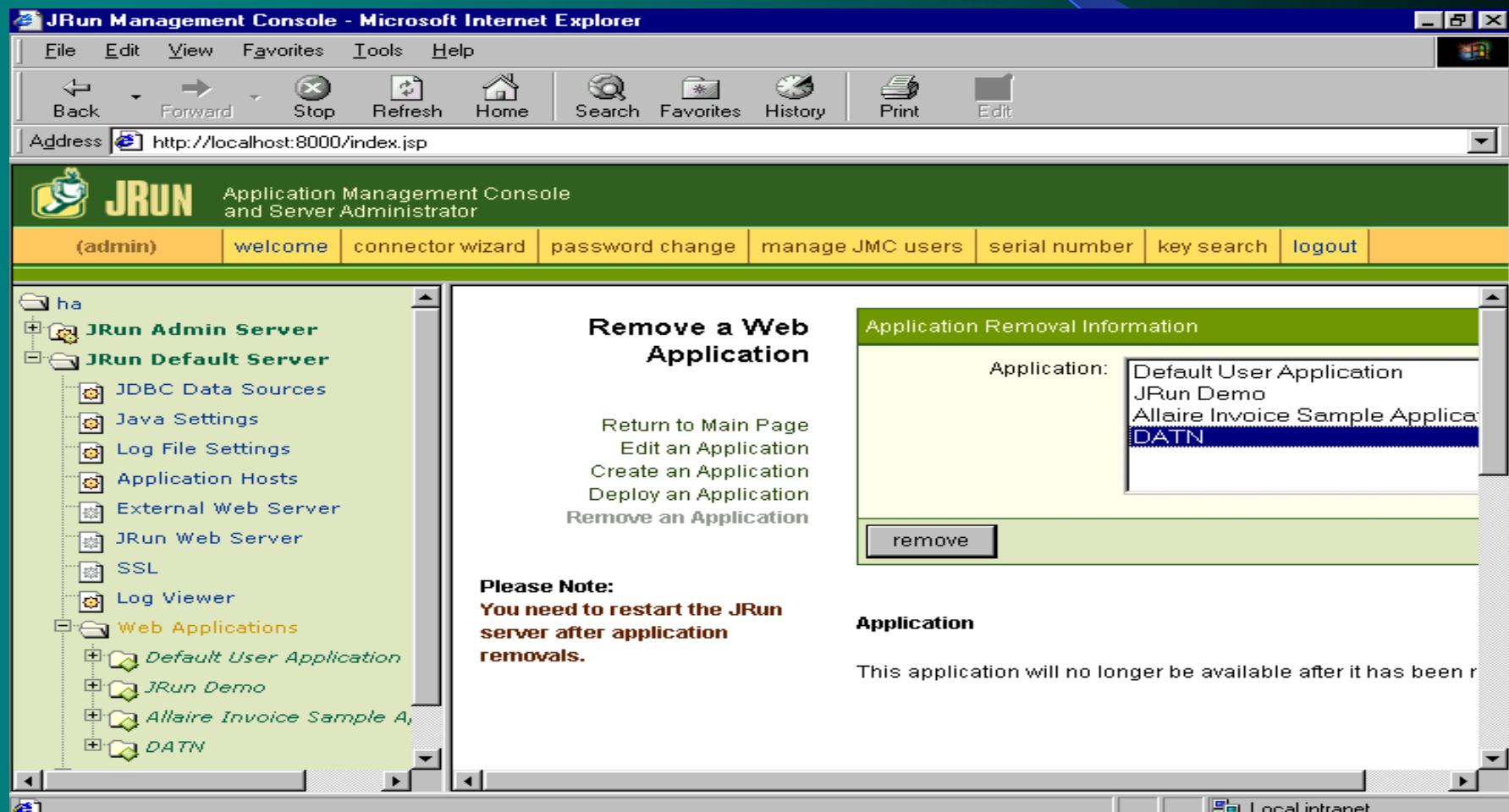
Do nhu cầu sử dụng đôi khi bạn tạo ra một ứng dụng web, nhưng có thể sau một thời gian sử dụng bạn không cần dùng đến nó nữa, ta có thể xóa nó đi. Để xóa một ứng dụng web:

- + Chọn Web Application từ màn hình quản trị của Jrun
- + Tiếp đến chọn Web Application. Jrun sẽ hiển thị danh sách các ứng dụng ta đã đăng ký hoặc tạo ra trước đó.
- + Để loại bỏ ứng dụng Web, ta chọn mục Remove Application.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.3 Cập nhật một ứng dụng Web

Sau khi tạo ra một ứng dụng Web nếu cần bổ sung hoặc thay đổi, ta có thể cập nhật lại nội dung ứng dụng theo các bước sau:

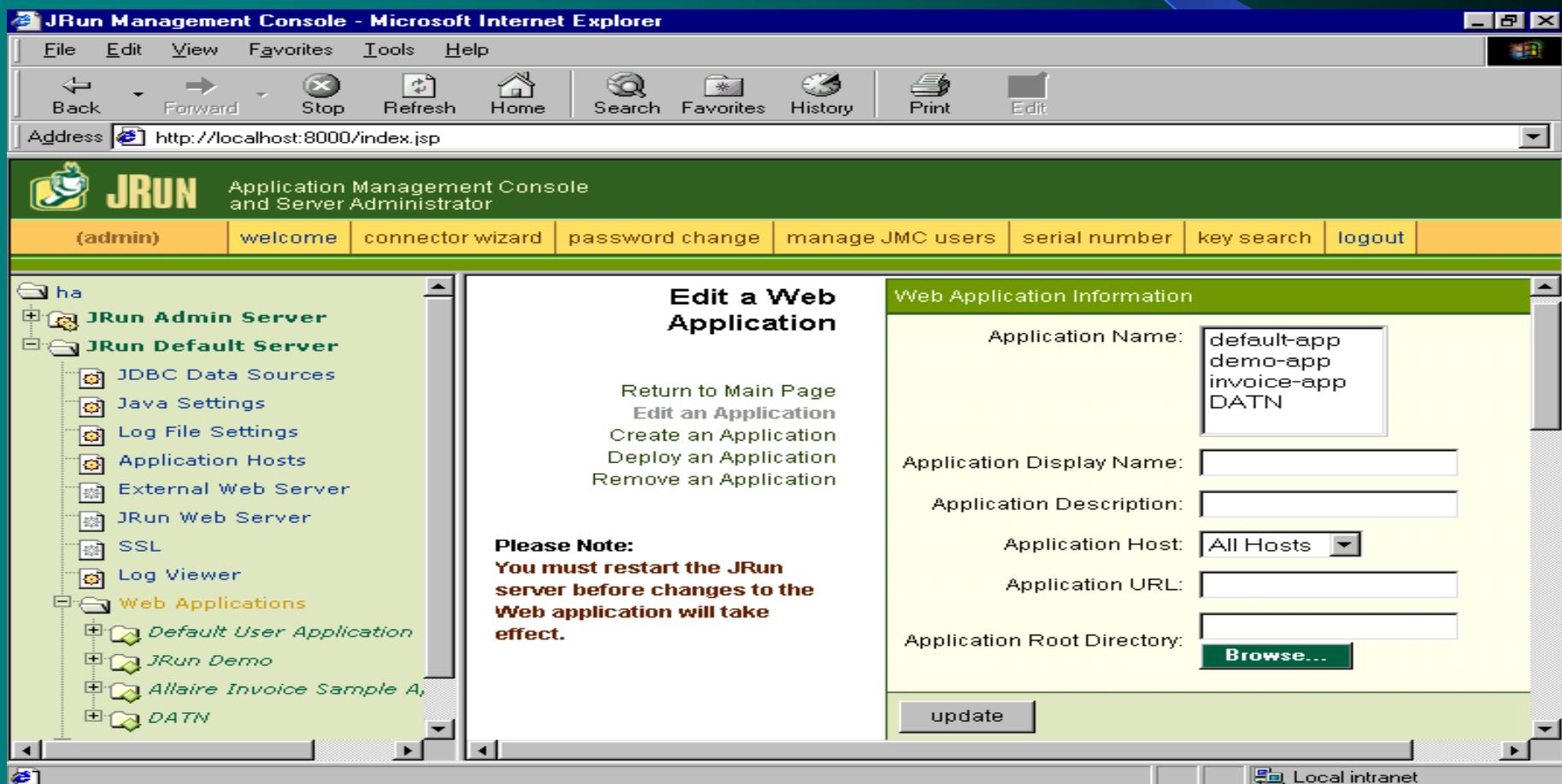
- + Chọn Web Application từ màn hình quản trị của Jrun Web Server.
- + Jrun sẽ hiển thị danh sách các ứng dụng đã đăng ký trước đó.
- + Chọn mục Edit Application.
- + Khi Jrun hiển thị danh sách ứng dụng, ta cần chọn lấy ứng dụng muốn cập nhật.
- + Thông tin ứng dụng sẽ hiện ra để ta bổ sung hoặc sửa đổi.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

