

## Chương 1

# GIỚI THIỆU MẠNG INTERNET

## 1. SƠ LƯỢC VỀ MẠNG INTERNET

### 1.1. Lịch sử mạng Internet

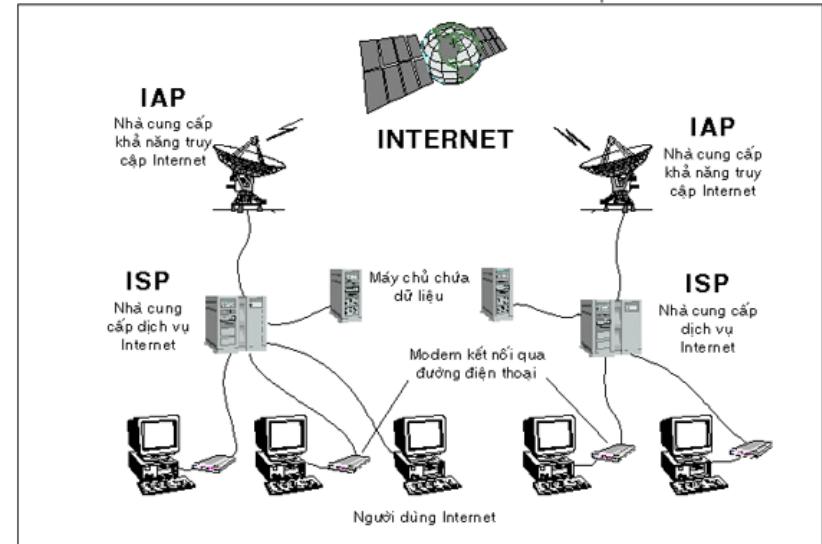
Mạng Internet xuất phát từ mạng ARPANET của Mỹ và ngày nay đã trở thành mạng toàn cầu kết nối hàng trăm triệu người trên thế giới. Mạng máy tính toàn cầu Internet có thể được xem như là mạng của tất cả các mạng (network of networks), trong đó người dùng tại bất cứ máy tính nào đều có thể truy cập tới các thông tin của các máy khác (nếu được phép). Mạng Internet ban đầu được biết dưới tên là ARPANET do tổ chức Advanced Research Projects Agency (ARPA) của Mỹ thiết lập năm 1969.

Mục đích chính của mạng này là tạo cơ sở cho các máy tính nghiên cứu của các trường đại học có thể kết nối và trao đổi với nhau. Thiết kế của ARPANET độc đáo ở chỗ là mạng vẫn có thể hoạt động khi một phần của nó bị phá hủy trong các trường hợp chiến tranh hoặc thiên tai.

Ngày nay mạng Internet là một mạng công cộng kết nối hàng trăm triệu người trên thế giới. Về mặt vật lý, mạng Internet sử dụng một phần của toàn bộ các tài nguyên của các mạng viễn thông công cộng đang tồn tại (public telecommunication networks). Về mặt kỹ thuật, mạng Internet sử dụng tập các giao thức gọi chung là TCP/IP (Transmission Control Protocol/Internet Protocol). Hai mô phỏng của công nghệ mạng Internet là intranet and extranet cũng sử dụng các giao thức này.

Sự ra đời của giao thức HTTP và HTML đã đánh dấu một bước ngoặc mới trong việc sử dụng Internet. Cho tới năm 1990 các dịch vụ cơ bản của Internet vẫn chỉ là e-mail, listserv, telnet, và ftp. Năm 1992, Tim Berners-Lee, một nhà vật lý học tại CERN đã phát triển các giao thức cho World Wide Web (WWW). Trong khi tìm kiếm cách để liên kết các tài liệu khoa học lại với nhau, anh ta đã tạo ra Hypertext Markup Language (HTML), một tập con của Standard Generalized Markup Language (SGML).

Từ một chuẩn cho các tài liệu văn bản, HTML ngày nay có thể chứa hình ảnh, âm thanh, video, và cho phép phát triển các ứng dụng thông qua Common Gateway Interface (CGI), ASP, JSP, PHP, Java Servlet, ....



Hình 1.1 – Mô hình mạng Internet

### 1.2. Một số kiến thức liên quan đến mạng Internet

#### 1.2.1. Địa chỉ IP (IP Address)

Địa chỉ IP là một con số 32-bit dùng để xác định đối tượng nhận và gửi thông tin trên Internet. Khi bạn yêu cầu một trang HTML hay gửi e-mail, địa chỉ IP của bạn sẽ được gửi đi cùng các gói tin đến địa chỉ IP của người nhận. Một khi người nhận được yêu cầu của bạn, họ sẽ căn cứ trên địa chỉ IP mà bạn gửi kèm để gửi kết quả trả về.

Để đơn giản hóa người ta phân con số này thành 4 con số 8 bit viết cách nhau bởi dấu chấm ". ". Vì mọi máy là một phần của mạng nên người ta chia địa chỉ IP thành 2 phần là phần mô tả mạng (network) mà máy đó thuộc về và phần mô tả máy (local host). Nếu tất cả các bit của vùng mô tả máy bằng 0, thì địa chỉ IP dùng để mô tả địa chỉ mạng (network address); Nếu tất cả các bit của vùng mô tả máy bằng 1, thì địa chỉ IP này chính là địa chỉ broadcast (broadcast

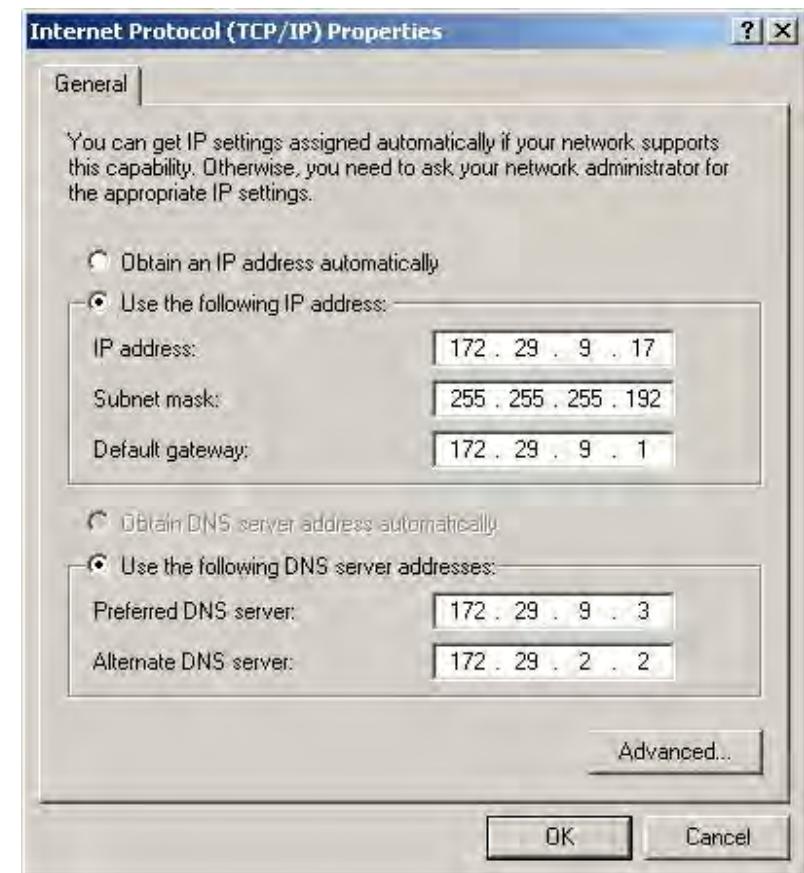
address); Nếu không thuộc hai trường hợp trên, địa chỉ IP này dùng để mô tả địa chỉ máy (host address).

Địa chỉ IP tự nó không chứa thông tin về phần nào mô tả mạng, phần nào mô tả máy mà thành phần subnet mask đi kèm với mỗi địa chỉ sẽ cung cấp thông tin này. Theo qui ước, vùng các bit 1 xác định vùng mô tả mạng, và vùng các bit 0 xác định vùng mô tả máy. Trong subnet mask chỉ gồm 2 dãy liên tục các bit 1 và dãy liên tục các bit 0 nằm liền tiếp nhau tính từ trái sang.

Các địa chỉ IP được phân thành 3 lớp A, B, C tùy theo giá trị của 3 bytes đầu tiên. Theo đó thì:

Class	Available Network Address	# of Networks	# of Hosts	Default Subnet
A	1 - 126	126	16.7 million	255.0.0.0
B	128.xxx – 191.xxx	16.384	65.536	255.255.0.0
C	192.xxx.xxx 223.xxx.xxx	–	2.097.152	255.255.255.0

Việc phân chia địa chỉ IP trên Internet do ICANN chịu trách nhiệm. Địa chỉ IP thường được quản lý bởi các nhà cung cấp dịch vụ Internet (ISP – Internet Service Provider). Các địa chỉ IP này thường được gọi nôm na là địa chỉ IP thực. Nếu bạn muốn thiết lập một website mà người dùng Internet có thể truy cập vào được, ít nhất máy chủ chứa website phải có địa chỉ IP thực. Nếu bạn không có địa chỉ IP thực, bạn phải sử dụng dịch vụ webhosting để thuê chỗ đặt website trên các máy chủ có địa chỉ IP thực được kết nối với Internet.



Hình 1.2 - Minh họa thiết lập giao thức TCP/IP

### 1.2.2. Tên miền (Domain name)

Tên miền có thể được xem như là tên giao dịch của công ty hay tổ chức trên Internet. Tên miền của các công ty thương mại thường có dạng **yourcompany.com**. Ví dụ, công ty Intel sẽ lấy tên là miền là **intel.com**; công ty Microsoft lấy tên miền là **microsoft.com**; ...

Việc đưa ra khái niệm tên miền giúp cho việc truy cập đến các tài nguyên trên Internet dễ dàng hơn. Việc ánh xạ qua lại giữa tên miền và địa chỉ IP của máy phục vụ được thực hiện bởi DNS Server.

Ví dụ một địa chỉ [www.intel.com](http://www.intel.com) cho ta một số thông tin sau: đây là địa chỉ của một máy thuộc tổ chức sở hữu tên miền **intel.com** (chính là công ty Intel). Địa chỉ IP của máy này sẽ là **192.102.198.160** (sử dụng chương trình **nslookup** để tra) và máy này có tên là “**www**” (thông thường là máy cung cấp dịch vụ web). Phần **“com”** trong tên miền mô tả mục đích của tổ chức (trong trường hợp này là “commercial” – thương mại) và được gọi là tên miền cấp 1 (top-level domain name). Phần ngay trước dấu “.” trong tên miền trên thông thường là tên của tổ chức (ví dụ như **intel**) được gọi là tên miền cấp 2 (second-level domain name). Tên miền cấp 3 thông thường được dùng để định nghĩa một máy phục vụ cụ thể nào đó và toàn bộ chúng sẽ được ánh xạ tới một địa chỉ Internet.

Tên miền cấp 1 bao gồm các mã quốc gia của các nước tham gia Internet được quy định bằng hai chữ cái theo tiêu chuẩn ISO - 3166 như Việt nam là VN , Anh Quốc là UK, Nhật bản là JP, ... và 7 lĩnh vực dùng chung, trong đó có 5 dùng chung cho toàn cầu và 2 chỉ dùng ở Mỹ.

Năm tên miền cấp 1 dùng chung là:

- **.com**: công ty thương mại (commercial).
- **.edu**: các trường học, tổ chức giáo dục (education).
- **.net**: các mạng (network)
- **.int**: các tổ chức quốc tế (international organizations)
- **.org**: các tổ chức khác (other organizations).

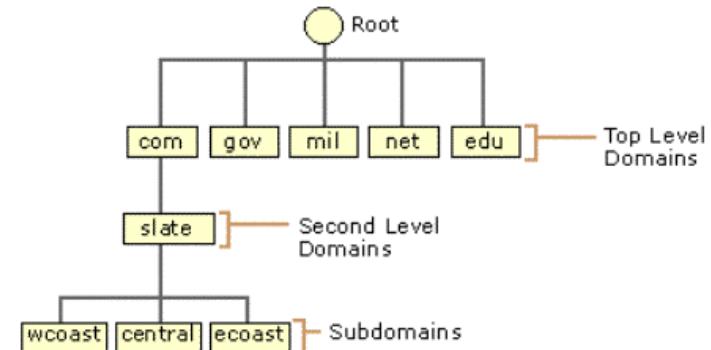
Hai tên miền cấp 1 chỉ dùng ở Mỹ là:

- **.gov**: các tổ chức chính phủ (government).
- **.mil**: các tổ chức quân sự (military).

Một địa chỉ IP có thể được ánh xạ cho nhiều tên miền. Điều này cho phép nhiều cá nhân, công ty và các tổ chức chia sẻ cùng một Internet server.

Do tầm quan trọng của tên miền nên một trong các bước đầu tiên của việc xây dựng website là thiết lập tên miền bằng cách mua từ các công ty được ủy quyền bán tên miền (<http://www.internic.net>). Các tên miền có đuôi là **.com** rất thông dụng trong các giao dịch quốc tế. Ngoài ra, các công ty Việt nam cũng thường hay lấy tên miền có đuôi là **.com.vn**. Để sở hữu tên miền, bạn phải trả chi phí

hàng năm. Nếu hết thời hạn sở hữu mà bạn không trả chi phí, tên miền này có thể bị người khác mua lại (và sẽ rất khó khăn khi bạn muốn đòi lại tên miền này từ tay người khác). Trên Internet hiện nay có nhiều người thường mua trước các tên miền của các công ty mà họ dự đoán sẽ nổi tiếng nhưng chưa lập trang web. Sau này khi các công ty này cần tên miền, họ sẽ bán lại với giá rất đắt. Tuy nhiên, bạn vẫn có thể mua trước tên miền mà chưa cần lập website.



**Hình 1.3 – Phân cấp của các tên miền**

## 2. CÁC DỊCH VỤ CƠ BẢN CỦA MẠNG INTERNET

### 2.1. World Wide Web (WWW)

Đây là dịch vụ thông dụng nhất trên Internet. Để sử dụng dịch vụ này, người dùng cần có một trình duyệt web thường được gọi là browser. Hai trình duyệt thông dụng nhất hiện nay là Internet Explorer của công ty Microsoft và Netscape Navigator của công ty Netscape.

Để truy cập vào một trang web, bạn cần phải biết địa chỉ (URL - Uniform Resource Locator) của trang web đó . Ví dụ, để truy cập vào trang web của công ty Microsoft, bạn gõ vào: <http://www.microsoft.com/ms.htm>. Do dịch vụ này sử dụng giao thức HTTP (Hypertext Transfer Protocol) nên mỗi địa chỉ trang web luôn được bắt đầu là **http://**



**Hình 1.4 – Dùng IE để truy cập trang web công ty Microsoft**

Trong mỗi trang web mà bạn truy cập vào, bạn có thể thấy được văn bản, hình ảnh, âm thanh, ... được trang trí và trình bày hết sức đẹp mắt. Ngoài ra, để có thể di chuyển tới các trang web khác, bạn có thể sử dụng các hyperlink (siêu liên kết). Do con trỏ chuột thường thay đổi hình dạng ngang qua một đối tượng có chứa hyperlink nên đây là cách đơn giản để nhận diện chúng.

Sự ra đời của www thực sự là một bước ngoặt lớn của mạng Internet bởi vì nó tạo cơ hội cho bạn truy cập đến một kho thông tin khổng lồ với hàng triệu triệu trang web. Điều này mở ra nhiều cơ hội và thách thức lớn cho công việc của bạn trong hiện tại và tương lai.

## 2.2. Thư điện tử - Email

Email (electronic mail) là dịch vụ trao đổi các thông điệp điện tử bằng mạng viễn thông. Các thông điệp này thường được mã hóa dưới dạng văn bản ASCII. Tuy nhiên bạn cũng có thể gửi các tập tin hình ảnh, âm thanh cũng như các tập tin chương trình kèm theo email. Email là một trong những dịch vụ nguyên thủy của Internet và được sử dụng rất rộng rãi. Chiếm phần lớn thông lượng trên mạng Internet là email.

Giao thức thường dùng để gửi/nhận email là SMTP(Simple Mail Transfer Protocol)/POP3 (Post Office Protocol 3).

Để sử dụng dịch vụ email, bạn cần phải có:

- Địa chỉ email. Một địa chỉ email thường có dạng [name@domainname](mailto:name@domainname). Ví dụ, trong địa chỉ email [duyld@yahoo.com](mailto:duyld@yahoo.com), duyld đóng vai trò là tên, yahoo.com là tên miền.
- Địa chỉ email được quản lý bởi mail server. Tại Việt Nam, các nhà cung cấp dịch vụ email thường là các ISP như VDC, FPT, SaigonNet. Do đó, tên miền trong các địa chỉ email của bạn thường là: hcm.vnn.vn, hn.vnn.vn, hcm.fpt.vn, fpt.vn, saigonnet.vn, ... Tuy nhiên, có rất nhiều website trên Internet cung cấp dịch vụ email miễn phí. Thông dụng nhất vẫn là Yahoo, Hotmail, ...
- Tên đăng nhập và mật khẩu để truy cập vào hộp thư: Điều này đảm bảo rằng chỉ có bạn mới có thể đọc và gửi các thư của chính mình.

## 2.3. Truyền, tải tập tin - FTP

FTP (File Transfer Protocol) là dịch vụ dùng để trao đổi các tập tin giữa các máy tính trên Internet với nhau. FTP thường được dùng để tải các trang web từ những người thiết kế đến các máy chủ. Nó cũng thường được dùng để tải các chương trình và các tập tin khác từ các máy chủ trên mạng.

Một số chương trình FTP với giao diện đồ họa thân thiện hữu ích hiện nay là:

- WS\_FTP (<http://www.ipswitch.com>)
- CuteFTP (<http://www.cuteftp.com>)
- FTP Explorer (<http://www.winsite.com>)

## 2.4. Tán gẫu - Chat

Dịch vụ tán gẫu cho phép người dùng có thể trao đổi trực tuyến với nhau qua mạng Internet. Cách thông dụng nhất là trao đổi bằng văn bản. Nếu đường truyền tốt, bạn có thể trò chuyện tương tự như nói chuyện điện thoại.

Các chương trình hỗ trợ tán gẫu thông dụng hiện nay là:



- AOL Instant Messenger



- Yahoo Messenger

Các chương trình này không những hỗ trợ tán gẫu qua văn bản thông thường mà còn hỗ trợ trò chuyện bằng âm thanh (voice chat) và hình ảnh (webcam)

### 3. KHAI THÁC CÁC TÀI NGUYÊN CỦA MẠNG INTERNET

#### 3.1. Một số website hữu ích

##### 3.1.1. Tra cứu thông tin

**Yahoo (<http://www.yahoo.com>)**

**Google (<http://www.google.com>)**

**Altavista (<http://www.altavista.com>)**

### 3.1.2. Thư viện phần mềm

ZDNet (<http://www.zdnet.com/downloads/>)

The screenshot shows the ZDNet Downloads homepage. It features a search bar at the top and a sidebar with news, offers, and links to PC Remote Access, Internet, Business & Finance, and Home & Education. The main content area displays a "Hot File of the Day" (Paint Shop Pro With Animation Shop 7.04) and a "Today in Downloads" section. Categories include Utilities, Games, MP3 & Audio, Desktop Enhancements, Business & Finance, Multimedia & Design, and Home & Education.

CNet (<http://download.cnet.com>)

The screenshot shows the CNet Download.com homepage. It features a search bar and a sidebar with links to CNET tech sites. The main content area displays a "Today on Download.com" message about the September 11 attacks, followed by sections for MP3 & Audio, Web Authoring, and Software Development.

### 3.1.3. Nghiên cứu, khoa học, giáo dục

Codeguru (<http://www.codeguru.com>)

The screenshot shows the CodeGuru homepage. It features a banner for EarthWeb, developer.com, and codeguru. The main content area includes a search bar, a sidebar for CodeGuru Sites (Visual C++/MFC, C#/.NET, Visual Basic), and a list of "Last 5 Visual C++/MFC Articles".

CodeProject (<http://www.codeproject.com>)

The screenshot shows the CodeProject homepage. It features a banner for Xtreme Toolkit and integrates with existing MFC applications. The main content area includes a search bar, a sidebar for Options (Sign in, New User, Add to Favorites, Tell a friend, Newsletters), and sections for Top Articles updated in the last week and Last 10 updates at the Code Project.

## MSDN (<http://msdn.microsoft.com>)

## Expert exchange (<http://www.experts-exchange.com>)

## ProgrammersHeaven (<http://www.programmersheaven.com>)

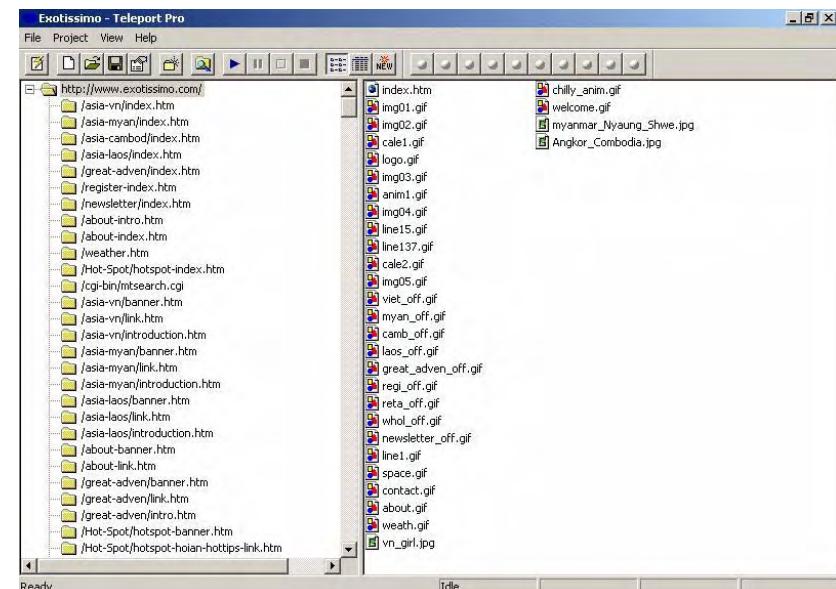
### 3.1.4. Mua bán trực tuyến

## Amazon (<http://www.amazon.com>)

**eBay (<http://www.ebay.com>)****Yahoo Shopping (<http://shopping.yahoo.com>)****3.2. Một số công cụ khai thác tài nguyên Internet****3.2.1. Teleport Pro (<http://www.tenmax.com>)**

Công cụ rất hữu ích cho phép tải toàn bộ website. Điều này giúp chúng ta xem website offline ngay trên máy của mình, tiết kiệm thời gian truy cập Internet.

Teleport Pro hoạt động hoàn toàn tự động, chương trình tự chia làm nhiều nhánh hoạt động đồng thời (multithreaded). Ngoài ra chương trình này còn có chế độ tự động tải theo lịch (schedule).

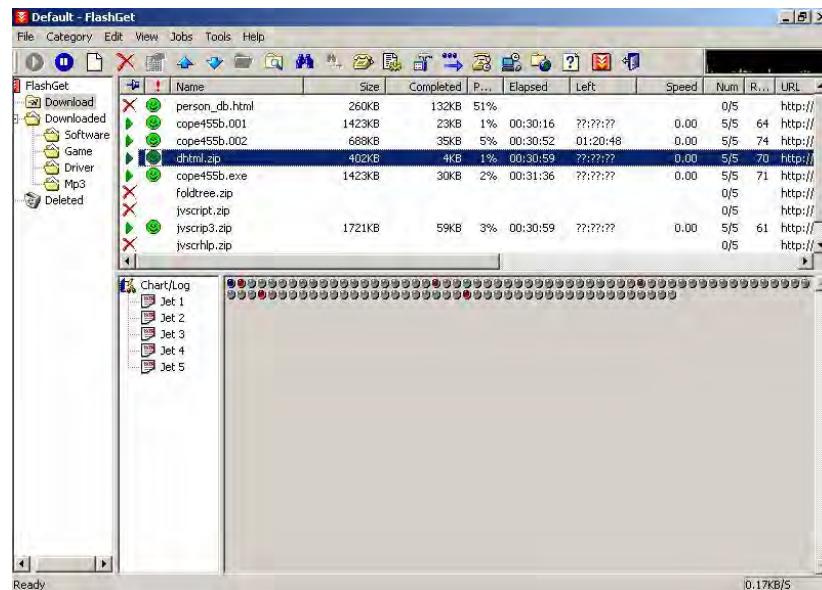
**Hình 1.5 - Màn hình giao diện của Teleport Pro****3.2.2. FlashGet (<http://www.flashget.com>)**

FlashGet là công cụ nhằm hỗ trợ cho tải các tập tin trên Internet và quản lý chúng một cách hiệu quả. Thông thường, việc tải một tập tin từ Internet không thành công do đường truyền chậm hay một sự cố nào đó.

FlashGet đặc biệt hiệu quả ở hai khả năng đó là tải các tập tin lớn và quản lý các tập tin đã tải về để phục vụ cho nhu cầu cập nhật hoặc tải lại sau này. FlashGet tải các tập tin bằng cách chẻ nhỏ (split) các tập tin thành các phần nhỏ và tải các phần này đồng thời.

Điều này tận dụng tối đa đường truyền rảnh và giảm thời gian tải một tập tin rất đáng kể. Trong trường hợp đường truyền chậm hoặc hay bị đứt quãng nửa chừng, FlashGet duy trì được trạng thái hiện hành các tập tin này để có thể tiếp tục tải một khi đường truyền hoạt động tốt trở lại.

FlashGet tích hợp với các trình duyệt để một khi bạn click vào một hyperlink có tải tập tin, chương trình FlashGet sẽ tự động bật lên và tự động tải tập tin về.



Hình 1.6 - Màn hình giao diện của FlashGet

### 3.2.3. Copernic (<http://www.copernic.com>)

Copernic là phần mềm tìm kiếm thông minh và đa năng vì nó cho phép người dùng điều khiển nhiều máy tìm kiếm (search engines) cùng một lúc để tìm kiếm thông tin mà mình yêu cầu. Copernic có thể sử dụng đến 1000 máy tìm kiếm nói tiếng và quan trọng nhất trên Internet.

Copernic tìm kiếm cùng một lúc trên tất cả các máy tìm kiếm trong nhóm nên có kết quả rất nhanh. Trong khi tìm kiếm nó tự động loại bỏ các kết quả trùng lặp. Khi có kết quả, người dùng có thể

nhấn nút Validate để loại bỏ các links; các websites không còn hoạt động và nút Refine để tinh chỉnh kết quả tìm kiếm hơn nữa.

Kết quả tìm thấy được trình bày trong cửa sổ bên dưới trong những ô theo thứ tự mức độ liên quan với chủ đề tìm kiếm, có tên và địa chỉ các trang web, thứ tự, ngày tìm được và hệ thống dò tìm nào đã tìm được, và vài dòng văn bản mô tả các trang web đó. Các font chữ và màu sắc trong các ô này có thể được điều chỉnh theo ý người dùng. Kết quả được tự động lưu lại để sau này xem lại hoặc để cập nhật (update), tức là tìm kiếm lại nhưng chỉ tìm những thông tin mới nên có kết quả rất nhanh.

Khi có kết quả, Copernic đề nghị ta xem bằng trình duyệt có sẵn được gọi là Results Explorer, hoặc khi ta nhấn vào nút Results. Lúc này kết quả tìm thấy được trình bày chi tiết hơn và đẹp hơn dưới dạng HTML. Trong Results Explorer, các địa chỉ tìm được có thể được duyệt, tải về một cách dễ dàng thuận tiện, và được lưu lại để sau này xem lại.

Nếu bạn sử dụng Internet Explorer, nút Search trên Tool Bar sẽ được Copernic sử dụng, và như thế trong khi lướt trên Internet bất cứ lúc nào bạn cũng có thể nhấn nút Search để tìm kiếm thông tin bằng Copernic.

## 4. MỘT SỐ THUẬT NGỮ THÔNG DỤNG

- **Internet:** Mạng máy tính toàn cầu kết nối các mạng máy tính khắp nơi trên thế giới. Tập các giao thức được dùng gọi chung là TCP/IP.
- **Intranet:** Mạng cục bộ có kiến trúc tương tự mạng Internet.
- **Website:** Tập hợp các trang web. Website của các tổ chức hay cá nhân trên mạng bao gồm tập hợp các trang web liên quan đến tổ chức này.
- **Webpage:** Là trang web. Có thể hiển thị các thông tin dưới dạng văn bản, hình ảnh, âm thanh, ...
- **Web browser:** Trình duyệt web, dùng để hiển thị các trang web. Các web browser thông dụng hiện nay là Internet Explorer của công ty Microsoft và Netscape Navigator của công ty Netscape.
- **Homepage:** Trang chủ hay còn gọi là trang nhà. Thường là trang đầu tiên (mặc định) khi truy cập một website.

- **Hyperlink:** siêu liên kết. Dùng để liên kết các trang web và dịch vụ của các website trên Internet.
- **IAP (Internet Access Provider):** Nhà cung cấp đường truyền Internet.
- **ISP (Internet Service Provider):** Nhà cung cấp dịch vụ Internet. Thông thường, bạn phải thông qua một ISP nào đó để đăng ký dịch vụ truy cập Internet qua mạng điện thoại. Các nhà cung cấp ISP hiện nay tại Việt Nam là VDC, FPT, SaigonNet, NetNam, ...
- **Search engines:** Máy tìm kiếm. Do số lượng các trang web trên Internet rất lớn nên các máy tìm kiếm sẽ hỗ trợ cho bạn khi cần tìm các trang web liên quan tới vấn đề mà bạn quan tâm trong tập hợp khổng lồ các trang web này. Các máy tìm kiếm thông dụng hiện nay là Yahoo, Google, Altavista, ...
- **HTTP, FTP, SMTP, POP3, ...:** Đây là các giao thức được dùng cho các dịch vụ web, ftp, email trên Internet.

## 5. MỘT SỐ CÂU HỎI THƯỜNG GẶP

### 5.1. Cách lưu một trang web từ Internet

**Cách 1:** Nếu bạn dùng trình duyệt Internet Explorer từ 5.0 trở lên hãy chọn chức năng Favorites/Add to Favorites sau đó đánh dấu chọn vào Make available offline. Khi đó toàn bộ trang Web cùng các tập tin đi kèm sẽ được tải về. Lần sau, bạn chỉ cần vào lại Favorites và chọn trang đã tải về.

**Cách 2:** Nếu bạn muốn lưu tất cả các thành phần của trang web (bao gồm hình ảnh, ...) dưới dạng một tập tin thì trong hộp thoại Save As, chọn dòng Save As Type là Web Archive, single file (\*.mht).

### 5.2. Làm thế nào để sau khi check mail xong, mail vẫn còn trên server

Sau khi check mail bằng Outlook Express mà mail vẫn không bị xóa trên mail server. Bạn vào Outlook Express chọn menu Tools/Accounts. Chọn account mà bạn đã setup để nhận email từ Yahoo. Chọn mục Properties/Advanced. Trong mục Delivery, check vào ô Leave a copy of messages on server.

### 5.3. Tải tập tin bằng email

Thông thường trong các mạng mà kết nối không được duy trì liên tục, việc tải một tập tin qua giao thức ftp hầu như rất khó khăn. Tuy nhiên bạn có thể vẫn tải được tập tin lưu ở ftp server thông qua email với sự trợ giúp của một FTPMail Server. Có thể tóm tắt như sau :

#### 5.3.1. Hoạt động của FTPMail Server:

Thay vì bạn phải kết nối trực tuyến tới FTP server (theo email của bạn là host.domain) và thực hiện các lệnh tải tập tin về (cách này thường ít khi thành công vì đường kết nối của chúng ta quá chậm), FTPMail server sẽ nhận yêu cầu của bạn qua email dưới dạng các lệnh của giao thức ftp, sau đó nó sẽ truy cập tới FTP server mà bạn yêu cầu để tải tập tin về. Cuối cùng nó sẽ chuyển tập tin sau khi tải về thành dạng email rồi gửi về cho bạn.

#### 5.3.2. Các bước để thực hiện:

Bạn cần phải biết địa chỉ của FTPMail Server để gửi yêu cầu đến đó, ví dụ bạn có thể sử dụng địa chỉ sau [ftpmail@ftp.sunet.se](mailto:ftpmail@ftp.sunet.se). Các địa chỉ khác có thể là: [ftpmail@doc.ic.ac.uk](mailto:ftpmail@doc.ic.ac.uk), [ftpmail@ieunet.ie](mailto:ftpmail@ieunet.ie), [ftpmail@ftp.uni-stuttgart.de](mailto:ftpmail@ftp.uni-stuttgart.de) , [bitftp@vm.gmd.de](mailto:bitftp@vm.gmd.de).

Bạn phải tìm hiểu cách viết các lệnh ftp để yêu cầu FTPMail server tải tập tin về: có thể làm điều này bằng cách gửi email tới FTPMail Server, phần body gõ help. Các lệnh cơ bản là OPEN, DIR, GET, QUIT, ...

Đối với các tập tin có kích thước lớn, FTPMail Server sẽ tự động "chè" (split) ra thành các mảnh nhỏ và kết quả là bạn sẽ nhận được rất nhiều email kết quả. Để có được tập tin mà bạn đã yêu cầu, cần phải "nối" (merge) các kết quả này lại. Nếu sử dụng Outlook Express, chương trình này sẽ tự động làm cho bạn.

Các FTPServer Mail thường rất bận, nên sau khi gửi yêu cầu đi bạn phải kiên nhẫn chờ kết quả trả về (nếu chậm có thể vài ngày).

Ví dụ sau minh họa việc tải một tập tin thông qua FTP Mail Server bằng cách gửi email đến địa chỉ [ftpmail@ftp.sunet.se](mailto:ftpmail@ftp.sunet.se) với nội dung sau:

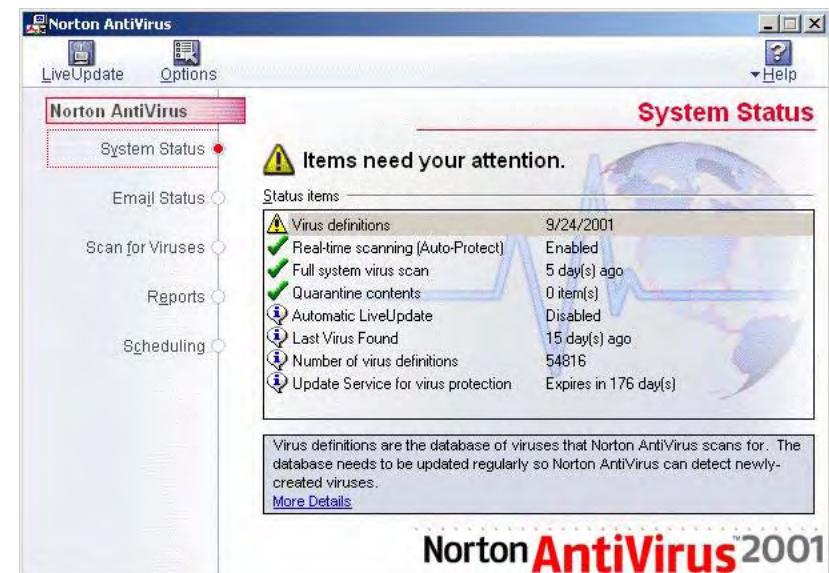
```
open ftp.ets.org
cd /pub/toefl
mode binary
uuencode
get 993007listb.pdf
quit
```

Lệnh open đầu tiên sẽ yêu cầu FTP Mail Server kết nối với FTP Server ftp.ets.org. Lệnh tiếp theo cd /pub/toefl yêu cầu chuyển đến thư mục có tập tin cần tải. Lệnh mode binary yêu cầu server truyền dữ liệu theo dạng nhị phân thay vì dạng văn bản ASCII. Lệnh uuencode yêu cầu server mã hóa dữ liệu trước khi truyền đi. Lệnh get yêu cầu server chuyển cho tập tin cần lấy về và lệnh quit chấm dứt kết nối với server ftp.ets.org. Sau khi FTP Mail Server kết nối với FTP Server để thực hiện các lệnh trên xong, tập tin lấy về sẽ được FTP Mail Server chuyển về cho bạn qua đường email.

#### 5.4. Phòng chống các virus bằng Norton AntiVirus

Hiện nay, virus lây lan qua đường email rất nhanh. Do đó để có thể bảo vệ hệ thống của bạn khỏi bị các virus lây qua email, bạn nên cài đặt chương trình phòng chống virus thông dụng nhất hiện nay là Norton Antivirus. Đặc điểm của chương trình này là nó có thể kiểm tra virus toàn bộ các email tải về máy của bạn trước khi bạn đọc nó. Khi gặp virus, chương trình sẽ tự động thông báo hoặc diệt ngay tùy theo yêu cầu của bạn.

Do virus mới xuất hiện ngày càng nhiều, nên bạn phải thường xuyên tải các cập nhật mới nhất về các virus mới tại website của công ty Symantec sở hữu phần mềm này. <http://www.symantec.com/avcenter/download.html> hoặc cho chạy chương trình Live Update để tự động cập nhật các thay đổi mới nhất của phần mềm này.



Hình 1.7 - Màn hình giao diện của phần mềm Norton Antivirus 2001

#### 5.5. Phòng chống virus lây lan qua đường email

Đây là một mẹo vặt giúp tránh lây virus qua đường email khi máy của bạn bị nhiễm virus. Lưu ý là nó không giúp cho máy bạn tránh bị nhiễm worm nhưng nó giúp ngăn chặn việc sử dụng số địa chỉ email của bạn để lây tiếp cho những người khác.

Đầu tiên, mở Address Book ra và chọn New Contact tương tự như thêm vào một tên mới. Trong cửa sổ, thay vì đánh tên của bạn bè, gõ vào: !000 (dấu chấm than và 3 số không).

Ở cửa sổ bên dưới, nơi mà thay vì gõ địa chỉ email của bạn bè, gõ vào dòng chữ sau: WormAlert. Sau cùng, hoàn tất công việc bằng cách click Add, Enter và Ok

"!000" sẽ được đặt đầu tiên trong Address Book và nó được đánh số 1. Đây sẽ là "người" đầu tiên mà con Worm sẽ bắt đầu lây. Nhưng người này lại có địa chỉ email là "WormAlert", không đúng quy cách nên nó không thể phát tán thông qua việc gửi cho những người tiếp theo có địa chỉ trong Address Book của bạn.

Khi email không được gửi đi, bạn sẽ nhận lại một thông báo ngay lập tức trong Inbox. Như vậy, nếu bạn nhận được một email

nói rằng "Email addressed to WormAlert could not be delivered", bạn biết ngay là "Worm" đang nằm ngay trong máy bạn.

## 5.6. Sử dụng Internet trong học tập

Khi sử dụng các máy tìm kiếm như Yahoo, Altavista, ... trong việc tìm các trang web, một trong các những điều bất tiện cho chúng ta là số kết quả trả về quá nhiều.

Để có thể tìm kiếm hiệu quả hơn trong lĩnh vực tin học, bạn có thể vào một trong 2 site giúp tra cứu thuật ngữ tin học đó là: <http://www.webopedia.com> và <http://www.whatis.com>. Ngoài việc giải thích nghĩa của mỗi thuật ngữ Tin học, các bạn sẽ tìm được ở đây các liên kết tới các site chứa thông tin liên quan. Ví dụ, nếu bạn muốn tìm các site chứa các hướng dẫn lập trình, cũng như các kỹ thuật, ... về ASP, bạn hãy vào một trong hai site trên gõ từ cần tìm kiếm là ASP. Trong trang giải thích thuật ngữ về ASP, bạn sẽ nhận được các liên kết tới các site cần thiết để tìm hiểu về ASP.

## 5.7. Dùng tiếng Việt Unicode trong các email client

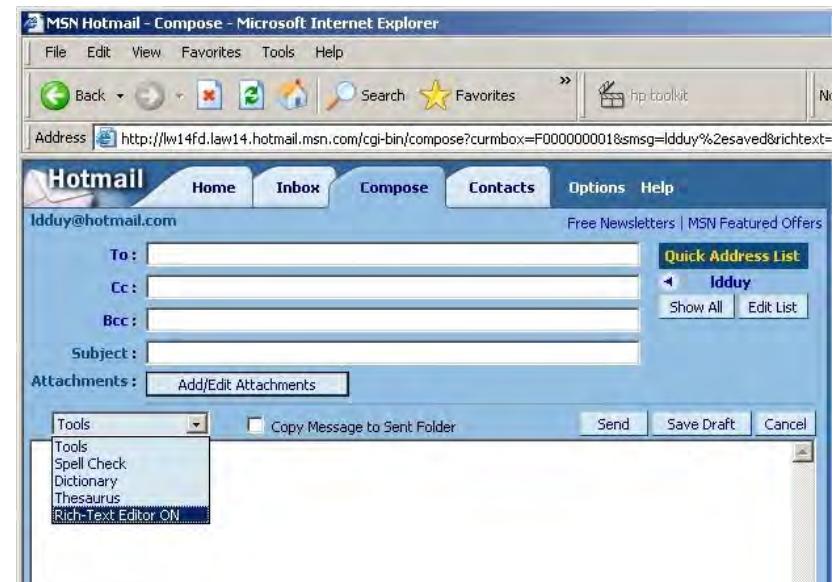
Để xem được tiếng Việt Unicode trong các email gửi đến các địa chỉ email Yahoo, Hotmail, bạn phải đặt Encoding của trình duyệt là UTF-8 (Chọn View/Encoding=UTF-8).



Hình 1.8 – Thiết lập Encoding là UTF-8 trong trình duyệt IE

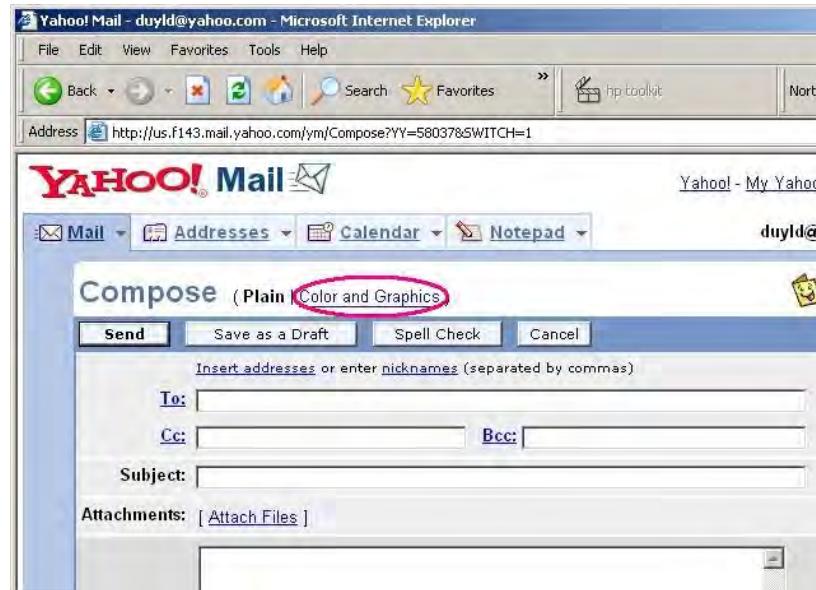
Để gửi email bằng tiếng Việt, trong các ứng dụng như Yahoo Mail, Hotmail, Outlook, Outlook Express, bạn phải thực hiện như sau:

- Trong Hotmail: Đặt chế độ **Rich Text Editor ON** trong mục chọn bên trái ngay trên phần nội dung của email



Hình 1.9 – Thiết lập chế độ hỗ trợ tiếng Việt Unicode trong Hotmail

- Trong Yahoo Mail: Đặt chế độ Color and Graphics trong mục Compose khi soạn thảo email.