3.2 Cấu hình và tạo ứng dụng Web (Web Application)

3.2.3 Cập nhật một ứng dụng Web



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

4. Tạo và quản lý user trong Jrun Web Server

Đối với Jrun Web Server ta có thể tạo ra một số thành viên giới hạn cho phép truy cập trình chủ Web Server. Ta có thể thực hiện theo các bước sau :

- + Chọn mục manage JMC users từ trang quản trị của Jrun.
- + Nhập tên user và password của thành viên đó. Mỗi lần ta chỉ cần đăng ký cho hai thành viên.
- + Ta chọn nút Update JMC Uesrs để cập nhật user vào cơ sở dữ liệu của Jrun.

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

4. Tạo và quản lý user trong Jrun Web Server

The screenshot shows the JRun Management Console interface within a Microsoft Internet Explorer window. The title bar reads "JRun Management Console - Microsoft Internet Explorer". The address bar shows "http://localhost:8000/index.jsp". The main content area is titled "Manage JMC Users". On the left, a navigation tree includes "JRun Admin Server", "JRun Default Server", "JDBC Data Sources", "Java Settings", "Log File Settings", "Application Hosts", "External Web Server", "JRun Web Server", "SSL", "Log Viewer", and "Web Applications" (with sub-options for "Default User Application", "JRun Demo", "Allaire Invoice Sample A", and "DATN"). The central panel contains instructions for managing users, a note about leaving passwords as "*****" if no changes are needed, and a note about trimming leading and trailing spaces in user names. To the right, a table titled "Currently Available JMC Users" lists two users with checkboxes for selecting multiple rows. The table has columns for "User Name", "Password", and "Permissions". The "Permissions" column shows "All servers", "JRun Admin Server", and "JRun Default Server" for both users. A "Update JMC Users" button is at the bottom of the table. The bottom status bar shows "Local intranet".

	User Name	Password	Permissions
<input type="checkbox"/>			All servers JRun Admin Server JRun Default Server
<input type="checkbox"/>			All servers JRun Admin Server JRun Default Server

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

I. Cấu Hình Trình Chủ Web Server

5. Kiểm tra lỗi thông qua Log Files

Thông thường bất kỳ lỗi nào xảy ra đối với Jrun đều được lưu vào file logs. Những file logs này nằm trong thư mục [Jrun_Home]\logs.

Mỗi file log lưu lại những sự kiện và lỗi phát sinh khác nhau. Thông thường có 3 file log sau :

- + default-err.log : file này lưu lại những lỗi của Jrun xảy ra trong quá trình dịch
- + default-even.log : lưu lại những lỗi trong quá trình thực hiện truy xuất tài nguyên của Java.
- + default-out.log : lưu lại những lỗi liên quan đến môi trường bên ngoài. Ví dụ : Khi bạn gửi thông tin đến một địa chỉ không có thì lỗi sẽ được ghi vào default-out.log.

- *Tóm lại* : Để theo dõi quá trình làm việc của Jrun, ta có thể dựa vào nội dung của file log. File log là nơi giúp ta phát hiện và khắc phục

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Poll

1. Giới thiệu :

- + Nhu cầu về kết nối cơ sở dữ liệu trung tâm để truy vấn dữ liệu trong các ứng dụng của Web ngày càng tăng.
- + Các hệ cơ sở dữ liệu phải tiêu tốn rất nhiều tài nguyên để duy trì liên tục kết nối trong quá trình truy vấn.
- + Ta chỉ nên mở kết nối truy vấn đến cơ sở dữ liệu chỉ khi nào thật sự cần thiết. Một khi đã hoàn tất truy xuất, bạn phải trả lại kết nối cho người dùng khác truy cập. Mặc dù vậy, mỗi một kết nối đến cơ sở dữ liệu đều phải đòi hỏi có thời gian thiết lập, phân bổ tài nguyên,...
- + Các trình chủ Web server đa ra cơ chế POLL cho phép các yêu cầu về kết nối đến cơ sở dữ liệu chỉ được mở

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Pool

1. Giới thiệu :

- + Vì vậy, các yêu cầu về truy vấn dữ liệu thường chỉ diễn ra trong thời gian ngắn
- + Những trang JSP tróc khi truy xuất dữ liệu sẽ yêu cầu hệ thống cung cấp một kết nối, hệ thống sẽ kiểm tra xem kết nối đã được thiết lập hay chưa. Nếu chưa có hệ thống sẽ tạo một kết nối cung cấp cho trang JSP sử dụng.
- + Trang JSP sau khi sử dụng xong không đóng kết nối mà trả lại kết nối cho hệ thống. Hệ thống sẽ lưu kết nối trở lại ngăn xếp hoặc hàng đợi.
- + Trong lời triệu gọi trang JSP ở phiên làm việc khác, nếu cần đến truy xuất cơ sở dữ liệu, hệ thống sẽ cung cấp kết nối đã có tróc đó mà không cần sử dụng tài nguyên của

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Poll

1. Giới thiệu :

Ta sẽ cài đặt một hệ thống như vậy cho cơ chế Pool (cơ chế chia sẻ kết nối). Trước hết hệ thống của ta cần hỗ trợ hai phương thức chính cho yêu cầu kết nối đó là cung cấp kết nối (getConnection()) và nhận lại kết nối khi trang JSP đã sử dụng (releaseConnection()). Khai báo giao tiếp interface mang tên IconnectionPool sẽ cung cấp thông tin này.

```
// Interface định nghĩa các phương thức cơ bản cung cấp và chia sẻ kết nối
public interface IconnectionPool
```

```
{ public Connection getConnection() throws SQLException;
  public void releaseConnection(Connection conn) throws SQLException;
}
```

Ta hình dung Poll tương tự cài đặt của ngăn xếp hoặc hàng đợi dùng để lưu trữ các phần tử cho mục đích sử dụng lại. Pool không đòi hỏi thứ tự lấy ra của các phần tử. Các phần tử lu trong Pool thường là đối tượng Connection

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Poll

2. Cơ Chế Pool Hỗ Trợ Bởi Web Server

- + Hiện nay hầu hết các trình chủ Web server đều cung cấp sẵn cơ chế Pool gọi là Data Source.
- + Việc quản lý và điều hành Pool sẽ do trình chủ Web server đảm nhiệm và bạn chỉ cần cung cấp cho trình chủ những thông tin tạo Pool là đủ.
- + Chúng ta sẽ cấu hình và sử dụng cơ chế Pool hỗ trợ bởi trình chủ Jrun. Trước hết ta mở trang <http://localhost:8100/> để vào trang quản trị của Jrun. Tiếp đến ta chọn Default Jrun Server\JDBC Data Source. Nhấn nút Add trang cung cấp thông tin để cấu hình cho DataSource hiển thị như sau :

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Poll

2. Cơ Chế Pool Hồ Trợ Bởi Web Server

JRun Management Console - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Edit

Address http://localhost:8000/index.jsp? Go Links >

JRUN Application Management Console and Server Administrator

(admin) welcome connector wizard password change manage JMC users serial number key search logout

JRun Default Server > JDBC Data Sources

JDBC data sources currently available to Web applications. Ensure that the JDBC driver is in the classpath of the **default** server. If not, you can use the Java Settings. In addition, testing the data source setting requires the JDBC driver to be in the classpath of the administrator user.