**Hình 1.10 – Thiết lập chế độ hỗ trợ tiếng Việt Unicode trong Yahoo Mail**

## 5.8. Chat tiếng Việt trong Yahoo Messenger

Trên Windows NT/2000/XP, sử dụng UniKey 3.55 (<http://unikey.sourceforge.net/bdownload.php#uk355>) ta có thể chat bằng tiếng Việt Unicode trong Yahoo Messenger. Để thực hiện, hãy chọn bảng mã Vietnamese Locale CP 1258 trong UniKey.

Để dùng CP 1258 máy của bạn cần cài đặt hỗ trợ tiếng Việt của Windows Vietnamese locale. Với Windows XP, hãy Control Panel/Regional and Language Options/Languages/ Details. Sau đó, nếu trong hộp thoại "Text Services and Input languages" chưa thấy Vietnamese keyboard trong đó thì chọn "Add", sau đó chọn Input Language là Vietnamese.

# CÂU HỎI

1. Hãy cho biết các dịch vụ Internet thông dụng và các giao thức được sử dụng cho các dịch vụ này.
2. Tìm hiểu cách phân lớp trong địa chỉ IP. Hãy cho biết các địa chỉ IP của Việt Nam thuộc lớp nào.
3. Tại sao cần phải có sự ánh xạ giữa tên miền và địa chỉ IP? Hãy cho biết tên của dịch vụ thực hiện việc này.
4. Cho biết các tên miền được dùng thông dụng hiện nay.
5. Tìm hiểu một số công ty được ủy quyền cấp tên miền trên Internet.
6. Chỉ ra một lợi ích của mạng Internet mà bạn hay công ty của bạn đã hoặc có thể có được.

# THỰC HÀNH

## 1. MỘT SỐ THAO TÁC CƠ BẢN SỬ DỤNG MẠNG INTERNET

### 1.1. Kết nối

Để kết nối được mạng Internet, cần chuẩn bị các bước sau:

- Modem (internal hoặc external).
- Đường điện thoại.
- Đăng ký dịch vụ với các nhà cung cấp dịch vụ Internet để biết được số điện thoại truy cập, tên đăng nhập và mật khẩu. Tại Việt Nam, bạn có thể đăng ký với các nhà cung cấp sau VDC, FPT, SaiGonNet, .... Sau khi đăng ký xong, bạn sẽ được nhận một số tên đăng nhập, mật khẩu và số điện thoại mà bạn sẽ quay số từ máy của bạn vào máy chủ của nhà cung cấp dịch vụ.
- Đối với VDC, có thể sử dụng các dịch vụ VNN1260, VNN1269, VNN-1260P. Dịch vụ VNN1269 là dịch vụ được dùng khá phổ biến hiện nay với số điện thoại quay vào là **1269**, tên đăng nhập là **vnn1269**, mật khẩu là **vnn1269**.

Sau khi đăng ký xong, vào mục Network Connections để tạo một kết nối với nhà cung cấp dịch vụ. Tham khảo thêm tại: <http://support.vnn.vn/tailieu/#dichvu>

### 1.2. Sử dụng Internet Explorer để truy cập web

Sau khi kết nối Internet thành công, hãy sử dụng Internet Explorer để thực hiện các thao tác sau:

- Truy cập các website đã được đề cập ở trên;
- Lần theo các hyperlink để truy cập đến các trang liên kết.
- Lưu các trang trên lại trong Favorites. Lần sau, sử dụng Favorites để có thể chuyển đến nhanh chóng các trang web này.
- Refresh (yêu cầu lại) các trang web mỗi khi truy cập trang web gặp lỗi do đường truyền.

- Lưu các trang web lại dưới dạng các tập tin để lần sau có thể theo dõi trên máy.
- Dùng chức năng Work Offline để xem lại các trang web đã truy cập mà không cần kết nối lại Internet.
- Khai báo proxy server nếu máy kết nối Internet thông qua proxy.

## 2. THIẾT LẬP DỊCH VỤ EMAIL TRONG OUTLOOK EXPRESS

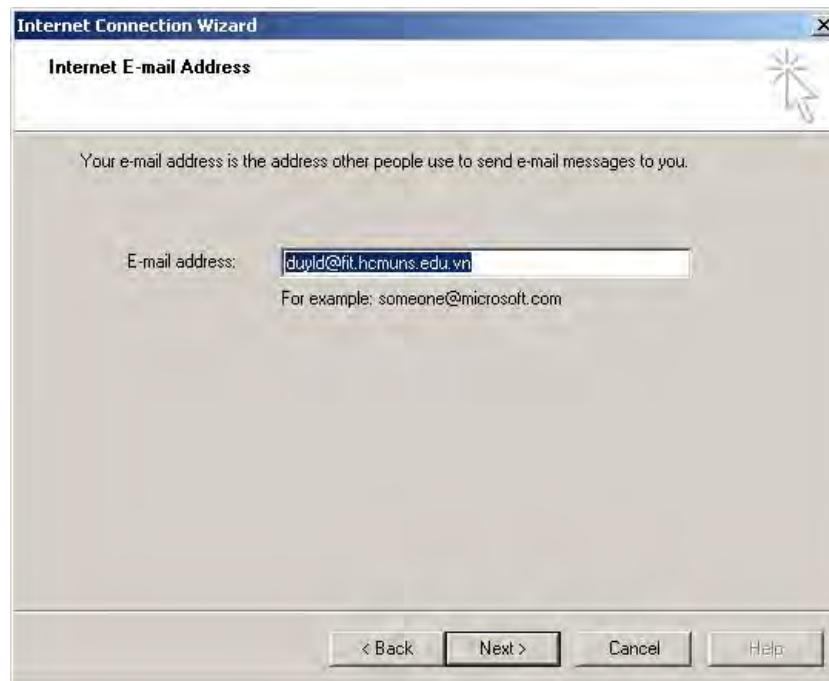
Để thiết lập dịch vụ này, bạn thực hiện các bước sau:

Khởi động Outlook Express. Vào mục Tools/Accounts.../Add/Mail

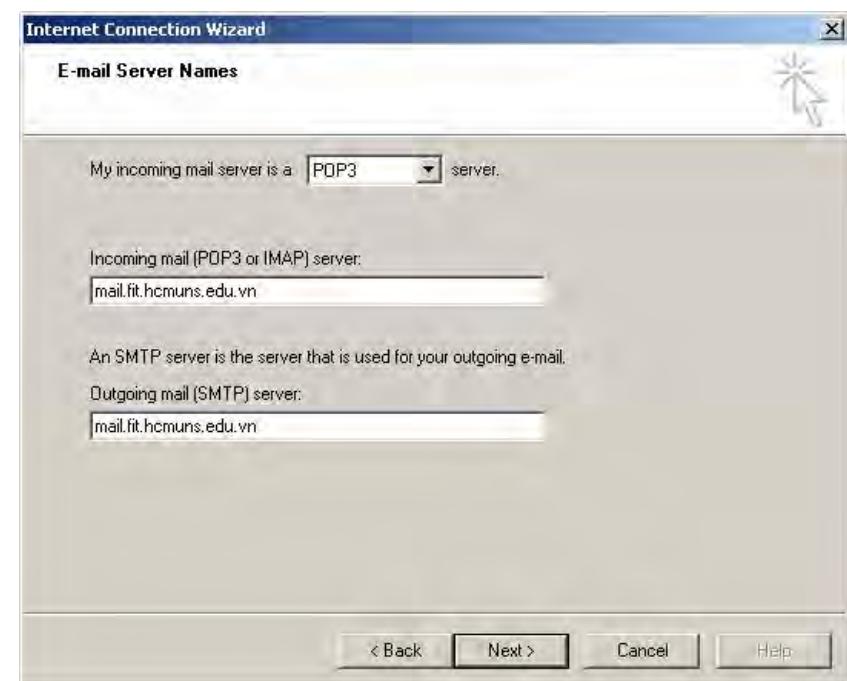
Màn hình sau sẽ hiện ra. Bạn điền vào Họ và tên. Tên này sẽ được dùng để hiển thị trong mục From trong các email mà bạn gửi đi.



Sau khi điền xong, bạn nhấn nút Next. Màn hình sau sẽ hiện ra. Bạn hãy điền vào địa chỉ email của mình.



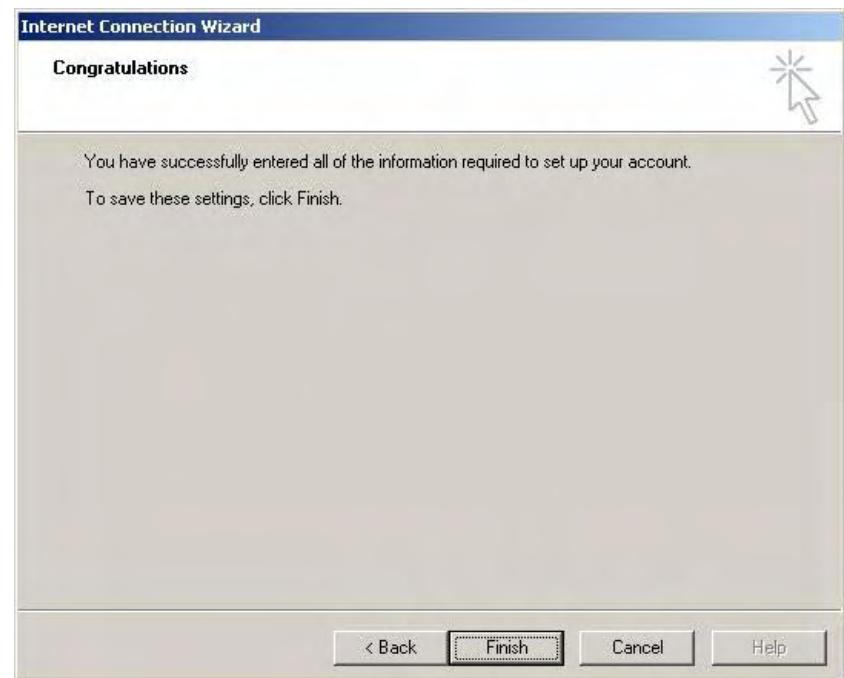
Sau khi điền xong, bạn nhấn nút Next. Màn hình sau sẽ hiện ra.  
Bạn hãy điền vào địa chỉ SMTP và POP3 mail server.



Sau khi điền xong, bạn nhấn nút Next. Màn hình sau sẽ hiện ra.  
Bạn hãy điền tên đăng nhập và mật khẩu dùng để truy xuất đến mail server.



Sau khi điền xong, bạn nhấn nút Next. Màn hình sau sẽ hiện ra.  
Bạn hãy nhấn nút Finish để kết thúc việc thiết lập.



### 3. ĐĂNG KÍ DỊCH VỤ EMAIL MIỄN PHÍ

- <http://mail.yahoo.com>
- <http://www.hotmail.com>
- <http://www.vol.vnn.vn>
- <http://mail.fptnet.com.vn/>
- <http://www.netsoft.vnn.vn/webmail/>

### 4. THAM KHẢO CÁC TRANG WEB SAU

#### 4.1. Các site bán hàng

- <http://shopping.yahoo.com>
- <http://www.amazon.com>
- <http://www.ebay.com>
- <http://www.barnesandnoble.com/>

#### 4.2. Các site giải trí

- <http://greetings.yahoo.com>
- <http://www.geraldstevens.com/>
- <http://www.1800flowers.com/>
- <http://www.123greetings.com/>
- <http://www.vnn.vn/ecards/>
- <http://www.fpt.vn/Postcard/main.asp>
- <http://www.confidantvcd.com>

#### 4.3. Các site thông tin và báo điện tử của Việt Nam

- <http://vnexpress.net/>
- <http://www.vnn.vn/>
- <http://www.vdc.com.vn/>
- <http://media.vdc.com.vn/>
- <http://www.nhandan.org.vn/>
- <http://www.sggp.org.vn/>
- <http://www.laodong.com.vn/>
- <http://www.nld.com.vn/>
- <http://www.tuoitre.com.vn>
- <http://website.fptnet.com.vn/>

#### 4.4. Tham khảo các site về lao động việc làm tại VN

- <http://www.vietnamworks.com/>
- <http://203.162.5.43/l2000/>
- <http://www.vietname-business.com/jobnld/>
- <http://www.jobsonline.saigonnet.vn/>
- <http://www.jobviet.com>

#### 5. CHIA SẺ ĐƯỜNG KẾT NỐI INTERNET

Một đơn vị (ví dụ như một công ty nhỏ, một quán cafe Internet, ...) muốn kết nối với Internet. Để tiết kiệm, đơn vị này chỉ muốn sử dụng một đường kết nối bằng điện thoại cho tất cả các thành viên muốn truy cập Internet. Giải pháp đề nghị như sau:

Thiết lập một mạng máy tính nối các máy của đơn vị. Các máy trạm cài đặt hệ điều hành Windows 9x, hoặc Windows 2000, sử dụng giao thức TCP/IP.

Cài đặt một phần mềm hỗ trợ ví dụ như WinGate (<http://www.wingate.com>) ngay tại máy có gắn modem, kết nối với Internet.

Thiết lập các thông số trong WinGate. Phần mềm này cung cấp các khả năng như sau:

- Cho phép chia sẻ kết nối Internet. Hệ thống sẽ tự động quay số mỗi khi có yêu cầu kết nối Internet và sẽ tự động ngắt khi không còn yêu cầu nào nữa.
- Cho phép kiểm soát các website truy cập. Bạn có thể cấm các thành viên truy cập các website mà bạn không cho phép.

## Chương 2

# THIẾT LẬP WEBSITE

### 1. CÀI ĐẶT WEB SERVER

WebServer là chương trình cung cấp dịch vụ www. Một webserver có thể phục vụ cho nhiều website. Port mặc định cho dịch vụ web là 80.

Có nhiều phần mềm hỗ trợ webserver. Trên môi trường của Microsoft, thông dụng nhất vẫn là IIS (Internet Information Server) với các phiên bản chạy trên hệ điều hành WindowsNT 4.0 và Windows 2000. Ngoài ra các máy sử dụng hệ điều hành Win9x, Microsoft cũng hỗ trợ cài đặt webserver thông qua phần mềm PWS (Personal Web Server).

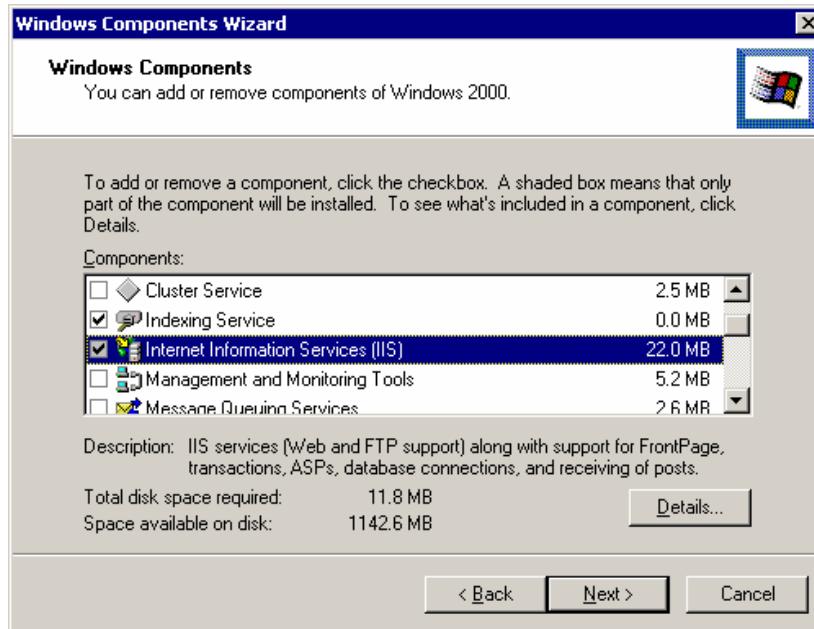
#### 1.1. Cài đặt IIS trên Windows/XP 2000 Server và Windows 2000/XP Professional

Để cài đặt IIS trên hệ điều hành Windows 2000 Server hay Windows 2000 Professional, bạn vào mục Start/Settings/Control Panel/Add Remove Programs. Chọn mục Add/Remove Windows Components.



**Hình 2.1 - Màn hình Control Panel/Add Remove Programs**

Trong cửa sổ Windows Components Wizard, chọn checkbox Internet Information Services.



**Hình 2.2 - Màn hình cài đặt các Windows Components**

Các dịch vụ mặc định được cài là Web Server, FTP Server, Virtual SMTP Server.

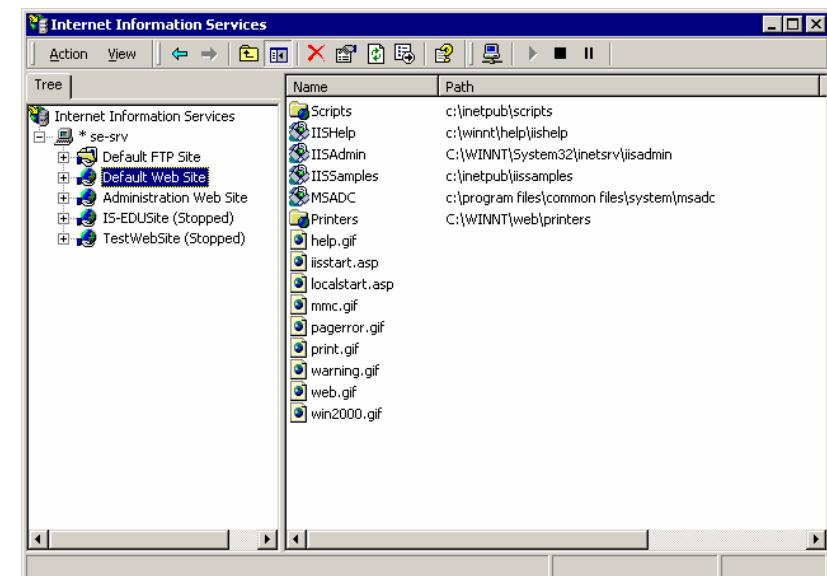
Sau khi cài đặt xong, các thư mục mặc định sau sẽ được tạo ra:

- \Inetpub\
- \Inetpub\wwwroot
- \SystemRoot\Help\iisHelp
- \SystemRoot\System32\inetsrv

*systemroot là thư mục system32 trong thư mục cài đặt Windows*

Để quản trị và cấu hình các dịch vụ này, bạn hãy chạy chương trình Internet Services Manager trong mục Start/Programs/Administrative Tools.

IIS cài đặt trên Windows 2000 Professional có một số giới hạn so cài đặt trên với Windows 2000 Server như chỉ được tạo một website, số kết nối bị giới hạn,...



**Hình 2.3 - Màn hình quản trị và cấu hình IIS**

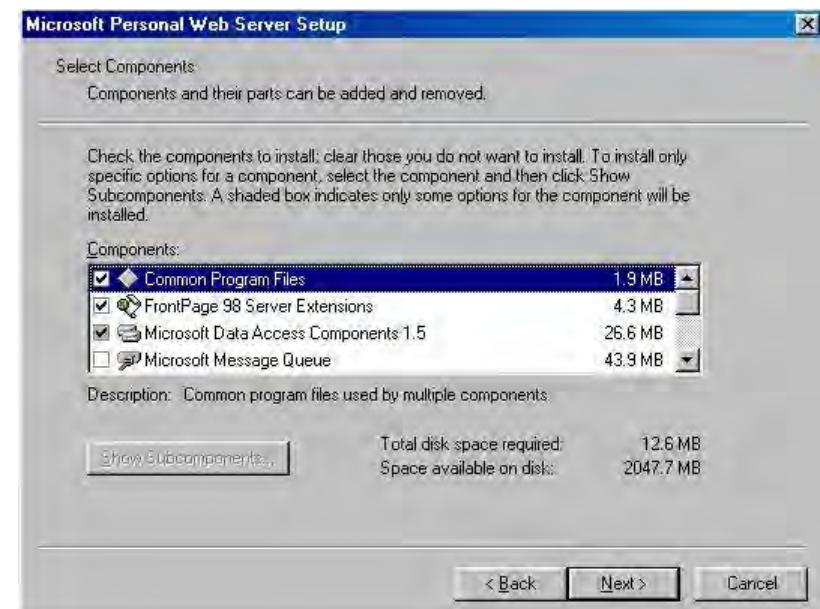
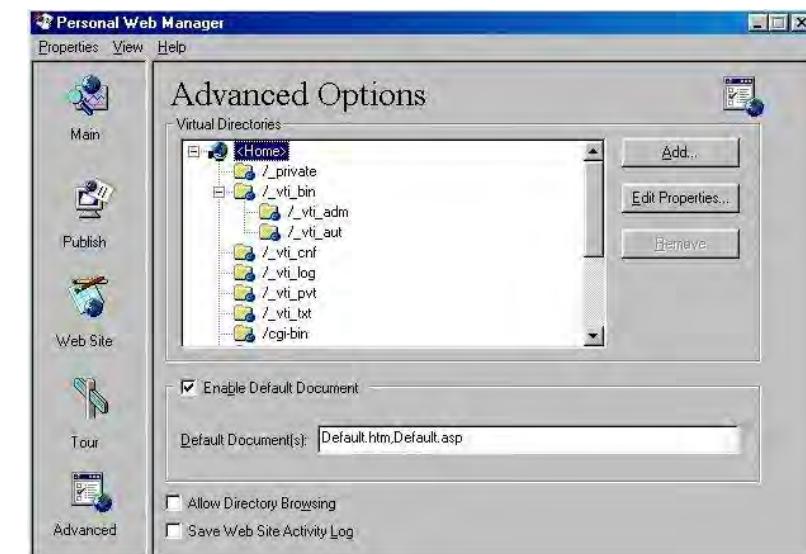
## 1.2. Cài đặt Personal Web Server (PWS) trên Win 98

Để cài đặt Personal Web Server trên hệ điều hành Windows 98, bạn vào vào thư mục \Add-onslpws trên đĩa CD dùng để cài đặt hệ điều hành này hoặc đĩa CD có phần NT Option Pack. Tìm và chạy tập tin setup.exe.

**Hình 2.4 - Màn hình cài đặt Personal Web Server**

Chọn các thành phần chính của Personal Web Server từ màn hình như hình 2.5.

Sau khi cài đặt thành công xong, hãy chạy cài thêm phiên bản mới nhất của ADO để có thể hỗ trợ tốt nhất cho các thao tác trên cơ sở dữ liệu (ví dụ MDAC 2.6). Để quản lý và cấu hình webserver này, hãy chạy chương trình Personal Web Manager.

**Hình 2.5 - Các lựa chọn các thành phần khi cài PWS****Hình 2.6 – Màn hình quản trị và cấu hình PWS**

### 1.3. Các cách thức truy cập đến website

Sau khi đã cài xong webserver, để truy cập đến website ta dùng các cách sau:

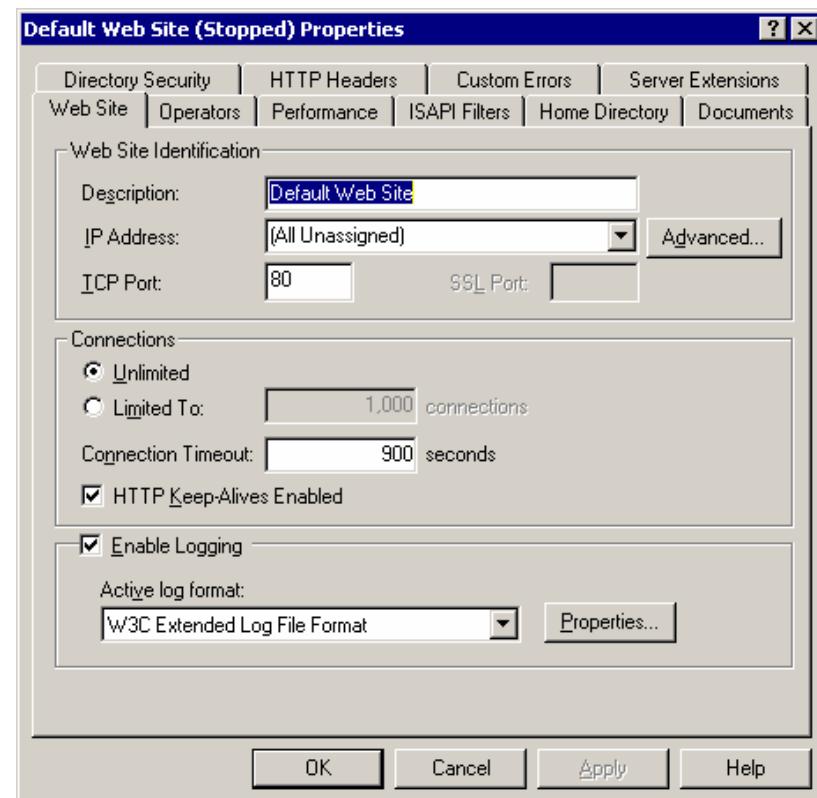
- <http://localhost/>
- <http://127.0.0.1/>
- http://<địa chỉ IP> ví dụ: <http://172.29.22.1>
- http://<tên máy> ví dụ: <http://vm-98se/>
- http://<tên DNS> ví dụ: <http://www.hcmuns.edu.vn>

## 2. CẤU HÌNH WEB SERVER

### 2.1. Các thông số chính

Các thông số mặc định chính cho webserver là tên mô tả website (description), địa chỉ IP và port gán cho website. Port mặc định cho website là 80. Nếu bạn sử dụng một port khác, ví dụ như port 81 cho website, bạn phải ghi rõ trong đường dẫn URL. Ví dụ: <http://www.visic-network.com:81/>. Ngoài ra, còn có các thông số liên quan đến số lượng các kết nối đồng thời (connections), đến định dạng và nơi lưu trữ tập tin nhật ký (log file) ghi lại quá trình truy cập website.

IIS mặc định thiết lập thư mục C:\InetPub\wwwroot là home directory cho website có tên là Default WebSite.



**Hình 2.7 - Màn hình thiết lập các thông số cho website**

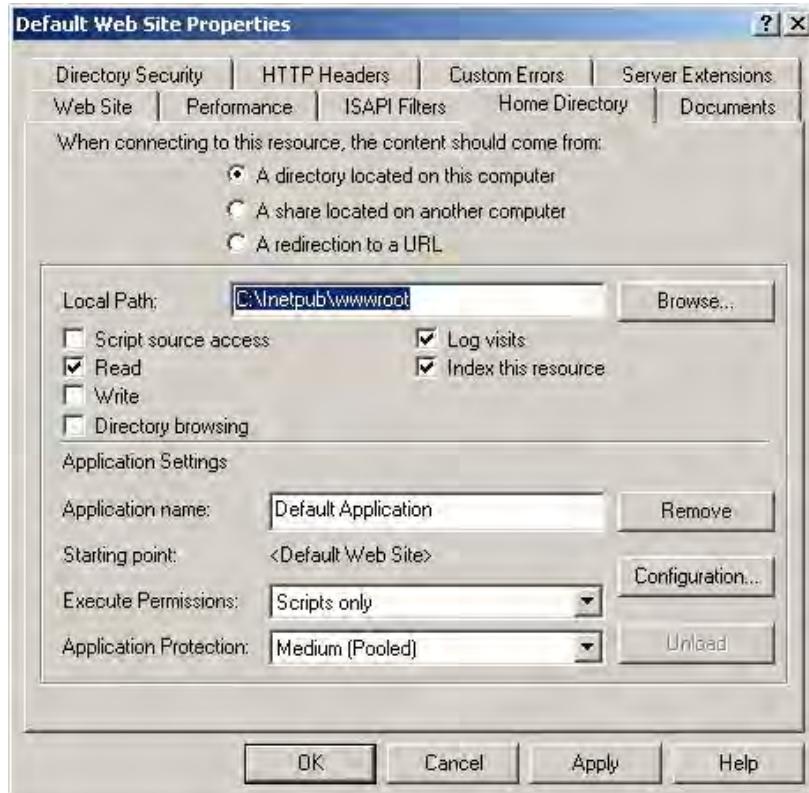
### 2.2. Thư mục Home Directory

Mỗi website phải có một home directory. Nó có thể được xem như là thư mục gốc của cây thư mục web bởi vì nó được ánh xạ với tên của máy chủ chứa website. Giả sử, nếu một máy có tên là [www.microsoft.com](http://www.microsoft.com) và home directory là C:\Website\Microsoft, khi đó để truy cập tới một tập tin, ví dụ như Search.htm đặt trong thư mục C:\Website\Microsoft, bạn phải gõ vào đường dẫn sau: <http://www.microsoft.com/search.htm>.

Nếu trong thư mục C:\Website\Microsoft, bạn tạo một thư mục con có tên là C:\Website\Microsoft\Products, thì để truy cập tới tập tin windowsnt.htm trong thư mục này, bạn phải gõ vào đường dẫn như sau: <http://www.microsoft.com/Products/windowsnt.htm>

IIS mặc định thiết lập thư mục C:\InetPub\wwwroot là home directory cho website có tên là Default WebSite.

Để xem, hiệu chỉnh thư mục home directory cho một website bạn click phải chuột vào website rồi chọn Properties. Sau đó chọn mục Home Directory



Hình 2.8 - Màn hình thiết lập thông số cho Home Directory

### 2.3. Thư mục ảo (Virtual Directory)

Để truy cập tới các thư mục không nằm trong thư mục home directory, bạn phải tạo một thư mục ảo. Một thư mục ảo là một thư mục mà về mặt vật lí nó không cần phải nằm trong thư mục home directory nhưng lại có dạng như thư mục thật trên đường dẫn truy cập.

Một thư mục ảo có một tên đại diện (alias). Đây là tên mà người dùng sẽ gõ vào trong đường dẫn khi muốn truy cập đến thư mục này. Do đó, khi tạo một thư mục ảo, bạn cần phải cung cấp 2 thông tin, đó là tên đại diện của thư mục ảo và đường dẫn vật lí của thư mục này.

Trong ví dụ sau, giả sử bạn đã thiết lập được một website có tên là [www.is-edu.hcmuns.edu.vn](http://www.is-edu.hcmuns.edu.vn), bảng sau sẽ minh họa việc ánh xạ giữa các thư mục vật lí và đường dẫn tương ứng khi muốn truy cập đến chúng

Physical Location	Alias	URL Path
C:\InetPub\wwwroot	(Home Directory)	<a href="http://www.is-edu.hcmuns.edu.vn">http://www.is-edu.hcmuns.edu.vn</a>
C:\InetPub\wwwroot\Scripts	Scripts	<a href="http://www.is-edu.hcmuns.edu.vn/Scripts/">http://www.is-edu.hcmuns.edu.vn/Scripts/</a>
D:\WebForum	Forum	<a href="http://www.is-edu.hcmuns.edu.vn/Forum/">http://www.is-edu.hcmuns.edu.vn/Forum/</a>

Trong Internet Services Manager, các biểu tượng tượng trưng cho thư mục ảo sẽ khác với các biểu tượng tượng trưng cho thư mục thực trong hiển thị cây thư mục của một website.

### 2.4. Tập tin truy cập mặc định (Default document)

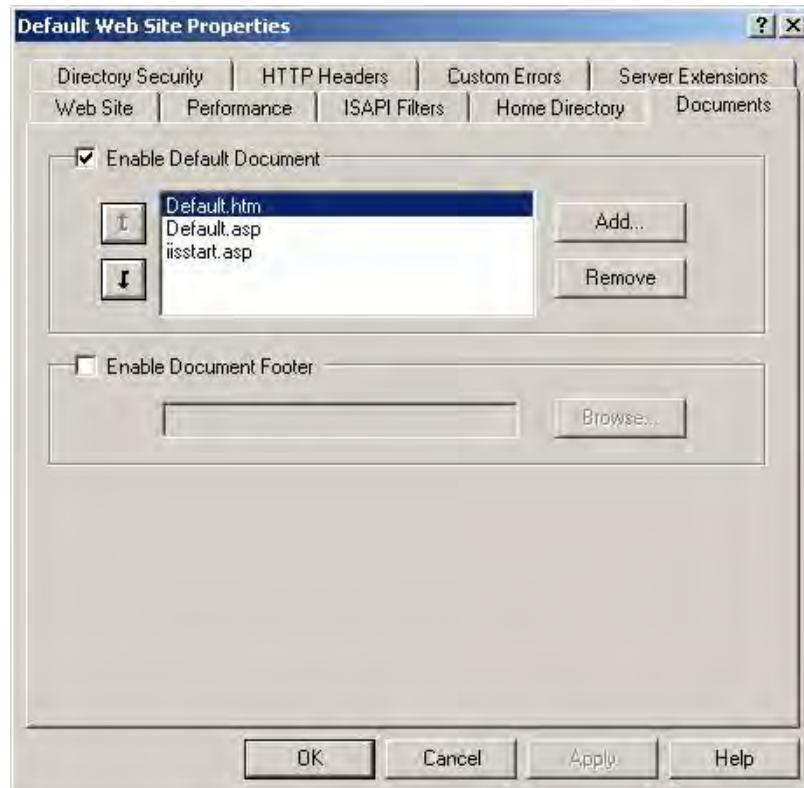
Để truy cập tới một trang web trên webserver, bạn phải chỉ định tên tập tin trong đường dẫn URL, ví dụ như: <http://www.microsoft.com/ms.htm>. Tuy nhiên trong một số trường hợp, bạn không cần phải chỉ định tên tập tin trong đường dẫn URL mà vẫn nhận được trang web. Điều này thực hiện được do cơ chế tập tin mặc định được thiết lập trên webserver.

Tập tin truy nhập mặc định là tập tin sẽ được webserver trả về khi người dùng không chỉ định tên tập tin cụ thể trong đường dẫn URL. Ví dụ, nếu bạn chỉ gõ đường dẫn <http://www.microsoft.com> trong ô Address của trình duyệt, bạn vẫn truy cập được vào trang chủ của Microsoft dù không chỉ định tập tin chứa trang web này. Điều này có được là do webserver trả về tập tin mặc định (trong trường hợp này là trang chủ của Microsoft) cho tất cả các yêu cầu không chỉ định tên tập tin.

Các tập tin truy nhập mặc định thường có tên là default.htm, default.asp, index.htm, index.asp, ... Tuy nhiên IIS cho phép bạn định nghĩa bất kì tên tập tin nào là tập tin mặc định, đồng thời cho phép người dùng khai báo nhiều tập tin mặc định cùng lúc. Khi một

yêu cầu gửi đến webserver mà không chỉ định tên tập tin, IIS sẽ tìm kiếm trong danh mục các tập tin mặc định đã được khai báo, tập tin tồn tại đầu tiên sẽ được trả về cho trình duyệt máy client. Ví dụ nếu bạn định nghĩa default.htm, default.asp là các tập tin mặc định, thì khi trên thư mục hiện hành không có tập tin default.htm mà chỉ có tập tin default.asp, tập tin default.asp sẽ được gửi trả về cho trình duyệt.

Để xem và hiệu chỉnh các tập tin mặc định, hãy chọn mục Properties của website. Sau đó chọn mục Documents.

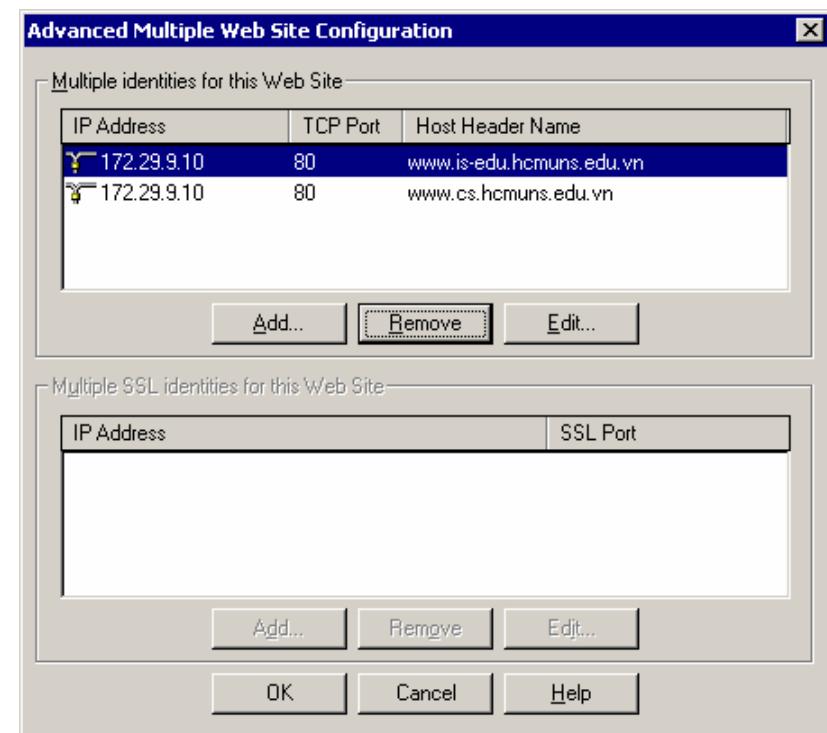


Hình 2.9 - Màn hình thiết lập danh sách các tập tin mặc định

## 2.5. Đặt nhiều website trên một web server (multiple hosting)

Trong IIS, bạn có thể thiết lập cấu hình cho nhiều website cùng hoạt động. Phần này sẽ đề cập đến vấn đề là làm thế nào để nhiều website có thể sử dụng cùng một port (giả sử port 80).

Để thực hiện điều này, bạn chỉ cần định rõ tên miền của website và tên port. Hai thông tin này sẽ giúp IIS xác định duy nhất một website. Bạn vào mục Advance trong mục xác định các thông số chính Web site (xem 2.1). Xem ví dụ sau:

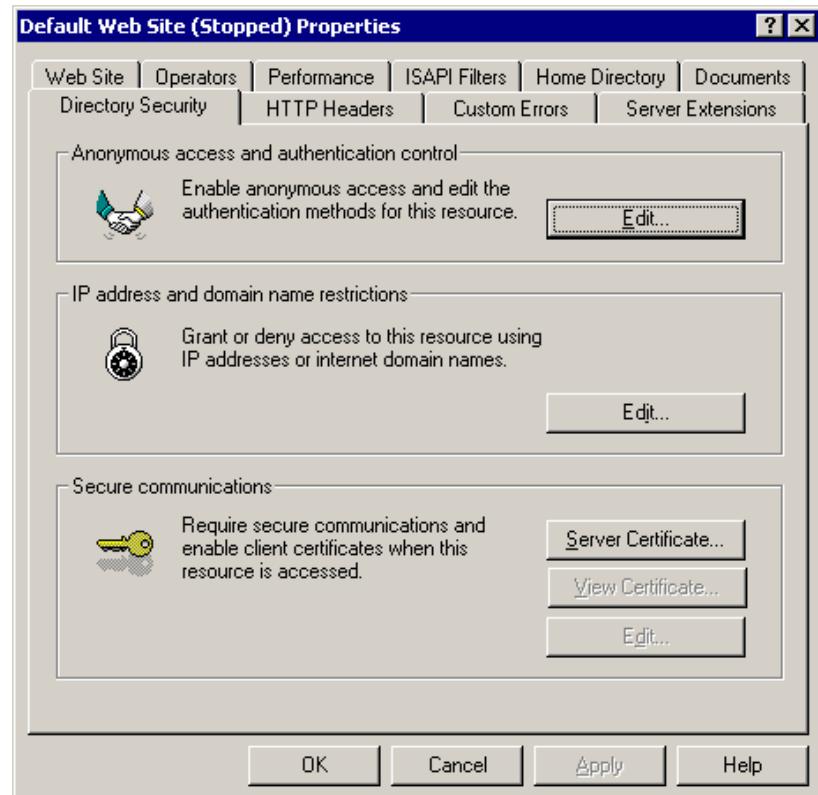


Hình 2.10 - Màn hình thiết lập nhiều website cùng hoạt động

### 3. MỘT SỐ VẤN ĐỀ LIÊN QUAN ĐẾN AN TOÀN

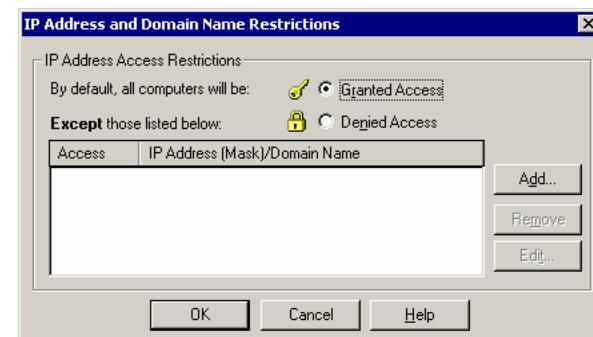
#### 3.1. Thiết lập quyền truy cập

Để thiết lập quyền truy cập vào website cũng như các thư mục và tập tin trên website, bạn vào mục Properties của website rồi chọn Directory Security,



**Hình 2.11 - Màn hình thiết lập quyền truy cập**

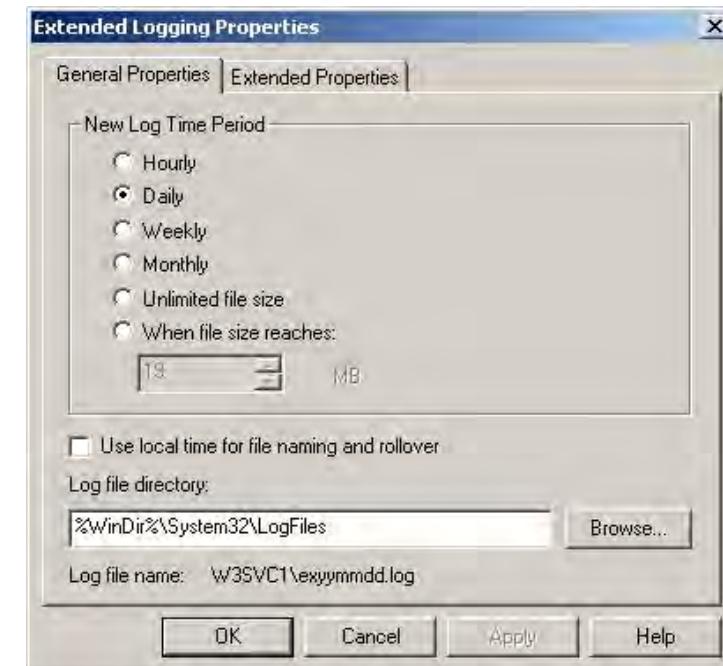
Để thiết lập hạn chế hay mở rộng quyền truy cập vào website theo vùng tên miền, địa chỉ IP, ... bạn chọn mục IP address and domain name restrictions



**Hình 2.12 - Màn hình thiết lập hạn chế theo tên miền, địa chỉ IP**

#### 3.2. Nhật ký sử dụng (website logging)

Bạn có thể lưu lại dấu vết của các truy cập tới website của bạn trong tập tin nhật ký. Để yêu cầu các thông tin nào của vết truy cập sẽ được lưu lại, bạn chọn mục Properties của Log File.