	Name	Driver	URL
	TaglibDataSource	com.pointbase.jdbc.jdbcUniversalDriver	jdbc:pointbase://embedded/sample
	Sample9bDataSource	com.pointbase.jdbc.jdbcUniversalDriver	jdbc:pointbase://embedded/sample
	KDPHUYENC	allaire.jrun.jdbc.JRunDriver	jdbc:jrun:sqlserver://127.0.0.1:1433;datebase=KDPHUYENC
	forum	allaire.jrun.jdbc.JRunDriver	jdbc:jrun:sqlserver://127.0.0.1:1433;datebase=forum
	Baomat	sun.jdbc.odbc.JdbcOdbcDriver	jdbc:odbc:Baomat

add

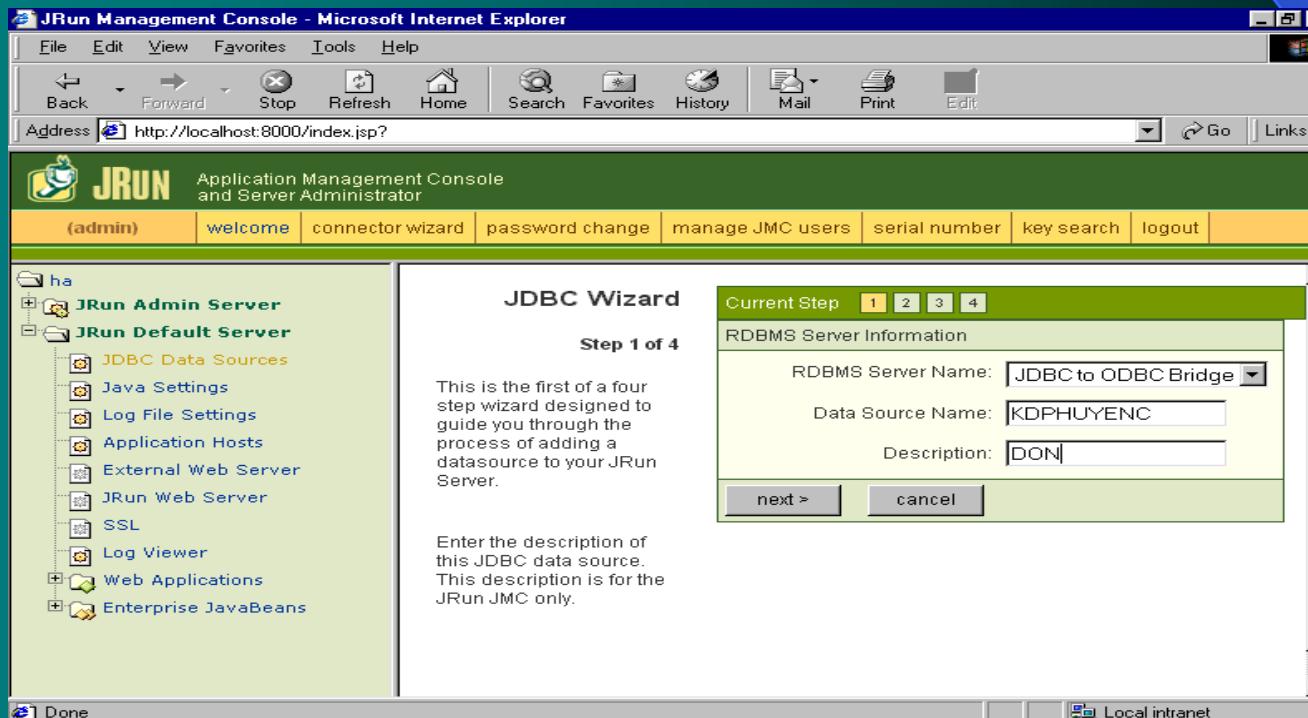
Done Local intranet

CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Pool

2. Cơ Chế Pool Hồ Trợ Bởi Web Server

+ Jrun cho phép sử dụng nhiều loại kết nối JDBC. Trong trường hợp này ta chọn mục danh sách JDBC to ODBC Bridge. Mục Data Source Name ta gõ vào tên *CSDL nguồn*. Đây là sõi chuỗi được dùng để tham chiếu đến Pool sau này.