**Hình 2.13 - Màn hình chỉ định các thông số được lưu trong logfile**

## 4. MỘT SỐ CÂU HỎI THƯỜNG GẶP

### 4.1. Không truy cập được trang web

Sau khi cài đặt xong IIS/PWS nhưng thử địa chỉ <http://localhost/> vẫn không chạy được. Trong trường hợp này hãy kiểm tra webserver đã được khởi động (Start) hay chưa hoặc kiểm tra trong Internet Explorer mục Tools/Internet Options/Connections/LAN settings để xem lại các thiết lập của Proxy server.

### 4.2. Lỗi The page cannot be found

Khi truy cập một trang web mà gặp thông báo lỗi như sau:



#### The page cannot be found

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable

Điều này có nghĩa rằng trang web mà bạn yêu cầu truy cập không tồn tại trên máy chủ. Hãy kiểm tra lại địa chỉ trang web.

### 4.3. Lỗi You are not authorized to view this page

Khi truy cập một trang web mà gặp thông báo lỗi như sau:



#### You are not authorized to view this page

You might not have permission to view this directory or page using the credentials you supplied.

Điều này có nghĩa rằng bạn không được phép truy cập trang web mà bạn đang yêu cầu truy cập. Hãy kiểm tra lại địa chỉ trang web và quyền truy cập mà bạn đang sử dụng.

### 4.4. Các địa chỉ cho phép đặt trang web (web hosting) miễn phí

Hãy truy cập vào website <http://www.freewebspace.net> khi bạn cần tìm kiếm một máy chủ để đặt trang web của mình. Website này liên kết đến hàng trăm địa chỉ cung cấp dịch vụ lưu trữ trang web miễn phí.

Các địa chỉ được sắp xếp theo chức năng hỗ trợ, rất tiện cho việc đánh giá ưu, nhược điểm của từng nhà cung cấp dịch vụ. Với chức năng “Tìm kiếm nâng cao”, bạn sẽ dễ dàng tìm được nhà cung cấp dịch vụ thỏa các yêu cầu của bạn. Ngoài ra bạn còn có thể đăng ký nhận bản tin qua e-mail giới thiệu về các dịch vụ webhosting mới nhất.

Sau đây là một số địa chỉ cho phép đặt các trang web viết bằng các ngôn ngữ ASP, PHP, JSP:

- ASP: <http://www.1asphost.com/>, <http://www.brinkster.com/>,  
<http://www.domainidx.com/> , <http://www.7host.com/>
- JSP: <http://www.mycqiserver.com/>
- PHP: [http://www.0php.com/free\\_PHP\\_hosting.php](http://www.0php.com/free_PHP_hosting.php),  
<http://www.free-php-hosting.com/>,

Sau khi đăng ký xong, bạn sẽ được cung cấp một địa chỉ để truy cập vào website của mình. Thông thường các freeserver chỉ cung cấp dịch vụ giới hạn và chèn quảng cáo vào các trang web của website của bạn.

### 4.5. Làm thế nào để tải một website lên máy chủ

Việc đưa trang web lên mạng thông thường bằng một trong hai cách sau:

- Dùng dịch vụ FTP. Nghĩa là nhà cung cấp sẽ cung cấp cho bạn một tài khoản để đăng nhập vào FTP server chứa trang web. Lúc đó, bạn sẽ dùng kết nối vào FTP server này thông qua các phần mềm FTP Client như Windows Commander, WS-FTP. Cách này thì rất thuận tiện và đơn giản, do đó được sử dụng rất phổ biến. Tuy nhiên dịch vụ này lại hầu như không được hỗ trợ khi bạn đăng ký miễn phí.
- Tại các website hosting cho phép người dùng sử dụng giao diện web của họ để upload tập tin lên. Tuy nhiên cách này bị hạn chế ở chỗ kích thước tập tin bạn upload trong 1 lần và số lượng tập tin được upload đồng thời. Nếu bạn có khoảng 100 tập tin thì việc upload sẽ rất vất vả.

### 4.6. Dùng cơ sở dữ liệu trên các máy chủ miễn phí

Thông thường các máy chủ miễn phí (freeserver) không bao giờ hỗ trợ cơ sở dữ liệu (CSDL) như SQL Server, Oracle cả, đơn giản là

phải trả chi phí về phần mềm. Thậm chí khi bạn trả tiền cho thuê chỗ (web hosting), cũng phải trả thêm tiền mới được hỗ trợ SQL Server, Oracle, .... Do đó, hầu hết các freeserver chỉ hỗ trợ tối đa cho bạn là Access như Brinkster mà thôi, hoặc là các CSDL miễn phí khác như MySQL, ... .

Thông thường do người dùng không được quyền thao tác trên máy chủ thuê (server hosting) nên các ứng dụng web không thể sử dụng các system DSN mà phải kết nối CSDL dưới dạng DSN-less. Cách khai báo này cũng khá đơn giản, bạn có thể tham khảo bất cứ tài liệu hướng dẫn lập trình kết nối web với CSDL nào.

Nếu bạn dùng Access thì trong phần đường dẫn đến tập tin .mdb phải dùng Server.MapPath để ánh xạ thư mục gốc của webroot, không được dùng đường dẫn tuyệt đối dạng: C:\mywebsite\db.

## CÂU HỎI

1. Hãy cho biết vai trò của các phần mềm như IIS và PWS.
2. Phân biệt thư mục vật lý và thư mục ảo. Ý nghĩa của thư mục ảo.
3. Cho biết các tập tin mặc định thông dụng đối với các website.
4. Làm thế nào để hạn chế các đối tượng được truy cập đến một tập tin hay một thư mục nào đó.
5. Làm thế nào để hạn chế các máy tính được phép truy xuất đến website.
6. Làm thế nào để có thể phát hiện các sự xâm nhập trái phép của các tin tặc vào máy chủ của bạn.

# THỰC HÀNH

## 1. CÀI ĐẶT VÀ CẤU HÌNH WEB SERVER

Cài đặt IIS hay PWS lên máy của bạn. Sau đó thực hiện tuân tự các bước cấu hình sau:

- Thay đổi thư mục Home Directory:** Tạo trên máy một thư mục để chứa website, ví dụ: C:\MyWebSite và cấu hình sao cho thư mục HomeDirectory của Website trả về thư mục này thay cho thư mục mặc định là C:\InetPub\wwwroot
- Truy cập theo đường dẫn đến thư mục vật lý:** Tạo thư mục BookStore trong thư mục C:\MyWebSite. Sau đó unzip tập tin BookStore.zip nằm trong AppSamples\BookStore của đĩa CD vào thư mục này. Truy cập ứng dụng BookStore từ địa chỉ: <http://localhost/BookStore/>
- Tạo thư mục ảo:** Tạo một thư mục RealEstate ở ngoài thư mục C:\MyWebSite, ví dụ như D:\RealEstate. Tiếp theo unzip tập tin RealEstate.zip nằm trong thư mục AppSamples\RealEstate của đĩa CD vào thư mục này. Sau đó tạo một thư mục ảo, lấy tên là RealEstate sao cho trả đến thư mục D:\RealEstate. Truy cập ứng dụng RealEstate từ địa chỉ: <http://localhost/RealEstate/>
- Kiểm tra web server có hỗ trợ ASP hay không:** Tạo thư mục D:\MyApp, sau đó tạo một thư mục ảo có tên là MyApp trả đến thư mục này. Tạo một tập tin default.asp với nội dung `<% Response.Write Date() %>` trong D:\MyApp rồi chạy thử bằng cách gõ vào địa chỉ: <http://localhost/MyApp/default.asp>
- Cấu hình tập tin mặc định:** Kiểm tra xem các tập tin mặc định của website là gì. Sau đó cấu hình để cho website truy cập các tập tin mặc định theo thứ tự sau: **default.asp, index.asp, default.htm, index.htm.**

Nếu là IIS cài đặt trên webserver, cấu hình FTP server sao cho các user khi kết nối với server sẽ được chuyển đến thư mục C:\MyWebSite.

Cấu hình IIS sao cho có thể hosting đồng thời nhiều website.

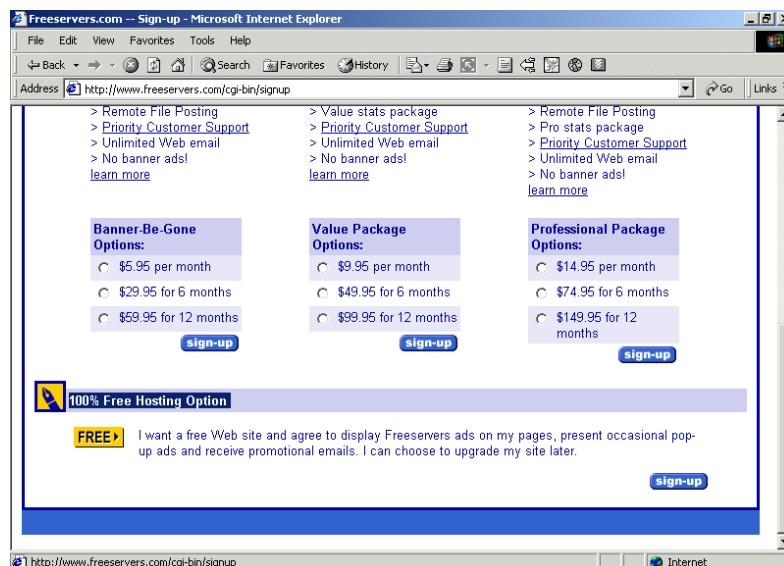
## 2. ĐĂNG KÍ WEBSITE MIỄN PHÍ TẠI FREESERVERS

Freeservers cho phép bạn đặt website của bạn lên server của họ miễn phí. Tuy nhiên dịch vụ này chỉ cho phép trang web tĩnh, do đó các trang web chứa các mã ASP, PHP, ... không thực hiện được

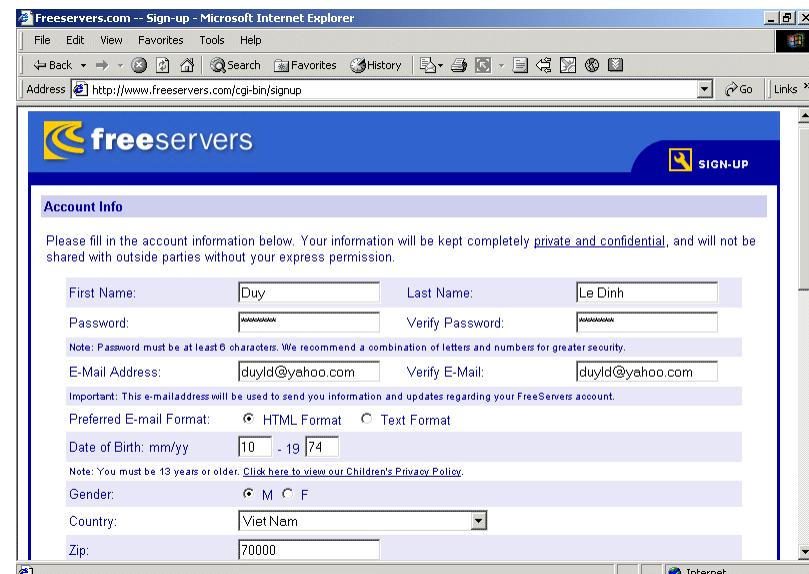
Vào địa chỉ <http://www.freeservers.com> để đăng kí website miễn phí với địa chỉ truy cập có dạng <http://yourname.freeservers.com>



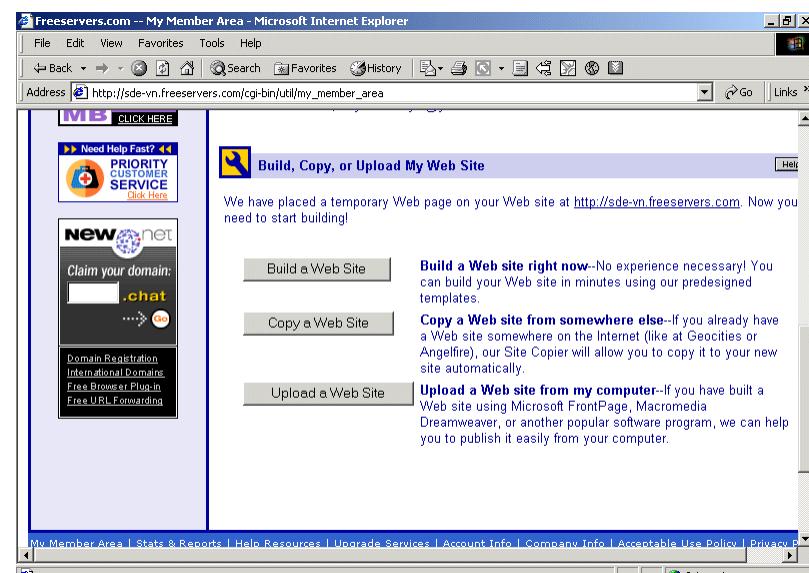
Gõ vào địa chỉ mà bạn dự định đăng ký tại ô **yourname**. Sau đó click **Go**. Màn hình tiếp theo sẽ hiện ra, bạn chọn mục cuối cùng **100% Free Hosting Option** và click vào **SignUp**



Sau đó bạn hãy điền vào các thông tin liên quan cần thiết như Họ tên, mật khẩu, email, ... Cuối cùng gõ vào chuỗi kí tự theo yêu cầu giống bên trái để chắc rằng bạn đã đồng ý với các qui định của FreeServers khi cung cấp dịch vụ miễn phí này.



Nếu đăng ký thành công, màn hình sau sẽ hiện ra để cung cấp cho bạn một số lựa chọn khi xây dựng website của mình.



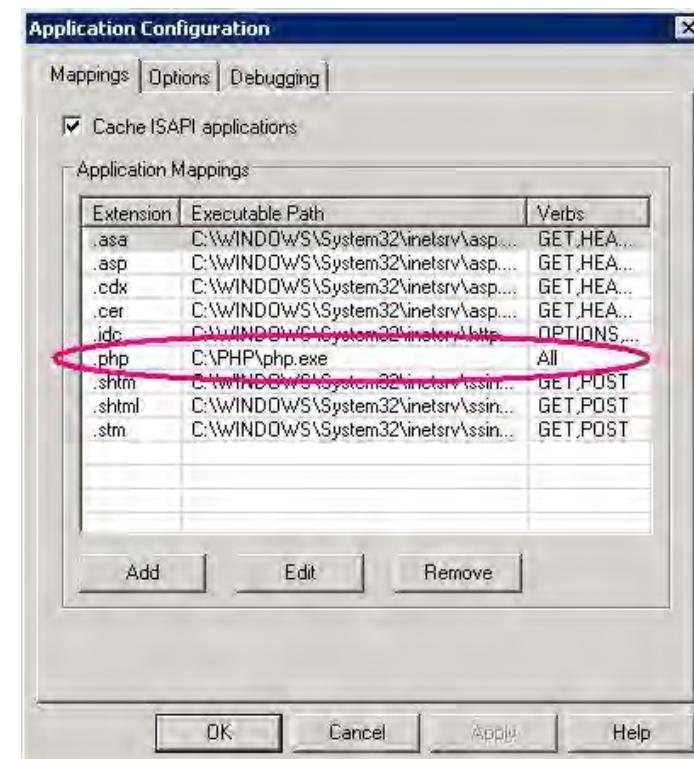
### 3. CÀI ĐẶT WEB SERVER IIS HỖ TRỢ PHP

Vào trang <http://www.php.net/downloads.php>, chọn trong mục Windows Binaries/PHP 4.3.1 installer để tải phần mềm mới nhất về. Thực hiện chương trình php-4.3.1-installer.exe để chương trình tự cài đặt và cấu hình webserver IIS hỗ trợ PHP.

Chú ý rằng phải cài IIS trước khi cài PHP và trong màn hình setup của PHP, chọn webserver tương ứng với webserver đang cài trên máy của bạn (mặc định là Microsoft IIS 4 or higher)



Để kiểm tra việc cài đặt có thành công hay không, hãy vào trình quản lý webserver IIS là Internet Information Services, chọn phần Web Sites. Trong mục Home Directory, chọn Configuration, kiểm tra xem trong Application Mappings có khai báo trình xử lý trang php hay không như hình sau:



### 4. ĐĂNG KÍ DỊCH VỤ ĐẶT TRANG WEB MIỄN PHÍ

Hãy vào các địa chỉ web hỗ trợ đặt trang web miễn phí được đề cập trong phần 4.4 để đăng ký. Thông thường các bước tuần tự như sau:

- Điền vào các thông tin đăng ký, tên đăng nhập, mật khẩu, địa chỉ email để chương trình gửi thông tin yêu cầu xác nhận đăng ký.
- Sau khi đăng ký thành công, bạn sẽ được cung cấp một nơi lưu trữ trên máy chủ và địa chỉ trang web để truy cập đến nơi này.
- Để đưa website mình đã thiết kế lên, bạn sử dụng các dịch vụ tải tập tin của nơi hỗ trợ, có thể là theo cách dùng FTP hoặc dùng trình duyệt.

## Chương 3

# THIẾT KẾ TRANG WEB

### 1. GIỚI THIỆU VỀ HTML

HTML là ngôn ngữ dùng để xây dựng trang web. Nó mô tả cách thức mà dữ liệu được hiển thị thông qua tập các ký hiệu đánh dấu thường được gọi là tag. Các tag này được bao quanh bởi các dấu "<" ">". Ví dụ: <HTML>, </BODY>, <IMG> là các tag HTML.

Một trang web thông thường gồm có hai thành phần chính đó là: dữ liệu của trang web (ví dụ như văn bản, hình ảnh, âm thanh, ...) và các tag HTML dùng để mô tả cách thức mà các dữ liệu này được hiển thị. Khi một trang web được tải về trình duyệt, trình duyệt sẽ căn cứ vào các tag HTML để định dạng dữ liệu được hiển thị. Ví dụ sau cho thấy, cùng một dữ liệu là dòng văn bản "This is webpage". Tuy nhiên nếu có thêm định dạng của tag <b> ở dòng thứ hai, hình thức hiển thị dữ liệu sẽ khác.

Mã HTML	Hiển thị trên trình duyệt
This is webpage <b>This is webpage</b>	This is webpage <b>This is webpage</b>

Hầu hết các tag của HTML đều có tag bắt đầu và tag kết thúc tương ứng. Tag kết thúc tương ứng với một tag chỉ khác ở chỗ có thêm ký tự "/". Ví dụ: tag <HTML> có tag kết thúc tương ứng là </HTML>.

Các tag không phân biệt chữ thường và chữ hoa. Nghĩa là các tag <html>, <Html> và <HTML> là như nhau.

Trong quá trình đọc trang web để hiển thị, trình duyệt sẽ bỏ qua các khoảng trắng thừa và các dấu ngắt dòng giữa văn bản dữ liệu và các tag. Đó chính là lí do mà ba đoạn sau sẽ cùng được hiển thị như nhau.

Đoạn 1	Đoạn 2	Đoạn 3
The browser will ignore new lines and extra spaces in the text	The browser will ignore new lines and extra spaces in the text	The browser will ignore new lines and extra spaces in the text

Mặc dù lúc soạn thảo bạn có thể ngắt dòng, hay thêm nhiều khoảng trắng vào nhưng khi hiển thị, ba đoạn trên đều cho kết quả như nhau là:

The browser will ignore new lines and extra spaces in the text

Khi bắt đầu soạn thảo một trang web, các tag cơ bản sau sẽ đóng vai trò là khung cho việc định nghĩa một trang HTML. Xét ví dụ của một trang HTML đơn giản sau:

```

<HTML>
  <HEAD>
    <TITLE>Title of the web page </TITLE>
  </HEAD>

  <BODY>
    An example of a simple web page.
  </BODY>

</HTML>
```

- <HTML> </HTML>: Định nghĩa phạm vi của văn bản HTML.
- <HEAD> </HEAD>: Định nghĩa các mô tả về trang HTML. Các thông tin này sẽ không hiển thị dưới dạng nội dung của trang web.
- <TITLE> </TITLE>: Mô tả tiêu đề của trang. Tiêu đề của trang web thường được hiển thị như là tiêu đề của cửa sổ hiển thị trang web. Thông tin này cũng có thể được dùng bởi một số máy tìm kiếm để xây dựng chỉ mục các trang web.
- <BODY> </BODY>: Xác định vùng "thân" của trang web. Đây là phần mà các dữ liệu cùng với các tag HTML sẽ được dùng để định dạng nội dung của trang web.

Một tag thông thường bao gồm hai thành phần: tên của tag (dùng để biết tag này định nghĩa cái gì) và thuộc tính của tag (dùng để biết dữ liệu được hiển thị như thế nào). Ví dụ sau minh họa việc thay đổi các thuộc tính FACE và SIZE của tag <FONT>

Mã HTML	Hiển thị trên trình duyệt
<FONT FACE="Arial" SIZE="2">This is a web page</FONT>	This is webpage
<FONT FACE="Times New Roman" SIZE="4">This is a web page</FONT>	This is webpage

## 2. MỘT SỐ TAG CƠ BẢN CỦA HTML

### 2.1. Định dạng trang

Định dạng trang thông thường bao gồm các thông tin về tiêu đề trang, bảng mã kí tự được dùng, màu nền, ảnh nền, từ khóa.

Để đặt tiêu đề cho trang web, ta dùng tag <TITLE>, ví dụ muốn có tiêu đề của trang web thiết kế là “Chào mừng các bạn đến với trang web này”, ta dùng: <TITLE> Chào mừng các bạn đến với trang web này</TITLE>

Để đặt bảng mã mà trang web dùng, ta sử dụng thiết lập các thông số cho tag <META>. Ví dụ thiết lập sau: <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8"> cho biết trang web sẽ dùng bảng mã Unicode-UTF-8.

Ta có thể đặt màu nền, ảnh nền cũng như màu chữ cho toàn bộ trang web bằng cách đặt các thuộc tính BGCOLOR, BACKGROUND và TEXT tương ứng trong tag <BODY>. Ví dụ sau đặt màu nền cho trang web là màu đỏ, màu chữ là màu trắng: <BODY BGCOLOR="#FF0000" TEXT="#FFFFFF">.

Ví dụ sau minh họa một trang web có các định dạng trang đã trình bày ở trên:

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Language" CONTENT="en-us">
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=utf-8">
<TITLE>Chào mừng các bạn đến với trang web này</TITLE>
</HEAD>

<BODY BGCOLOR="#FF0000" TEXT="#FFFFFF">

</BODY>
</HTML>
```

### 2.2. Định dạng văn bản

#### 2.2.1. Định dạng phân đoạn

Tag <P> được dùng để định dạng phân đoạn (paragraph). Thuộc tính thường gặp là ALIGN dùng để canh chỉnh dữ liệu trong phân đoạn là LEFT (trái), RIGHT (phải) CENTER (canh giữa) hoặc JUSTIFY (canh đều hai bên). Ví dụ sau minh họa việc hiển thị khi đặt các thuộc tính canh chỉnh cho tag <P> bằng cách lần lượt gán thuộc tính ALIGN trong tag này các giá trị “LEFT”, “CENTER”, “RIGHT”:

```
<P ALIGN="LEFT">This is webpage</P>
<P ALIGN="CENTER">This is webpage</P>
<P ALIGN="RIGHT">This is webpage</P>
```



Hình 3.1 – Các thuộc tính canh chỉnh của tag <P>

Trong một phân đoạn, việc ngắt dòng trong lúc soạn thảo không ảnh hưởng gì đến việc hiển thị. Trình duyệt chỉ hiểu ngắt dòng trong một phân đoạn thông qua tag <BR>. Ví dụ:

```
<P>
Họ tên: Lê Đình Duy
Nghề nghiệp: Giảng viên
Địa chỉ: Khoa CNTT – ĐH Khoa Học Tự Nhiên </P>
```



Hình 3.2 – Không thể xuống dòng như thiết kế

Để hiển thị mỗi mục trên mỗi dòng, ta phải chỉnh lại đoạn mã HTML trên bằng cách thêm vào cuối mỗi mục tag <BR> như sau:

```
<P>
Họ tên: Lê Đình Duy <BR>
Nghề nghiệp: Giảng viên <BR>
Địa chỉ: Khoa CNTT – ĐH Khoa Học Tự Nhiên <BR>
</P>
```



### Hình 3.3 – Dùng tag <BR> khi muốn xuống dòng

Ngoài ra, ta cũng có thể dùng đường kẻ ngang với tag <HR> để tạo sự phân cách giữa các thành phần trong trang web. Thuộc tính thường gặp là ALIGN (dùng để canh chỉnh vị trí của đường này) và COLOR (màu sắc).

```
Nội dung của trang web
<HR COLOR="#0000FF">
Cập nhật năm 2003. Mọi ý kiến, góp ý xin liên hệ Webmaster.
```



### Hình 3.4 – Phân cách bằng tag <HR>

## 2.2.2. Định dạng chữ

Tag <FONT> để định dạng font chữ, màu sắc, kích thước, ... của văn bản. Các thuộc tính của tag này thường là: FACE: loại font chữ, SIZE: kích thước, COLOR: màu chữ. Ngoài ra, để định dạng chữ **in đậm** ta dùng tag <B>, **in nghiêng** dùng tag <I>, **gạch dưới** dùng tag <U>.

Thuộc tính COLOR dùng trong các tag thường được định nghĩa bằng tên qui ước (ví dụ như RED: màu đỏ, BLUE: màu xanh da trời, ...) hoặc mã RGB dưới dạng 3 chữ số hệ 16 bắt đầu bằng dấu # (ví dụ như #FF0000).

Ví dụ sau định dạng đoạn văn bản với font Tahoma, in đậm, kích thước +2 (10pt)

```
<B><FONT FACE="Tahoma" SIZE="2">Thuật xử thế của người xưa</FONT></B>
```

Để có kết quả hiển thị là: **Thuật xử thế của người xưa**

## 2.3. Định dạng hình ảnh

Tag <IMG> được dùng để chèn một ảnh hoặc một video clip vào trong một trang web. Các định dạng tập tin ảnh và video sau được hỗ trợ trên Internet Explorer: \*.avi, \*.bmp, \*.emf, \*.gif, \*.jpg, \*.jpeg, \*.mov, \*.mpg, \*.mpeg, \*.png, \*.wmf, ...

Đối với trang web không chứa hình ảnh, thì nội dung văn bản sẽ được hiển thị sau này nằm toàn bộ trong trang web. Tuy nhiên với trang web có chứa hình ảnh, hình ảnh sẽ là một tập tin độc lập với tập tin chứa trang web. Tag <IMG> dùng trong trang web để thông báo cho trình duyệt đọc tập tin ảnh và hiển thị tại vị trí đặt tag này. Ví dụ, một trang web muốn hiển thị hình ảnh linh vật biểu tượng Seagames 22 sau đoạn văn bản "Linh vật Seagames 22", phải bao gồm hai tập tin; Một tập tin hình ảnh về linh vật, ví dụ linhvat.gif. Tập tin còn lại chứa đoạn văn bản "Linh vật Seagames 22" và tag <IMG SRC="linhvat.gif"> như sau:

```
<P><B><FONT FACE="Tahoma" SIZE="2"> Linh vật Seagames 22</FONT></B></P>
<IMG BORDER="0" SRC="linhvat.gif" WIDTH="512" HEIGHT="18">
```

Các thuộc tính chính bao gồm: SRC (tên tập tin hình ảnh), WIDTH, HEIGHT (kích thước ảnh), BORDER (đường viền khung quanh ảnh), ALT (văn bản thay thế dùng khi không hiển thị), ALIGN

(anh chỉnh), ... Thuộc tính ALT cũng được dùng khi ta muốn đưa chuột đến hình ảnh và có dòng chữ như tooltip hiển thị bên dưới.

#### 2.4. Định dạng hyperlink (siêu liên kết)

Tag `<A>` được dùng để đặt một hyperlink. Đoạn văn bản (hay hình ảnh, ...) nằm giữa các tag `<A>` và `</A>` sẽ đóng vai trò như là “dấu hiệu” (anchor) hyperlink. Thông thường, con trỏ chuột sẽ thay đổi hình dạng (thường là hình bàn tay) khi di chuyển ngang qua đoạn văn bản này.

Một hyperlink cho phép liên kết tới một vị trí khác ngay bên trong trang web chứa hyperlink (liên kết nội) hoặc tới một trang web khác (liên kết ngoại).

Để tạo một liên kết nội, ta cần thực hiện hai bước. Bước 1 là đánh dấu vị trí (bookmark) của nơi cần liên kết bằng thuộc tính NAME. Bước 2 là tạo hyperlink đến vị trí vừa được đánh dấu. Ví dụ sau minh họa việc tạo liên kết nội để khi người dùng nhấp vào hyperlink “Go to Chapter 1” thì trình duyệt sẽ chuyển đến vị trí của Chapter 1 trong cùng trang web

Mã HTML	Hiển thị trên trình duyệt
<code>&lt;A HREF="#Chapter1"&gt; Go to Chapter 2&lt;/A&gt;</code>	<u>Go to chapter 1</u>
...	
<code>&lt;A NAME="Chapter1"&gt; Chapter 1&lt;/A&gt;</code>	Chapter 1

Dấu # trong mục HREF là dấu hiệu của liên kết nội.

Để liên kết tới các trang web khác, điền địa chỉ URL của trang web vào mục HREF. Xem các ví dụ sau:

Mã HTML	Ý nghĩa
<code>&lt;A HREF="http://www.intel.com"&gt; Intel Home Page&lt;/A&gt;</code>	Liên kết đến trang ở máy chủ khác
<code>&lt;A HREF="home.htm"&gt; Home&lt;/A&gt;</code>	Liên kết đến trang khác trong cùng thư mục
<code>&lt;A HREF="javascript:window.open()"&gt; Click here&lt;/A&gt;</code>	Gọi một hàm JavaScript
<code>&lt;A HREF="FlashGet.zip"&gt; Download&lt;/A&gt;</code>	Hiển thị cửa sổ tải tập tin về.

Mã HTML	Ý nghĩa
<code>&lt;A HREF="ftp://ftp.microsoft.com/mouse.zip"&gt; Download&lt;/A&gt;</code>	Hiển thị cửa sổ tải tập tin theo giao thức FTP.
<code>&lt;A HREF="mailto:duyld@yahoo.com"&gt; Email me&lt;/A&gt;</code>	Gửi email
<code>&lt;A HREF="Forum.htm" TARGET=_blank&gt; Forum&lt;/A&gt;</code>	Mở tập tin trong cửa sổ định nghĩa bởi TARGET

Một số giá trị có thể được dùng để gán cho thuộc tính TARGET của tag `<A>` là:

Giá trị của TARGET	Ý nghĩa
Name	Nạp trang web trong cửa sổ hoặc FRAME có tên là name
_blank	Nạp trang web vào một cửa sổ trống mới. Cửa sổ mới này không có tên.
_parent	Nạp trang web vào cửa sổ cha gần nhất của trang web hiện hành.
_self	Nạp trang web vào cùng cửa sổ với trang web hiện hành. Đây là giá trị mặc định của hyperlink.
_top	Nạp trang web vào cửa sổ cao nhất (topmost).

#### 2.5. Định dạng một số kí tự đặc biệt

HTML sử dụng các kí tự đặc biệt (ví dụ như “<” và “>” để đánh dấu các tag, ...), do đó để hiển thị các kí tự đặc biệt này, ta phải dùng các nhóm kí tự thay thế sau:

Kí tự cần hiển thị	Mã thay thế
<	&lt;
>	&gt;
&	&amp;
"	&quot;
khoảng trắng	&nbsp;
¢, #, ¥	&cent, &pound, &yen
1/2, 1/3, 3/4	&frac14, &frac12, &frac34
÷, °	&divide, &deg
®, ©	&reg, &copy

## 2.6. Chuyển hướng trang tự động (Automatic redirect)

Để định nghĩa một trang sẽ tự động chuyển sang một địa chỉ khác sau một khoảng thời gian định trước, ta chèn ngay sau tag <HTML> một tag như sau: <META HTTP-EQUIV="Refresh" CONTENT="3;url=http://domain/directory/file.html">

Trang web chứa tag trên sẽ tự động chuyển sang trang mới <http://domain/directory/file.html> sau khoảng thời gian là 3 giây.

## 3. ĐỊNH DẠNG BẢNG BIỂU

Tag <TABLE> được dùng để định dạng bảng cùng với các tag <TR>, <TD> để định dạng các dòng, cột. Các dòng, cột trong bảng thường được gọi là cell.

Các thuộc tính thường dùng là: BORDER (định nghĩa đường viền, nếu đặt giá trị là 0 thì sẽ không có đường viền), BGCOLOR (màu nền), ALIGN (canh chỉnh), WIDTH (chiều rộng, có thể theo % của kích thước cửa sổ hoặc pixel), CELLPADDING (khoảng cách giữa nội dung và đường biên của cell), CELLSPACING (khoảng cách giữa các cell). Đoạn mã HTML sau minh họa một bảng dữ liệu gồm có 2 dòng, 2 cột, kích thước là 300 pixel, khoảng cách giữa nội dung và đường biên của cell là 5:

```
<TABLE BORDER="1" CELLPADDING="5" CELLSPACING="0"
WIDTH="300">
<TR>
<TD WIDTH="74" ALIGN="center">MSSV</TD>
<TD WIDTH="203" ALIGN="center">Họ và tên</TD>
</TR>
<TR>
<TD WIDTH="74">9901234</TD>
<TD WIDTH="203">Trần Đức Vũ</TD>
</TR>
</TABLE>
```

MSSV	Họ và tên
9901234	Trần Đức Vũ

**Hình 3. 6 – Minh họa một bảng đơn giản**

Để trộn (merge)/tách (split) các dòng hoặc cột lại với nhau ta dùng các thuộc tính tương ứng là ROWSPAN hoặc COLSPAN. Ví dụ sau minh họa trong một bảng vừa có sự trộn/tách trên dòng và cột:

```
<TABLE BORDER="1" CELLPADDING="5" CELLSPACING="0"
WIDTH="310">
<TR>
<TD WIDTH="94" ALIGN="center">Mã nhóm</TD>
<TD WIDTH="70" ALIGN="center">
    MSSV</TD>
<TD WIDTH="197" ALIGN="center" COLSPAN="2">
    Họ và tên</TD>
</TR>
<TR>
<TD WIDTH="94" ROWSPAN="2">Nhóm 01</TD>
<TD WIDTH="70">9901234</TD>
<TD WIDTH="123">Trần Đức </TD>
<TD WIDTH="74">Văn</TD>
</TR>
<TR>
<TD WIDTH="70">9901235</TD>
<TD WIDTH="123">Hoàng Minh </TD>
<TD WIDTH="74">Vũ</TD>
</TR>
</TABLE>
```

The screenshot shows a Microsoft Internet Explorer window with the title "Test table - Microsoft Internet Explorer". The address bar shows "C:\DuyLD\WebBook2003\PrePrint\test.htm". The table has three columns: "Mã nhóm", "MSSV", and "Họ và tên". The first row has a single cell for "Mã nhóm" and two cells for "MSSV" and "Họ và tên" respectively. The second row has a single cell for "Mã nhóm" and two cells for "MSSV" and "Họ và tên". The third row has a single cell for "Mã nhóm" and two cells for "MSSV" and "Họ và tên". The "Họ và tên" column is merged across all three rows.

Mã nhóm	MSSV	Họ và tên
Nhóm 01	9901234	Trần Đức Văn
	9901235	Hoàng Minh Vũ

**Hình 3. 7 – Minh họa một bảng có trộn/tách các cột**

Có hai cách để thiết lập bề rộng (WIDTH) của một cell hoặc một bảng, đó là tính theo % và tính theo pixel. Trong nhiều trường hợp, việc đặt theo % sẽ làm cho kích thước bảng thay đổi tùy theo dữ liệu và độ phân giải của màn hình máy người dùng, do đó để đảm bảo tính nhất quán trong hiển thị dữ liệu của bảng, nên chọn cách tính theo pixel.

Trong thiết kế trang web, các bảng biểu đóng vai trò rất quan trọng vì nó là công cụ chính dùng để thể hiện các trình bày phức tạp của trang web. Hiện nay, các phần mềm hỗ trợ soạn thảo trang web hỗ trợ vẫn chưa tốt các thao tác phức tạp trên bảng biểu. Do đó, cách tốt nhất là kết hợp cả hai. Nghĩa là, ngoài việc sử dụng các phần mềm này để phát sinh bảng biểu một cách trực quan, ta phải sử dụng mã HTML để can thiệp khi các phần mềm này không đáp ứng nổi các trình bày phức tạp nhất là khi có các bảng vừa lồng vào nhau, các dòng cột trộn/tách nhiều lần, ...

#### 4. FORM VÀ CÁC THÀNH PHẦN CỦA FORM

Form thường được dùng như là công cụ hỗ trợ nhập liệu cho các ứng dụng trên Web, tương tự như các hộp thoại (dialog) trong các ứng dụng trên Windows. Hoạt động của form thông thường là: ứng dụng hiển thị form để yêu cầu nhận thông tin từ người dùng, người dùng điền các thông tin và kết thúc việc nhập liệu bằng cách submit form. Sau đó dữ liệu sẽ được chuyển đến các chương trình xử lý tương ứng.

Tag **<FORM>** được dùng để thiết lập một form nhập liệu. Các thuộc tính chính là: ACTION (được dùng để chỉ định chương trình

nào sẽ xử lý dữ liệu nhập từ form), METHOD (phương thức chuyển dữ liệu), NAME (tên của form – dùng cho các xử lý sau này).

Người dùng nhập dữ liệu cho form thông qua các ô nhập liệu (thường gọi là các controls) như textbox, checkbox, radio button, push button, dropdown listbox, ... Các ô nhập liệu này được đặt trong phần bao quanh bởi cặp tag **<FORM>** và **</FORM>**. Khi form được submit, dữ liệu mà người dùng nhập vào trong các ô nhập liệu này sẽ được chuyển đến chương trình xử lý form theo dạng **< tên ô nhập liệu >=< giá trị >**. Chính vì vậy mà thuộc tính NAME là thuộc tính rất quan trọng khi khai báo các ô nhập liệu này.

Một khi form được submit, các dữ liệu sẽ được chuyển đến cho chương trình xử lý form theo dạng **< tên control >=< giá trị dữ liệu nhập vào >**. Người ta dùng dấu "&" để ngăn cách dữ liệu của 2 control khác nhau. Ví dụ sau minh họa một form và dữ liệu khi submit có dạng:

<http://localhost/Update.asp?USERID=ledduy&FULLNAME=Le+Dinh+Duy&EMAIL=ledduy@yahoo.com&btnSubmit=Update>

```
<FORM ACTION="Update.asp" METHOD="GET">
<INPUT TYPE="HIDDEN" NAME="USERID" VALUE="ledduy">
<P>Full Name: <INPUT TYPE="TEXT" NAME="FULLNAME" SIZE="25" VALUE="Le Dinh Duy"><BR>
Email: <INPUT TYPE="TEXT" NAME="EMAIL" SIZE="25" VALUE="ledduy@yahoo.com"> <BR>
<INPUT TYPE="SUBMIT" NAME="btnSubmit" VALUE="Update">
</P>
</FORM>
```

**Hình 3. 8 – Cách dữ liệu được chuyển đến trình xử lý form**

Một số kí tự đặc biệt sẽ được chuyển đổi trước khi dữ liệu được chuyển đi:

Kí tự nhập vào	Kí tự thay thế được chuyển đi
khoảng trắng	+
=	%3D
dấu xuống dòng (line feed)	%0A
về đầu dòng (carriage return)	%0D
%	%25
&	%38

Để tạo các ô nhập liệu dạng hộp văn bản, nút nhấn, checkbox, radio button, ... ta dùng tag <INPUT> và đặt giá trị tương ứng với các loại ô nhập liệu cho thuộc tính TYPE.

#### 4.1. Hộp văn bản (TextBox)

Dùng để nhập dữ liệu trên một dòng. Để tạo ô nhập liệu dạng này, ta chỉ định thuộc tính TYPE="TEXT" trong tag <INPUT>. Ví dụ sau minh họa mã HTML dùng để tạo ra một ô nhập liệu dạng hộp văn bản, có tên là USRNAME, kích thước là 25 kí tự, giá trị mặc định là ledduy:

```
Username: <INPUT TYPE="TEXT" NAME="USRNAME"
VALUE="ledduy" SIZE="20" MAXLENGTH="128">
```

Thuộc tính SIZE dùng để chỉ số kí tự hiển thị trong ô nhập liệu (đây cũng chính là chiều rộng của ô nhập liệu). Thuộc tính MAXLENGTH dùng để chỉ số kí tự tối đa có thể được nhập.

Khi muốn chỉ định dữ liệu nhập vào control dưới dạng mật khẩu (nghĩa là các kí tự nhập vào sẽ không được hiển thị mà thay vào đó là các dấu \*), ta chỉ định thuộc tính TYPE="PASSWORD". Ví dụ:

```
Password: <INPUT TYPE="PASSWORD" NAME="USRPASSWORD"
VALUE="123456" SIZE="20">
```

#### 4.2. Nút nhấn (Button)

Khi chỉ định thuộc tính TYPE="BUTTON" trong tag <INPUT> ta sẽ tạo được một nút nhấn nói chung.

Trong form thông thường có hai loại nút nhấn đặc biệt đó là nút nhấn SUBMIT và nút nhấn RESET. Nút nhấn RESET là nút nhấn mà khi người dùng nhấn chuột vào, các dữ liệu của các ô nhập liệu khác đều được đặt về giá trị mặc định lúc khởi tạo. Để tạo ta sẽ chỉ

định thuộc tính TYPE="RESET" trong tag <INPUT>. Trong khi đó nút nhấn SUBMIT là nút nhấn dùng để kết thúc việc nhập liệu của form. Để tạo nút nhấn dạng này, ta chỉ định thuộc tính TYPE="SUBMIT" trong tag <INPUT>.

Giá trị gán cho thuộc tính VALUE chính là nhãn của nút nhấn.

Ví dụ sau minh họa việc tạo ra hai nút nhấn Submit và Reset:

```
<INPUT TYPE="SUBMIT" NAME="BTNSUBMIT"
VALUE="Submit">&nbsp; &nbsp;
<INPUT TYPE="RESET" NAME="BTNRESET" VALUE="Reset">
```

#### 4.3. Radio button

Dùng để chọn duy nhất một trong tập các lựa chọn được liệt kê.

Để tạo ra một nhóm các radio button, ta phải chỉ định thuộc tính TYPE="RADIO" trong tag <INPUT> của các radio button và các radio button này phải có cùng giá trị của thuộc tính NAME.

Khi form được submit, dữ liệu của radio button được chọn (giá trị gán cho thuộc tính VALUE) sẽ được chuyển đi cùng với tên của radio button này.

Để đặt một radio button là mặc định, ta thêm vào thuộc tính CHECKED. Ví dụ:

Ví dụ sau minh họa lựa chọn giới tính (Sex) thông qua hai radio button Nam (Male) và Nữ (Female):

```
Sex: <INPUT TYPE="RADIO" NAME="USRSEX" CHECKED
VALUE="M">Male
<INPUT TYPE="RADIO" NAME="USRSEX" VALUE="W">Female
```

#### 4.4. Checkbox

Dùng để chọn một hoặc nhiều trong tập các lựa chọn được đề nghị.

Khi ta chỉ định thuộc tính TYPE="CHECKBOX" trong tag <INPUT>, ta sẽ tạo ra được một ô nhập liệu kiểu checkbox. Tương tự như radio button, khi một checkbox được check, giá trị trả về của nó tương ứng với giá trị của thuộc tính VALUE.

Để đặt một checkbox là ON, ta thêm vào thuộc tính CHECKED.

Ví dụ sau minh họa một nhóm các checkbox dùng để lấy thông tin về các ngôn ngữ được chọn:

```
Language: <INPUT TYPE="CHECKBOX" NAME="USRLANG"
VALUE="ENG" CHECKED> English
<INPUT TYPE="CHECKBOX" NAME="USRLANG"
VALUE="FR">French
<INPUT TYPE="CHECKBOX" NAME="USRLANG" VALUE="JP"
CHECKED>Japanese
```

#### 4.5. Dropdown listbox

Dùng để lựa chọn Ta sử dụng tag `<SELECT>` để tạo. Mỗi mục của dropdown listbox sẽ được định nghĩa bằng tag `<OPTION>`. Ví dụ sau minh họa việc chọn một nghề nghiệp đã được liệt kê sẵn:

```
Occupation: <SELECT SIZE="1" NAME="Occupation">
<OPTION SELECTED VALUE="0">Other</OPTION>
<OPTION VALUE="1">Engineer</OPTION>
<OPTION VALUE="2">Teacher</OPTION>
<OPTION VALUE="3">Doctor</OPTION>
<OPTION VALUE="4">Worker</OPTION>
</SELECT>
```

Để đặt một mục chọn trong dropdown listbox là mặc định, ta thêm vào thuộc tính `SELECTED`.

Để tạo một multi-select listbox ta đặt thêm thuộc tính `SIZE` và `MULTIPLE` vào trong tag `<SELECT>`.

Khi form được submit, dữ liệu của tương ứng với mục chọn (giá trị gán cho thuộc tính `VALUE` của tag `<OPTION>`) trong dropdown listbox sẽ được chuyển đi cùng với tên của dropdown listbox này. Ví dụ trong trường hợp ví dụ trên, khi người dùng chọn Worker, giá trị trả về của ô nhập liệu này sẽ là 4.

#### 4.6. Ô dữ liệu ẩn (Hidden field)

Dùng để lưu trữ các thông tin của form cần thiết cho chương trình xử lý sau này nhưng lại không hiển thị dưới dạng một control nào. Ta lấy ví dụ form cập nhật thông tin một sinh viên. Các thông tin mà người dùng có thể cập nhật thường là Họ tên, Ngày tháng năm sinh, ... sẽ được hiển thị trên các ô nhập liệu của form. Tuy nhiên để chương trình xử lý form cập nhật này có thể biết cần cập nhật sinh viên nào, cần phải có thêm thông tin về Mã số sinh viên (giả sử đóng vai trò là khóa chính trong cơ sở dữ liệu). Thông tin về Mã số sinh

viên sẽ được lưu trữ trong một ô nhập liệu của form có kiểu là `HIDDEN`. Thông tin này sẽ không được hiển thị trên màn hình của người dùng nhưng sẽ được chuyển đi mỗi khi form submit.

Để tạo một ô nhập liệu có kiểu là `HIDDEN`, ta chỉ định thuộc tính `TYPE="HIDDEN"` trong tag `<INPUT>` như ví dụ sau:

```
<INPUT TYPE="HIDDEN" NAME="USRID" VALUE="ledduy">
```

Mỗi khi form được submit, ngoài các ô nhập liệu đã được hiển thị trên màn hình, ta cũng sẽ có thêm một ô nhập liệu có tên là "USRID" và giá trị là "ledduy" được chuyển đến cho trình xử lí

#### 4.7. Vùng văn bản (TextArea)

Dùng để nhập dữ liệu trên nhiều dòng. Để tạo ô nhập liệu dạng này ta dùng tag `<TEXTAREA>`. Dữ liệu nằm giữa cặp tag `<TEXTAREA>` và `</TEXTAREA>` chính là giá trị trả về của control dạng này. Ví dụ:

```
Other information: <TEXTAREA ROWS="3" NAME="OtherInfo"
COLS="20"></TEXTAREA>
```

Thuộc tính `ROWS` và `COLS` dùng để chỉ số dòng và cột của vùng dữ liệu nhập.

Ví dụ sau minh họa một form nhập liệu gồm có đầy đủ các ô nhập liệu đã kể ở trên:

```
<FORM ACTION="URL">
Username: <INPUT TYPE="TEXT" NAME="USRNAME"
VALUE="ledduy" SIZE="20" MAXLENGTH="128"> <BR>
Password: <INPUT TYPE="PASSWORD" NAME="USRPASSWORD"
VALUE="123456" SIZE="20"><BR>
Sex: <INPUT TYPE="RADIO" NAME="USRSEX" CHECKED
VALUE="M">Male
<INPUT TYPE="RADIO" NAME="USRSEX" VALUE="W">Female
<BR>
Language: <INPUT TYPE="CHECKBOX" NAME="USRLANG"
VALUE="ENG" CHECKED> English
<INPUT TYPE="CHECKBOX" NAME="USRLANG"
VALUE="FR">French
<INPUT TYPE="CHECKBOX" NAME="USRLANG" VALUE="JP"
CHECKED>Japanese<BR>
Occupation: <SELECT SIZE="1" NAME="Occupation">
<OPTION SELECTED VALUE="0">Other</OPTION>
<OPTION VALUE="1">Engineer</OPTION>
<OPTION VALUE="2">Teacher</OPTION>
<OPTION VALUE="3">Doctor</OPTION>
```

```

<OPTION VALUE="4">Worker</OPTION>
</SELECT>&nbsp; <BR>
Other information: <TEXTAREA ROWS="3" NAME="OtherInfo"
COLS="20"></TEXTAREA><P><INPUT TYPE="SUBMIT"
NAME="BTNSUBMIT" VALUE="Submit">&nbsp;&nbsp;&nbsp;
<INPUT TYPE="RESET" NAME="BTNRESET" VALUE=" Reset
"></P>
</FORM>

```

Form Sample - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Media

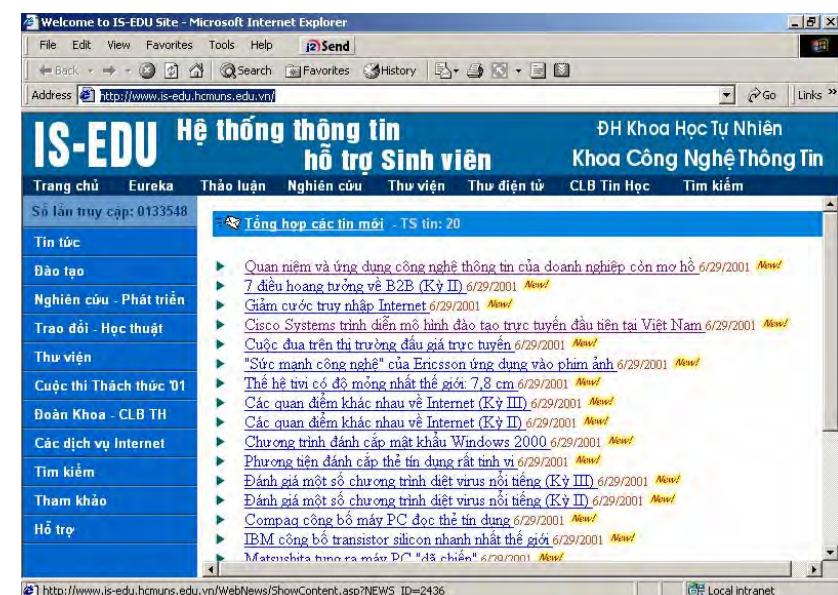
Address C:\DuyLD\WebBook2003\PrePrint\test.htm

Username: led duy  
Password: \*\*\*\*\*  
Sex: ♂ Male ♀ Female  
Language: English French Japanese  
Occupation: Other  
Other information:  
Submit Reset

Hình 3.9 – Minh họa một form nhập liệu

## 5. FRAME

Nội dung hiển thị trên trình duyệt có thể được tích hợp từ nhiều cửa sổ khác nhau, mỗi cửa sổ chứa một URL tương ứng với một trang web. Ví dụ sau cho ta thấy có ba cửa sổ, một cửa sổ chứa thực đơn nằm ngang, một cửa sổ chứa thực đơn bên trái và một cửa sổ chứa nội dung bên phải. Trong trường hợp này chúng ta dùng tag <FRAMESET> và <FRAME> để định nghĩa.



Hình 3.10 - Minh họa frame

Tag <FRAMESET> định nghĩa cách tổ chức của các frame. Tag <FRAME> định nghĩa chi tiết từng frame.

Các thuộc tính của tag <FRAMESET> là: ROWS (định nghĩa số frame được phân theo chiều dọc), COLS (định nghĩa số frame được phân theo chiều ngang), FRAMEBORDER (định nghĩa đường viền khung của các frame con, giá trị là YES hoặc NO)

Các thuộc tính cơ bản của tag <FRAME> là: SRC (địa chỉ URL của trang web sẽ hiển thị trong frame), NAME (tên của frame, có thể dùng trong thành phần TARGET của hyperlink), BORDER (đường viền khung, nếu không muốn có đường viền thì đặt giá trị này về 0), MARGINHEIGHT, MARGINWIDTH (canh chỉnh lề), SCROLLING (có hiển thị scrollbar hay không, các giá trị thường dùng là YES, NO, AUTO), NORESIZE (không cho phép người dùng hiệu chỉnh kích thước của frame window)

Để định nghĩa các frame lồng nhau ta dùng các tag <FRAMESET> lồng nhau

Ví dụ sau minh họa định nghĩa của trang web trên:

```

<HTML>
<HEAD>
<TITLE>Welcome to IS-EDU Site</TITLE>
</HEAD>
<FRAMSET BORDER="0" ROWS="80,*" FRAMEBORDER="0">
  <FRAME NAME="Banner" SCROLLING="NO" NORESIZE
TARGET="Contents" SRC="GlobalMenu.asp" MARGINWIDTH="0"
MARGINHEIGHT="0">
  <FRAMESET COLS="175,*">
    <FRAME NAME="Contents" TARGET="Main" SRC="MainMenu.asp"
SCROLLING="AUTO">
      <FRAME NAME="Main" SCROLLING="AUTO"
SRC="WebNews/LastNewsAll.asp">
    </FRAMESET>
  </FRAMESET>
</HTML>

```

## 6. CASCADING STYLE SHEET (CSS)

Sử dụng style sheet giúp cho người soạn thảo trang web dễ dàng hơn trong thiết kế và hiệu chỉnh các trang web đồng thời đảm bảo tính nhất quán trong trình bày của website.

Một style sheet là một mẫu định dạng (template) của các HTML tag. Khái niệm style sheet trong các trang Web rất tương tự với khái niệm templates trong MS Word. Bạn có thể thay đổi sự trình bày của một văn bản Word thông qua việc thay đổi các style trong văn bản này. Một cách tương tự, bạn có thể thay đổi sự trình bày của các trang Web bằng cách thay đổi các style sheet được gán cho các HTML tag.

### 6.1. Các lợi ích của style sheet:

Sử dụng được các thuộc tính như leading, margins, indents, point sizes, and text background colors trong trang web. Đây là các thuộc tính mà các tag HTML không hỗ trợ trực tiếp.

Thay đổi thuộc tính của từng trang web hoặc toàn bộ các trang web trong website mà không cần phải hiệu chỉnh các dòng liên quan đến định dạng trong các tập tin HTML. Ví dụ, nếu ta đã dùng tag <FONT> với thuộc tính FACE=Arial để chỉ định font chữ cho các văn bản của mọi trang web trong website, ta sẽ phải hiệu chỉnh từng tag <FONT> một trong toàn bộ các trang web nếu ta muốn đổi tất cả các văn bản này sang font chữ khác, ví dụ như Tahoma.

### 6.2. Các cách sử dụng style sheets

Có 3 cách sử dụng style sheets tùy vào nhu cầu thiết kế:

- Bằng cách liên kết (**linking**) đến một tập tin chứa các style sheet. Cách này cho phép bạn thay đổi cách trình bày của nhiều trang web một cách dễ dàng thông qua việc sửa đổi một tập tin chứa các style sheet này mà thôi.
- Bằng cách nhúng (**embedding**) style sheet vào trong tập tin HTML. Cách này cho phép bạn thay đổi cách trình bày của từng trang web một thông qua việc sửa định nghĩa style sheet ban đầu.
- Bằng cách thêm các **inline styles** vào trang HTML. Cách này cho phép bạn thay đổi một cách nhanh chóng cách thể hiện của từng tag, hoặc nhóm các tag hoặc một khối thông tin trên trang web.

### 6.3. Cú pháp cơ bản:

Cả hai kiểu linked and embedded style sheets đều chứa một hoặc nhiều định nghĩa style. Một định nghĩa style (style definition) gồm có một HTML tag, sau đó là danh sách các thuộc tính của tag đó được đặt bên trong các dấu "{} và {}". Mỗi thuộc tính được xác định bằng tên thuộc tính, theo sau là dấu ":" và giá trị của thuộc tính. Các thuộc tính được phân cách với nhau bởi dấu ",". Ví dụ sau minh họa một style definition gán cho tag <FONT> : **<FONT{font-size: 15pt; font-weight: bold}>**

Sau định nghĩa một style mới, để dùng nó trong các tag, ta gán tên style cho thuộc tính class trong tag đó. Ví dụ sau minh họa một style có tên là txtInputText, sau đó được dùng để chỉ định font chữ cho dropdown listbox:

```

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
<TITLE>Test</TITLE>
<STYLE>
<!--
.txtInputText { font-family: Arial; font-size: 10pt; color: #0000FF }
-->
</STYLE>
</HEAD>

```

```
<BODY>
<P>
<SELECT NAME="NEWS" CLASS="txtInputText">
<OPTION VALUE="TT">Tuổi Trẻ</OPTION>
<OPTION VALUE="LD">Lao Động</OPTION>
<OPTION VALUE="TN">Thanh Niên</OPTION>
</SELECT>
</P>

</BODY>
</HTML>
```

#### 6.4. Liên kết đến một Style Sheet

Để có thể sử dụng cùng một style sheet cho bất kì trang web nào của site, trước tiên bạn phải tạo một tập tin .css lưu trữ các định nghĩa style, sau đó trong bất kì các trang web nào muốn sử dụng các style sheet này, bạn phải tiến hành thủ tục liên kết. Ví dụ, nếu bạn có một tập tin style sheet có tên là MYSTYLES.CSS đặt tại địa chỉ <http://internet-name/mystyles.css>, để liên kết vào trang web của mình, bạn thêm các dòng sau nằm giữa tag <HEAD>

```
<HEAD>
<TITLE>Title of article</TITLE>
<LINK REL=STYLE
      HREF="http://internet-name/mystyles.css"
      TYPE="text/css">
</HEAD>
```

#### 6.5. Nhúng một STYLE Block vào trang HTML

Để nhúng a style sheet, bạn thêm <STYLE> </STYLE> block vào đầu trang web giữa tag <HTML> và <BODY>. Cách này cho phép bạn thay đổi trình bày của chỉ trang web hiện tại mà thôi. Theo sau tag <STYLE> là bất kì định dạng nào và kết thúc bởi tag </STYLE>. Ví dụ sau định nghĩa các style cho các tag <BODY>, <H1>, <H2>, và <P>:

```
<HTML>
<STYLE TYPE="text/css">
<!--
BODY {font: 10pt "Arial"}
H1 {font: 15pt/17pt "Arial";
   font-weight: bold;
   color: maroon}

H2 {font: 13pt/15pt "Arial";
   font-weight: bold;
   color: blue}
P {font: 10pt/12pt "Arial";
   color: black}
-->
</STYLE>
<BODY>
...
</BODY>
</HTML>
```

#### 6.6. Sử dụng Inline Styles

Các định nghĩa Inline style ảnh hưởng trực tiếp đến tag hiện hành. Chúng được nhúng vào bên trong tag bằng cách sử dụng tham số STYLE . Ví dụ:

```
<SELECT STYLE="font-family: Arial; font-size: 10pt; color: #0000FF"
        NAME="NEWS">
    <OPTION VALUE="TT">Tuổi Trẻ</OPTION>
    <OPTION VALUE="LD">Lao Động</OPTION>
    <OPTION VALUE="TN">Thanh Niên</OPTION>
</SELECT>
```

# THỰC HÀNH

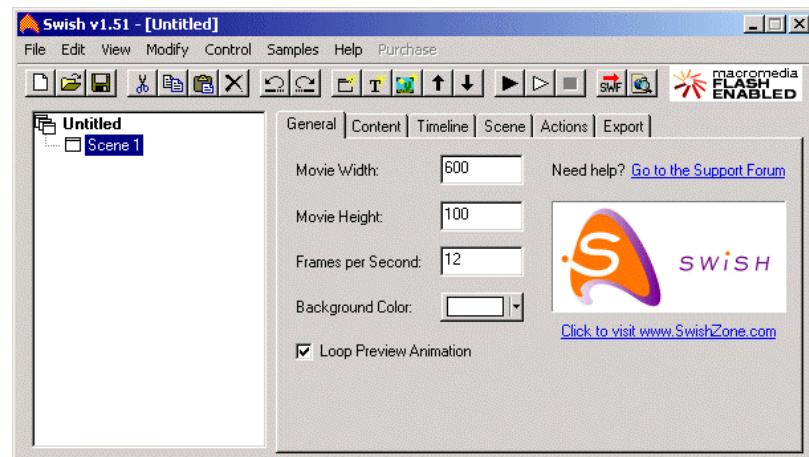
## 1. SỬ DỤNG PHẦN MỀM SWISH ĐỂ TẠO CÁC HIỆU ỨNG ĐỒ HỌA ĐƠN GIẢN

Phần mềm Swish cho phép tạo các hiệu ứng động từ đơn giản đến phức tạp. Kết quả cuối cùng của ứng dụng này là tập tin .swf có thể chạy được trên các trình duyệt có cài đặt sẵn Macromedia Flash Player.

Đối tượng chính khi soạn thảo một hiệu ứng động trên Swish được gọi là Movie (đoạn phim). Một movie là tập hợp của nhiều Scene (phân cảnh) khác nhau, mỗi scene bao gồm tập hợp các Objects (đối tượng). Hai đối tượng chính được hỗ trợ trong phiên bản Swish 1.51 là Text Objects (tập các ký tự) và Image Objects (hình ảnh). Các đối tượng của scene có thể hoạt hình (animate) thông qua việc thiết lập các Effects (hiệu ứng). Các hiệu ứng đã được lập trình sẵn để có thể sinh ra các Frames (khung hình) tương ứng.

### 1.1. Cài đặt Swish

Cài đặt Swish từ đĩa CD. Khởi động ứng dụng Swish. Màn hình làm việc chính xuất hiện cùng với View Window



Hình 3. 11 – Màn hình làm việc chính của Swish



Hình 3. 12 – Cửa sổ View Window

Sau khi cài đặt xong, phải đăng kí sử dụng. Một đăng kí sử dụng hợp lệ sẽ làm mờ đi mục Purchase trong màn hình làm việc của Swish.

### 1.2. Thiết lập các thông số cơ bản cho movie

Trong tab General, hiệu chỉnh các thông số liên quan đến chiều rộng, chiều cao, màu nền, tốc độ hiển thị của các khung hình của movie. Lưu ý rằng, ta cũng có thể hiệu chỉnh chiều rộng và chiều cao bằng cách dùng chuột hiệu chỉnh View Window.



Hình 3. 13 – Hiệu chỉnh các thông số cơ bản của movie

### 1.3. Tạo lập một movie đơn giản

Ngay khi khởi động, ứng dụng tự động tạo ra một scene mới có tên là Scene1. Nếu muốn chèn thêm scene khác, hãy nhấn nút bên trái nút hình chữ T trên thanh công cụ hoặc chọn trên thực đơn Modify/Insert Scene.

Ta chèn một đối tượng văn bản vào scene bằng cách nhấn nút hình chữ T trên thanh công cụ hoặc chọn trên thực đơn Modify/Insert Text. Trong trường hợp có nhiều đối tượng văn bản

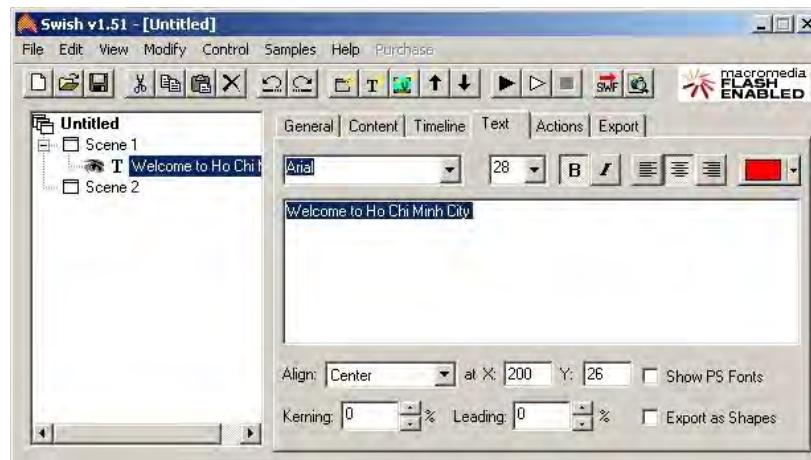
trong cùng một scene, để thiết lập thứ tự, ta chọn đối tượng rồi nhấn các nút mũi tên lên hoặc mũi tên xuống cho đến khi đạt được vị trí mong muốn.

Để thay đổi vị trí mặc định của văn bản vừa hiện ra, click vào nó trong cửa sổ View Window và kéo đến vị trí mong muốn. Ví dụ ta kéo đoạn văn bản về phía phải của View Window:



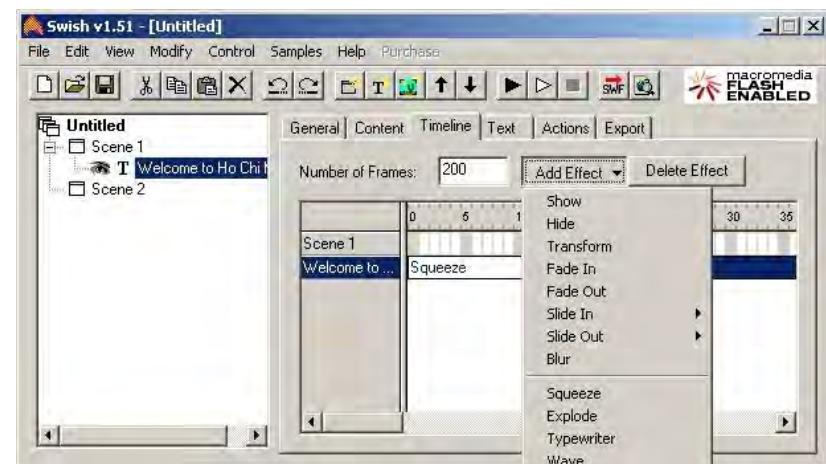
**Hình 3. 14 – Thay đổi vị trí của văn bản vừa chèn vào**

Để thay đổi nội dung đoạn văn bản, chọn tab Text trong màn hình làm việc chính và thay thế nó. Chúng ta cũng có thể thay đổi font chữ, kích thước, màu sắc, kiểu chữ (bold, italic, ...).



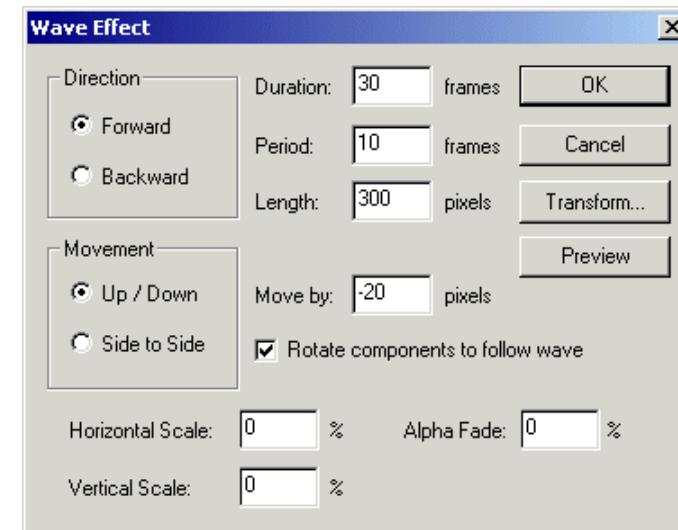
**Hình 3. 15 – Hiệu chỉnh nội dung và các thuộc tính của văn bản**

Để thêm vào các hiệu ứng cho đối tượng nào trong scene, chọn Timeline Tab và click trên đối tượng cần đặt hiệu ứng, sau đó nhấn nút Add Effect



**Hình 3. 16 – Thiết lập các hiệu ứng cho các đối tượng**

Đối với một số hiệu ứng, sẽ có hộp thoại yêu cầu bạn nhập các thuộc tính của hiệu ứng. Nếu muốn giữ các thuộc tính mặc định, hãy chọn OK khi màn hình thiết lập hiệu ứng hiện ra. Ví dụ:



**Hình 3. 17 – Thiết lập các thông số cho một hiệu ứng**

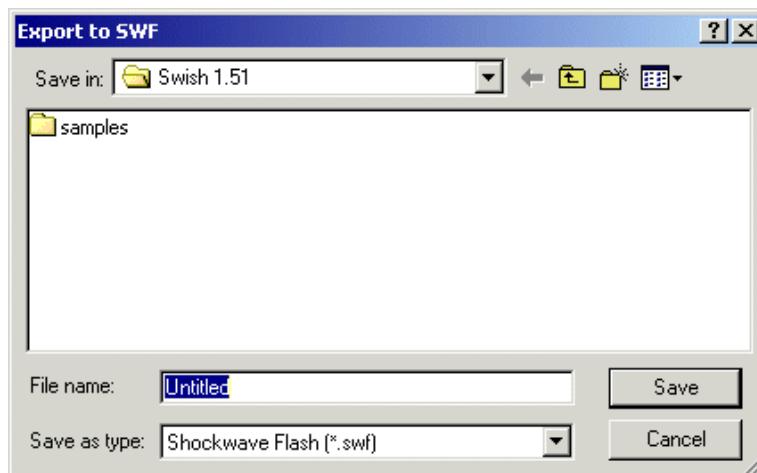
Để xem thử sự trình diễn của các hiệu ứng, ta nhấn nút Play Movie trên Toolbar. Muốn dừng để tiếp tục hiệu chỉnh, nhấn nút Stop.

Sau khi làm xong, ta nhấn nút Save Movie trên thanh công cụ hoặc chọn trên thực đơn File/Save. Tập tin sẽ được lưu với phần mở rộng là .swi. Sau này, muốn hiệu chỉnh, ta có thể mở lại tập tin này.

Để kiểm tra movie sau này sẽ thực hiện như thế nào trên trình duyệt, nhấn vào nút Test In Browser nằm ngoài cùng trên thanh công cụ.

#### 1.4. Đưa các movie vào trang web

Để đưa movie vừa tạo vào trang web, trước tiên phải kết xuất movie ra dạng .swf bằng cách nhấn nút SWF trên thanh công cụ hoặc chọn trên thực đơn File/Export to SWF.



**Hình 3. 18 – Kết xuất ra tập tin .swf**

Sau đó, nhấn vào tab Export, click vào copy HTML to Clipboard

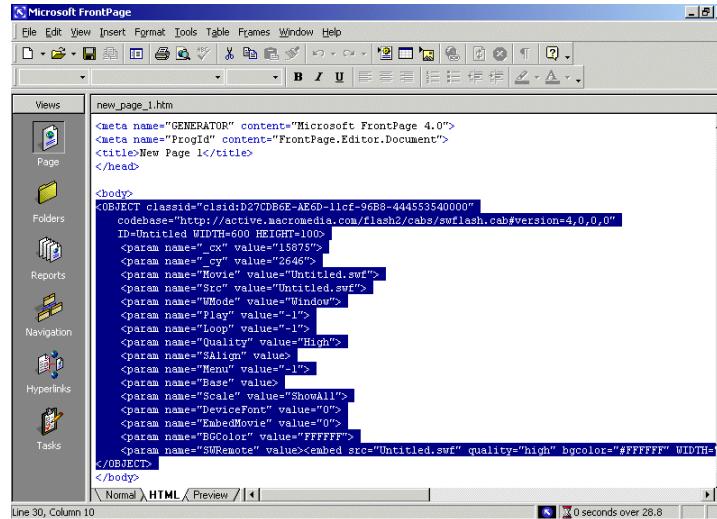


**Hình 3. 19 – Màn hình Export**

Thao tác này sẽ đưa vào clipboard một đoạn mã HTML dùng để mô tả movie vừa rồi như sau:

```
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
        codebase="http://active.macromedia.com/flash2/cabs/swfsh.cab#version=4,0,0,0"
        ID=Intro WIDTH=400 HEIGHT=50>
<PARAM NAME=movie VALUE="Intro.swf">
<PARAM NAME=quality VALUE=high>
<PARAM NAME=bgcolor VALUE=#FFFFFF>
<EMBED src="Intro.swf" quality=high bgcolor=#FFFFFF
        WIDTH=400 HEIGHT=50 TYPE="application/x-shockwave-flash"
        PLUGINSPAGE="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=ShockwaveFlash">
</EMBED>
</OBJECT>
```

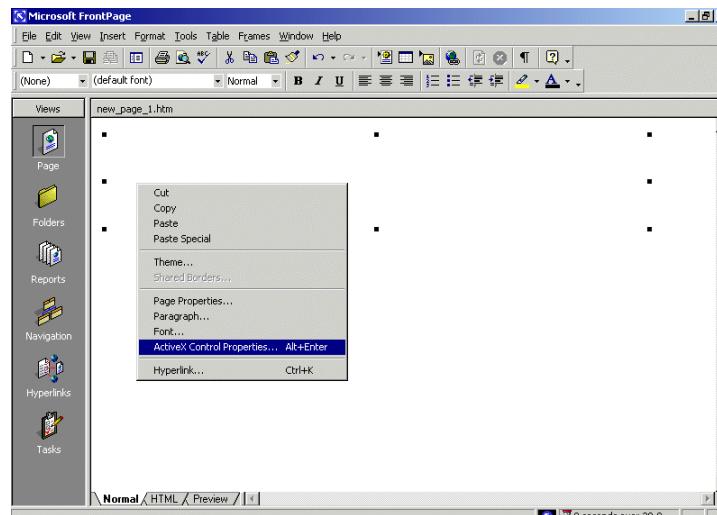
Sau đó, dán đoạn code HTML này vào vị trí cần chèn movie trên trang web mà đang soạn thảo (chú ý sử dụng View HTML).



Hình 3. 20 – Dán đoạn mã HTML mô tả movie vào trang web

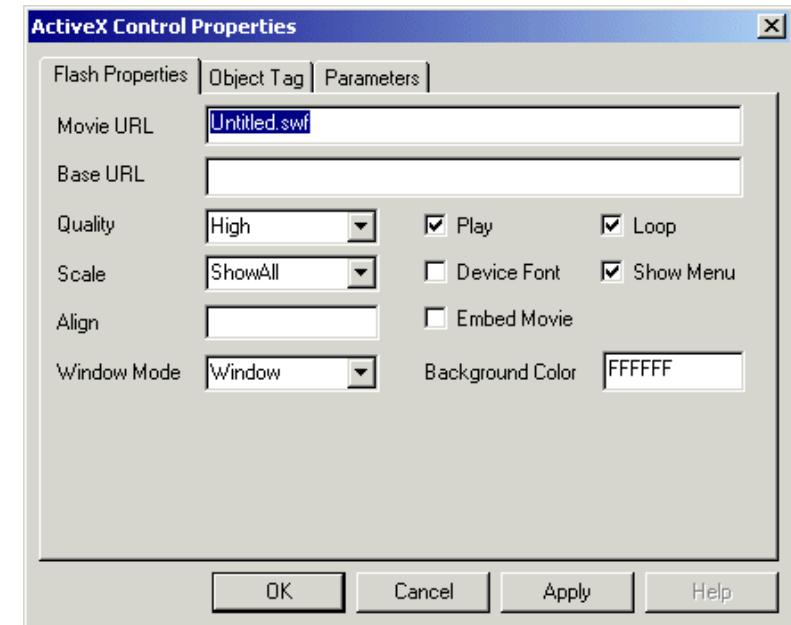
### 1.5. Đặt lại các thông số của movie trong FrontPage

Chuyển qua View Normal, click phải chuột lên đối tượng rồi chọn ActiveX Control Properties...



Hình 3. 21 – Đặt lại thông số cho movie trên trang web

Thuộc tính quan trọng nhất là Movie URL. Nếu ta đặt movie ở thư mục khác với thư mục của trang web hiện hành, ta phải chỉ định đường dẫn đến movie trong ô nhập liệu này



Hình 3. 22 – Các thuộc tính của movie

## 2. SỬ DỤNG PHẦN MỀM BUTTON STUDIO ĐỂ TẠO CÁC NÚT NHẤN DẠNG ĐỒ HỌA ĐƠN GIẢN

Button Studio là phần mềm hỗ trợ thiết kế và tạo các button một cách nhanh chóng và dễ dàng. Các button được tạo ra có thể được dùng cho các trang web, presentations, hay các ứng dụng cần dùng chúng.

Bước đầu tiên nên chọn hình dáng của nút nhấn bằng cách chọn vào tab Shape.

Sau đó ta chọn kích thước và định vị của văn bản trong nút nhấn bằng cách chọn vào nút Size+Positioning. Không nên chọn Auto Size vì lúc đó kích thước của nút nhấn sẽ phụ thuộc vào nội dung văn bản hiển thị trên đó.



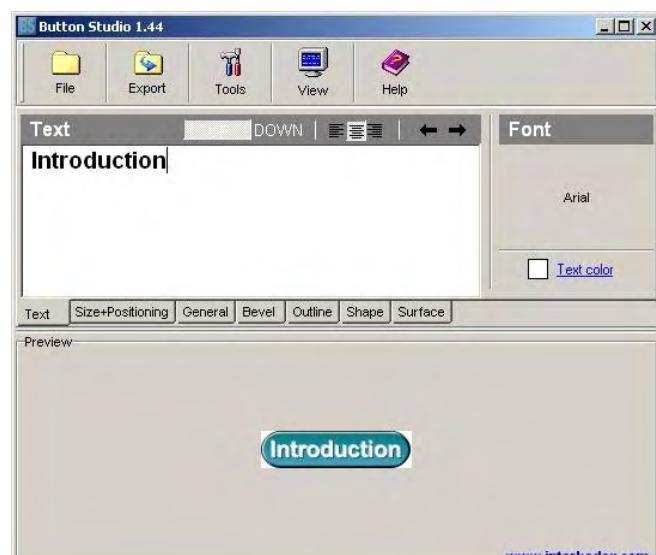
Hình 3. 23 – Màn hình làm việc của phần mềm Button Studio



Hình 3. 25 – Thiết lập kích thước và định vị



Hình 3. 24 – Các dạng nút nhấn được hỗ trợ sẵn



Hình 3. 26 – Văn bản hiển thị trên nút nhấn

Sau đó sang tab Text để gõ nội dung văn bản sẽ hiển thị trên nút nhấn. Đồng thời thiết lập các thuộc tính của văn bản như là font chữ, kích thước, màu sắc.

Sau khi làm xong, ta có thể lưu thiết kế lại dưới dạng tập tin .bs bằng cách chọn thực đơn File/Save As. Sau này muốn sửa lại, ta có thể mở tập tin và tiếp tục chỉnh sửa.

Để có thể dùng được trên trang web, ta phải chọn chức năng Export trên thực đơn ra các dạng tập tin hình ảnh .gif, .jpg.

### 3. SỬ DỤNG PHẦN MỀM XARA WEBSTYLE ĐỂ TẠO CÁC HIỆU ỨNG ĐỒ HỌA

Xara WebStyle là phần mềm hỗ trợ tạo các hiệu ứng đồ họa từ đơn giản đến phức tạp để có thể đưa vào trang web một cách nhanh nhất. Xara WebStyle có thể được dùng để tạo các Headings, banners, nút nhấn, ảnh nền, ...



Hình 3. 27 – Màn hình làm việc của Xara WebStyle

Để tạo đối tượng nào, hãy click vào đối tượng đó. Ví dụ như Heading, Buttons, Bullets, Background. Trong màn hình tạo từng đối tượng nếu muốn trở về màn hình chính thì chọn nút Graphic Type.

Giả sử muốn tạo một đối tượng đồ họa Headings, ta click vào biểu tượng Headings. Màn hình hỗ trợ tạo các Heading xuất hiện.



Hình 3. 28 – Màn hình hỗ trợ tạo các Headings

Nhấn vào nút Style bên tay trái, các mẫu Headings được xây dựng sẵn sẽ hiện ra. Chọn một mẫu, ví dụ như Web Style.

Để thay đổi văn bản của Heading, ta nhấn vào nút Text và gõ vào đoạn văn bản ví dụ như: Welcome to Ho Chi Minh City. Ta có thể thay đổi font chữ, đặt chữ in đậm, in nghiêng tùy thích ở đây.

Các nút tiếp theo là Color và Texture cho phép hiệu chỉnh màu và mẫu tô của văn bản.

Sau đó, ta có thể thiết lập bóng bằng cách nhấn vào nút Shadow.

Mục Themes cho phép lưu lại các thiết lập về kiểu, màu sắc, tạo bóng, ... kể trên để có thể dùng lại cho thiết kế tương tự sau này.

Mục Save để kết xuất đối tượng ra tập tin ảnh .GIF hoặc .JPG dùng trong trang web



Hình 3.29 – Hiệu chỉnh văn bản của Heading



Hình 3.31 – Thiết lập shadow



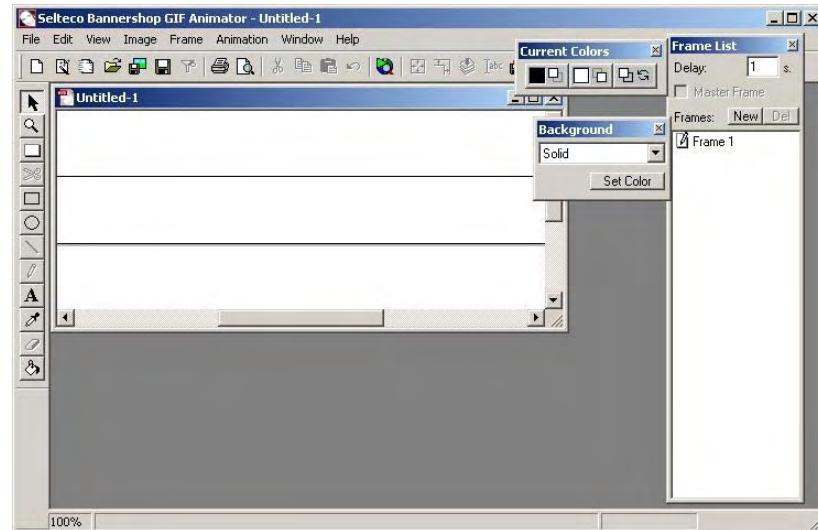
Hình 3.30 – Thiết lập texture



Hình 3.32 – Kết xuất ra tập tin ảnh dùng trong trang web

#### 4. SỬ DỤNG PHẦN MỀM BANNERSHOP GIF ANIMATOR ĐỂ TẠO CÁC ẢNH ĐỘNG

Bannershop GIF Animator là phần mềm hỗ trợ tạo các ảnh web động. Các ảnh web động được lưu dưới định dạng .GIF có thể được dùng trong các trang web và các chương trình khác.

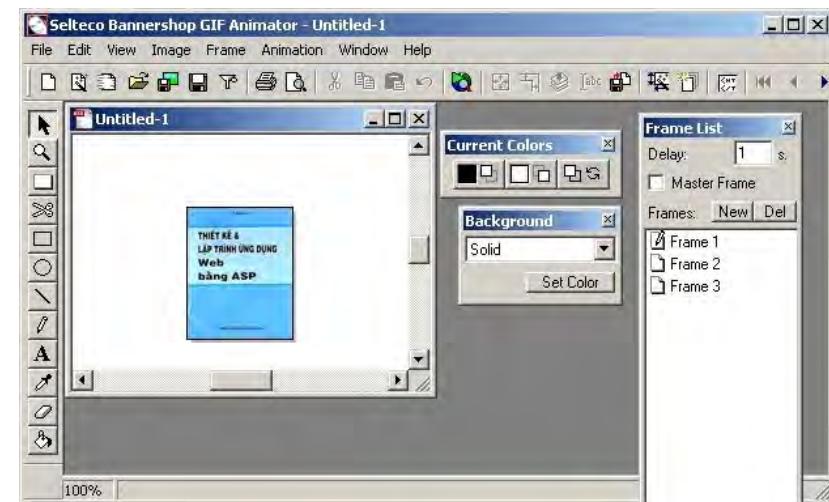


Hình 3.33 – Màn hình làm việc của BannerShop Gif Animator

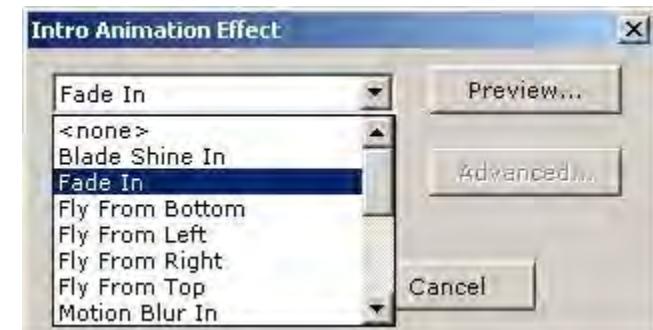
Để tạo một ảnh web động, trước tiên hãy thiết lập các thông số của frame như kích thước, màu nền, ... bằng cách chọn trên thực đơn Animation/Frame Size, Frame/Set Background Color phù hợp với các kích thước các ảnh sẽ được chèn vào từng frame.

Sau đó, chọn trên thực đơn chức năng Image/Import Image để chèn các ảnh vào từng frame một. Sau khi làm xong một frame, hãy chọn trên thực đơn chức năng Frame/Insert New Frame để chèn thêm một frame mới rồi lại tiếp tục chèn ảnh vào từng frame này.

Tiếp theo ta thiết lập các hiệu ứng Animation cho các frame. Intro Animation để chỉ hiệu ứng lúc bắt đầu hiển thị frame. Outro Animation để chỉ hiệu ứng kết thúc frame trước khi chuyển sang frame mới.



Hình 3.34 – Tạo ra các frame và chèn ảnh vào các frame



Hình 3.35 – Các hiệu ứng của frame

Sau khi thực hiện xong việc đặt hiệu ứng cho các frame, nhấn nút F9 hoặc chọn trên thực đơn chức năng Animation/Play Animation để xem kết quả.

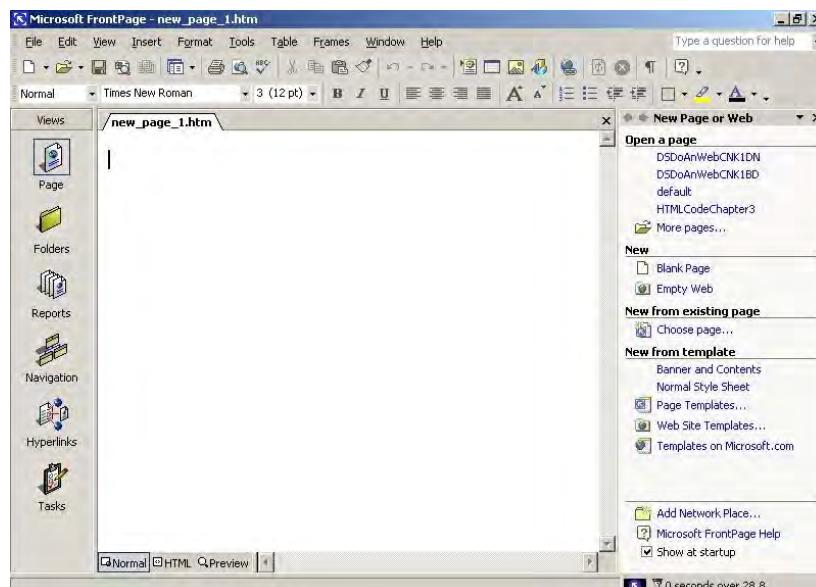
Cuối cùng, ta chọn trên thực đơn chức năng File/Save Document để lưu thiết kế trên dưới dạng tập tin .bsd. Tập tin này sau đó có thể mở lại để hiệu chỉnh. Chọn chức năng File/Save Gif Animation để kết xuất ra tập tin .gif dùng trong các ứng dụng web.

## 5. MỘT SỐ THAO TÁC CƠ BẢN KHI THIẾT KẾ TRANG WEB BẰNG MICROSOFT FRONTPAGE

### 5.1. Giới thiệu về FrontPage

MS FrontPage cùng với Macromedia Dreamweaver MX hiện đang là các phần mềm hỗ trợ thiết kế web thông dụng nhất hiện nay. Chức năng chính của các phần mềm này là hỗ trợ người dùng tạo các trang web một cách dễ dàng mà không cần thiết phải biết nhiều kiến thức về HTML.

Khi khởi động, màn hình làm việc của MS FrontPage như sau:



**Hình 3. 36 – Màn hình làm việc của MS FrontPage**

Ở bên tay trái là cửa sổ Views. Để xem cấu trúc của các website dưới dạng các tập tin và cây thư mục, ta click vào biểu tượng Folders. Để làm việc với một trang web, ta click vào biểu tượng Page. Click vào biểu tượng Hyperlinks sẽ cho ta hình dung ra sự liên kết giữa tập tin hiện hành với các tập tin khác.

### 5.2. Tổ chức website

Để thuận tiện cho việc xuất bản (publish) website sau này, trước tiên nên tập trung tất cả các thành phần của website vào một thư

mục, ví dụ như C:\MyWebSite. Trong thư mục này, tổ chức một số thư mục con như **images** dùng để lưu các hình ảnh, **db** dùng để lưu các tập tin cơ sở dữ liệu, ...

Sau khi tạo xong thư mục có cấu trúc như trên, trong FrontPage ta tạo mới một website trả đến thư mục C:\MyWebSite bằng cách chọn trên thực đơn chức năng File/New/Page or Web/Empty Web, sau đó gõ vào đường dẫn C:\MyWebSite trong ô của mục Options/Specify the location of the new web



**Hình 3. 37 – Tạo mới một website**

Với việc tạo website này, mỗi khi ta soạn thảo mới bất kì tập tin nào, mặc định nó sẽ được lưu trong thư mục này. Ngoài ra, ta có thể chỉ định bảng mã UTF-8 là bảng mã được dùng mặc định khi tạo mới các tập tin bằng cách chọn trên thực đơn chức năng Tools/Web Settings/Language.



**Hình 3. 38 – Thiết lập bảng mã mặc định cho toàn bộ các trang của website là UTF-8**

### 5.3. Các thao tác chính khi soạn thảo một trang web

Để tạo mới một trang web, chọn chức năng File/New/Page or Web/Blank Page. Ở bên dưới trang web, ta thấy 3 tab là Normal, HTML và Preview cho ta nhìn trang web đang thiết kế ở 3 góc độ khác nhau. Tab Normal cho thấy trang web ở góc độ thiết kế, tab HTML cho thấy mã HTML của thiết kế ở tab Normal và tab Preview cho xem trước kết xuất thực sự của trang web trên trình duyệt.

Để thiết kế các trang web nhất quán và hiệu quả nên thực hiện tuần tự các bước bắt đầu thiết kế một trang web trong MS FrontPage

- Đặt các thuộc tính chung cho toàn bộ trang web như bảng mã dùng là UTF-8, tiêu đề trang, màu nền, ...

- Sử dụng frames, bảng biếu, ... để xếp đặt (layout) các thành phần của trang web
- Tạo và thêm vào các thành phần của trang web như văn bản (text), đồ họa (graphics), bảng biếu, hyperlink, form, banner, ...
- Định dạng văn bản bằng cách dùng các style hoặc style sheets
- FrontPage tự động tạo ra các tag khi ta thực hiện các thao tác trên trong tab Normal. Tuy nhiên để đạt hiệu quả cao trong việc thiết kế trang web, cần kết hợp với các kiến thức về HTML và chỉnh sửa trong tab HTML
- Thử nghiệm trang web bằng cách chọn tab Preview, hoặc nút Preview in Browser

Một số lưu ý khác:

- Không nên dùng font mặc định (default font) khi thiết kế trang web vì nó sẽ bị phụ thuộc và thiết lập cấu hình của trình duyệt ở phía người dùng
- Trang chủ của website nên đặt là default.htm hoặc default.asp.



**Hình 3. 39 – Các thuộc tính chung cho một trang web**

#### 5.4. Định dạng văn bản (text) và phân đoạn (paragraph)

Một phân đoạn là một khối văn bản được phân định với các phân đoạn khác bằng phím Enter khi soạn thảo. Nếu chỉ muốn xuống dòng mà không muốn chuyển sang phân đoạn khác thì dùng phím Shift+Enter.

Một phân đoạn sẽ được bao quanh bởi cặp tag <P> </P>, xuống dòng tương ứng với tag <BR>, khoảng trắng tương ứng với 6 ký tự: &nbsp;

Ta có thể định dạng theo nhóm các từ được đánh dấu chọn hoặc định dạng cho toàn bộ văn bản. Các định dạng cơ bản bao gồm: font chữ, kích thước, màu sắc, in đậm/nghiêng, gạch dưới, canh chỉnh trái/phải/giữa/đều hai bên.

#### Thực hành:

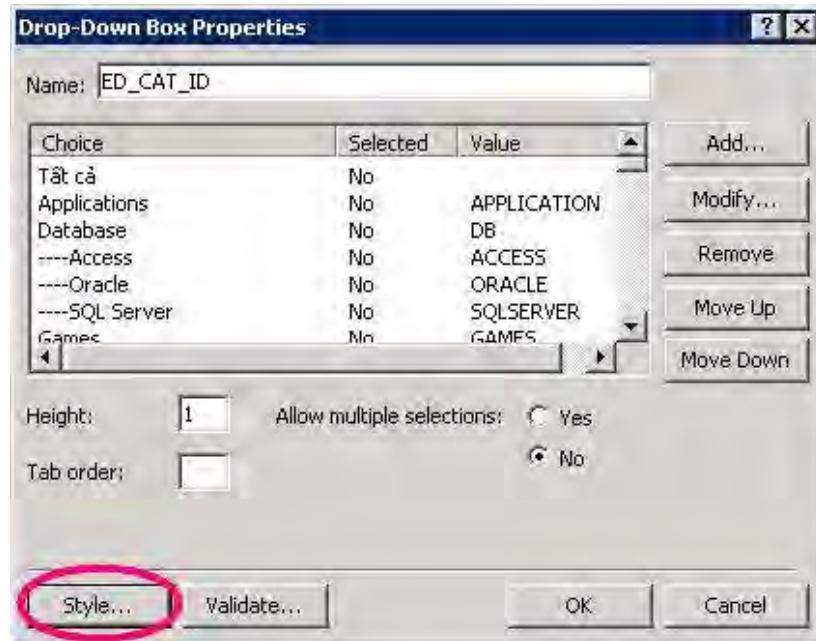
Gõ vào một đoạn văn bản, sau đó thực hiện định dạng theo các bước sau:

- Dùng chuột đánh dấu chọn một nhóm các từ, sau đó thực hiện định dạng font chữ, kích thước chữ, màu chữ, hiệu ứng chữ (như in đậm, in nghiêng, gạch dưới).
- Dùng chuột đánh dấu chọn toàn bộ phân đoạn, sau đó thực hiện định dạng font, border and shading, canh chỉnh (alignment).
- Chuyển sang tab HTML để xem các tag và các thuộc tính tương ứng được FrontPage phát sinh.
- Gõ vào các đoạn văn bản, sau đó sử dụng nút Bullets hoặc nút Numbering để đặt các đoạn văn bản này về định dạng Bullets hoặc Numbering. Thay đổi thuộc tính của Bullets và Numbering vừa tạo ra

#### 5.5. Dùng style và style sheet để định dạng

##### 5.5.1. Sử dụng inline style

Để chèn các inline style vào các đối tượng của trang web, ta click phải chuột lên đối tượng đó, chọn mục Properties, sau đó chọn Style. Ví dụ:

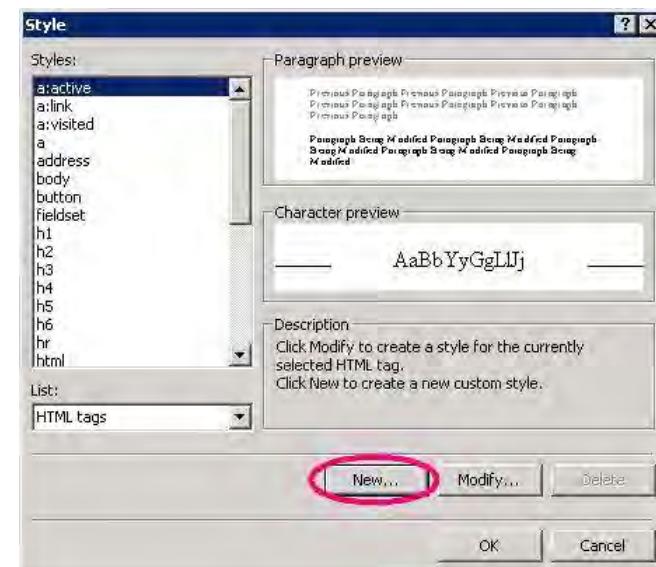


Hình 3.40 – Sử dụng Inline Style

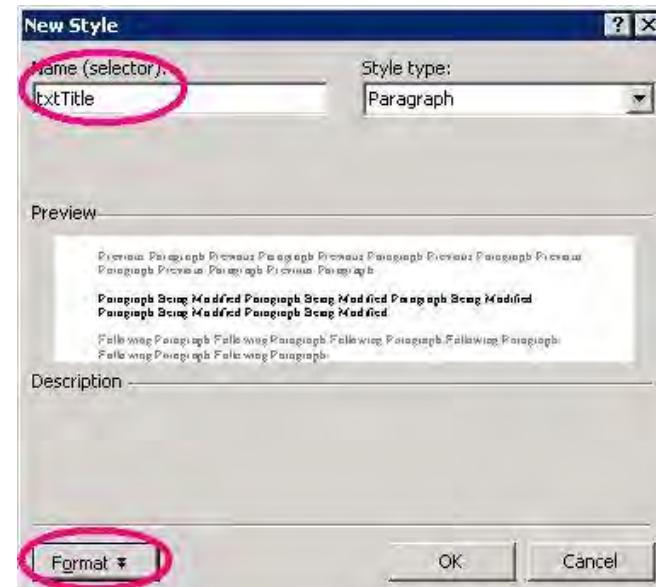
### 5.5.2. Sử dụng embeded style

Để sử dụng embeded style, ta thực hiện tuân tự các bước sau:

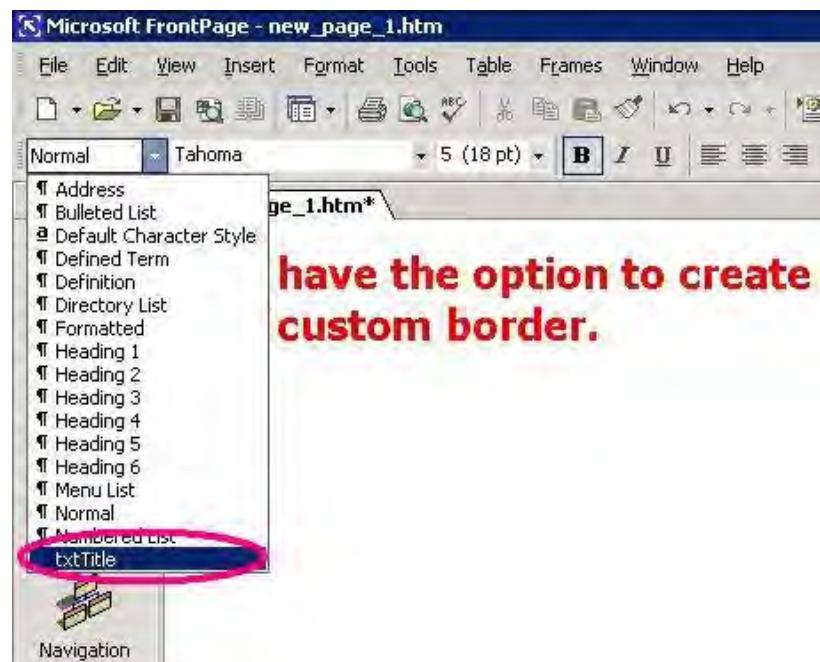
- Tạo ra các style cho trang web hiện hành bằng cách chọn trên thực đơn chức năng Format/Style.
- Chọn mục New để tạo mới một style, sau đó điền vào các thông tin cho style. Ví dụ: gõ vào ô Name (selector) là txtTitle, sau đó chọn mục Format để định dạng cho style có tên txtTitle với Font là Tahoma, kích thước 18pt, màu đỏ, in đậm, ...
- Để áp dụng style vừa tạo vào một đối tượng trong trang web, đánh dấu đối tượng rồi chọn mục Normal, tìm mục có tên txtTitle



Hình 3.41 – Tạo các style cho trang web



Hình 3.42 – Tạo mới một style



**Hình 3. 43 – Áp dụng style vào một đối tượng trong trang web**

### 5.5.3. Sử dụng Linking style

Để sử dụng linking style, ta thực hiện tuần tự các bước sau:

- Tạo mới tập tin dạng CSS bằng cách chọn trên thực đơn chức năng File/New/Page or Web/Normal Style Sheet. Giả sử lưu tập tin này lại với tên mystyle.css
- Tạo mới các style như trong bước embeded style trên rồi lưu lại.
- Trong tập tin mypage.htm, để dùng các style được định nghĩa trong tập tin mystyle.css, ta chọn trên thực đơn chức năng Format/Style Sheet Links, rồi chỉ định tập tin mystyle.css.



**Hình 3. 44 – Liên kết một tập tin .css vào trang web**

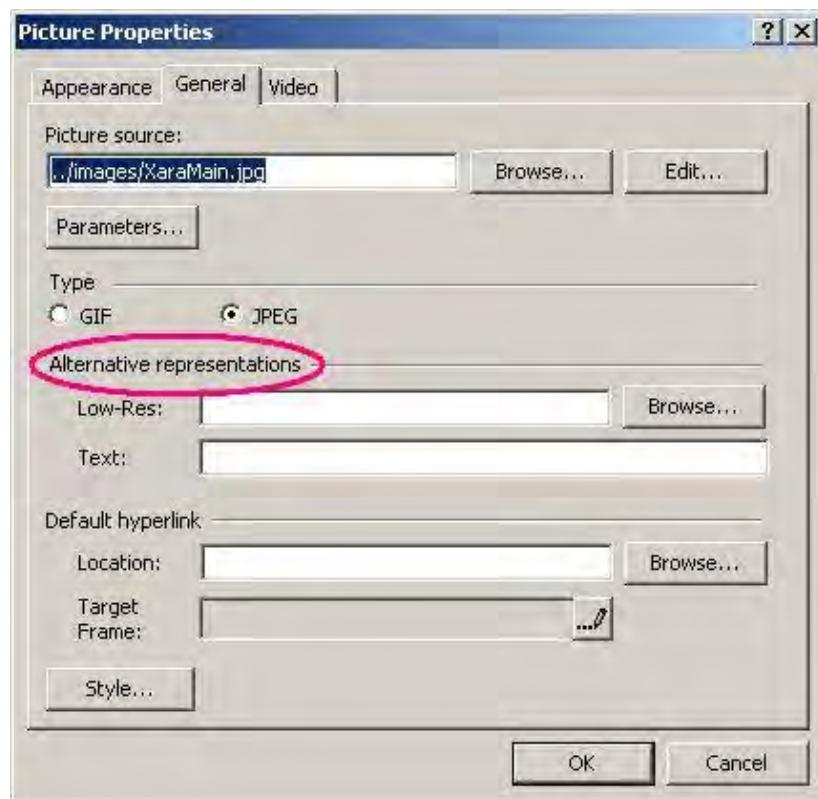
### 5.6. Chèn hình ảnh và ảnh động Flash vào trang web

Định dạng ảnh thông dụng cho thiết kế các trang web là GIF và JPEG. Ảnh GIF tuy bị giới hạn ở 256 màu nhưng cho phép tạo các hoạt hình (gif animation), ảnh JPEG cho phép ảnh dùng hơn 256 màu và kích thước khá nhỏ. Các phần mềm Button Studio, Banner Gif Animation, Paint Shop Pro, Adobe Photoshop là các phần mềm thông dụng để tạo và xử lý các ảnh dùng cho thiết kế web.

Ảnh động Flash là các tập tin dạng .swf, có ưu điểm là kích thước nhỏ trong khi có thể mô tả được nhiều hoạt hình, được dùng khá thông dụng trong thiết kế web hiện nay. Phần mềm Macromedia Flash MX, Swish là các phần mềm thông dụng để tạo các ảnh động này.

Để chèn một tập tin ảnh vào trang web, chọn chức năng Insert/Picture/From File ... FrontPage sẽ tạo ra tag dạng <IMG SRC="mypicture.gif" WIDTH="200" HEIGHT="100" ALT="My picture">.

Để hiển thị văn bản dạng tooltip khi đưa chuột đến hình ảnh, hãy đặt giá trị cho thuộc tính ALT của tag IMG hoặc chỉnh trong Picture Properties/General/Alternative presentations

**Hình 3. 45 – Dùng thuộc tính ALT**

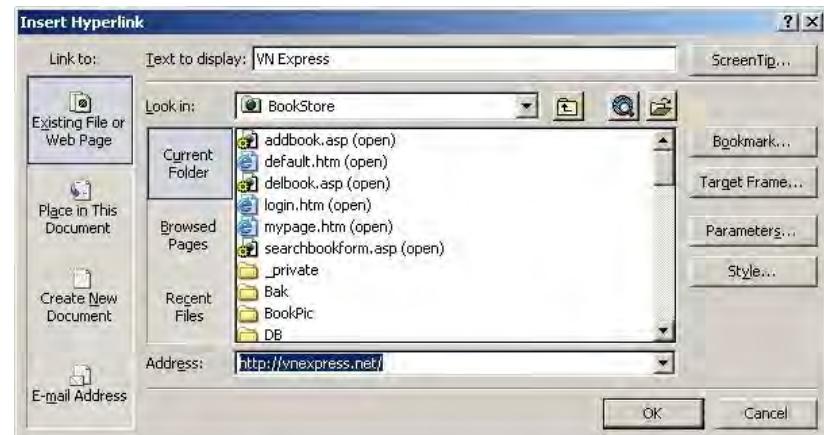
Để chèn một ảnh động Flash, ta phải chèn một đoạn mã mô tả nằm trong cặp tag <OBJECT>, </OBJECT> như đã hướng dẫn trong phần sử dụng phần mềm Swish.

Lưu ý rằng, cần chép ảnh vào thư mục chỉ định, ví dụ là images, trong thư mục của website trước khi tiến hành thao tác chèn ảnh.

## 5.7. Chèn hyperlink vào trang web

Hyperlink được dùng để kết nối các trang web từ trang này sang trang khác. Khi người dùng click vào hyperlink, trang web liên kết sẽ được hiển thị trên trình duyệt.

Để tạo một hyperlink, đánh dấu đoạn văn bản hoặc hình ảnh được dùng để mô tả sự liên kết, rồi chọn trên thực đơn chức năng Insert/Hyperlink (hoặc nhấn Ctrl+K). Hộp thoại mô tả hyperlink xuất hiện:

**Hình 3. 46 – Thiết lập hyperlink**

Ta có thể đặt hyperlink sao cho liên kết đến các thành phần trong cùng một trang, hoặc đến trang khác trong cùng một website, hoặc đến trang của website khác. Ngoài ra, ta cũng có thể dùng hyperlink để tạo ra các hộp thoại tải dữ liệu về, hoặc khởi động trình gửi email, ...

### Thực hành:

- Hyperlink liên kết đến một trang web khác trong cùng website. Ví dụ hãy tạo trong tập tin default.htm một button Giới thiệu, sao cho khi người dùng nhấp vào nó, trang web Intro.htm sẽ hiển thị ra để giới thiệu về nhóm hoặc công ty nào đó.
- Hyperlink liên kết đến một trang web của website khác. Ví dụ hãy tạo trong tập tin default.htm một button Liên kết, sao cho khi người dùng nhấp vào nó, trang web về môn học này <http://www.is-edu.hcmuns.edu.vn/~ldduy/WebCourse/> sẽ hiển thị.
- Trong trường hợp sử dụng frame, phải chỉ định tên frame mà trang web liên kết sẽ được hiển thị.

- Hyperlink cho phép gọi trình gửi email để tạo email. Ví dụ hãy tạo trong tập tin default.htm một button Góp ý, sao cho khi người dùng nhấp vào nó, chương trình gửi email sẽ xuất hiện để gửi email về địa chỉ ldduy@fit.hcmuns.edu.vn
- Hyperlink cho phép hiển thị cửa sổ Save để một tập tin về máy người dùng. Ví dụ hãy tạo trong tập tin default.htm một button Font tiếng Việt, sao cho khi người dùng nhấp vào nó, một hộp thoại lưu tập tin sẽ hiện lên để yêu cầu tải tập tin FontViet.zip về

## 5.8. Sử dụng bảng biểu

Ngoài chức năng thông thường là hiển thị dữ liệu dưới dạng bảng, bảng biểu còn thường được dùng cho việc sắp đặt (layout) các thành phần trong trang web.

Để chèn một bảng biểu vào trang web, chọn trên thực đơn chức năng Table/Insert. Các thông số cần lưu ý:

- Border: Để không hiển thị đường viền chọn Border size là 0.
- Width: Để đảm bảo table không bị resize khi độ phân giải màn hình của máy người dùng thay đổi, đặt kích thước của table dưới dạng pixel bằng cách chọn radio button In pixels
- Cell padding dùng để chỉnh khoảng cách từ đường viền đến văn bản trong một cell
- Cell spacing dùng để chỉnh khoảng cách giữa các cell
- Background dùng để chỉnh màu nền của table
- Canh chỉnh trong mỗi Cell Properties: Horizontal alignment dùng để chỉnh các canh chỉnh theo chiều ngang. Ví dụ như Left, Right, Center, Justify. Vertical alignment dùng để chỉnh các canh chỉnh theo chiều dọc. Ví dụ như Top, Middle, Bottom
- Chỉnh border của từng cell: vào Cell Properties/Style/Border

**Lưu ý:** FrontPage không hỗ trợ tốt khi thao tác trên các bảng phức tạp dạng như nhiều bảng lồng nhau, có nhiều cell merge và split. Do đó, cần kết hợp với sự hiểu biết của các tag HTML về TABLE để xử lý sao cho đạt được hiệu quả tối ưu.

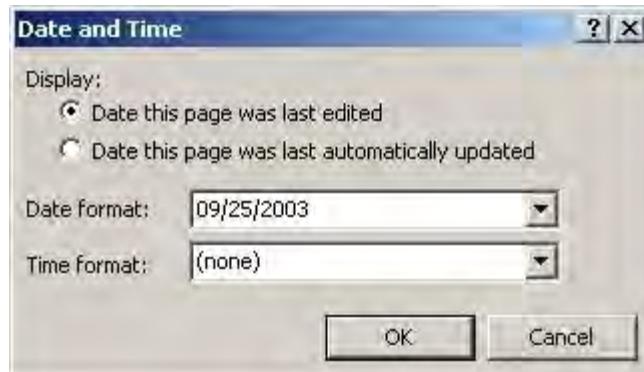


Hình 3.47 – Các thuộc tính của một bảng

## 5.9. Chèn các web components

### 5.9.1. Thời gian cập nhật (Time stamp)

Chọn Insert/Date and Time. Sau đó chọn định dạng ngày tháng và thời gian phù hợp với nhu cầu.



Hình 3. 48 – Thiết lập thông số cho Date, Time

### 5.9.2. Đường kẻ ngang (Horizontal line)

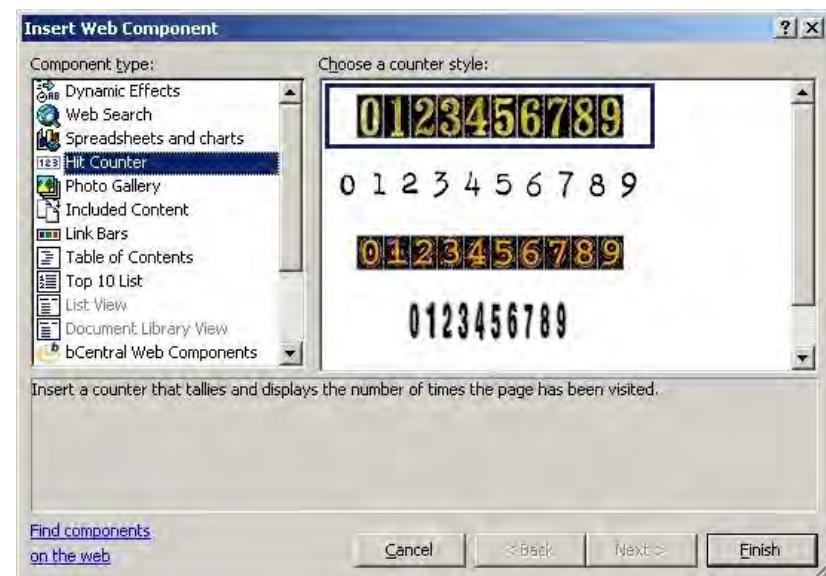
Chọn Insert/Horizontal line. Đặt các thuộc tính cho đường kẻ ngang bằng cách double click chuột lên nó. Sau đó chọn các thông số về Width, Height, Color, Alignment.



Hình 3. 49 – Các thông số cho Horizontal line

### 5.9.3. Các thành phần khác

Chọn trong thực đơn Insert/Web component.



Hình 3. 50 – Các thành phần khác được hỗ trợ sẵn

### 5.10. Một số hiệu ứng đặc biệt

- Chuyển trang (Page transition): Chọn Format/Page Transition...
- Hiệu ứng chữ chuyển động theo chiều ngang (Marquees): Chọn Insert/Component/Marquee. Sau khi hộp thoại hiện ra, bạn gõ vào dòng chữ cần chuyển động và đặt các thuộc tính khác như màu sắc, font chữ, ...
- Thêm hiệu ứng font chữ cho các hyperlink: Chọn Format/Background, check vào Enable hyperlink rollover effects. Sau đó bạn có thể chọn các màu theo ý muốn.

### 5.11. Tạo form nhập liệu

Dùng table để layout trước các thành phần chính của form.

Tạo các văn bản và các control (textbox, dropdown listbox, button, radio, checkbox, text area, ...). Các control được tạo bằng cách gõ trực tiếp các tag vào (dùng tab HTML).

Để chỉnh các thuộc tính của control, có thể chuyển sang tab Normal, click vào từng control, chọn Form Field Properties để chỉnh các thuộc tính.

Trong mỗi control của form, cần lưu ý các thuộc tính như Type (dùng để chỉ loại control nào), Name (dùng để định danh control và tham chiếu đến dữ liệu mà control lưu giữ sau này), Value (dùng để chỉ giá trị dữ liệu mà control hiển thị), Size (kích thước control), ...

Gắn tag <FORM> </FORM> bao trùm lên các control của form (kể cả table)

Trong tag FORM, cần lưu ý các thuộc tính như Name (dùng để định danh form và tham chiếu đến các control trong form), Action (dùng để trả về tên xử lý dữ liệu từ form), Method (dùng để chỉ phương thức gửi dữ liệu cho server).

Mỗi form thông thường có 2 button control, một button kiểu submit và một button kiểu reset. Button reset dùng để reset dữ liệu đã nhập, button submit dùng để submit dữ liệu của form sau khi người dùng nhập xong.

Sau khi tạo xong các thành phần của form, thiết lập phần trình bày cho form.

### Thực hành:

Thực hiện tuân tự các bước trên để tạo một form như hình sau:

Hình 3. 51 – Một mẫu form

## 6. THIẾT KẾ CÁC WEBSITE THEO MẪU

### 6.1. Yahoo Greetings (<http://greetings.yahoo.com>)



### 6.2. Flowers (<http://www.flowers.com>)



### 6.3. Yahoo Shopping (<http://shopping.yahoo.com>)

The screenshot shows the Yahoo! Shopping homepage. It features a search bar, a sidebar with categories like Apparel, Accessories, and Shoes, and a main area with promotional banners for Summer SALE, AVON, and Barnes & Noble. There are also featured stores like GAP, COMPAQ, and TARGET.

### 6.4. IS –EDU (<http://www.is-edu.hcmuns.edu.vn>)

The screenshot shows the IS-EDU website homepage. It features a navigation menu with links like Trang chủ, Eureka, Thảo luận, Nghiên cứu, Thư viện, Thu điện tử, CLB Tin Học, and Tim kiếm. A news section titled "Tổng hợp các tin mới" displays recent articles on topics such as Băng chứng điện tử, Phát hiện lỗ hổng trong cơ sở dữ liệu Oracle8i, Hacker tấn công nhiều lần thi trường bảo mật dữ liệu sinh học, Các công ty có thể tự đổi phó vớ hacker?, Các công ty có thể tự đổi phó vớ hacker?, Thiết bị không dây trước Hong Kong của hacker, Mạng IRC Efnet bị tấn công theo kiểu DDoS, Ping có thể là dấu hiệu của các hacker nguy hiểm, Nhóm hacker Poison Box với nan nhán mới, Hacking 101, 679 Website bị tấn công trong vòng một phút, Liên minh hacker chia phát hành Peekabooty, Phòng vấn hacker gửi Viagra cho Bill Gates, Hacking 101, Chương trình tấn công khai thác lỗ hổng IIS của Microsoft, Loài worm thứ hai nấp dưới dạng bản tin bảo mật của Microsoft, and Đề phòng tiêu diệt worm Code Red.

### 6.5. VN Express(<http://vnexpress.net>)

The screenshot shows the VN Express homepage. It features a sidebar with links like Home, Thế giới, Kinh doanh, Văn hóa, Thể thao, Sức khỏe, Khoa học, Vị trí, Pháp luật, Xe máy, Cuối, Thời tiết, and Chứng khoán. The main content includes a news article about the IT industry and a sidebar for Lao Dong Com.VN.

### 6.6. Code Project (<http://www.codeproject.com>)

The screenshot shows the Code Project homepage. It features a sidebar with links like Options, Sign in, New User, Add to Favorites, Tell a friend, Newsletters, Your Email, and .NET. The main content includes a banner for Xtreme Toolkit, a welcome message, a chart for Dundas Chart, and sections for Top Articles updated in the last week and Last 10 updates at the Code Project.

Để tham khảo cách thiết kế các trang web trên, hãy:

- Tải về máy bằng cách chọn chức năng Save của IE.
- Dùng MS Front Page để mở tập tin đã lưu lên
- Chuyển qua lại giữa các màn hình Normal View và HMTL View để biết cách thiết kế

Với mỗi trang web đã xem hãy chú ý các vấn đề sau:

- Cách bố trí các bảng (table)
- Cách sử dụng font chữ
- Cách chèn vào các hình ảnh để trang trí
- Cách định nghĩa các thuộc tính như màu chữ, màu nền, ...
- Phần tĩnh (không thay đổi) mỗi khi click vào các hyperlink chuyển qua một nội dung mới

Từ các trang web trên, hãy tự thiết kế các trang cho website của chính mình.

## 7. Những câu hỏi thường gặp

### 7.1. Sử dụng tiếng Việt Unicode trong các trang web

Để có thể tạo ra các trang web tiếng Việt sử dụng bảng mã Unicode, trước tiên phải đặt thuộc tính của trang sử dụng bảng mã Unicode. Trong FrontPage, chọn File/Page Properties/Language/HTML Encoding/Save the document as=UTF-8 hoặc xem meta tag trong phần đầu của trang web có thuộc tính charset là UTF-8 hay chưa. Ví dụ: <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">.

Để gõ được tiếng Việt, hãy sử dụng một trong các phần mềm hỗ trợ như UniKey, VietKey và chọn bảng mã sử dụng là Unicode.

Khi dùng bảng mã Unicode, chúng ta chỉ có thể dùng các font hỗ trợ Unicode như Arial, Tahoma, Times New Roman, Verdana, ...

### 7.2. Định dạng font trong các ô nhập liệu dropdown listbox, text area

Để thiết lập định dạng font mong muốn cho từng ô nhập liệu dạng này, click vào ô nhập liệu, chọn Form Field Properties/Style/Format/Font. Thiết lập bằng cách này sẽ tạo ra

thuộc tính style cho ô nhập liệu, ví dụ sau thiết lập font chữ cho một dropdown list box:

```
<SELECT SIZE="1" NAME="Occupation" STYLE="font-family: Arial; font-size: 10pt; color: #0000FF">
<OPTION SELECTED VALUE="0">Other</OPTION>
<OPTION VALUE="1">Engineer</OPTION>
<OPTION VALUE="2">Teacher</OPTION>
<OPTION VALUE="3">Doctor</OPTION>
<OPTION VALUE="4">Worker</OPTION>
</SELECT>
```

### 7.3. Tại sao khi chọn Encoding của một trang web là UTF-8 thì các ký tự Unicode tiếng Việt không hiển thị đúng trong tab HTML của FrontPage 2000

FrontPage 2000 của bộ Office 2000 không hỗ trợ tính năng này, hãy dùng FrontPage trong bộ Office XP. Với FrontPage của Office XP, ta có thể xem được tiếng Việt trong tab HTML của trang web có Encoding là UTF-8.

### 7.4. Chuỗi ký tự &nbsp; dùng để làm gì

Đây là chuỗi ký tự dùng để chỉ khoảng trắng. Lưu ý rằng trình duyệt sẽ tự động bỏ qua các khoảng trắng dư trong quá trình soạn thảo do đó nếu muốn giữa 2 đoạn văn bản A và B có 3 khoảng trắng, ta soạn: A&nbsp;&nbsp;&nbsp;B.

### 7.5. Trong trang web có dùng frame, làm thế nào để biết được khi click vào một hyperlink trong cửa sổ bên này, trang tương ứng sẽ được hiển thị ở cửa sổ nào?

Mỗi frame đều có tên định nghĩa trong tập tin chung có tag <FRAMESET>, do đó trong mỗi tập tin cơ sở của frame, nếu muốn chỉ định cửa sổ sẽ hiển thị nội dung tập tin khi người dùng click vào hyperlink ta đặt thuộc tính TARGET trong tag <BASE>. Ví dụ: <BASE TARGET="frContent">. Điều này có nghĩa là khi click vào bất kỳ hyperlink trong tập tin này, nội dung của tập tin do hyperlink chỉ tới sẽ được hiển thị trong frame có tên là "frContent"

## 7.6. Tiếng Việt trong các phần mềm Swish, Button Studio, Xara WebStyle

Do trong các phiên bản của các phần mềm này chưa hỗ trợ Unicode nên ta không thể dùng font Arial để hiển thị được các kí tự tiếng Việt được. Để hiển thị tiếng Việt, phải dùng các font VNI (dạng VNI-xxx) hoặc TCVN3 (dạng .vnxss).

## 7.7. Các câu hỏi thường gặp khi dùng phần mềm MS FrontPage

**Hỏi:** Tại sao khi hiển thị trang web, font chữ không hiển thị giống như trong lúc thiết kế của FrontPage

**Trả lời:** Có thể font hiển thị lúc thiết kế của FrontPage là Default font, do đó lúc hiển thị trên trình duyệt, trình duyệt sẽ lấy font mặc định được thiết lập (ví dụ VNI-Times) trái với font mà FrontPage đã dùng cho default font (ví dụ Arial). Để tránh trường hợp này, bạn nên đặt font cho các văn bản lúc thiết kế, tránh dùng default font.

**Hỏi:** Làm thế nào để vừa xem danh sách các thư mục khi đang ở Page View

**Trả lời:** Nhấn vào nút Toggle Pane.



Hình 3. 52 – Vừa xem cấu trúc website trong Page View

## 7.8. Làm thế nào để rút ngắn thời gian tải ảnh

Khi đưa các đối tượng hình ảnh, video, ... vào trang web, cần cân nhắc kĩ vì nó sẽ làm tăng kích thước của website, đồng thời sẽ làm tăng thời gian tải trang web về. Các thủ thuật sau sẽ giúp cho việc đưa hình ảnh vào trang web hiệu quả hơn, tiết kiệm được thời gian tải về:

- Hiệu chỉnh kích thước của hình ảnh: Với các ảnh quá lớn, cần chỉnh lại kích thước (chiều rộng, chiều cao) của ảnh để làm kích thước ảnh nhỏ lại. Lưu ý là nếu ta chỉnh thông số kích thước chiều rộng, chiều cao của ảnh bằng cách đặt giá trị cho thuộc tính WIDTH và HEIGHT trong tag <IMG> thì thời gian tải về không đổi, chỉ khác là kích thước của ảnh hiển thị trên màn hình trình duyệt mà thôi.
- Chọn độ phân giải thấp hơn, màu ít hơn khi quét các ảnh vào từ máy quét.
- Chuyển sang lưu trữ dưới dạng JPEG và giảm chất lượng của ảnh đến mức cho phép. Với cách này, dữ liệu được nén tốt hơn làm kích thước tập tin ảnh nhỏ hơn.
- Tạo thumbnail: Một thumbnail là một phiên bản của hình ảnh nhưng cho phép tải nhanh hơn. Nhìn vào phiên bản này, người dùng sẽ quyết định có cần thiết phải tải phiên bản hình ảnh với kích thước thật hay không. Cách này thường dùng khi hiển thị các trang web nhiều hình ảnh.
- Sử dụng thuộc tính ALT trong tag <IMG>: thuộc tính này sẽ hiển thị văn bản mô tả về hình ảnh, giúp người xem có thể hình dung ra ý nghĩa của hình ảnh trong khi nó đang được tải về.

## 7.9. Các câu hỏi thường gặp khi dùng phần mềm Swish

**Hỏi:** Tại sao movie hiện ra khi chọn chức năng Test in Browser ra đoạn văn bản không giống như đoạn văn bản đã gõ trong lúc thiết kế mà toàn là mã

**Trả lời:** Bạn chưa đăng ký sử dụng phần mềm. Hãy đăng ký và export lại.

**Hỏi:** Có hai đoạn văn bản cùng hiển thị trên movie, làm thế nào để hai đoạn văn bản này không bị hiển thị chồng lên nhau

**Trả lời:** Chọn Effect/Hide hoặc Effect>Show để điều khiển đoạn văn bản tương ứng là hiển thị hoặc dấu đi

**Hỏi:** Tại sao không chạy được movie tạo bằng Swish trên Internet Explorer?

**Trả lời:** Có thể máy chưa cài đặt Flash Player. Có thể cài đặt Flash Player từ phần mềm Macromedia Flash MX từ đĩa CD WebCourse.

**Hỏi:** Tại sao trang web có sử dụng movie tạo từ Swish hiển thị rất lâu?

**Trả lời:** Trong trường hợp rất lâu thì có lẽ tập tin .swf chưa tồn tại hoặc đường dẫn đến tập tin .swf khai báo trong đoạn mã HTML (giữa cặp tag <OBJECT>) chưa chính xác. Hãy kiểm tra lại tập tin .swf tương ứng đã tồn tại hay chưa và đường dẫn đến tập tin này đã được khai báo đúng hay chưa.

## Bài tham khảo

# BÀY BƯỚC THIẾT KẾ TRANG WEB ẤN TƯỢNG VÀ HIỆU QUẢ

Mặc dù việc thiết kế trang web đòi hỏi phải có nhiều khả năng về mỹ thuật, tuy nhiên vẫn có thể đưa ra một quá trình thiết kế để có thể giúp bạn tăng khả năng tạo ra các trang web hiệu quả và ấn tượng. Có thể nhiều người sẽ cho rằng việc đưa ra các bước trong quá trình thiết kế sẽ làm mất đi quá trình sáng tạo, điều này có lẽ đúng đối với những nhà thiết kế giỏi, những người có quá trình thiết kế và sáng tạo của riêng họ. Nhưng với những người không được đào tạo bài bản và thực hành nhiều thì việc nghiên cứu xem người khác thiết kế và ứng dụng các bước đó như thế nào có thể sẽ giúp ích rất nhiều cho họ.

### 1. Phân tích người dùng và tự đánh giá chính mình

Nếu bạn không biết người dùng dự định là ai, thì tất cả việc thiết kế, cho dù có được thực hiện kĩ lưỡng đến đâu cũng chỉ dẫn đến thất bại. Bạn cần phải biết các thông tin về người dùng như trình độ, sở thích, các lĩnh vực quan tâm, cấu hình trang thiết bị, phần mềm, ... để tránh đưa ra một trang web vô tích sự.

Bạn cũng cần phải phân tích các mối quan tâm và khả năng của chính bạn. Bạn có khả năng thiết kế các trang web có hiệu quả và ấn tượng không? Bạn có đủ trình độ chuyên môn để tạo ra được các trang có lượng thông tin phong phú dựa trên các tài nguyên sẵn có không?

Sau đây là một số kĩ thuật giúp cho bước này :

- Mô tả mục tiêu: Hãy xác định chính xác mục tiêu của trang này một cách ngắn gọn. Mục tiêu cần được mô tả một cách súc tích, rõ ràng, không quá rườm rà, chi tiết.
- Xác định vấn đề giải quyết: Từ mô tả mục tiêu ở bước trên, nêu ra các vấn đề cần giải quyết để đạt được mục tiêu, tóm tắt phương pháp giải quyết, ...

- Xác định người dùng: Liệt kê các đặc điểm của khách hàng như tuổi tác, nghề nghiệp, giới tính, thu nhập, trình độ, vùng cư trú, cấu hình trang thiết bị, phần mềm, ... vào một danh sách để phân tích và xử lí sau này.
- Liệt kê các nguồn tài nguyên: Bạn có sẵn những gì để hoàn thành công việc cả về mặt trang thiết bị, công cụ phần mềm, ... và cả về trình độ chuyên môn ? Bạn có thể làm được những gì, và bạn sẽ nhờ giúp đỡ những gì ?
- Xây dựng bảng tiến độ thực hiện: Xác định thời gian cần để hoàn thành sản phẩm với các tài nguyên sẵn có, thời gian cần để thực hiện từng bước của quá trình, ...

## 2. Thiết kế các chức năng và cấu trúc trang

Có thể lúc này bạn rất muốn ngồi ngay vào máy và bắt tay vào việc xây dựng trang web nhưng đừng vội! Hãy dành thời gian cho việc thiết kế các chức năng và cấu trúc của các trang chính, vì đây là bước quan trọng nhất trong quá trình thiết kế. Sau đây là một số hướng dẫn để thực hiện bước này :

- Chọn cách làm việc sao cho có thể phác thảo thiết kế một cách thoải mái: Bạn có thể dùng bút để vẽ sơ đồ trên giấy, hay có thể dùng các chương trình máy tính để phác thảo. Tuy nhiên việc sử dụng các chương trình máy tính có thể sẽ làm hạn chế năng suất làm việc vì các công cụ có sẵn thường bị giới hạn.
- Việc thiết kế nên đi từ trừu tượng đến cụ thể: Việc đưa ra các chi tiết ngay từ đầu có thể sẽ làm mất đi tổng quan của vấn đề. Phải xác định khung của chức năng trước rồi sau đó mới lựa chọn nội dung để điền vào.

## 3. Tìm cách trình bày ấn tượng và hiệu quả

Ngay cả một cấu trúc tốt nhất cũng sẽ thất bại nếu việc trình bày thông tin không trực quan và ấn tượng. Một cách trình bày có ấn tượng và hiệu quả được đánh giá không chỉ bằng cách trông nó như thế nào mà còn xem nó đóng góp như thế nào vào quá trình đạt mục tiêu ban đầu.

Sau đây là một số cách để tìm ra nguồn cung cấp cho các trình bày tốt:

- Đưa ra càng nhiều cách trình bày từ khả năng của chính bạn. Luôn quan sát và sưu tập các trình bày tốt đã đoạt giải, đã được nhiều người công nhận, hay các trình bày mà bạn thích, ...
- Luôn cập nhật các thay đổi về công nghệ web. Bạn nên luôn có các tài liệu mới nhất về HTML, cũng như các thông tin về các dạng tập tin và các thiết bị mới được hỗ trợ bởi các nhà sản xuất.
- Luôn ghi nhớ: Đối tượng đánh giá cách trình bày là người dùng chứ không phải bạn.
- Thủ càng nhiều giải pháp càng tốt và hãy ghi nhận các nhận xét, phản hồi của những người cộng tác để hoàn chỉnh thiết kế.

## 4. Xây dựng nội dung

Là một người thiết kế trang web, bạn có thể có hoặc không chịu trách nhiệm tạo nội dung (như văn bản, hình ảnh, âm thanh, video, ...). Vì việc tạo nội dung thường không thể đợi đến lúc thiết kế hình thành, bạn có thể tiến hành các bước sau để đảm bảo rằng nội dung và thiết kế của bạn là tương thích với nhau:

- Sửa đổi, hiệu chỉnh các nội dung đã có hoặc các nội dung mà bạn có quyền sửa.
- Xin hỗ trợ và cố vấn của những chuyên gia đối với các chủ đề ngoài lĩnh vực chuyên môn của bạn.
- Thiết lập đường dây liên lạc giữa bạn (người thiết kế) và những người tạo nội dung. Đưa ra các qui ước, các đặc tả cho nội dung như môi trường hỗ trợ, định dạng tập tin, cách né, qui ước đặt tên tập tin, ...
- Đảm bảo càng nhiều thông tin càng tốt. Cần phải có sự kết hợp chặt chẽ giữa văn bản và đồ họa, hình ảnh, âm thanh trong nội dung.
- Tạo một cấu trúc cây thư mục hợp lý cho nội dung và thường xuyên sao lưu để đảm bảo an toàn.

## 5. Thiết kế và kiểm tra khung trang web

Trong khi đang tiến hành xây dựng nội dung, đây là lúc kiểm tra các chức năng và cấu trúc được xây dựng trong bước 2 xem nó hoạt động như thế nào. Đây là bước mà bạn chuyển các mô tả về chức năng, về thiết kế ban đầu sang một thể hiện là các trang web cụ thể. Sau đây là một số hướng dẫn để thực hiện bước này :

- Liên lạc với người quản trị server để xem việc tổ chức các tập tin như thế nào và các đặc tả nào có sẵn. Cho người quản trị biết các loại tập tin nào mà bạn đang sử dụng chưa được hỗ trợ.
- Sử dụng các liên kết trong các trang tới các cấu trúc thư mục tương tự như cấu trúc thư mục trên server.
- Ghi nhận các ảnh thường được dùng trong việc truy xuất các trang thông thường để đưa vào cache. Bằng cách này bạn có thể tăng tốc độ truy xuất các trang.
- Thủ nghiệm trên server để kiểm tra xem nó hoạt động đúng như thiết kế hay không.

## 6. Đưa nội dung vào

Trong trường hợp tốt nhất, các khung dành cho văn bản và đồ họa sẽ được điền vào bằng nội dung thực sự của nó một cách dễ dàng và ăn khớp. Tuy nhiên điều này hiếm khi xảy ra vì một lẽ, hình ảnh và văn bản đưa vào có thể không vừa với khung thiết kế dành cho nó như dự định ban đầu. Cần phải thêm một số thao tác nữa mới có thể thực hiện xong chuyện này.

Để việc đưa nội dung vào thật sự đơn giản, ăn khớp, cần phải giữ mối liên lạc tốt giữa các thành viên liên quan như người thiết kế, người minh họa, người viết nội dung, người biên tập, và người quản trị server, ... Sau đây là một số hướng dẫn cho việc thực hiện tốt bước này :

- Trước tiên hãy cho các trang hoạt động cục bộ, riêng lẻ để dễ kiểm tra, hiệu chỉnh, đánh giá, ...
- Làm việc theo module, nghĩa là cho nhóm các trang liên quan nhau hoạt động trôi chảy trước khi mở rộng ra.

- Đừng ngại thay đổi một quyết định thiết kế trước đó. Có thể bạn đã giả định sai, hoặc là công nghệ đã thay đổi vào lúc đưa nội dung vào, ...

## 7. Kiểm tra và đánh giá

Các trang hiệu quả nhất là kết quả của việc thiết kế và đánh giá cẩn thận. Một web site trị giá nửa triệu đô la có thể có đến 70% tổng chi phí dành cho việc thiết kế và đánh giá. Sau đây là một số bước để thực hiện việc này:

- Kiểm tra hoạt động của các liên kết nội bộ và các nguồn tài nguyên.
- Kiểm tra độ chính xác của các liên kết ngoại. Không có gì tệ hơn là các liên kết với các trang bên ngoài không còn tồn tại nữa, hoặc là được chuyển đến nơi khác, hoặc là không còn phù hợp nữa.
- Thủ các trang với nhiều trình duyệt khác nhau. Thực hiện điều này để kiểm tra tính tương thích của trang với các trình duyệt, xem thử thiết kế trang đã tận dụng hết các hỗ trợ của trình duyệt chưa, ...
- Thủ các trang bằng nhiều cách kết nối khác nhau. Thủ xem việc hiển thị các trang có ảnh hưởng như thế nào nếu kết nối bằng mạng cục bộ, đường kết nối tốc độ cao, đường điện thoại, ...
- Thủ các trang ở tình trạng mức độ truy cập cao. Nếu server của bạn chạy tốt trong các giờ cao điểm thì những giờ khác có thể chấp nhận được.
- Thủ các trang với nhiều dạng người dùng khác nhau. Nếu trang của bạn đề cập về các mối quan tâm chung thì hãy tranh thủ thử trang web với những đồng sự, bạn bè, ... Hãy ghi chú và quan sát. Có thể bạn sẽ không cần thay đổi phiên bản của trang web nhưng bạn sẽ cần các thông tin vì trang liên tục được cập nhật hóa.

## Chương 4

# CÁC NGÔN NGỮ SCRIPT HỖ TRỢ LẬP TRÌNH WEB

## 1. GIỚI THIỆU VỀ VBSCRIPT VÀ JAVASCRIPT

VBScript và JavaScript là các ngôn ngữ lập trình kiểu thông dịch dùng cho Web. Các đoạn chương trình viết bằng các ngôn ngữ này nhúng trong các trang HTML sẽ được các trình duyệt thông dịch để thực hiện. Các ngôn ngữ script cho phép phát triển nhanh và dễ dàng các chương trình đơn giản hơn là các ngôn ngữ lập trình dạng biên dịch như C, C++.

VBScript do Microsoft phát triển trong khi JavaScript do Netscape phát triển. Cú pháp của VBScript gần giống với VisualBasic trong khi cú pháp của JavaScript lại gần giống với C++.

VBScript và JavaScript là hai ngôn ngữ script dùng để viết các đoạn chương trình ở phía client rất thông dụng hiện nay. JavaScript được hỗ trợ tốt trên hầu hết các trình duyệt trong khi VBScript chỉ được hỗ trợ tốt nhất trên trình duyệt Internet Explorer của Microsoft mà thôi.

Các lệnh, hàm, biến ... trong VBScript không phân biệt chữ thường và chữ hoa, trong khi JavaScript thì ngược lại.

Các vấn đề chính sẽ được trình bày khi giới thiệu các ngôn ngữ lập trình VBScript và JavaScript:

- Các định nghĩa của các kiểu dữ liệu và cách khai báo các biến.** Hầu hết các ngôn ngữ lập trình đều định nghĩa các kiểu dữ liệu thông dụng như kiểu số, kiểu chuỗi, ... Điểm khác nhau giữa chúng chỉ là cách đặt tên kiểu dữ liệu mà thôi. Ví dụ: để lưu trữ số nguyên trong Pascal, ta dùng kiểu dữ liệu có tên là integer, còn trong C là kiểu dữ liệu có tên là int.
- Cú pháp của các lệnh, các cấu trúc điều khiển trong chương trình.** Ví dụ: lệnh gán trong Pascal, chúng ta dùng ký hiệu “:=”, trong khi ở C là “=”. Có 4 loại cấu trúc điều khiển chính: lệnh gán, lệnh điều kiện, lệnh lặp, lời gọi hàm/thủ tục.

- Cách sử dụng các hàm thư viện sẵn có và cách tự xây dựng các hàm/thủ tục.** Các hàm thư viện sẵn có thường gặp là: xử lý các kiểu dữ liệu (xử lý chuỗi, xử lý số, xử lý ngày, tháng, ...), tương tác với các thiết bị nhập xuất chuẩn (xuất dữ liệu ra màn hình, nhập dữ liệu từ bàn phím, ...), ...

Phần trình bày về sử dụng VBScript và JavaScript chỉ là phần tóm tắt (Ngầm định rằng người đọc đã biết đến cách lập trình ở một ngôn ngữ lập trình khác như Pascal, C, ...)

## 2. SỬ DỤNG VBSCRIPT

### 2.1. Các kiểu dữ liệu cơ bản

VBScript chỉ có một kiểu dữ liệu duy nhất là **Variant**. Variant là kiểu dữ liệu đặc biệt có thể chứa các loại dữ liệu khác nhau từ những kiểu dữ liệu đơn giản như kiểu số cho đến các kiểu dữ liệu phức tạp như kiểu bản ghi (record). Vì **Variant** là kiểu dữ liệu duy nhất của VBScript nên đây là cũng là kiểu dữ liệu được trả về từ các hàm/thủ tục viết bằng VBScript.

Nói một cách dễ hiểu hơn là: nếu trong Pascal bạn phải lưu trữ dữ liệu số trong kiểu dữ liệu **integer**, dữ liệu chuỗi trong kiểu dữ liệu **string**, thì trong VBScript bạn có thể vừa lưu trữ dữ liệu số, vừa lưu trữ dữ liệu chuỗi (hay bất kì dữ liệu kiểu nào khác) trong kiểu dữ liệu **Variant**. Việc xem một biến **Variant** là số hay chuỗi tùy vào ngữ cảnh sử dụng. Ví dụ: trong biểu thức **b=a+1234** thì **a**, **b** được xem như là kiểu số còn trong biểu thức **b=a+"1234"**, thì **a**, **b** được xem như là kiểu chuỗi.

**Variant Subtypes:** là các dạng thông tin khác nhau mà kiểu dữ liệu **Variant** có thể lưu trữ.

Subtypes	Mô tả
Empty	Dữ liệu chưa được khởi tạo.
Null	Null
Boolean	Dữ liệu luận lí True hoặc False
Byte	Số nguyên có giá trị từ 0 đến 255
Integer	Số nguyên có giá trị từ -32768 đến 32767
Currency	Từ -922,337,203,685,477.5808 đến 922,337,203,685,477.5807.
Long	Số nguyên có giá trị từ -2,147,483,648 đến 2,147,483,647.
Single	Số thực, có giá trị từ 3.402823E38 đến 1.401298E-

Subtypes	Mô tả
	45 cho các giá trị âm; từ 1.401298E-45 đến 3.402823E38 cho các giá trị dương.
Double	Số thực, có giá trị từ 1.79769313486232E308 đến 4.94065645841247E-324 cho các giá trị âm; từ 4.94065645841247E-324 đến 1.79769313486232E308 cho các giá trị dương.
Date (Time)	Chứa giá trị số biểu diễn ngày trong khoảng 01.01.100 đến 31.12.9999.
String	Chuỗi kí tự có độ dài thay đổi. Số kí tự tối đa có thể lưu trữ là 2 tỉ
Object	Chứa đối tượng
Error	Chứa mã lỗi

Để chuyển dữ liệu từ một subtype này sang subtype khác ta dùng các hàm chuyển như **CByte**, **CDate**, **CInt**, **CStr**, .... Trong khi đó, để biết một biến **Variant** đang lưu trữ dữ liệu kiểu subtype nào, ta dùng hàm **VarType**.

## 2.2. Khai báo biến

Người ta sử dụng từ khóa Dim để khai báo biến. Để khai báo nhiều biến, người ta dùng dấu “,” để phân cách giữa các biến. Ví dụ: **Dim Left, Top, Right, Bottom**

Trong VBScript, không nhất thiết phải khai báo các biến trước khi sử dụng. Để yêu cầu các biến cần phải được khai báo trước khi sử dụng, ta dùng lệnh “**Option Explicit**” đặt ngay trước lệnh đầu tiên của đoạn mã chương trình.

### Qui tắc đặt tên biến:

- Phải bắt đầu bằng 1 kí tự chữ cái.
- Không chứa dấu “.”
- Không vượt quá 255 kí tự.

Để khai báo mảng, ta dùng cặp dấu “(“ và “)” để định nghĩa các số chiều và kích thước mỗi chiều của mà. Ví dụ sau dùng để khai báo một mảng một chiều có 11 phần tử: **Dim A(10)**.

Phần tử bắt đầu của mảng một chiều là phần tử có chỉ số là 0. Để truy xuất tới phần tử thứ i của mảng một chiều A, ta dùng như sau: **A(i)**.

VBScript có thể hỗ trợ mảng lên đến 60 chiều. Tuy nhiên, thường dùng nhất trong mảng nhiều chiều là mảng 2 chiều hoặc 3

chiều. Ví dụ sau khai báo một mảng 2 chiều gồm 6 dòng, 11 cột: **Dim MyTable(5, 10)**.

Để truy xuất tới một phần tử của mảng 2 chiều, ví dụ như phần tử ở dòng i, cột j, ta dùng: **MyTable(i, j)**.

Đối với các mảng không khai báo kích thước trước, người ta gọi mảng động (dynamic). Nghĩa là kích thước của mảng có thể thay đổi được trong quá trình chạy chương trình. Người ta dùng từ khóa **ReDim** để thay đổi kích thước mảng.

## 2.3. Lệnh điều kiện

Có hai dạng lệnh điều kiện là **If...Then...Else** và **SelectCase**

- Các cú pháp của lệnh **If...Then...Else**

**If <biểu thức điều kiện> Then <Lệnh>**

Ví dụ:

```
Sub FixDate()
    Dim myDate
    myDate = #2/13/95#
    If myDate < Now Then myDate = Now
End Sub
```

Chú ý: Lệnh được viết trên cùng một hàng với **If ... Then**

**If <biểu thức điều kiện> Then**

```
...
<Lệnh i>
...
```

**End If**

Ví dụ:

```
Sub AlertUser(value)
    If value = 0 Then
        AlertLabel.ForeColor = vbRed
        AlertLabel.Font.Bold = True
        AlertLabel.Font.Italic = True
    End If
End Sub
```

**If <biểu thức điều kiện> Then**

```
...
<Lệnh i>
```

```

    ...
Else
    ...
    <Lệnh i>
    ...
End If

```

Ví dụ:

```

Sub AlertUser(value)
If value = 0 Then
    AlertLabel.ForeColor = vbRed
    AlertLabel.Font.Bold = True
    AlertLabel.Font.Italic = True
Else
    AlertLabel.Forecolor = vbBlack
    AlertLabel.Font.Bold = False
    AlertLabel.Font.Italic = False
End If
End Sub

```

```

If <biểu thức điều kiện 1> Then
    ...
ElseIf <biểu thức điều kiện 2> Then
    ...
ElseIf <biểu thức điều kiện i> Then
    ...
Else
    ...
End If

```

Ví dụ:

```

Sub ReportValue(value)
If value = 0 Then
    MsgBox value
ElseIf value = 1 Then
    MsgBox value
ElseIf value = 2 then
    Msgbox value
Else
    Msgbox "Value out of range!"
End If

```

- Cú pháp của lệnh **SelectCase**

```

Select Case <tên biến>
    Case <giá trị1>
        ...
        <Lệnh i>
    Case <giá trị i>
        ...
        <Lệnh i>
    Case Else
        ...
        <Lệnh i>
    ...
End Select

```

Ví dụ:

```

Select Case
Document.Form1.CardType.Options(SelectedIndex).Text
    Case "MasterCard"
        DisplayMCLogo
        ValidateMCAccount
    Case "Visa"
        DisplayVisaLogo
        ValidateVisaAccount
    Case "American Express"
        DisplayAMEXCOLogo
        ValidateAMEXCOAccount
    Case Else
        DisplayUnknownImage
        PromptAgain
End Select

```

## 2.4. Lệnh lặp

Có bốn dạng lệnh lặp là **Do...Loop**, **While...Wend**, **For...Next**, **For Each...Next**.

- Các cú pháp của lệnh **Do...Loop**

**Do While** <biểu thức điều kiện>

```
...
<Lệnh i>
...
```

**Loop**

**Do**

```
...
<Lệnh i>
...
```

**Loop While** <biểu thức điều kiện>

Ví dụ:

```
Sub ChkFirstWhile()
    Dim counter, myNum
    counter = 0
    myNum = 20
    Do While myNum > 10
        myNum = myNum - 1
        counter = counter + 1
    Loop
    MsgBox "The loop made " & counter & " repetitions."
End Sub
```

```
Sub ChkLastWhile()
    Dim counter, myNum
    counter = 0
    myNum = 9
    Do
        myNum = myNum - 1
        counter = counter + 1
    Loop While myNum > 10
    MsgBox "The loop made " & counter & " repetitions."
End Sub
```

**Do Until** <biểu thức điều kiện>

```
...
<Lệnh i>
...
```

**Loop**

**Do**

```
...
<Lệnh i>
...
```

**Loop Until** <biểu thức điều kiện>

Ví dụ:

```
Sub ChkFirstUntil()
    Dim counter, myNum
    counter = 0
    myNum = 20
    Do Until myNum = 10
        myNum = myNum - 1
        counter = counter + 1
    Loop
    MsgBox "The loop made " & counter & " repetitions."
End Sub
```

**Sub ChkLastUntil()**

```
Dim counter, myNum
counter = 0
myNum = 1
Do
    myNum = myNum + 1
    counter = counter + 1
Loop Until myNum = 10
MsgBox "The loop made " & counter & " repetitions."
End Sub
```

- Cú pháp của lệnh **While...Wend**

**While** <biểu thức điều kiện>

```
...
<Lệnh i>
...
```

**Wend**

Nên dùng **Do...Loop** thay cho **While...Wend**

- Các cú pháp của lệnh **For...Next**

**For** <biến đếm> = <chỉ số đầu> **To** <chỉ số cuối>

```
...
<Lệnh i>
...
```

**Next**

Ví dụ:

```
Sub DoMyProc50Times()
    Dim x
    For x = 1 To 50
        MyProc
    Next
End Sub
```

**For** <biến đếm> = <chỉ số đầu> **To** <chỉ số cuối> **Step** <bước nhảy>

```
...
<Lệnh i>
...
```

**Next**

Ví dụ:

```
Sub TwosTotal()
    Dim j, total
    For j = 2 To 10 Step 2
        total = total + j
    Next
    MsgBox "The total is " & total
End Sub
```

```
Sub NewTotal()
    Dim myNum, total
    For myNum = 16 To 2 Step -2
        total = total + myNum
    Next
    MsgBox "The total is " & total
End Sub
```

- Các cú pháp của lệnh **For Each...Next**

Tương tự **For ...Next** nhưng thay vì lặp theo một số lần đã định trước, lệnh **For Each...Next** được dùng để lặp tương ứng với mỗi thành phần của các biến dạng collection hoặc mỗi thành phần của mảng.

Ví dụ:

```
<HTML>
<HEAD><TITLE>Forms and Elements</TITLE></HEAD>
<SCRIPT LANGUAGE="VBScript">
```

```
<!--
Sub cmdChange_OnClick
```

```
Dim d 'Create a variable
```

```
Set d = CreateObject("Scripting.Dictionary")
d.Add "0", "Athens" 'Add some keys and items
d.Add "1", "Belgrade"
d.Add "2", "Cairo"
```

```
For Each I in d
    Document.frmForm.Elements(I).Value = D.Item(I)
```

```
Next
```

```
End Sub
```

```
-->
```

```
</SCRIPT>
```

```
<BODY>
```

```
<CENTER>
```

```
<FORM NAME="frmForm"
```

```
<Input Type = "Text"><p>
```

```
<Input Type = "Button" NAME="cmdChange" VALUE="Click Here"><p>
```

```
</FORM>
```

```
</CENTER>
```

```
</BODY>
```

```
</HTML>
```

## 2.5. Hàm và thủ tục

- Khai báo thủ tục

```
Sub <tên thủ tục>(<danh sách các tham số>)
```

```
...
```

```
End Sub
```

Ví dụ:

```
Sub ConvertTemp()
    temp = InputBox("Please enter the temperature in deg F.", 1)
    MsgBox "The temperature is " & Celsius(temp) & " deg C."
End Sub
```

- Khai báo hàm

```
Function <tên hàm>(<danh sách các tham số>)
    ...
End Function
```

Ví dụ:

```
Function Celsius(fDegrees)
    Celsius = (fDegrees - 32) * 5 / 9
End Function
```

- Minh họa cách gọi hàm, thủ tục bên trong chương trình

```
Temp = Celsius(fDegrees)

MsgBox "The Celsius temp. is " & Celsius(fDegrees) & " deg."

Call MyProc(firstarg, secondarg)

MyProc firstarg, secondarg
```

### 3. SỬ DỤNG JAVASCRIPT

#### 3.1. Các kiểu dữ liệu cơ bản

JavaScript định nghĩa các kiểu dữ liệu cơ bản bao gồm: kiểu số, kiểu chuỗi, kiểu luận lí, kiểu object, kiểu mảng, null, undefined.

Trình thông dịch JavaScript chỉ có thể tính toán giá trị của một biểu thức một khi kiểu dữ liệu của các toán hạng là như nhau. Do đó, đôi lúc bạn cần phải “ép kiểu” cho các toán hạng trong các biểu thức để tránh các hiệu ứng lè.

Chú ý rằng trong Javascript, có phân biệt chữ thường và chữ hoa.

#### 3.2. Khai báo biến

Sử dụng từ khóa **var** để khai báo biến. JavaScript đòi hỏi các biến phải được khai báo trước khi sử dụng. Ví dụ:

```
var count; // a single declaration.
var count, amount, level; // multiple declarations with a single var keyword.
var count=0, amount=100; //variable declaration and initialization in 1 statement.
```

Nếu một biến không được khởi tạo trong lúc khai báo, nó sẽ lấy giá trị là **undefined**.

#### Qui tắc đặt tên biến:

- Phải bắt đầu bằng 1 kí tự chữ hoặc dấu gạch dưới “\_”.
- Theo sau chỉ là hoặc kí tự chữ, hoặc là kí tự số, hoặc là dấu gạch dưới.
- Không được trùng với các từ dành riêng ví dụ như từ khóa, ...

#### Tạo dữ liệu kiểu mảng

- Ví dụ sau dùng để khai báo một mảng một chiều có 10 phần tử: **var A = new Array(10)**. Ta dùng toán tử **new** và **Array()** constructor.
- Phần tử bắt đầu của mảng một chiều là phần tử có chỉ số là 0. Để truy xuất tới phần tử thứ i của mảng một chiều A, ta dùng như sau: **A[i]**.

#### 3.3. Lệnh điều kiện

Có hai dạng lệnh điều kiện là **if/else** và **switch**

- Các cú pháp của lệnh **if/else**

```
if (<biểu thức điều kiện>)
{
    ...
    <Lệnh i>;
    ...
}
else
{
    ...
    <Lệnh i>;
    ...
}
```

## Ví dụ

```
// The smash() function is defined elsewhere in the code.
// Boolean test of whether newShip is true.
if (newShip)
    smash(champagneBottle,bow);

// In this example, the test fails unless both conditions are true.
if (rind.color == "deep yellow " && rind.texture == "large and
small wrinkles")
{
    theResponse = ("Is it a Crenshaw melon?");
}

// In this example, the test succeeds if either condition is true.
var theReaction = "";
if ((dayOfWeek == "Saturday") || (dayOfWeek == "Sunday"))
{
    theReaction = ("I'm off to the beach!");
}
else
{
    theReaction = ("Hi ho, hi ho, it's off to work I go!");
}
```

- Cú pháp của lệnh **switch**

```
switch (<tên biến>) {
    case <giá trị1>:
        ...
        break;
    case <giá trị i>:
        ...
        break;
    default:
        ...
        break;
}
```

## Ví dụ

```
function MyObject() {
    ...
}

switch (object.constructor){
    case Date:
        ...
    case Number:
```

```
...
case String:
...
case MyObject:
...
default:
...
}
```

## 3.4. Lệnh lặp

Có bốn dạng lệnh lặp là **while**, **do/while**, **for**, **for/in**.

- Các cú pháp của lệnh **while**

```
while (<biểu thức điều kiện>) {
    ...
    <Lệnh i>;
    ...
}
```

## Ví dụ

```
var x = 0;
while ((x != 42) && (x != null))
{
    x = window.prompt("What is my favourite number?", x);

    if (x == null)
        window.alert("You gave up!");
    else
        window.alert("Yep - it's the Ultimate Answer!");
```

- Các cú pháp của lệnh **do/while**

```
do {
    ...
    <Lệnh i>;
    ...
} while (<biểu thức điều kiện>)
```

## Ví dụ

```
var x = 0;
do
{
    x = window.prompt("What is my favourite number?", x);
} while ((x != 42) && (x != null));
```

```

if (x == null)
    window.alert("You gave up!");
else
    window.alert("Yep - it's the Ultimate Answer!");

```

- Các cú pháp của lệnh **for**

```

for ( <biểu thức khởi tạo>; <biểu thức điều kiện>; <biểu thức
thay đổi>) {
    ...
}

```

Ví dụ

```

var howFar = 10; // Sets a limit of 10 on the loop.
// Creates an array called sum with 10 members, 0 through 9.
var sum = new Array(howFar);
var theSum = 0;
sum[0] = 0;
// Counts from 0 through 9 in this case.
for(var icount = 0; icount < howFar; icount++) { theSum += icount;
sum[icount] = theSum;
}
// This isn't executed at all, since icount is not greater than
howFar
var newSum = 0;
for(var icount = 0; icount > howFar; icount++) { newSum += icount;
}
var sum = 0;
// This is an infinite loop.
for(var icount = 0; icount >= 0; icount++) { sum += icount;
}

```

### 3.5. Hàm

Khai báo hàm

```

function <tên hàm>(<danh sách các tham số>){
    ...
}

```

Ví dụ

```

function add(x, y)
{
    return(x + y);      //Perform addition and return results.
}

```

## 4. SỬ DỤNG VBSCRIPT VÀ JAVASCRIPT TRONG CÁC TRANG WEB

Thuở ban đầu, các trang web thường là tĩnh. Nghĩa là nội dung của trang web tải về mọi máy là như nhau và bất biến. Một trong các cách để tăng tính động cho các trang web tĩnh đó là chèn vào các đoạn chương trình viết bằng các ngôn ngữ lập trình script như VBScript và JavaScript. Các đoạn chương trình như vậy thường được gọi đoạn mã thực hiện ở phía client (client-side script). Nghĩa là các đoạn chương trình này sẽ được tải về client và trình duyệt sẽ đóng vai trò như là trình thông dịch các đoạn mã này mỗi khi thực hiện chúng.

Các đoạn chương trình thực hiện ở client thường thực hiện các công việc không quá phức tạp như: thực hiện một số thao tác đơn giản trên các đối tượng của trang web, kiểm tra tính hợp lệ của form nhập liệu, thực hiện việc trình bày động của các đối tượng trên trang web (thay đổi màu sắc, kích thước, ...) ...

### 4.1. Cách chèn các đoạn chương trình VBScript/JavaScript vào trang HTML

Sử dụng tag <SCRIPT>, trong đó ghi rõ ngôn ngữ lập trình dùng để viết mã chương trình. Các tag <!--> dùng để nhắc các trình duyệt không hiển thị các đoạn mã bên trong nếu nó không hiểu tag <SCRIPT>. Ví dụ:

```

<SCRIPT LANGUAGE="JavaScript">
<!--
document.writeln("Hello World!")
-->
</SCRIPT>

```

Các đoạn mã script có thể đặt giữa cặp tag <HEAD> và </HEAD>, hay giữa cặp tag <BODY> và </BODY>. Tuy nhiên nên đặt tất cả các đoạn mã script trong tag <HEAD> và </HEAD> để dễ kiểm soát và chắc chắn rằng các đoạn mã này đã được đọc và thông dịch trước các thành phần trong <BODY>. Ví dụ:

```

<HTML>
<HEAD>
<TITLE>Place Your Order</TITLE>
<SCRIPT LANGUAGE="VBScript">
<!--

```

```

Function CanDeliver(Dt)
    CanDeliver = (CDate(Dt) - Now()) > 2
End Function
-->
</SCRIPT>
</HEAD>
...

```

## 4.2. Tương tác với các đối tượng trên trang web

Mọi thành phần trong trang web đều được xem là đối tượng lập trình: toàn bộ trang web được xem là một đối tượng, các form, ô nhập liệu, nút nhấn, ... đều là đối tượng. Chính vì vậy mà ta có thể tham chiếu đến các thuộc tính cũng như gọi các phương thức của các đối tượng này theo cách thông thường của lập trình hướng đối tượng.

Đối tượng **document** là đối tượng quan trọng nhất đối với các chương trình thực hiện tại client vì nó đại diện cho chính trang web hiện hành. Nghĩa là mọi thao tác muốn thay đổi, tham chiếu, ... đến các đối tượng, các thành phần trên trang web hiện hành như form, văn bản, ... đều phải thông qua đối tượng này.

Ví dụ: Để kết xuất dữ liệu ra trang web hiện hành, ta dùng các hàm **write** và **writeln** của đối tượng **document**.

```

<HTML>
<HEAD>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
!---
    document.writeln("Welcome to my site!");
//-->
</SCRIPT>
<br>
<SCRIPT LANGUAGE="VBScript">
!---
    document.write "This site is under construction"
//-->
</SCRIPT>
</BODY>
</HTML>

```

Đối tượng **window** là đối tượng đại diện cho cửa sổ mà trong đó trang web hiển thị. Thông thường, trình duyệt sẽ tạo ra đối tượng **window** khi nó mở một trang HTML.

Để hiển thị các hộp thông báo, ta dùng các hàm **alert**, **confirm** và **prompt** của đối tượng **window**. Ví dụ:

```

<HTML>
<HEAD>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!--
    x = window.confirm("Are you sure you want to quit?");
    if (x)
        window.alert("Thank you.")
    else
        window.alert("Good choice.")
//-->
</SCRIPT>
</BODY>
</HTML>

```

Ví dụ sau dùng để đặt thông báo ngay trong thanh trạng thái, ta dùng: **window.status="Welcome to my website"**

## 4.3. Xử lý các sự kiện khi tương tác với các thành phần của trang web

Sự kiện **onclick** là sự kiện được phát sinh khi người dùng nhấp chuột vào một đối tượng trên trang web ví dụ như button, hyperlink, ... . Để gắn các hàm xử lý sự kiện này vào đối tượng mỗi khi nó được phát sinh, thông thường ta thực hiện theo cách sau:

- Tạo một hàm để xử lý sự kiện đó. Hàm này đặt trong cặp tag <SCRIPT>.
- Trong đối tượng cần gắn hàm xử lý sự kiện, ta thêm dòng có cú pháp: <đối tượng>=<hàm xử lý sự kiện> vào bên trong. Lưu ý thêm thuộc tính LANGUAGE để xác định ngôn ngữ của script chèn vào

Ví dụ sau minh họa việc gắn hàm xử lý sự kiện **onclick** trên hai đối tượng nút nhấn:

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
    Sub vbsEventHandler

```

```

        window.alert("VBScript Event Handler")
    End Sub
-->
</SCRIPT>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function jsEventHandler() {
        window.alert ("JavaScript Event Handler");
    }
-->
</SCRIPT>
</HEAD>
<BODY>
<INPUT TYPE=BUTTON NAME="vbsBtn" VALUE="VBS Test"
LANGUAGE="VBScript" onclick="vbsEventHandler()">&nbsp;
<INPUT TYPE=BUTTON NAME="jsBtn" VALUE=" JS Test "
LANGUAGE="JavaScript" onclick="jsEventHandler();"><br>
</BODY>
</HTML>

```

Đối với VBScript, nếu bạn đặt tên một hàm có dạng <tên đối tượng>\_<tên sự kiện>, thì hàm này sẽ được xem như là hàm xử lý sự kiện cho đối tượng đã nêu trên. Ví dụ:

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
    Sub vbsBtn_OnClick
        window.alert("VBScript Event Handler")
    End Sub
-->
</SCRIPT>
</HEAD>
<BODY>

<INPUT TYPE=BUTTON NAME="vbsBtn" VALUE="VBS Test">
</BODY>
</HTML>

```

## 5. Dấu mã VBScript/JavaScript

Về mặt lí thuyết thì các javascript là các client-side script nghĩa là nó sẽ được thi hành tại máy của người dùng chứ không phải là tại server. Điều đó có nghĩa là nó phải được tải về máy của người dùng lúc trang web được gọi. Do đó, ta không thể dấu được.

Tuy nhiên, vẫn có một số cách để hạn chế những người ít hiểu biết bằng cách:

- Các javascript không code sẵn trong các tập tin .htm gọi nó mà được tổ chức thành các tập tin .js và include vào tập tin .htm. Tuy nhiên, nếu người dùng save toàn bộ file htm xuống thì vẫn có thể tìm ra được tập tin .js vì chắc chắn nó nằm trên máy của người dùng.
- Viết một số đoạn mã để mã hóa javascript sao cho người dùng không thể thấy source code của javascript một cách dễ dàng nhưng trình duyệt vẫn hiểu được. Có thể xem một demo tại <http://www.designerwiz.com/generator/encryptHTML.htm>

Tuy nhiên cách này cũng vẫn không che mắt được người dùng chuyên nghiệp.

## 6. MỘT SỐ VÍ DỤ MINH HỌA

### 6.1. Đổi màu nền của trang web hiện hành

Ví dụ sau minh họa việc thay đổi màu nền của trang web hiện hành động. Lệnh dùng để thay đổi màu nền của tài liệu là **document.bgcolor="rrggbb"**

```

<HTML>
<HEAD>
<TITLE>Change BG Color</TITLE>
</HEAD>
<BODY>
<FORM>
    <SELECT onChange=
        "document.bgColor=this.options[this.selectedIndex].value">
        <OPTION VALUE="40E0D0"> Torquoise
        <OPTION VALUE="2E8B57"> Sea Green
        <OPTION VALUE="87CEEB"> Sky Blue
        <OPTION VALUE="F4A460"> Sandy Brown
        <OPTION VALUE="FFF0F5"> Lavender Blush
        <OPTION VALUE="FF1493"> Deep Pink
        <OPTION VALUE="FFFFFF" SELECTED> White
    </SELECT>
</FORM>
</BODY>
</HTML>

```

## 6.2. Chữ chuyển động trên thanh trạng thái (scroller)

Ví dụ sau minh họa việc cho dòng chữ chạy trên thanh trạng thái của cửa sổ. Lệnh dùng để thay đổi nội dung của thanh trạng thái là **window.status=str**

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
  <!-- Start of scroller script
  var scrollCounter = 0;
  var scrollText   = "Welcome to my lesson!";
  var scrollDelay  = 70;
  var i = 0;

  while (i ++ < 140)
    scrollText = " " + scrollText;

  function Scroller()
  {
    window.status = scrollText.substring(scrollCounter++,
      scrollText.length);
    if (scrollCounter == scrollText.length)
      scrollCounter = 0;
    setTimeout("Scroller()", scrollDelay);
  }

  Scroller();
  // End of scroller script -->
</SCRIPT>
<TITLE>StatusScroller</TITLE>
</HEAD>
<BODY>
<P>See at your status bar!!!</P>
</BODY>
</HTML>
```

## 6.3. Đồng hồ

Ví dụ sau minh họa việc hiển thị thời gian hệ thống hiện hành. Đối tượng cung cấp thông tin về ngày giờ là **Date**

```
<HTML>
<HEAD>
<TITLE>Clock</TITLE>
</HEAD>
<BODY>
<TABLE BORDER=0 BGCOLOR=BLUE>
```

```
<TR><TD>
<FORM NAME="clock_form">
  <INPUT TYPE=TEXT NAME="clock" SIZE=26>
</FORM>

<SCRIPT LANGUAGE="JavaScript">
  <!-- Hide from non JavaScript browsers
  function clockTick()
  {
    currentTime = new Date();
    document.clock_form.clock.value = " "+currentTime;
    document.clock_form.clock.blur();
    setTimeout("clockTick()", 1000);
  }
  clockTick();
  // End of clock -->
</SCRIPT>
</TD></TR>
</TABLE>
</BODY>
</HTML>
```

## 6.4. Kiểm tra tính hợp lệ của dữ liệu nhập từ form

Ví dụ sau minh họa một form nhập liệu. Khi người dùng chọn nút Submit thì đoạn chương trình sẽ kiểm tra các ô dữ liệu có được nhập vào hay không. Nếu có một ô dữ liệu nào chưa nhập, chương trình sẽ hiện thông báo yêu cầu nhập lại

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
  <!-- Hide code from non-js browsers
  function validateForm()
  {
    formObj = document.sample;
    if (formObj.yourname.value == "")
    {
      alert("You have not filled in the name field.");
      formObj.yourname.focus();
      return false;
    }
    else if (formObj.yourage.value == "")
    {
```

```

        alert("You have not filled in the age field.");
        formObj.yourage.focus();
        return false;
    }
    else if (formObj.yourdob.value == "")
    {
        alert("You have not filled in your date of birth.");
        formObj.yourdob.focus();
        return false;
    }
    // end hiding -->
</SCRIPT>
<TITLE>FormValidation</TITLE>
</HEAD>
<BODY>
<FORM NAME="sample" METHOD=POST
ACTION="adduser.asp" onSubmit="return validateForm()">
    Enter your name :
<INPUT TYPE=TEXT NAME="yourname" SIZE=30><BR>
    Enter your age :
<INPUT TYPE=TEXT NAME="yourage" SIZE=3><BR>
    Date of birth :
<INPUT TYPE=TEXT NAME="yourdob" SIZE=10><BR>
<INPUT TYPE=SUBMIT>&nbsp;&nbsp;<INPUT TYPE=RESET>
</FORM>
</BODY>
</HTML>

```

## 6.5. Kiểm tra chuỗi nhập vào có phải là địa chỉ email hợp lệ hay không

Ví dụ sau minh họa việc sử dụng Javascript để kiểm tra chuỗi người dùng nhập vào có phải là một địa chỉ email hợp lệ hay không.