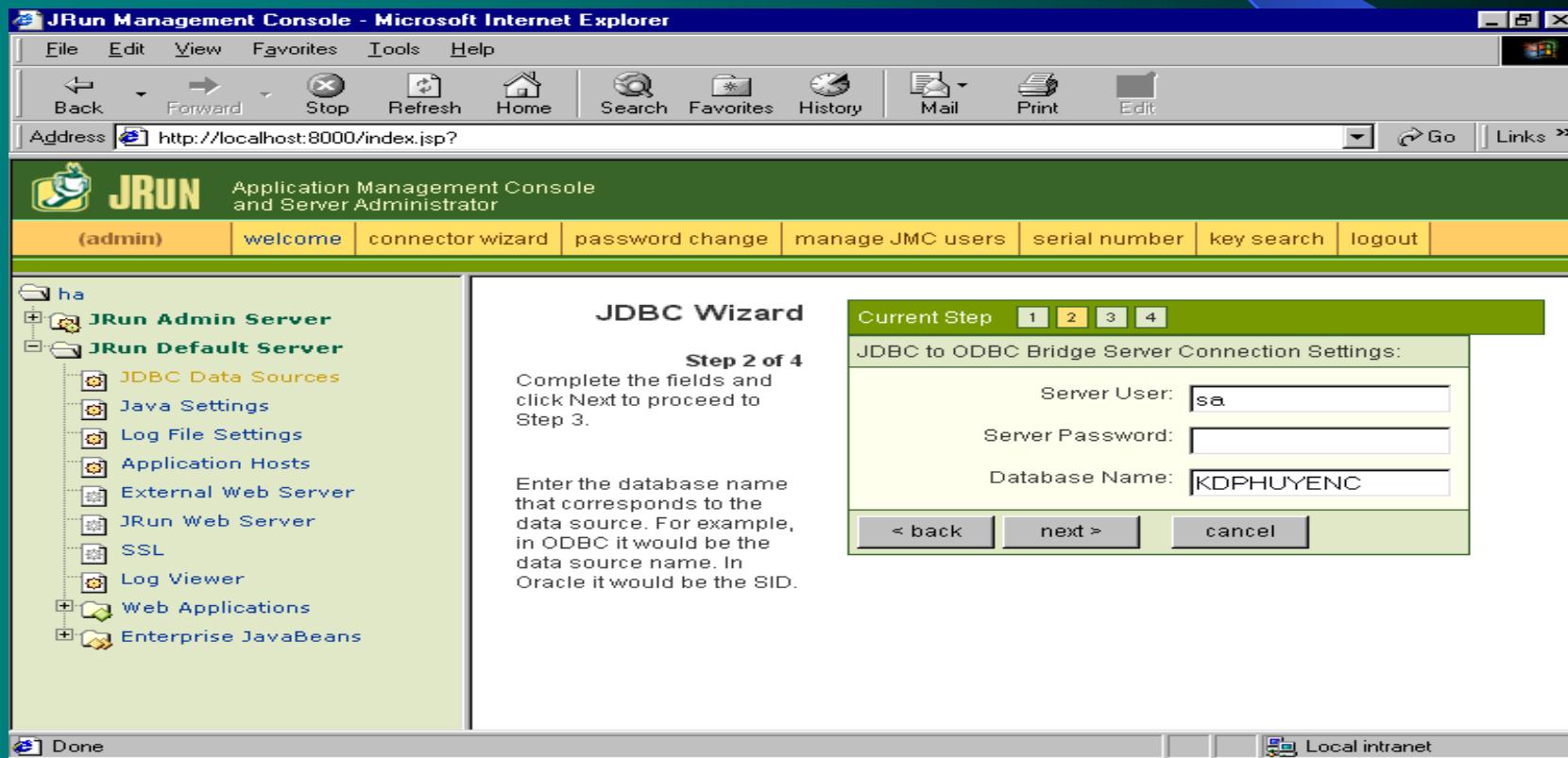


CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Pool

2. Cơ Chế Pool Hồ Trợ Bởi Web Server

Tiếp tục ta nhập vào các thông tin liên quan đến chuỗi kết nối cơ sở dữ liệu.

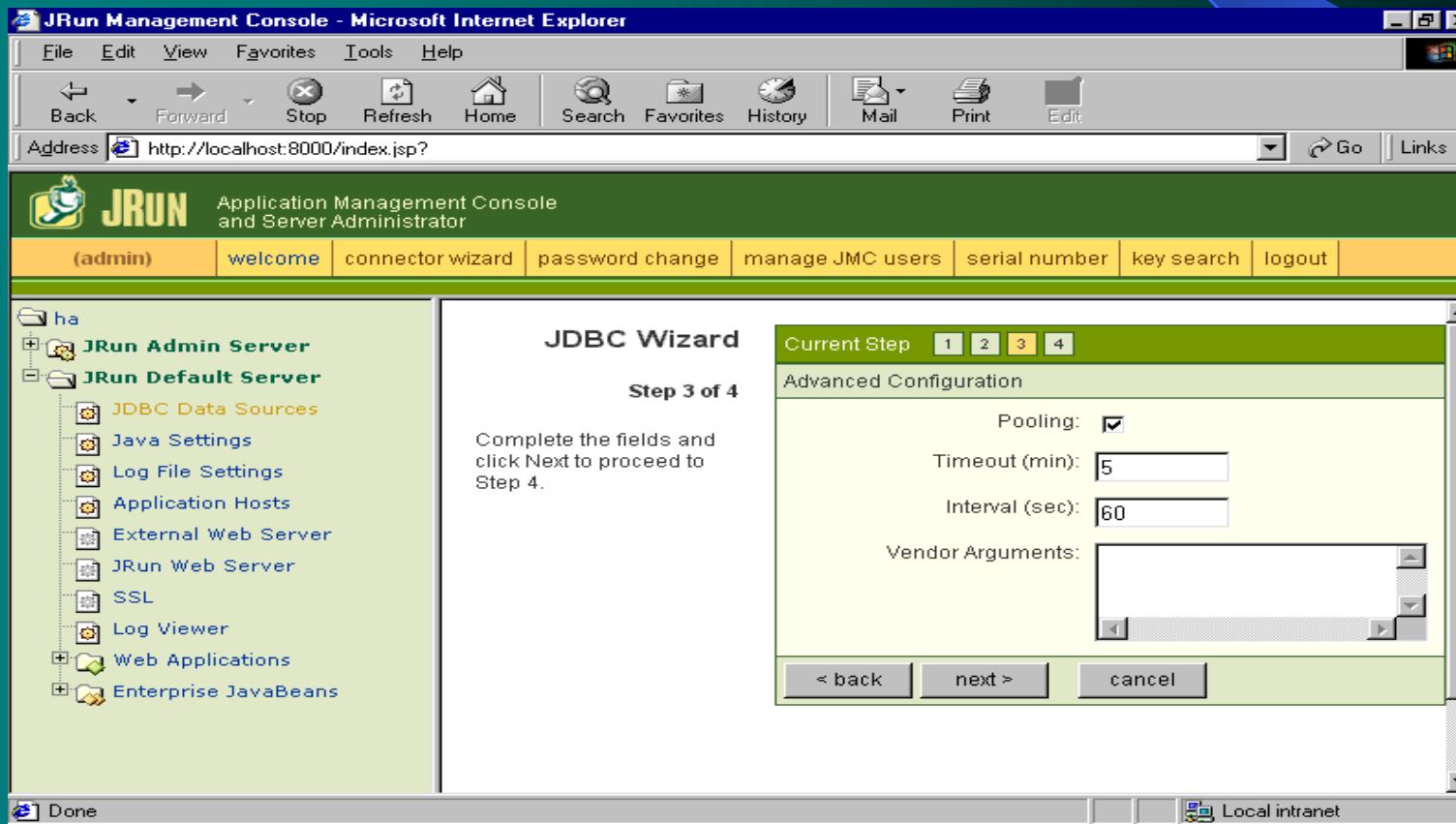


CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Poll

2. Cơ Chế Pool Hồ Trợ Bởi Web Server

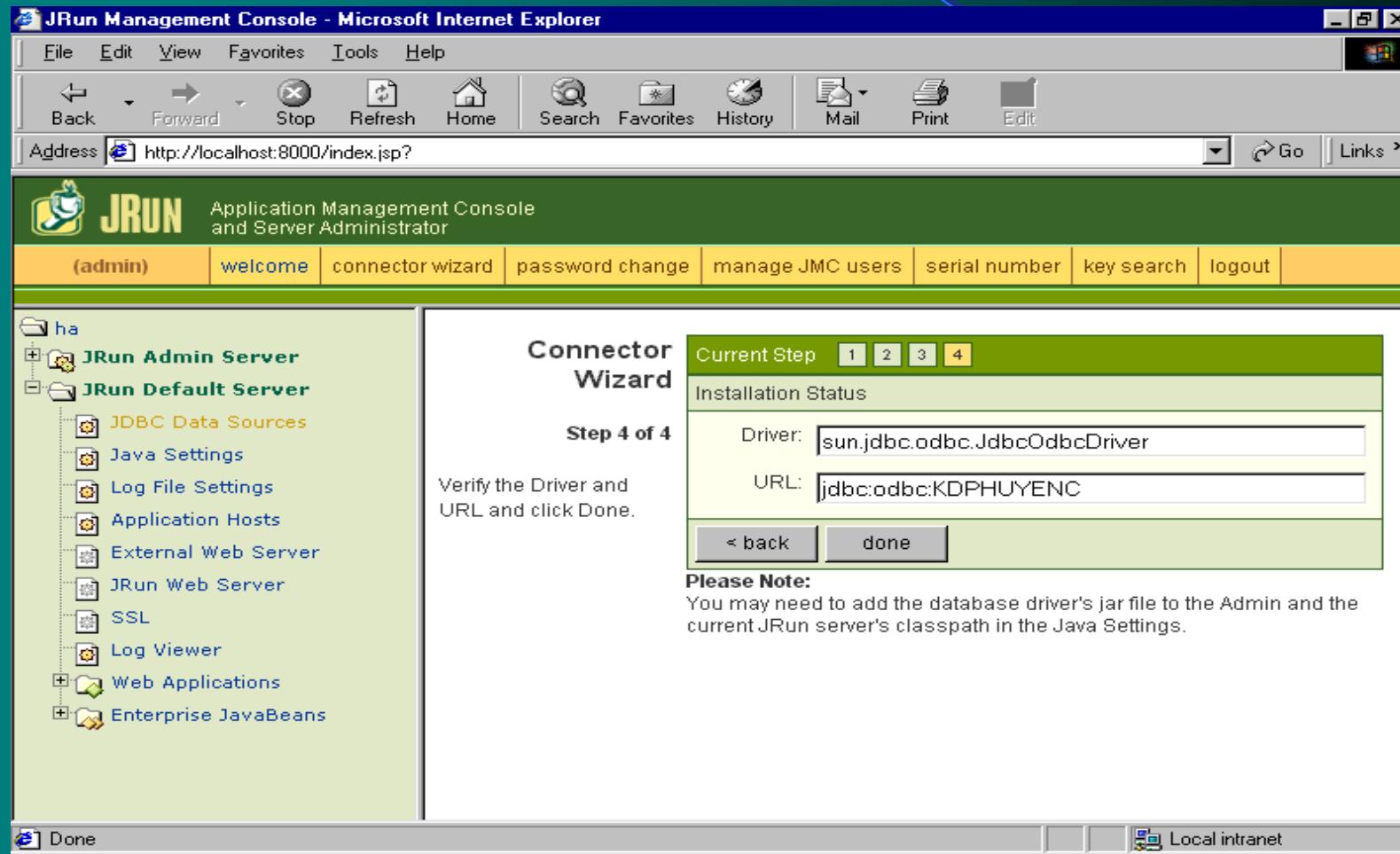
Tiếp tục nhấn Next Jrun sẽ hoàn tất những thao tác còn lại.



CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

II. Xây Dựng Cơ Chế Connection Pool

2. Cơ Chế Pool Hồ Trợ Bởi Web Server



Hêt !!!