```

<HTML>
<HEAD>
<TITLE>Email Check</TITLE>
</HEAD>
<BODY>
<form name="validation" onSubmit="return checkbae()">
Please input a valid email address: <br>
<input type="text" size=18 name="emailcheck">
<input type="submit" value="Submit">
</form>
<script language="JavaScript1.2">

```

```

//Advanced Email Check credit-
//By JavaScript Kit (http://www.javascriptkit.com)
//Over 200+ free scripts here!

var testresults
function checkemail(){
var str=document.validation.emailcheck.value
var filter=/^([\w-]+(?:\.[\w-]+)*)@((?:[\w-]+\.)*\w[\w-]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)$/i
if (filter.test(str))
testresults=true
else{
alert("Please input a valid email address!")
testresults=false
}
return (testresults)
}
</script>

<script>
function checkbae(){
if (document.layers||document.getElementById||document.all)
return checkemail()
else
return true
}
</script>
</BODY>
</HTML>

```

## 6.6. Menu

Ví dụ sau minh họa một menu dropdown.

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript1.2">
/*
Drop down menu link © Dynamic Drive (www.dynamicdrive.com)
*/
//Contents for menu 1
var menu1=new Array()

```

```

menu1[0]='<a href=http://home.vdc.vn>VDC News</a><br>
menu1[1]='<a href=http://vnexpress.net>VN Express</a><br>
menu1[2]='<a href=http://www.is-edu.hcmuns.edu.vn>IS-EDU
Site</a><br>'
//Contents for menu 2
var menu2=new Array()
menu2[0]='<a href=http://www.yahoo.com>Yahoo</a><br>
menu2[1]='<a href=http://www.altavista.com>Altavista</a><br>
menu2[2]='<a href=http://www.lycos.com>Lycos</a><br>

</SCRIPT>

<STYLE>
<!--
.iewrap1{
position:relative;
height:30px;
}
.iewrap2{
position:absolute;
}
#dropmenu0, #dropmenu1{
visibility:hide;
z-index:100;
}
-->
</STYLE>
</HEAD>

<BODY>
<SCRIPT LANGUAGE="JavaScript1.2">
//reusable///////////
var zindex=100
function dropit2(whichone){
    if (window.themenu&&themenu.id!=whichone.id)
        themenu.style.visibility="hidden"
    themenu=whichone
    if (document.all){
        themenu.style.left=document.body.scrollLeft+event.clientX-
event.offsetX
        themenu.style.top=document.body.scrollTop+event.clientY-
event.offsetY+18
        if (themenu.style.visibility=="hidden"){

```

```

            themenu.style.visibility="visible"
            themenu.style.zIndex=zindex++
        }
    } else{
        hidemenu()
    }
}

function dropit(e,whichone){
    if (window.themenu&&themenu.id!=eval(whichone).id)
        themenu.visibility="hide"
    themenu=eval(whichone)
    if (themenu.visibility=="hide")
        themenu.visibility="show"
    else
        themenu.visibility="hide"
    themenu.zIndex++
    themenu.left=e.pageX-e.layerX
    themenu.top=e.pageY-e.layerY+19
    return false
}

function hidemenu(whichone){
    if (window.themenu)
        themenu.style.visibility="hidden"
}

function hidemenu2(){
    themenu.visibility="hide"
}

if (document.all)
    document.body.onclick=hidemenu

//reusable///////////
</SCRIPT>

<!-------Menu 1 starts here----->

<ilayer height=35px>
<layer visibility=show>
<span class=iewrap1>
<span class=iewrap2
onClick="dropit2(dropmenu0); event.cancelBubble=true; return

```

```

false">
<font face=Verdana><b><a href="alternate.htm"
onClick="if(document.layers) return dropit(event,
'document.dropdown0')>Local Links</a></b></font>
</span>
</span>
</layer>
</layer><br>
<!-----Menu 1 ends here----->

<!-----Menu 2 starts here----->

<ilayer height=35px>
<layer visibility=show>
<span class=iewrap1>
<span class=iewrap2>
onClick="dropit2(dropmenu1); event.cancelBubble=true; return
false">
<font face=Verdana><b><a href="alternate.htm"
onClick="if(document.layers) return dropit(event,
'document.dropdown1')>Global Links</a></b></font>
</span>
</span>
</layer>
</layer><br>
<!-----Menu 2 ends here----->

<P>&nbsp;</P>
<DIV id=dropdown0 STYLE="position: absolute; left: 0; top: 0; layer-
background-color: lightyellow; background-
color: lightyellow; width: 120; visibility: hidden; border: 2px solid
black; padding: 0px">

<SCRIPT LANGUAGE="JavaScript1.2">
    if (document.all)
        dropdown0.style.padding="4px"
    for (i=0;i<menu1.length;i++)
        document.write(menu1[i])
</SCRIPT>
</DIV>

<SCRIPT LANGUAGE="JavaScript1.2">
    if (document.layers){
        document.dropdown0.captureEvents(Event.CLICK)
        document.dropdown0.onclick=hidemenu2
    }
</SCRIPT>

```

```

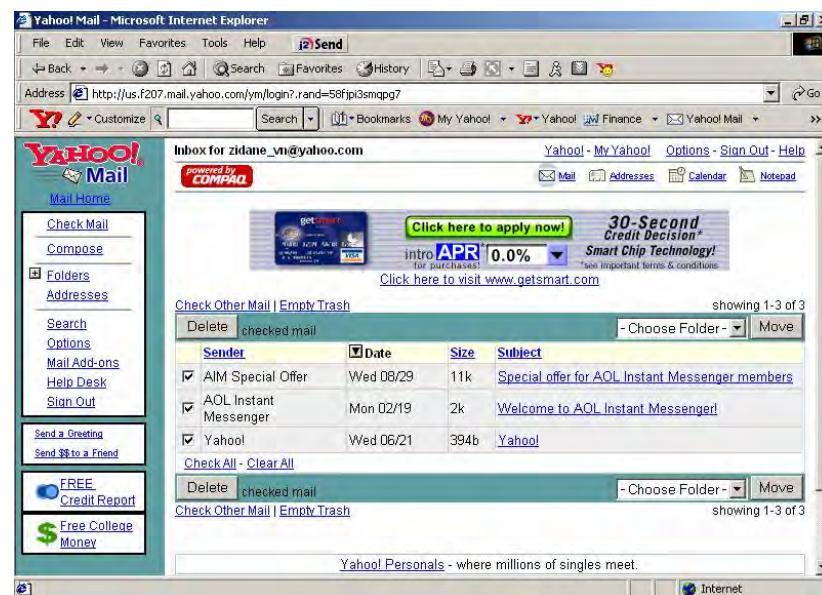
}
</SCRIPT>
<DIV id=dropdown1 STYLE="position: absolute; left: 0; top: 0; layer-
background-color: lightyellow; background-
color: lightyellow; width: 120; visibility: hidden; border: 2px solid
black; padding: 0px">
<SCRIPT LANGUAGE="JavaScript1.2">
    if (document.all)
        dropmenu1.style.padding="4px"
    for (i=0;i<menu2.length;i++)
        document.write(menu2[i])
</SCRIPT>
</DIV>
<SCRIPT LANGUAGE="JavaScript1.2">
    if (document.layers){
        document.dropdown1.captureEvents(Event.CLICK)
        document.dropdown1.onclick=hidemenu2
    }
</SCRIPT>
</BODY>
</HTML>

```

# THỰC HÀNH

## 1. Kiểm tra tính hợp lệ của dữ liệu trong form

- Thiết kế một form nhập Username và Password từ người dùng. Form chỉ được submit một khi người dùng có nhập thông tin cả 2 ô Username và Password. Người ta yêu cầu các thông tin nhập vào 2 ô này chỉ được phép là chữ cái hoặc chữ số thôi. Đưa ra các thông báo tương ứng và yêu cầu người dùng nhập lại cho đến khi các điều kiện được kiểm tra thành công.
- Thiết kế một form có nhập dữ liệu dạng ngày, kiểm tra xem ngày mà người dùng nhập vào có hợp lệ hay không.
- Tạo một form nhập liệu, trong đó có 10 checkbox và 1 checkbox có tên Check All. Khi người dùng nhấn nút CheckAll, trạng thái của 10 checkbox trên sẽ thay đổi theo trạng thái của nút này. Xem 1 ví dụ của Yahoo Mail sau:



## 2. Tham khảo các mã JavaScript

- [www.javascriptkit.com](http://www.javascriptkit.com)
- <http://javascript.internet.com/>
- [www.dynamicdrive.com](http://www.dynamicdrive.com)
- <http://javascript.com/>
- <http://www.mjtnet.com/resources.htm>

### Tham khảo các website trên để ứng dụng vào:

- =Tạo một lịch (calender), cho phép người dùng chọn ngày, tháng, năm cho các ứng dụng liên quan đến xếp lịch hay chọn ngày tháng năm sinh. Tham khảo: <http://javascript.internet.com/calendars/dynamic.html>
- Chuyển tất cả thành chữ in hoa khi người dùng vừa nhập xong một ô nhập liệu. Tham khảo: <http://javascript.internet.com/forms/all-upper-case.html>
- Chọn theo 2 cấp. Giả sử ứng dụng muốn lấy thông tin về quận huyện mà người dùng muốn làm việc. Chương trình đầu tiên sẽ hiển thị danh mục các tỉnh thành, sau khi chọn xong tỉnh, chương trình sẽ hiển thị danh mục các quận huyện tương ứng với tỉnh đó. Tham khảo: <http://javascript.internet.com/forms/country.html>
- Giải thích ô nhập liệu. Bên cạnh một ô nhập liệu, tạo một hyperlink đến một cửa sổ hướng dẫn cách thức nhập liệu cho ô đó. Khi người dùng xem xong, focus trở lại ô nhập liệu để người dùng nhập. Tham khảo: <http://javascript.internet.com/forms/field-explanation.html>

## Chương 5

# NHẬP MÔN ASP

## 1. GIỚI THIỆU VỀ ASP

Active Server Page (ASP) do Microsoft phát triển là môi trường lập trình ứng dụng phía server (server side scripting) hỗ trợ mạnh trong việc xây dựng các ứng dụng Web. Các ứng dụng ASP rất dễ viết và sửa đổi, đồng thời có thể tích hợp các công nghệ sẵn có của Microsoft như COM, ... một cách dễ dàng.

ASP được hỗ trợ mặc định khi cài đặt Internet Information Server. Để thực hiện ASP trên các môi trường khác, bạn phải cài đặt một thư viện hỗ trợ ASP. Thông dụng nhất là Sun Chili!Soft (<http://www.chillisoft.com>)



Một trang ASP thường có một số đặc điểm sau:

- Là một tập tin văn bản (text file) có phần mở rộng là **.asp**: Phần mở rộng này sẽ giúp webserver yêu cầu trình xử lý trang ASP (ASP engine) trước khi trả về cho trình duyệt.
- Ngôn ngữ script thông dụng nhất dùng để viết các mã của ASP là VBScript. Ngoài ra, ta cũng có thể viết các mã bằng các ngôn ngữ như JScript, Perl, Python, ... nếu trên webserver có cài đặt các bộ xử lý ngôn ngữ này (script engine).
- Các đoạn mã viết trong trang ASP sẽ được các bộ xử lý ngôn ngữ trên webserver xử lý tuân tự từ trên xuống dưới. Kết quả của việc xử lý này là trả về trang HTML cho webserver và webserver sẽ gửi trang này về cho trình duyệt. Đó là lí do tại sao, tại trình duyệt ta không thấy được các đoạn mã chương trình đã được viết trong trang ASP.

## 2. CÁC KHÁI NIỆM CƠ BẢN VỀ ASP

Một trang ASP thông thường gồm có bốn thành phần:

- Dữ liệu văn bản (text)
- Các tag HTML
- Các đoạn mã chương trình phía client đặt trong cặp tag **<SCRIPT>** và **</SCRIPT>**
- Mã chương trình ASP được đặt trong cặp tag **<%** và **%>**

Ba thành phần ban đầu là cấu trúc của một trang HTML thông thường, do đó có thể xem một trang ASP là một trang HTML được nhúng thêm phần xử lý viết bằng mã ASP (VBScript, JScript, Perl, ...)

Ví dụ sau minh họa một trang ASP, dữ liệu văn bản là "Welcome to my website. Today is:", các tag HTML là **<P>**, **<B>**, ... và đoạn mã chương trình đặt giữa **<%** và **%>**

```
<HTML>
<BODY>
<P> <B>Welcome to my website</B>. Today is
<%
    Response.Write Date()
%>
</BODY>
</HTML>
```

Trong trang ASP, ta có thể trộn lẫn mã ASP và mã HTML vào trong các cấu trúc điều khiển. Xét ví dụ sau:

```
<HTML>
<BODY>
<%
    Dim dtmHour

    dtmHour = Hour(Now())

    If dtmHour < 12 Then
        Response. Write "Good Morning!"
    Else
        Response. Write "Hello!"
    End If
%>
</BODY>
</HTML>
```

Đoạn mã chương trình trên sẽ in ra màn hình lời chào “Good Morning” nếu thời điểm truy cập vào trang này là buổi sáng, ngược lại nó sẽ in ra màn hình lời chào “Hello”. Lệnh Response.Write trong ví dụ trên đóng vai trò kết xuất nội dung dữ liệu ra trang web, do đó ta có thể thay bằng đoạn mã HTML có chức năng tương tự như minh họa sau:

```
<HTML>
<BODY>
<%
Dim dtmHour

dtmHour = Hour(Now())

If dtmHour < 12 Then
%>
  Good Morning!
<%
Else
%>
  Hello!
<%
End If
%>
</BODY>
</HTML>
```

Do ta có thể sử dụng nhiều ngôn ngữ script khác nhau để viết mã ASP, nên để chỉ định ngôn ngữ nào là ngôn ngữ đang được dùng trong một trang ASP, ta đặt đoạn mã sau vào đầu trang ASP: **<%LANGUAGE=ScriptingLanguage%>**. Ví dụ, để chỉ định VBScript là ngôn ngữ dùng cho trang ASP, ta viết như sau: **<%LANGUAGE=VBScript%>**

#### Cách viết các chú thích cho trang ASP:

- Đối với VBScript, ta dùng kí tự '. Ví dụ:

```
<%
'This line and the following two are comments.
'The PrintTable function prints all
'the elements in an array.
PrintTable MyArray()
%>
```

- Đối với JScript, ta dùng kí tự //. Ví dụ:

```
<%
var x
x = new Date()
// This line sends the current date to the browser,
// translated to a string.
Response.Write(x.toString())
%>
```

Các đoạn mã ASP trong tài liệu này từ nay trở về sau sẽ được minh họa bằng VBScript.

Để có thể viết được một chương trình bằng ASP, chúng ta cần phải biết:

- Cách khai báo và sử dụng các biến.
- Cách viết các cấu trúc điều khiển như lệnh điều kiện, lệnh lặp, ...
- Cách viết hàm, thủ tục, ...
- Cách sử dụng các hàm thư viện cơ bản hỗ trợ cho việc nhập, xuất dữ liệu. Ví dụ, nếu bạn học C, bạn phải học cách sử dụng các hàm như scanf, printf, ... hay nếu bạn học Pascal, bạn phải học cách sử dụng các hàm như readln, writeln, ... Trong ASP, bạn phải học cách sử dụng chúng thông qua các đối tượng được xây dựng sẵn như Request, Response, ...
- Các hàm thư viện hỗ trợ cho các thao tác phức tạp khác như truy xuất tới cơ sở dữ liệu, hệ thống tập tin, các tiện ích của hệ thống, ...

Ba phần đầu liên quan đến việc sử dụng một ngôn ngữ script đã được đề cập trong bài trước.

Hầu hết các đối tượng xây dựng sẵn của ASP đều là các đối tượng kiểu **Collection**. Collection là cấu trúc dữ liệu tương tự như mảng, nó có thể lưu trữ bất kì loại dữ liệu nào như số, chuỗi kí tự, các đối tượng, ... Khác với mảng, kích thước các Collection sẽ thay đổi một cách tự động mỗi khi có các thao tác như thêm hoặc xóa dữ liệu trên biến kiểu này.

Có 3 cách để truy cập vào một mục dữ liệu (item) được lưu trữ trong biến Collection:

- Thông qua chuỗi kí tự khóa duy nhất (unique string key). Trong ví dụ sau, **Age** là chuỗi kí tự khóa được dùng để truy cập vào biến collection Session.Contents: <% Session.Contents("Age") = 29 %>
- Thông qua vị trí của mục dữ liệu đó (index). Ví dụ: <% Session.Contents(2) = 29 %>
- Bằng cách duyệt qua hết các thành phần trong collection. Ví dụ:

```
<%
'Declare a counter variable.
Dim strItem

'For each item in the collection, display its value.
For Each strItem In Session.Contents
    Response.Write Session.Contents(strItem) & "<BR>"
    Next
%>
```

### 3. XỬ LÍ DỮ LIỆU NHẬP TỪ NGƯỜI DÙNG

Trong các ứng dụng web, người dùng nhập dữ liệu thông qua đối tượng form. Sau khi người dùng submit form, các dữ liệu trong form sẽ được chuyển đến cho chương trình xử lý được khai báo trong mục ACTION. Ví dụ sau minh họa một form nhập liệu, sau khi người dùng nhấn nút Search để submit form, dữ liệu nhập sẽ được chuyển cho trang search.asp:

```
<FORM ACTION="search.asp" NAME=f1 METHOD=GET>
<INPUT TYPE=TEXT NAME="KEYWORD" SIZE=20>
<INPUT TYPE=SUBMIT NAME=b1 VALUE="Search">
</FORM>
```

Trong trang search.asp, để có thể lấy dữ liệu được chuyển từ form trên. Ta dùng đối tượng Request theo hai cách sau:

- Nếu khai báo METHOD trong tag <FORM> là GET, ta dùng **Request.QueryString**
- Nếu khai báo METHOD trong <FORM> là POST, ta dùng **Request.Form**

Để lấy dữ liệu từ ô nhập liệu nào, ta dùng **tên** của ô nhập liệu đó (khai báo trong thuộc tính NAME của đối tượng) như là chuỗi kí tự khóa theo cách truy cập tới dữ liệu của biến Collection ở trên. Ví dụ,

để lấy dữ liệu từ ô nhập liệu của ví dụ trên (có tên là **KEYWORD**), ta dùng **Request.QueryString("KEYWORD")**.

Ví dụ sau minh họa một ứng dụng nhỏ, người dùng sẽ nhập vào họ, tên, giới tính trong form nằm trong tập tin **input.htm**. Sau khi người dùng submit form, chương trình nằm trong tập tin **hello.asp** sẽ in ra câu chào tương ứng với giới tính của họ.

**input.htm**

```
<HTML>
<HEAD>
<TITLE>Input data</TITLE>
</HEAD>
<BODY>
<FORM ACTION="hello.asp" NAME="myform" METHOD="GET">
    First Name: <INPUT TYPE="TEXT" NAME="FirstName"
    SIZE=20><br>
    Last Name: <INPUT TYPE="TEXT" NAME="LastName"
    SIZE=20><br>
    Gender: <SELECT NAME="Gender" >
        <OPTION VALUE="Mr">Male</OPTION>
        <OPTION VALUE="Ms">Female</OPTION>
    </SELECT>
    <INPUT TYPE="SUBMIT" NAME="b1" VALUE="Submit">
</FORM>
</BODY>
</HTML>
```

**hello.asp**

```
<HTML>
<HEAD>
<TITLE>Output data</TITLE>
</HEAD>
<BODY>
<%
    FName = Request.QueryString("FirstName")
    LName = Request.QueryString("LastName")
    Gender = Request.QueryString("Gender")
%> <p>Hello
<%
    Response. Write Gender & " " & FName & " " & LName
%>
</BODY>
</HTML>
```

## 4. XỬ LÍ CÁC KẾT XUẤT DỮ LIỆU

Trong mã chương trình phía client (đặt trong cặp tag <SCRIPT> và </SCRIPT>), để gửi dữ liệu ra trang web hiện hành, bạn dùng phương thức document.write. Tương tự như vậy, để gửi dữ liệu ra trang web ngay trong đoạn mã ASP (đặt trong cặp tag <% và %>), bạn dùng phương thức Response.Write

Sau đây là một số ví dụ khi sử dụng phương thức này:

- In một hàng kiểu số, ví dụ như 5: **Response.Write 5**
- In một hàng chuỗi, ví dụ như "Hello World": **Response.Write "Hello World"**. Lưu ý rằng, hàng chuỗi bao giờ cũng được rào quanh bởi cặp dấu nháy đơn ('') hoặc nháy kép ("").
- In một giá trị của một biến, ví dụ như biến FirstName: **Response.Write FirstName**
- In kết hợp các dạng dữ liệu trên, ta dùng toán tử &. Ví dụ: **Response.Write "No " & 1 & FirstName**

Khi muốn xuất ra một chuỗi dài, ta có thể chia nhỏ chuỗi này thành nhiều dòng bằng cách sử dụng chuỗi kí tự: "&\_". Ví dụ:

```
<%
    Response.Write "Chào mừng các bạn đến với website" &_
        "của Trường ĐH Khoa Học Tự Nhiên"
%>
```

Ngoài ra, thay vì sử dụng phương thức Response.Write để kết xuất ra trình duyệt, ta cũng có thể dùng dạng tương tự <%==%>. Ví dụ sau minh họa hai đoạn mã cho kết quả tương đương:

```
<%
    Response.Write "Trường ĐH Khoa Học Tự Nhiên"
%>
<%= "Trường ĐH Khoa Học Tự Nhiên" %>
```

Cách dùng <%==%> chỉ thích hợp cho trường hợp dùng cho các đoạn mã chỉ thuần các kết xuất đơn giản mà thôi.

### 4.1. Sử dụng các dấu nháy

VB Script dùng các dấu nháy để đánh dấu sự bắt đầu và kết thúc của một chuỗi. Nếu trong bản thân chuỗi có chứa dấu nháy thì vấn đề sẽ nảy sinh. Ví dụ: nếu dùng đoạn mã sau để in chuỗi: He said, "This doesn't work!":

```
<%
    Response.Write "He said, "This doesn't work!""
%>
```

Thì chương trình sẽ báo lỗi sai cú pháp vì lúc đó chương trình sẽ hiểu "He said, " là chuỗi đầu tiên This doesn't work!"" là chuỗi thứ hai nhưng viết không đúng cú pháp vì thiếu dấu " trước This.

Có ba cách để giải quyết vấn đề này. Cách thứ nhất là ghi liền tiếp hai dấu nháy để VB Script dịch lại thành còn một dấu nháy. Ví dụ như, câu lệnh trên nên viết lại với hai dấu nháy trước This và ba dấu nháy sau work!:

```
<%
    Response.Write "He said, ""This doesn't work!"""
%>
```

Cách thứ hai là sử dụng hàm CHR(34) để in dấu nháy ra. Ví dụ:

```
<%
    Response.Write "He said, " & CHR(34) & "This doesn't work!" &
    CHR(34)
%>
```

Cách thứ ba là thay thế dấu nháy kép bằng dấu nháy đơn. Nghĩa là:

```
<%
    Response.Write "He said, 'This doesn't work! ' "
%>
```

### 4.2. Hiển thị các kí tự đặc biệt

Khi làm việc với lệnh Response.Write, có một số kí tự trong chuỗi có thể sẽ không được hiển thị chính xác, ví dụ như dấu >, <, bởi vì HTML xem các dấu này như là các kí tự bắt đầu và kết thúc của một tag. Do đó, để khắc phục ta phải chuyển các kí tự đặc biệt này sang mã thay thế bằng hàm Server.HTMLEncode. Ví dụ, ta muốn in ra một chuỗi: "<B>Hyperion</b> by <i>Dan Simmons</i> is a great novel", ta phải dùng như sau:

```
<%
    Response.Write Server.HTMLEncode("<B>Hyperion</b> by
    <i>Dan Simmons</i> is a great novel ")
%>
```

Xem rõ hơn qua ví dụ minh họa sau:

```
<HTML>
<HEAD>
</HEAD>
<BODY>
    Response.Write "<B>Hyperion</b> by <i>Dan Simmons</i> is
a great novel"
    Response.Write "<p>"
    Response.Write server.htmlencode("<B>Hyperion</b> by
<i>Dan Simmons</i> is a great novel")
    Response.Write "<p>"

    Response.Write "Joe Smith & Hilda = a team"
    Response.Write "<p>"
    Response.Write server.URLEncode("Joe Smith & Hilda = a team")
    %>

</BODY>
</HTML>
```

### 4.3. Một số ví dụ minh họa

Sau đây là một số ví dụ minh họa việc nhúng các đoạn mã ASP cùng với mã HTML để cho các kết xuất thường gặp trong lập trình.

#### 4.3.1. Ví dụ 1: Trộn lẫn các dạng kết xuất trong trang ASP

Ví dụ sau minh họa việc trộn lẫn các dạng kết xuất bằng cách dùng phương thức Response.Write, <%=> hay trộn lẫn mã HTML giữa các lệnh điều khiển ASP, đồng thời minh họa việc sử dụng uyển chuyển các phương pháp dùng dấu nháy.

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<%
    Response.Write "<FORM>"
    Response.Write "Hello, Joe<br>"
    UsrName="Joe"
    Response.Write "Hello, " & UsrName & "<br>"
%>

Hello, <%=UsrName%><br>
Which Book? <input type="TEXT" name="book" value="The
```

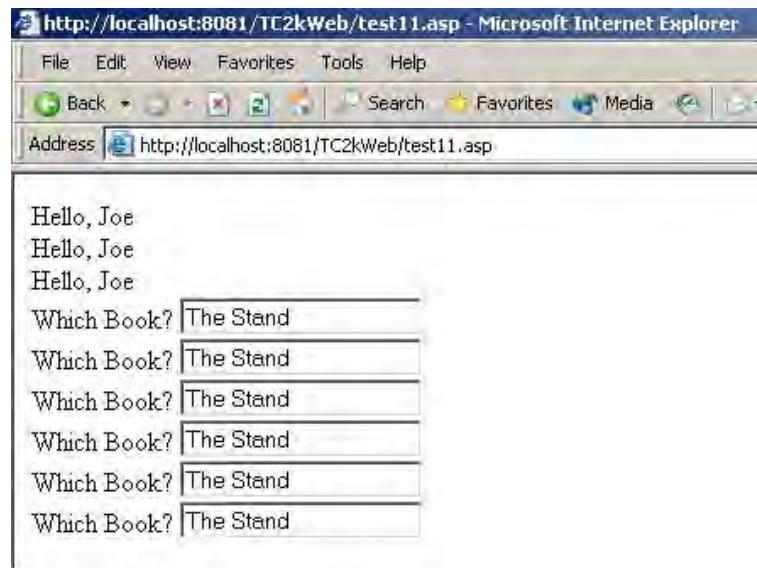
```
Stand"><br>
```

```
<%
    Response.Write "Which Book? <input type=""TEXT"""
name=""book"" value=""The Stand""><br>"%
>

<%
    Response.Write "Which Book? <input type='TEXT' name='book'
value='The Stand'><br>"%
>
<%
    quote=chr(34)
    Response.Write "Which Book? <input type=" & quote & "TEXT" &
quote & " name=" & quote & "book" & quote & " value=" & quote &
"The Stand" & quote & "><br>"%
>

<%bookname="The Stand"%>
Which Book? <input type="TEXT" name="book"
value="<%=bookname%>"><br>

<%
    Response.Write "Which Book? <input type=""TEXT"""
name=""book"" value=""" & bookname & """><br>"%
>
</FORM>
</BODY>
</HTML>
```



Hình 1 – Kết quả của đoạn mã trên

### 4.3.2. Tạo một dropdown listbox động

Giả sử ta muốn cho phép người dùng nhập vào năm sinh từ một dropdown listbox. Mã HTML của dropdown listbox này có dạng:

```
<FORM>
Year <SELECT NAME= "Year" >
<OPTION VALUE="1930">1930</OPTION>
<OPTION VALUE="1931">1931</OPTION>
<OPTION VALUE="1932">1932</OPTION>
...
<OPTION VALUE="1980">1980</OPTION>
</SELECT>
</FORM>
```

Dễ thấy rằng nếu soạn bằng HTML, ta phải gõ vào rất nhiều để có thể có được một dropdown listbox gồm các năm từ 1930 đến 1980. Trong khi đó, dropdown listbox này có thể sinh ra dễ dàng bằng cách sử dụng một vòng lặp in tuần tự các dòng dạng: <OPTION VALUE="1930">1930</OPTION> như sau:

```
<FORM>
Year <SELECT NAME= "Year" >
<%
    for i=1930 to 1980
        Response.Write "<OPTION VALUE=" & i & ">"
        Response.Write i & "</OPTION>"
    next
%>
</SELECT>
</FORM>
    Hay là đoạn mã dùng <%==%>
```

```
<FORM>
Year <SELECT NAME= "Year" >
<%
    for i=1930 to 1980
%>
    <OPTION VALUE="<%=i%>"><%= i %></OPTION>
next
%>
</SELECT>
</FORM>
```

### 4.3.3. Tạo bảng dữ liệu động

Giả sử ta cần hiển thị một bảng dữ liệu có dạng sau:

TT	MSSV	Họ và Tên	Ghi chú
1	99001	A1	
2	99002	A2	
..			
99	990099	A99	

Mã HTML của bảng dữ liệu này có dạng:

```
<TABLE BORDER="1" CELLPADDING="5" CELLSPACING="0"
STYLE="border-collapse: collapse" BORDERCOLOR="#111111"
WIDTH="400" >
<TR>
<TD WIDTH="30" ALIGN="center">TT</TD>
<TD WIDTH="60" ALIGN="center">MSSV</TD>
<TD WIDTH="210" ALIGN="center">Họ và Tên</TD>
<TD WIDTH="100" ALIGN="center">Ghi chú</TD>
</TR>
<TR>
<TD>1</TD>
<TD>99001</TD>
```

```

<TD>A1</TD>
<TD>&nbsp;</TD>
</TR>
<TR>
<TD>2</TD>
<TD>99002</TD>
<TD>A2</TD>
<TD>&nbsp;</TD>
</TR>
...
<TR>
<TD>99</TD>
<TD>990099</TD>
<TD>A99</TD>
<TD>&nbsp;</TD>
</TR>
</TABLE>

```

Ta sẽ dùng một vòng lặp để in tuần tự các dòng của bảng, với nội dung chứa trong cặp tag `<TR>` và `</TR>`

```

<TABLE BORDER="1" CELLSPACING="5" CELLPADDING="0"
STYLE="border-collapse: collapse" BORDERCOLOR="#111111"
WIDTH="400" >
<TR>
<TD WIDTH="30" ALIGN="center">TT</TD>
<TD WIDTH="60" ALIGN="center">MSSV</TD>
<TD WIDTH="210" ALIGN="center">Họ và Tên</TD>
<TD WIDTH="100" ALIGN="center">Ghi chú</TD>
</TR>
<%
for i= 1 to 99
%>
<TR>
<TD><%=i%></TD>
<TD>9900<%=i%></TD>
<TD>A<%=i%></TD>
<TD>&nbsp;</TD>
</TR>
<%
next
%>
</TABLE>

```

## 5. KẾT NỐI VỚI CƠ SỞ DỮ LIỆU

ADO (ActiveX Data Object) được dùng để cung cấp khả năng kết nối và xử lý trên cơ sở dữ liệu. Ta có thể sử dụng ADO để kết nối với nguồn dữ liệu tương thích với OLE DB như cơ sở dữ liệu, bảng tính, tập tin tuần tự, ... OLE DB là giao diện lập trình mức hệ thống cung cấp tập các giao tiếp COM chuẩn để khai thác các chức năng quản trị hệ cơ sở dữ liệu. Ngoài ra ta cũng có thể sử dụng ADO để truy xuất đến các cơ sở dữ liệu tương thích với ODBC.

Việc sử dụng ADO để truy cập và xử lý cơ sở dữ liệu trong các trang ASP có thể chia làm các bước chính sau:

- Kết nối với dữ liệu thông qua OLE DB hoặc ODBC
- Xây dựng câu truy vấn dữ liệu và yêu cầu thực hiện câu truy vấn để thực hiện các thao tác xử lý trên cơ sở dữ liệu như thêm, xóa, sửa, lọc dữ liệu, ...
- Xử lý kết quả trả về từ câu truy vấn trong trường hợp câu truy vấn lọc dữ liệu (SELECT queries)
- Ngắt kết nối với cơ sở dữ liệu, giải phóng các tài nguyên của hệ thống đã dùng.

### 5.1. Kết nối với cơ sở dữ liệu

#### 5.1.1. Tạo connection string

Giống như khi bạn muốn trò chuyện bằng điện thoại, bước đầu tiên là phải quay số điện thoại của người mà bạn cần trò chuyện. Để có thể xử lý dữ liệu, bước đầu tiên, bạn phải cung cấp các thông tin cần thiết để hệ thống biết bạn muốn truy cập tới cơ sở dữ liệu nào. Một chuỗi kí tự gọi là connection string sẽ được dùng để lưu trữ các thông tin này, bao gồm:

- Thông tin về hệ quản trị cơ sở dữ liệu được dùng để quản lý cơ sở dữ liệu của bạn. Ví dụ đó là MS Access hoặc MS SQL hay Oracle, ...
- Thông tin về vị trí của cơ sở dữ liệu của bạn. Ví dụ: nếu bạn dùng MS Access, bạn phải chỉ ra cơ sở dữ liệu của bạn được lưu trong tập tin .mdb nào.

Bảng sau liệt kê các OLE DB connection string cho một số hệ quản trị cơ sở dữ liệu thông dụng:

Data Source	OLE DB Connection String
Microsoft® Access	Provider=Microsoft.Jet.OLEDB.4.0;Data Source=physical path to .mdb file
Microsoft SQL Server	Provider=SQLOLEDB.1;Data Source=path to database on server
Oracle	Provider=MSDAORA.1;Data Source=path to database on server
Microsoft Indexing Service	Provider=MSIDXS.1;Data Source=path to file

Bảng sau liệt kê các ODBC connection string cho một số hệ quản trị cơ sở dữ liệu thông dụng:

Data Source Driver	ODBC Connection String
Microsoft® Access	Driver={Microsoft Access Driver (*.mdb)};DBQ=physical path to .mdb file
Microsoft SQL Server	DRIVER={SQL Server};SERVER=path to server
Oracle	DRIVER={Microsoft ODBC for Oracle};SERVER=path to server
Microsoft Excel	Driver={Microsoft Excel Driver (*.xls)};DBQ=physical path to .xls file;DriverID=278
Microsoft Excel 97	Driver={Microsoft Excel Driver (*.xls)};DBQ=physical path to .xls file;DriverID=790
Paradox	Driver={Microsoft Paradox Driver (*.db)};DBQ=physical path to .db file;DriverID=26
Text	Driver={Microsoft Text Driver (*.txt; *.csv)};DefaultDir=physical path to .txt file

Data Source Driver	ODBC Connection String
Microsoft Visual FoxPro® (with a database container)	Driver={Microsoft Visual FoxPro Driver};SourceType=DBC;SourceDb=physical path to .dbc file
Microsoft Visual FoxPro (without a database container)	Driver={Microsoft Visual FoxPro Driver};SourceType=DBF;SourceDb=physical path to .dbf file

Ta cũng có thể dễ dàng xây dựng connection string bằng cách tạo Data Source Name trong ODBC.

Để đảm bảo tính hiệu quả và tin cậy, bạn nên dùng các hệ quản trị cơ sở dữ liệu kiểu client-server (client-server database engine) khi phát triển các ứng dụng web vì nó đòi hỏi số lượng kết nối đồng thời lớn. Mặc dù ADO hỗ trợ được các nguồn dữ liệu tương thích với OLE DB nhưng nó chủ yếu được thiết kế để làm việc tốt với các hệ quản trị cơ sở dữ liệu kiểu client-server như MS SQL, Oracle, ...

### 5.1.2. Kết nối với cơ sở dữ liệu

ADO cung cấp đối tượng **Connection** để hỗ trợ cho việc tạo và quản lý kết nối với cơ sở dữ liệu cần xử lý. Các thuộc tính và phương thức của đối tượng này cho phép bạn mở, đóng kết nối, đồng thời cho phép thực hiện các câu truy vấn dữ liệu,

Để thiết lập kết nối với cơ sở dữ liệu bằng ADO, bạn thực hiện tuần tự các bước sau:

- Tạo một thê hiện của đối tượng Connection bằng lệnh: Server.CreateObject("ADODB.Connection").
- Sử dụng phương thức Open để mở kết nối với cơ sở dữ liệu. Tham số cho phương thức này là chuỗi connection string.

Ví dụ sau minh họa việc tạo kết nối đến cơ sở dữ liệu được lưu bằng MS Access:

```
<%
strDSN = "Driver={Microsoft Access Driver (*.mdb)};DBQ=" &
server.mappath("Test.mdb")

'Create a connection object.
Set Conn = Server.CreateObject("ADODB.Connection")

'Open a connection using the ODBC connection string.
Conn.Open strDSN
%>
```

## 5.2. Xây dựng câu truy vấn và yêu cầu thực hiện

Cách đơn giản nhất để thực hiện các thao tác trên cơ sở dữ liệu là xây dựng các câu truy vấn SQL (SQL queries) và yêu cầu hệ thống thực hiện. Phương thức **Execute** do đối tượng Connection cung cấp cho phép thực hiện một câu truy vấn SQL.

Ví dụ sau minh họa việc thực hiện thêm một mẫu tin vào cơ sở dữ liệu

### insert.asp

```
<%
strDSN = "Driver={Microsoft Access Driver (*.mdb)};DBQ=" &
server.mappath("Test.mdb")

'Create a connection object.
Set Conn = Server.CreateObject("ADODB.Connection")

'Open a connection using the ODBC connection string.
Conn.Open strDSN

'Define SQL statement.
strSQL = "INSERT INTO Customers (FirstName, LastName)
VALUES ('Jose','Lugo')"

'Use the Execute method to issue a SQL query to database.
Conn.Execute strSQL
%>
```

Các lệnh như DELETE, UPDATE cũng được thực hiện theo cách tương tự. Ví dụ:

### update.asp

```
<%
strDSN = "Driver={Microsoft Access Driver (*.mdb)};DBQ=" &
server.mappath("Test.mdb")
```

```
'Create a connection object.
Set Conn = Server.CreateObject("ADODB.Connection")
```

```
'Open a connection using the ODBC connection string.
Conn.Open strDSN
```

```
'Define SQL statement.
strSQL = "UPDATE Customers SET FirstName = 'Jeff' WHERE
LastName = 'Smith'"
```

```
'Use the Execute method to issue a SQL query to database.
Conn.Execute strSQL
%>
```

### delete.asp

```
<%
strDSN = "Driver={Microsoft Access Driver (*.mdb)};DBQ=" &
server.mappath("Test.mdb")
```

```
'Create a connection object.
Set Conn = Server.CreateObject("ADODB.Connection")
```

```
'Open a connection using the ODBC connection string.
Conn.Open strDSN
```

```
'Define SQL statement.
strSQL = "DELETE FROM Customers WHERE LastName = 'Smith'"
```

```
'Use the Execute method to issue a SQL query to database.
Conn.Execute strSQL
%>
```

Một số lưu ý khi xây dựng câu truy vấn cơ sở dữ liệu

- Dữ liệu kiểu chuỗi phải được đặt trong cặp dấu nháy đơn '' vì cặp dấu nháy kép "" được dùng cho việc định nghĩa hằng chuỗi. Ví dụ: Để mô tả LastName sẽ lấy giá trị là chuỗi Smith, trong câu lệnh SQL ta dùng LastName='Smith'.
- Các dữ trường dữ liệu có khoảng trắng ở giữa, trong câu lệnh SQL ta dùng cặp dấu []. Ví dụ, trong cơ sở dữ liệu của bạn có định nghĩa trường "Last Name", trong câu lệnh SQL, ta dùng [Last Name]='Smith' chứ không dùng Last Name=Smith.

### 5.3. Xử lý kết quả trả về từ câu truy vấn lọc dữ liệu

#### 5.3.1. Lưu trữ dữ liệu trả về

ADO sử dụng đối tượng **Recordset** để lưu trữ kết quả trả về từ câu truy vấn lọc dữ liệu SELECT. Vì kết quả trả về của một câu truy vấn SELECT có thể có nhiều mẫu tin, cho nên có thể xem Recordset như là một mảng các mẫu tin.

Thông thường ta dùng 2 cách sau để lấy dữ liệu từ câu truy vấn vào biến Recordset

- Lấy kết quả trả về từ việc thực hiện câu truy vấn bằng phương thức **Execute** của đối tượng Connection. Ví dụ: **Set rs = Conn.Execute (strSQL)**. Trong trường hợp này rs là thể hiện của đối tượng Recordset lưu trữ kết quả trả về từ việc thực hiện câu truy vấn strSQL.
- Tạo một thể hiện của đối tượng Recordset và sử dụng phương thức **Open**, kết hợp với thể hiện của đối tượng Connection đã tạo. Ví dụ:

```
<%
strDSN = "Driver={Microsoft Access Driver (*.mdb)};DBQ=" &
server.mappath("Test.mdb")

'Create a connection object.
Set Conn = Server.CreateObject("ADODB.Connection")

'Instantiate a Recordset object.
Set rsCustomers = Server.CreateObject("ADODB.Recordset")

'Open a recordset using the Open method
'and use the connection established by the Connection object.
strSQL = "SELECT FirstName, LastName FROM Customers WHERE
LastName = 'Smith' "

rsCustomers.Open strSQL, Conn
%>
```

#### 5.3.2. Hiển thị dữ liệu trả về

Có ba vấn đề lưu ý:

- Để truy cập đến dữ liệu của một trường (field) của mẫu tin hiện hành, ta chỉ định tên của trường đó như là chuỗi kí tự khóa khi truy cập đến đối tượng Recordset. Ví dụ: để truy cập

đến dữ liệu của trường FirstName trong ví dụ trên ta dùng: **rsCustomers("FirstName")**

- Để di chuyển đến các mẫu tin được lưu trong đối tượng Recordset, ta dùng phương thức **MoveNext**, **MovePrevious**, **MoveFirst**, **MoveLast**
- Để kiểm tra vị trí của con trỏ mẫu tin hiện hành là trước mẫu tin đầu hay sau mẫu tin cuối trong Recordset, ta dùng các thuộc tính **BOF** hoặc **EOF** để kiểm tra.

Ví dụ sau minh họa việc thể hiện dữ liệu từ biến Recordset theo dạng tuần tự:

```
<%
strDSN = "Driver={Microsoft Access Driver (*.mdb)};DBQ=" &
server.mappath("Test.mdb")

'Create a connection object.
Set Conn = Server.CreateObject("ADODB.Connection")

'Instantiate a Recordset object.
Set rsCustomers = Server.CreateObject("ADODB.Recordset")

'Open a recordset using the Open method
'and use the connection established by the Connection object.
strSQL = "SELECT FirstName, LastName FROM Customers WHERE
LastName = 'Smith' "

rsCustomers.Open strSQL, Conn

i = 1
do while not rsCustomers.EOF
    Response.Write "Customer No" & i & ". "
    Response.Write rsCustomers("FirstName") & " " &
rsCustomers("LastName") & "<br>"
    i = i+1
    rsCustomers.MoveNext
loop
%>
```

Kết quả thể hiện sẽ có dạng:

Customer No1. Michael Platini  
Customer No2. Dino Zoff  
Customer No3. Ruddi Voller

## 5.4. Ngắt kết nối với cơ sở dữ liệu và giải phóng tài nguyên hệ thống đã dùng

Để ngắt kết nối với cơ sở dữ liệu, ta dùng phương thức **Close** có trong các đối tượng Connection và Recordset. Sau đó để giải phóng tài nguyên hệ thống đã dùng cho các đối tượng này, ta dùng lệnh gán giá trị Nothing cho các biến đối tượng này. Ví dụ:

```
<%
rsCustomers.Close
Set rsCustomers = Nothing
Conn.Close
Set Conn= Nothing
%>
```

Đoạn chương trình sau tổng hợp các vấn đề đã học liên quan đến việc kết nối với cơ sở dữ liệu dùng trong ứng dụng web

```
<%
' Build connection string
strDSN = "Driver={Microsoft Access Driver (*.mdb)};DBQ=" &
server.mappath("Test.mdb")

'Create a connection object.
Set Conn = Server.CreateObject("ADODB.Connection")

'Instantiate a Recordset object.
Set rsCustomers = Server.CreateObject("ADODB.Recordset")

'Open a recordset using the Open method
'and use the connection established by the Connection object.
strSQL = "SELECT FirstName, LastName FROM Customers WHERE
LastName = 'Smith' "

rsCustomers.Open strSQL, Conn

i = 1
do while not rsCustomers.EOF
    Response.Write "Customer No" & i & ".&nbsp;"
    Response.Write rsCustomers("FirstName") & "&nbsp;" &
rsCustomers("LastName") & "<br>"
    i = i +1
    rsCustomers.MoveNext
loop

' Free resource
```

```
rsCustomers.Close
Set rsCustomers = Nothing
Conn.Close
Set Conn= Nothing
```

```
%>
```

## 6. NHỮNG CÂU HỎI THƯỜNG GẶP

### 6.1. Làm thế nào để bắt đầu học ASP

Microsoft Active Server Pages (ASP) là môi trường lập trình ứng dụng phía server (server side scripting) hỗ trợ mạnh trong việc xây dựng các ứng dụng Web. Các ứng dụng ASP có thể làm việc với bất kỳ cơ sở dữ liệu nào tương thích với ODBC như SQL, Access, Oracle, Informix,... đồng thời rất dễ viết và sửa đổi. Hơn nữa nó có thể tích hợp các công nghệ sẵn có của Microsoft như COM, ... một cách dễ dàng.

Để có thể viết ứng dụng web bằng ASP, cần phải biết các kiến thức cơ bản sau:

- Kiến thức về thiết kế web, HTML để giúp thiết kế các trang web.
- Kiến thức về các ngôn ngữ lập trình VB script, Java script. VB Script là ngôn ngữ lập trình thông dụng cho ASP.
- Thông thường các ứng dụng web có liên quan nhiều đến việc quản lý, truy xuất, cập nhật cơ sở dữ liệu nên cần phải nắm thêm kiến thức về cú pháp các câu truy vấn SQL, kiến thức về kết nối và lập trình cơ sở dữ liệu với ADO.

Các tài liệu và website tham khảo:

- Thiết kế và Lập trình ứng dụng web bằng ASP – Lê Đình Duy – NXB Thống kê, 2001
- Xây dựng trang web động với ASP – Nhóm tác giả ELICOM - 2001
- ASP Databases – Nhóm tác giả SAIGONBOOK - 2001
- Professional Active Server Pages 3.0 – Alex Homer et al - 1999

- MSDN – Active Server Pages Tutorial
- <http://www.learnasp.com>
- <http://www.4guysfromrolla.com>
- <http://www.15seconds.com>
- <http://asp.superexpert.com>
- <http://www.aspfaqs.com>

## 6.2. Làm thế nào để ngăn cache trang ASP ?

Cache là kĩ thuật thường dùng để đảm bảo cho việc tiết kiệm thời gian và băng thông mạng. Về mặt bản chất đó là việc lưu trữ tạm thời lại các trang web mà bạn đã truy cập trên đĩa cứng và khi nhận được yêu cầu tương tự, hệ thống sẽ trả về trang lưu trữ này thay vì phải yêu cầu thực hiện lại, ví dụ như kết nối với webserver để trả về. Có 2 mức độ cache: cache tại trình duyệt và cache tại webserver.

Để ngăn không cho trình duyệt cache lại các trang ASP, bạn đặt đoạn mã sau vào đầu trang :

```
<%
  Response.ExpiresAbsolute = #2000-01-01#  ' có thể dùng một
ngày khác nhỏ hơn năm hiện hành.
  Response.AddHeader "pragma", "no-cache"
  Response.AddHeader "cache-control", "private, no-cache, must-
revalidate"
%>
```

Để ngăn không cho webserver IIS cache lại các trang ASP, bạn phải vào màn hình quản trị của IIS (Microsoft Management Console (MMC)) và bỏ mục (unchecked) "Cache ISAPI Applications".

## 6.3. Xử lí lỗi HTTP 500 - Internal server error

Lỗi HTTP 500 – Internal server error là thông báo mà bạn rất hay gặp khi trang ASP đang viết gấp lỗi. Tuy nhiên với thông báo chung chung như lỗi này, bạn rất khó có thể biết trang ASP mình sai chỗ nào để sửa:

```
The page cannot be displayed
There is a problem with the page you are trying to reach and it
cannot be displayed.
HTTP 500 - Internal server error
Internet Explorer
```

Thông thường thông báo lỗi này bắt nguồn từ trình duyệt IE 5.0 trở lên và cách cấu hình IIS. Bạn khắc phục như sau:

- Trong IE, bạn vào mục Tools/Internet Options, chọn Advanced tab, bỏ mục (unchecked) "Show friendly HTTP error messages."
- Trên IIS, bạn vào IIS Admin, trong mục Configuration/App Debugging, chọn "Send detailed ASP error messages to client".

## 6.4. Làm thế nào để trang ASP có thể lấy thông tin từ người dùng như địa chỉ IP, tên trình duyệt?

Thông thường, các thông tin liên quan đến hệ thống và người dùng đều được lưu trữ trong biến Request.ServerVariables, do đó, bạn có thể truy cập đến các thành phần trong này để lấy ra. Ví dụ, địa chỉ IP của người dùng là Request.ServerVariables ("REMOTE\_ADDR").

Đoạn mã sau liệt kê thông tin của các biến chứa trong Request.ServerVariables.

```
<table>
<%
for each x in Request.ServerVariables
  Response.Write("<tr><td>" & x & "</td><td>")
  Response.Write(Request.ServerVariables(x))
  Response.Write("</td></tr>" & vbCrLf)
next
%>
</table>
```

## 6.5. Làm thế nào để bảo vệ đoạn mã ASP

Ở phía người dùng thông thường sẽ không biết được nội dung của đoạn mã ASP vì khi truy cập một trang ASP, webserver đã thông dịch và xử lý các đoạn mã trong đó và người dùng chỉ thấy được kết quả trả về dưới dạng trang HTML. Chính vì vậy, cách duy nhất có thể xem được mã của trang ASP là bạn phải có quyền trên webserver chứa trang đó. Nếu bạn muốn tránh không muốn cho biết nội dung đoạn mã này luôn, thì có một trong hai cách sau:

- Viết ứng dụng của bạn dưới dạng các component .DLL. Đây là cách mà các nhà cung cấp dịch vụ cộng thêm hay làm ví dụ như component hỗ trợ tải tập tin lên server, component hỗ trợ gửi email, ... Hạn chế của cách làm này là bạn phải có quyền

cài đặt component trên server. Điều này sẽ không dễ dàng nếu bạn hosting trang web trên các máy chủ của người khác.

- Dùng chương trình mã hóa của Microsoft tên là Windows Script Encoder (tải về tại [http://msdn.microsoft.com/scripting/vbscript/download/x86/sce\\_10en.exe](http://msdn.microsoft.com/scripting/vbscript/download/x86/sce_10en.exe)). Chương trình này sẽ chuyển các đoạn mã ASP của bạn thành các kí tự “khó đọc”, ví dụ sau minh họa một trang web trước và sau khi mã hóa:

#### Trang web nguyên thủy:

```
<HTML>
<HEAD>
<TITLE>Page with secret information</TITLE>
<SCRIPT LANGUAGE="JScript">
<!--/
/**Start Encode**
 alert ("this code should be kept secret!!!!");
//-->
</SCRIPT>
</HEAD>
<BODY>
This page contains secret information.
</BODY>
</HTML>
```

#### Trang web sau khi được mã hóa bằng Windows Script Encoder:

```
<HTML>
<HEAD>
<TITLE>Page with secret information</TITLE>
<SCRIPT LANGUAGE="JScript.Encode">
<!--/
/**Start
Encode**#@~^QwAAAA==@#@&P~,|^+DDPvEY4kdP1W[n,/tK;V9
P4
~V+aY,/nm.nD"Z"eE#p@#@&&JOO@*#@&#&qhAAAA==^#~@&
It;/SCRIPT>
</HEAD>
<BODY>
This page contains secret information.
</BODY>
</HTML>
```

Cách làm này chỉ che mắt được những người không chuyên nghiệp bởi vì cách mã hóa này vẫn có thể bị phá, xem thêm chi tiết tại <http://www.klaphek.nl/nr6/scrdec.html>

# THỰC HÀNH

## 1. SỔ GÓP Ý

Trong các website, để tiếp thu ý kiến của khách ghé thăm, người ta thường tạo một sổ góp ý trên web, cho phép các chức năng sau:

- Cho phép người dùng nhập vào các thông tin họ tên, địa chỉ email, và nội dung góp ý.
- Cho phép người dùng xem toàn bộ các nội dung đã góp ý. Thứ tự các góp ý được sắp xếp theo thời gian.
- Cho phép quản trị hiệu chỉnh, xóa các góp ý.

## 2. GỬI THÔNG TIN MỚI CHO KHÁCH HÀNG

Một siêu thị muốn thông báo cho các khách hàng của mình mỗi khi có mặt hàng mới về. Hãy viết một ứng dụng web hỗ trợ công việc trên bằng các chức năng sau:

- Cho phép người dùng điền vào các thông tin của khách hàng như tên đăng nhập, mật khẩu, họ tên, địa chỉ liên lạc, điện thoại, địa chỉ email, ... và các chủng loại mặt hàng mà người dùng muốn thông tin khi có mặt hàng mới thuộc chủng loại này.
- Cho phép người dùng cập nhật lại các thông tin đã đăng ký. Để thực hiện được thao tác này, người dùng phải đăng nhập đúng với tên và mật khẩu đã đăng ký.
- Cho phép quản trị liệt kê danh sách các khách hàng đã đăng ký theo từng chủng loại mặt hàng.

## Bài tham khảo

# LẬP TRÌNH WEB ASP VỚI TIẾNG VIỆT UNICODE

## 1. Một số khái niệm căn bản về biểu diễn kí tự bên trong máy tính

### 1.1. Khái niệm về điểm mã, đơn vị mã, bảng mã

Về mặt bản chất, máy tính chỉ làm việc với các con số, do đó để biểu diễn các kí tự trên máy tính cần phải có một qui ước nhất quán giữa các kí tự cần biểu diễn và các con số tương ứng mà máy tính xử lí. Qui ước này được thể hiện qua các bước sau:

- Chọn tập các kí tự cần mã hóa (character set).
- Gán cho mỗi kí tự cần mã hóa một giá trị nguyên không âm, gọi là điểm mã (code point).
- Chuyển các điểm mã thành dãy các đơn vị mã (code units) để phục vụ cho việc lưu trữ và mã hóa. Một đơn vị mã là một đơn vị của bộ nhớ, có thể là 8, 16, hay 32 bit. Các điểm mã không nhất thiết phải có cùng số đơn vị mã.

Tập hợp những điểm mã của một tập các kí tự được gọi là một trang mã (code page) hay còn gọi là bảng mã hay bộ mã. Như vậy khi nói về một bảng mã, chúng ta quan tâm đến hai điều chính, số lượng các kí tự được mã hóa, và cách mã hóa chúng thành các đơn vị mã.

Lấy ví dụ bảng mã ASCII, tập kí tự cần mã hóa có 128 kí tự bao gồm các kí tự tiếng Anh, kí tự số, kí tự tiền tệ Anh, Mỹ và các kí tự điều khiển hệ thống ngoại vi. Các điểm mã có giá trị nằm trong khoảng từ 0-127. Mỗi điểm mã được mã hóa bằng đúng một đơn vị mã 8 bit, có nghĩa là đúng một byte.

Việc quyết định chọn cách mã hóa như thế nào sẽ quyết định số lượng kí tự được mã hóa. Ví dụ, nếu chọn cách mã hóa các điểm mã bằng đúng một đơn vị mã 8-bit thì số lượng điểm mã của một bảng mã (tạm gọi là bảng mã 8 bit) chỉ có thể tối đa là 256.

Do bảng mã ASCII không đủ để biểu diễn các kí tự của các ngôn ngữ khác, ví dụ như tiếng Việt, nên Microsoft đã nới rộng bảng mã ASCII bằng cách sử dụng 128 điểm mã có giá trị từ 128-255 để mã hóa cho các kí tự ngoài ASCII này. Tuy nhiên do chỉ có 128 điểm mã, trong khi số lượng các kí tự của các ngôn ngữ khác nhiều hơn, nên Microsoft đã tạo ra nhiều bảng mã khác nhau cho từng loại ngôn ngữ [1]. Ví dụ:

code page	1250	1251	1252	1253	1254	1258	etc.,
upper 128	Eastern Europe	Cyrillic	West Euro ANSI	Greek	Turkish	Vietnamese	etc.,
lower 128	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	etc.,

Tuy nhiên trong từng bảng mã này, không phải tất cả các kí tự của một ngôn ngữ đều có trong bảng mã. Hay nói chính xác hơn là không phải tất cả các kí tự đều được biểu diễn bằng duy nhất một điểm mã. Lấy ví dụ tiếng Việt chúng ta có 134 kí tự tổ hợp từ 28 chữ cái và 5 dấu thanh. Do chỉ có 128 điểm mã nên bảng mã windows-1258 dành cho tiếng Việt biểu diễn một số kí tự thành hai điểm mã liên tiếp, một điểm mã dành cho kí tự cơ sở và một điểm mã dành cho dấu thanh. Ví dụ: kí tự “é” được biểu diễn bằng hai điểm mã tương ứng với các kí tự ê và kí tự dấu sắc: é = ê + ’. Cách biểu diễn như vậy được gọi là cách biểu diễn tách rời (decomposed) mà thuật ngữ chúng ta hay gọi là tổ hợp.

Bảng mã TCVN3-ABC dùng 134 điểm mã để biểu diễn hết các kí tự tiếng Việt, chính điều này đã dẫn đến phải sử dụng một số điểm mã của bảng mã ASCII. Đây chính là lí do mà các trang web sử dụng bảng mã này không hiển thị được kí tự ư trong các trình duyệt Internet Explorer 5.0 trở lên. Cách biểu diễn như vậy được gọi là cách biểu diễn kết hợp sẵn (precomposed) mà thuật ngữ chúng ta hay gọi là dựng sẵn.

### 1.2. Bảng mã Unicode

Về mặt bản chất các bảng mã trên của Windows là bảng mã 8-bit, nghĩa là mỗi điểm mã được mã hóa bằng đúng một đơn vị mã 8-bit. Chính điều này đã giới hạn số lượng các kí tự được mã chỉ là 256. Do đó trong một văn bản không thể cùng hiển thị nhiều kí tự của các ngôn ngữ khác nhau được.

Unicode ra đời nhằm thống nhất chung các kí tự của mọi ngôn ngữ trong một bảng mã duy nhất [2]. Hai vấn đề nên lưu ý khi đề cập đến thuật ngữ Unicode đó là:

- Tập kí tự mà Unicode biểu diễn: ở đây muốn nói đến tập kí tự và cách ánh xạ các kí tự bằng các điểm mã tương ứng.
- Cách mã hóa các điểm mã thành các đơn vị mã.

Unicode dùng 16 bit để biểu diễn các điểm mã, do đó nó có thể biểu diễn được đến 65,536 kí tự có điểm mã nằm trong khoảng từ 0-65,535. Do vậy với Unicode người ta có thể biểu diễn được hầu hết các kí tự của các ngôn ngữ.

Cách đơn giản nhất để mã hóa các kí tự Unicode là biểu diễn mỗi điểm mã bằng đúng một đơn vị mã 16-bit. Đây chính là cách mã hóa nguyên thủy của Unicode trong phiên bản 2.0 được ISO/IEC chuẩn hóa thành ISO/IEC 10646 hay còn gọi là UCS-2. Tuy nhiên, để tương thích với các hệ thống xử lí trước kia Unicode ra đời cũng như tối ưu hóa trong quá trình lưu trữ và truyền dữ liệu, người ta dùng các cách khác nhau để mã hóa các điểm mã thành các đơn vị mã. Mỗi cách mã hóa như vậy được gọi là một dạng biến đổi của Unicode (UTF – Unicode Transformation Format). Thông dụng nhất hiện nay là UTF-8 và UTF-16 dùng dãy các đơn vị mã có độ dài khác nhau để mã hóa các điểm mã. UTF-8 dùng 1 đến 4 đơn vị mã 8-bit trong khi UTF-16 dùng 1 đến 2 đơn vị mã 16-bit để mã hóa. Ví dụ sau minh họa cách mã hóa của UTF-8:

- 128 kí tự đầu tiên của Unicode từ điểm mã U+0000 đến U+007F, được mã hóa thành 1 byte.
- Từ điểm mã U+0080 đến U+07FF, được mã hóa thành 2 byte.
- Từ điểm mã U+0800 đến U+FFFF, được mã hóa thành 3 byte.
- Từ điểm mã U+0800 đến U+FFFF, được mã hóa thành 4 byte.

Như vậy khi đề cập đến Unicode trong lập trình, cần phải xác định rõ chúng ta dùng bảng mã Unicode theo dạng biến đổi nào: UCS-2, UTF-8, hay UTF-16, ... UCS-2 được dùng trong các hệ quản trị cơ sở dữ liệu như SQL Server 7.0/2000, Microsoft Access 2000, UTF-8 thường được dùng trong các ứng dụng web, trong khi UTF-

16 lại được dùng trong các hệ thống như Windows 2000/XP, Java, ...

## 2. Lập trình web với tiếng Việt Unicode

### 2.1. Chỉ định bảng mã dùng trong trang web

Khi một trang web được server chuyển xuống cho client, trình duyệt sẽ dùng thông tin về bảng mã mà trang web đó sử dụng để chuyển dãy các byte trong tài liệu đó thành các kí tự tương ứng để hiển thị lên màn hình. Ngoài ra, một khi dữ liệu trong các FORM được gửi đi sau khi người dùng submit, trình duyệt cũng sẽ cẩn cứ vào bảng mã này để chuyển đổi dữ liệu khi truyền đi. Ví dụ, nếu trang web được chỉ định dùng bảng mã windows-1252 thì khi FORM được submit, dữ liệu sẽ được mã hóa theo bảng mã này cho dù trước đó trong các hộp điều khiển của FORM, dữ liệu được gõ dưới dạng Unicode [3].

Việc chỉ định bảng mã có vai trò rất quan trọng trong việc hiển thị đúng nội dung mà người thiết kế mong muốn, bởi vì nếu không chỉ định bảng mã được dùng trong trang web hiện hành một cách rõ ràng, trình duyệt sẽ sử dụng bảng mã mặc định. Ví dụ, nếu dữ liệu chuyển đến cho trang web là E1 BB 81, nếu chỉ định bảng mã là UTF-8 thì 3 byte này chính là biểu diễn mã của kí tự “è” trong khi nếu hệ thống dùng bảng mã mặc định, ví dụ như windows-1252, thì 3 byte này lại được xem như là biểu diễn 3 kí tự khác nhau và sẽ được hiển thị là “á»»”.

Để chỉ định bảng mã mà trang web hiện hành sử dụng, ta dùng tag META với thuộc tính HTTP-EQUIV được gán là Content-Type, và chỉ định tên của bảng mã được dùng trong thuộc tính CONTENT (Thông tin về các bảng mã được dùng trên Windows có thể xem tại [4]). Trong ví dụ sau, tag META được dùng để chỉ định bảng mã windows-1252 cho một trang web:

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
CHARSET=windows-1252">
```

Để yêu cầu trình duyệt sử dụng một bảng mã cho toàn bộ trang web, ta phải đặt tag META này trước tag BODY. Thông thường là đặt tag META này trong tag HEAD như ví dụ sau:

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
CHARSET=windows-1252">
```

```
<TITLE>New Page 1</TITLE>
</HEAD>
<BODY>
...
</BODY>
</HTML>
```

Trong trường hợp bảng mã được chỉ định không có khả năng biểu diễn được tất cả các kí tự của trang web, người ta phải dùng đến số tham chiếu của kí tự (NCRs - numerical character references). Số tham chiếu kí tự là điểm mã của kí tự Unicode tương ứng mà nó biểu diễn. Số tham chiếu kí tự có hai dạng thập phân và thập lục phân. Dạng thập phân có cú pháp là “#D;”, với D là số thập phân. Dạng thập lục phân có cú pháp là “#H;”, với H là số thập lục phân. Ví dụ: &#229; và &#xE5; là các số tham chiếu của kí tự “a” trong bảng mã Unicode. Một khi gặp số tham chiếu của kí tự, trình duyệt sẽ tham chiếu trực tiếp đến kí tự có điểm mã tương ứng trong bảng mã Unicode mà không sử dụng đến bảng mã được chỉ định hiện hành [5].

Lấy ví dụ một trang web được mã hóa với bảng mã windows-1252, lúc đó để hiển thị đoạn văn bản: “Tiếng Việt”, dữ liệu cho trang web phải là “Ti&#7871;ng Vi&#7879;t”, trong đó &#7871; và &#7879; lần lượt là các số tham chiếu của các kí tự “é” và “è” trong bảng mã Unicode.

Điều này cho phép giải thích tại sao, các trang web không dùng bảng mã UTF-8, ví dụ như windows-1252, vẫn có thể hiển thị được các kí tự Unicode không thuộc bảng mã đó hay khi chuyển đổi từ bảng mã UTF-8 sang windows-1252, MS FrontPage 2000 lại tự động thêm vào các số tham chiếu kí tự theo cách trên.

## 2.2. Hoạt động của webserver

Khi trình duyệt yêu cầu một trang .asp, trình xử lí trang asp tại webserver sẽ thông dịch các mã lệnh ở trong trang web này và gửi kết quả về cho trình duyệt. Thông thường, lệnh Response.Write được dùng cho các kết xuất từ các hằng chuỗi hay từ các biến ra màn hình. Ví dụ như:

```
...
<%
    Response.Write "Chào mừng bạn đến với trang web này"
    'in một hằng chuỗi
    Response.Write rs("TEN_NV") 'in dữ liệu của một biến, ví
    dụ như là một trường của recordset
```

```
%>
```

...

Để yêu cầu webserver mã hóa các dữ liệu trong các hằng chuỗi và biến theo bảng mã sẽ được dùng để hiển thị tại client, ta cần phải đặt thuộc tính CodePage về bảng mã tương ứng. Các lệnh trong ví dụ sau sẽ yêu cầu webserver mã hóa các chuỗi dữ liệu theo bảng mã UTF-8 (Thông tin về các codepage tương ứng với các bảng mã xem tại [4]):

```
<%Session.CodePage=65001%> // Dùng cho toàn bộ các trang
trong Session hiện hành
<%@CodePage=65001%> // Dùng cho trang hiện hành
```

Lấy ví dụ trong trường hợp dùng cơ sở dữ liệu SQL Server 7.0, dữ liệu được trả về từ các câu truy vấn theo bảng mã UCS-2. Nếu ta chỉ định CodePage là 65001, webserver sẽ tự động chuyển dữ liệu từ UCS-2 sang UTF-8, ngược lại nếu không chỉ định thuộc tính CodePage, webserver sẽ chuyển dữ liệu đó đến client theo bảng mã mặc định (ví dụ như windows-1252). Điều này giải thích cho trường hợp một số trang web asp hiển thị không đúng dữ liệu Unicode được lưu trong các cơ sở dữ liệu như SQL Server 7.0/2000, MS Access 2000.

Ngoài ra, các trang asp có sử dụng đoạn mã lệnh thiết lập CodePage là 65001 phải được lưu theo định dạng tương ứng là UTF-8 [6].

Như vậy, việc thiết lập thuộc tính CodePage trong trang asp sẽ giúp cho webserver hiểu được các dữ liệu được lưu trong các cơ sở dữ liệu, hằng chuỗi kí tự, ... theo bảng mã nào để mã hóa (encode) nó trước khi chuyển đến cho trình duyệt. Việc chỉ định bảng mã dùng trong trang web bằng tag META sẽ giúp cho trình duyệt diễn dịch (decode) dữ liệu được chuyển đến từ webserver đúng nhất khi hiển thị [7].

## 2.3. Các bước cơ bản của lập trình web asp sử dụng tiếng Việt Unicode

- Soạn và lưu trữ tập tin .asp dưới dạng mã hóa UTF-8.
- Trong các tập tin asp, chèn các đoạn mã chỉ định cho web server và trình duyệt xử lí dữ liệu trong trang web như là UTF-8. Các đoạn mã này phải đặt ở đầu trang asp. Sử dụng ví dụ mẫu sau:

```
<%@CODEPAGE=65001%>
<%Session.Codepage=65001%>
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;
CHARSET=UTF-8">
</HEAD>
<BODY>
</BODY>
</HTML>
```

- Sử dụng các hệ quản trị CSDL hỗ trợ Unicode như SQLServer 7.0/2000, MS Access 2000. Nếu dùng SQL Server thì phải khai báo kiểu dữ liệu cho các trường lưu dữ liệu Unicode là NCHAR, NVARCHAR, NTEXT, ... Các kiểu dữ liệu như TEXT, MEMO, HYPERLINK trong MS Access 2000 mặc định là hỗ trợ lưu dữ liệu Unicode.
- Truy xuất cơ sở dữ liệu thông qua JScript/VBScript/ODBC.
- Khi làm việc trên hệ quản trị CSDL SQL Server 7.0/2000, nếu dùng các hằng chuỗi trong các câu lệnh SQL, phải thêm tiếp đầu ngữ N (bắt buộc là chữ in hoa) vào [8]. Nếu không sử dụng tiếp đầu ngữ này, SQL Server sẽ tự động chuyển chuỗi dữ liệu sang bảng mã mặc định hiện hành trước khi sử dụng nó trong các thao tác cập nhật CSDL. Ví dụ, nếu bạn dùng câu lệnh sau: INSERT INTO SINHVIEN(TEN\_SV) VALUES('Trần Nam Hải') thì hằng chuỗi dữ liệu 'Trần Nam Hải' sẽ được SQL Server xem như là chuỗi kí tự thường chứ không phải là chuỗi Unicode. Điều này sẽ dẫn đến hậu quả là dữ liệu sẽ được lưu trữ không chính xác. Ví dụ như dữ liệu của kí tự "à" trong chuỗi trên là E1 BA A7, sẽ được lưu thành 3 kí tự khác nhau. Trong khi đó nếu dùng câu lệnh INSERT INTO SINHVIEN(TEN\_SV) VALUES(N'Trần Nam Hải') thì 3 byte E1 BA A7 sẽ được xem như là một kí tự khi lưu xuống [9].

### 3. Hỗ trợ Unicode của các phần mềm

#### 3.1. Các phần mềm hỗ trợ soạn thảo trang web

- Visual Studio.NET, Notepad, MS FrontPage2002: Hỗ trợ lưu tập tin dưới dạng UTF-8
- Visual InterDev 6.0: Nếu trong trang asp ta sử dụng các hằng chuỗi được gõ vào dưới dạng Unicode, ví dụ như:

Response.Write "Chào mừng bạn " thì lúc lưu tập tin, chương trình sẽ phát hiện ra trong trang asp này có xuất hiện kí tự Unicode và yêu cầu lưu xuống dưới dạng Unicode, nếu không các kí tự Unicode sẽ bị mất. Tuy nhiên, nếu chọn lưu dưới dạng Unicode thì chương trình sẽ lưu tập tin này dưới dạng mã hóa UCS-2. Hiện nay webserver IIS không thể xử lý được trang asp này [10]. Do đó không nên dùng Visual InterDev 6.0 để soạn thảo các trang asp trong các ứng dụng Unicode tiếng Việt.

- Các phần mềm thông dụng hỗ trợ gõ tiếng Việt Unicode: UniKey, VietKey.

#### 3.2. Các phần mềm hệ thống khác

- SQL Server 7.0/2000 và MS Access 2000 hỗ trợ Unicode. Với mỗi kí tự Unicode, hệ thống sẽ sử dụng bảng mã UCS-2 để lưu trữ, nghĩa là dùng cố định 2 byte cho một kí tự. SQL 6.5 và MS Access 97 không hỗ trợ Unicode.
- IIS 5.0 không thể đọc được các tập tin lưu dưới dạng UCS-2 [10], không hỗ trợ CodePage của bảng mã UTF-16 là 1200 [11]. IIS 4.0 không hỗ trợ CodePage của bảng mã UTF-8 là 65001 [11].

### 4. Sử dụng Unicode trong các trang web hiện nay

Có hai cách có thể hiển thị được kí tự theo bảng mã Unicode trong trình duyệt.

- Hiển thị theo cách lưu trữ Unicode
- Hiển thị theo qui ước Unicode (tạm gọi là vậy)

Các trang web hiển thị đúng theo cách lưu trữ unicode là các trang web khi View Source lên, bạn sẽ thấy có dòng định nghĩa charset=utf-8 hoặc trình duyệt sẽ tự động dò ra nếu tập tin đó được lưu theo utf-8. Với các trang web này, khi bạn thay đổi bảng mã khi view trong trình duyệt (bằng cách chọn View-Encoding/..Windows), hầu như bạn sẽ thấy trang web trở nên hoàn toàn xa lạ

Các trang web hiển thị theo qui ước Unicode là các trang web khi view lên sẽ không có dòng định nghĩa charset=utf-8 mà thường là charset=windows-1252 và trong nội dung trang web bạn sẽ thấy rất nhiều kí tự &#. Trang web của IS-EDU là trang web dạng này.

Về mặt lí thuyết nếu charset là windows-1252 nghĩa là kí tự sẽ được hiển thị theo bảng mã 1252. Mà trong bảng mã 1252 chắc chắn không thể có các kí tự tiếng Việt được vì nó là bảng mã của Western European, thế nhưng tại sao ta vẫn thấy được tiếng Việt.

Lí do là ở chỗ, W3C đã qui ước về trang HTML, nếu trong 1 trang web mà bạn muốn tham chiếu đến một kí tự theo qui ước của bảng mã Unicode, bạn sẽ gõ trực tiếp vào &#<điểm mã>. Chính vì vậy mà trong một trang web có charset là 1252, muốn hiển thị được chữ "Tiếng Việt" bạn phải gõ vào Ti&#<điểm mã kí tự ê>ng Vi&#<điểm mã kí tự è>t.

Bạn có thể không biết điều này vì khi soạn một trang web tiếng Việt trong frontpage bằng các bộ gõ như Vietkey, Unicode, Frontpage tự động điền các điểm mã vào cho bạn.

Nói tóm lại: Khi xem một trang web có phải thực sự là unicode hay không, bạn chỉ cần View Source lên, nếu charset=utf-8 thì đó thực sự là dùng unicode thật theo nghĩa vừa là dùng bảng mã unicode và cách lưu trữ unicode.

Còn ngược lại, nếu bạn nhìn thấy charset=windows-1252 như trang IS-EDU này thì nó mới chỉ là dùng bảng mã unicode mà thôi, còn cách lưu trữ vẫn là windows-1252.

Có rất nhiều trang web tiếng Việt hiện nay gọi là dùng Unicode nhưng thực chất là Unicode theo dạng windows-1252 kể trên.

## 5. Kết luận

Unicode ra đời nhằm khắc phục hạn chế về số lượng kí tự được mã hóa của các bảng mã 8-bit trước đó, cho phép mọi ngôn ngữ có thể sử dụng chung một bảng mã duy nhất. Do vấn đề tương thích trong lưu trữ và truyền dữ liệu mà Unicode có các dạng mã hóa khác nhau như UCS-2, UTF-8, UTF-16. UTF-8 là dạng mã hóa Unicode thông dụng nhất trong các ứng dụng web hiện nay.

Để viết các ứng dụng web dùng tiếng Việt Unicode, cần chọn các phần mềm soạn thảo hỗ trợ lưu trữ tập tin dưới dạng mã hóa UTF-8 như Visual Studio.NET, MS FrontPage2000, NotePad, ... ; sử dụng các hệ quản trị CSDL hỗ trợ Unicode như SQL Server 7.0/2000, MS Access 2000, ...; đặt các đoạn mã chỉ định bảng mã mà webserver và trình duyệt dùng để mã hóa và giải mã dữ liệu.

## 6. Tài liệu trích dẫn

1. <http://www.microsoft.com/globaldev/articles/unicode.asp>
2. <http://www.unicode.org/unicode/standard/principles.html>
3. <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q303612>
4. <http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/charsets/charset4.asp>
5. <http://www.w3.org/TR/html401/charset.html#code-position>
6. <http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q295063&>
7. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/iisref/html/psdk/asp/vbob150i.asp>
8. <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q239530>
9. <http://support.microsoft.com/default.aspx?scid=kb;en-us;q232580>
10. <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q245000>
11. <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q245000>
12. [http://support.microsoft.com/default.aspx?scid=kb;\[LN\];Q254313](http://support.microsoft.com/default.aspx?scid=kb;[LN];Q254313)

## Bài tham khảo

# KẾT XUẤT DỮ LIỆU RA DẠNG EXCEL TỪ TRANG ASP

Thông thường, các ứng dụng web kết xuất dữ liệu của các báo cáo ra dưới dạng bảng biểu. Sẽ rất cần thiết nếu người dùng cũng nhận được dữ liệu này dưới dạng Excel để có thể sử dụng cho các mục đích khác.

Kỹ thuật để đạt được mục đích này khá đơn giản. Ý tưởng chính của kỹ thuật này là sử dụng thuộc tính ContentType của đối tượng Response trong ASP và thực hiện theo các bước tuần tự sau:

### Bước 1: Chỉ định dữ liệu sẽ được chuyển đi theo định dạng Excel

Đơn giản chỉ cần dùng câu lệnh: Response.ContentType = "application/vnd.ms-excel". Thông thường, cần phải đặt câu lệnh Response.Buffer = True và Response.Clear trước câu lệnh này để đảm bảo dữ liệu được chuyển xuống client chính xác.

### Bước 2: Kết xuất dữ liệu dưới dạng bảng theo cách làm thông thường.

Dòng đầu tiên của bảng chứa tên của các cột sẽ được hiển thị trong tập tin excel.

Hãy xem ví dụ minh họa sau:

```
<%
    Response.Buffer = True
    Response.Clear
    ' thiết lập định dạng sẽ kết xuất là Excel
    Response.ContentType = "application/vnd.ms-excel"

    ' kết nối với CSDL
    strDSN = "DRIVER=Microsoft Access Driver (*.mdb);DBQ="
    ' đường dẫn tương đối đến tập tin CSDL
    strDSN = strDSN & Server.MapPath("myDB.mdb")
    set Conn = Server.CreateObject("ADODB.Connection")
    Conn.Open strDSN
    strSQL = "SELECT * FROM myTable"
    set rs = Conn.Execute(strSQL)
```

```
    rowstart = "<tr>"
    rowend = "</tr>"
    cellstart = "<td>"
    cellend = "</td>"
    Response.Write "<TABLE border=1>" 
    ' dòng đầu tiên in tên các cột
    Response.Write rowstart
    For i = 0 to rs.Fields.Count -1
        Response.Write cellstart & "<b>" & rs.Fields(i).name &
        "</b>" & cellend
    Next
    Response.Write rowend
    ' in dữ liệu của từng dòng
    Do while not rs.EOF
        Response.Write rowstart
        For i = 0 to rs.Fields.Count -1
            Response.Write cellstart & rs.Fields(i)& cellend
        Next
        Response.Write rowend
        rs.MoveNext
    Loop
    rs.Close
    set rs = Nothing
    Conn.Close
    set Conn = nothing
    Response.Write "</TABLE>" 
    Response.End
%>
```

### Bàn luận:

Ưu điểm của cách làm trên là cho phép bạn sử dụng các đoạn mã sẵn có lúc kết xuất dữ liệu ra dạng bảng theo cách thông thường để chuyển sang định dạng Excel. Tuy nhiên việc kết xuất dữ liệu theo định dạng Excel như trong ví dụ trên có thể chiếm tài nguyên của webserver đặc biệt khi dữ liệu lớn do đó chỉ nên dùng cách này nếu trang này không được sử dụng thường xuyên.

Nếu muốn kết xuất dữ liệu lớn và thực hiện thường xuyên, ta có thể kết xuất thông qua định dạng CSV (Comma-Separated Values) để tối ưu hơn. Các tập tin theo định dạng CSV là các tập tin văn bản mà dữ liệu trong các cột được ngăn cách với nhau bởi dấu phẩy "," (comma), rất thường được dùng cho việc trao đổi dữ liệu giữa các hệ quản trị CSDL và các chương trình bảng tính như Excel. Ví dụ, nếu bạn mở một tập tin csv có nội dung như sau trong Excel, ta sẽ nhận được một bảng 3 dòng, 3 cột:

Doe,John,944-7077

Johnson,Mary,370-3920

Smith,Abigail,299-3958

Để chuyển dữ liệu sang định dạng CSV, vẫn với cách làm tương tự bằng cách thay đổi thuộc tính ContentType về dạng "application/csv" và thêm dòng lệnh sau để yêu cầu trình duyệt hiển thị hộp thoại tải tập tin về: Response.AddHeader "Content-Disposition", "filename=mydata.csv;". Sau đó, thay vì định dạng dữ liệu dưới dạng bảng, ta định dạng dữ liệu theo dạng dữ liệu các cột được phân cách với nhau bằng dấu phẩy ",". Xem ví dụ minh họa sau:

```
<%
    Response.Buffer = True
    Response.Clear
    ' thiết lập định dạng sẽ kết xuất là Excel
    Response.ContentType = "application/csv"
    Response.AddHeader "Content-Disposition",
    "filename=mydata.csv;"

    ' kết nối với CSDL
    strDSN = "DRIVER=Microsoft Access Driver (*.mdb);DBQ="
    ' đường dẫn tương đối đến tập tin CSDL
    strDSN = strDSN & Server.MapPath("myDB.mdb")
    set Conn = Server.CreateObject("ADODB.Connection")
    Conn.Open strDSN
    strSQL = "SELECT * FROM myTable"
    set rs = Conn.Execute(strSQL)
    ' dòng đầu tiên in tên các cột
    For i = 0 to rs.Fields.Count -1
        Response.Write rs.Fields(i).name & ","
    Next
    Response.Write vbNewLine ' kết thúc một dòng dữ liệu
    ' in dữ liệu của từng dòng
    Do while not rs.EOF
        For i = 0 to rs.Fields.Count -1
            Response.Write rs.Fields(i)& ","
        Next
        Response.Write vbNewLine ' kết thúc một dòng dữ liệu
        rs.MoveNext
    Loop
    rs.Close
    set rs = Nothing
    Conn.Close
%>
```

```
set Conn = nothing
Response.End
%>
```

Hạn chế của định dạng này là do dữ liệu chỉ là các kí tự ASCII nên sẽ không dùng được trong trường hợp dữ liệu sử dụng Unicode, ngoài ra cần phải có cách xử lí thích hợp trong trường hợp dữ liệu trong các cột có dấu phẩy ",". Excel xử lí trường hợp này bằng cách thay đổi toàn bộ dữ liệu trong dấu "". Ví dụ nếu bạn có dữ liệu là **Abc, xyz** thì dữ liệu sẽ được đổi thành "**Abc, xyz**"

## Chương 6

# ASP NÂNG CAO

## 1. ĐỐI TƯỢNG RESPONSE

### 1.1. Chuyển hướng trang web

Để yêu cầu trình duyệt chuyển sang nạp một web khác khi đang xử lí trang web hiện hành, ta dùng phương thức **Response.Redirect** với tham số là một địa chỉ URL.

Khi sử dụng phương thức này tắt cả những nội dung đã được kết xuất ra trình duyệt trước đó của trang web hiện hành đều bị bỏ qua.

Ví dụ sau minh họa việc chuyển hướng trang tùy theo yêu cầu của người dùng

```
<HTML>
<HEAD>
<TITLE>Page Redirect</TITLE>
</HEAD>
<BODY>
<%
Site = Request.QueryString("SiteName")
Select case Site
    case "Microsoft"
        Response.Redirect "http://www.microsoft.com"
    case "Intel"
        Response.Redirect "http://www.intel.com"
    case "Oracle"
        Response.Redirect "http://www.oracle.com"
End Select
%>
</BODY>
</HTML>
```

### 1.2. Kiểm soát việc kết xuất ra trình duyệt

Thuộc tính **Response.Buffer** cho phép yêu cầu server kiểm soát quá trình gửi kết quả về client hay không trong quá trình xử lí các đoạn mã trong trang ASP. Nếu thuộc tính này là **FALSE**, điều đó có

nghĩa là server sẽ gửi kết quả về client ngay mỗi khi xử lí xong từng dòng lệnh. Ngược lại, nếu thuộc tính này là **TRUE**, server sẽ không gửi các kết quả về client ngay mà gửi chúng vào vùng đệm cho đến khi **tắt cả** các đoạn mã trong trang asp hiện hành được xử lí xong, sau đó kết quả từ vùng đệm mới được chuyển đến client.

Việc thiết lập giá trị của thuộc tính này bắt này bắt buộc phải được thực hiện trước khi server gửi bất kì kết quả nào về client. Do đó, dòng lệnh **<%Response.Buffer=TRUE/FALSE%>** phải được đặt ngay từ dòng đầu tiên của trang ASP.

Tuy nhiên, trong trường hợp tập tin ASP quá dài, nếu để xử lí xong hết đoạn mã mới kết xuất ra client, sẽ kéo dài thời gian chờ đợi hiển thị trang web của người dùng. Phương thức **Response.Flush** cho phép ta chuyển các kết quả đang có trong vùng đệm đến client ngay lập tức. Lưu ý rằng lệnh này chỉ được dùng khi đã đặt **<%Response.Buffer=TRUE%>**.

Để yêu cầu server ngừng xử lí các đoạn mã tiếp theo và chuyển các kết quả đang có (có thể còn nằm trong vùng đệm) đến client, ta dùng phương thức **Response.End**. Phương thức này thường dùng trong các cấu trúc điều khiển như **If...Then...Else**. Có thể xem **Response.End** tương tự như lệnh **halt()** trong C.

Để xóa các kết xuất hiện hành trong vùng đệm, ta dùng phương thức **Response.Clear**. Lưu ý rằng lệnh này chỉ được dùng khi đã đặt **<%Response.Buffer=TRUE%>**.

Để chỉ định thời gian lưu tạm (cache) một trang tại trình duyệt hết hạn (expires) là bao lâu, ta đặt giá trị cho thuộc tính **Response.Expires**. Nếu đặt **<%Response.Expires=-1%>**, trang web này sẽ hết hạn lưu tạm ngay lập tức.

### 1.3. Kiểm tra client còn kết nối với máy chủ không

Để kiểm tra client còn kết nối với webserver hay không, ta dùng thuộc tính **Response.IsClientConnected**

Ví dụ sau minh họa các thao tác tương ứng với các trạng thái của client

```
<%
'Check to see if the client is connected.
If Not Response.IsClientConnected Then
    'Get the sessionid to send to the shutdown function.
    Shutdownid = Session.SessionID
```

```
'Perform shutdown processing.
Shutdown(Shutdownid)
End If
%>
```

## 2. ĐỔI TƯỢNG REQUEST

### 2.1. Request.Form và Request.QueryString

**Request.Form** được dùng cho trường hợp ta muốn lấy giá trị của các biến kết hợp với các control của form được gửi đi bằng method **POST**. Đối với trường hợp một control chứa nhiều giá trị, chúng ta phải chỉ định thứ tự (index) của giá trị cần lấy. Để biết số lượng các giá trị được gửi theo một control, ta tham chiếu đến thuộc tính Count. Ví dụ:

```
<%
For i = 1 to Request.Form("FavoriteFlavor").Count
%>
Request.Form(FavoriteFlavor) =
<%= Request.Form("FavoriteFlavor")(i)%> <BR>
<% Next %>
```

**Request.QueryString** được dùng cho trường hợp ta muốn lấy giá trị các biến được tạo ra trong **HTTP query string**. Các cặp tên biến và giá trị theo sau dấu ? trong URL sẽ hình thành một HTTP query string. Trong ví dụ sau:  
[string sample](example.asp?FirstName=Duy&LastName=LeDinh), HTTP query string là "FirstName=Duy&LastName=Le Dinh". HTTP query string sẽ tạo ra hai biến có tên là FirstName và LastName, các giá trị tương ứng là "Duy" và "Le Dinh". Để lấy các giá trị này, ta dùng Request.QueryString("FirstName") và Request.QueryString("LastName").

Có hai cách để tạo ra HTTP query string, đó là:

- Người dùng submit một form có method là GET, lúc này trình duyệt sẽ tự động sinh ra HTTP query string.
- Người dùng gõ trực tiếp chuỗi HTTP query string trong URL như ví dụ trên.

So sánh các method GET và POST trong form:

- Method **GET**: số lượng kí tự bị giới hạn (kéo theo kích thước dữ liệu của biến không thể lớn), form sẽ hiển thị các tham số

của nó dưới dạng HTTP query string ngay trong ô địa chỉ của trình duyệt.

- Method **POST**: số lượng kí tự không bị giới hạn, form sẽ không hiển thị các tham số trong ô địa chỉ của trình duyệt.

Chúng ta cũng có thể truy xuất vào giá trị của các biến thông qua đối tượng **Request(<tên biến>)** không cần phải chỉ định collection là **Request.QueryString(<tên biến>)** hay **Request.Form(<tên biến>)**. Nghĩa là, nếu bạn có một biến có tên là "PHONE" được form submit bằng method GET và một biến có tên là "ADDRESS" được form submit bằng method POST, bạn đều có thể truy cập vào cả hai biến này bằng **Request("PHONE")** và **Request("ADDRESS")** thay cho **Request.QueryString("PHONE")** và **Request.Form("ADDRESS")**.

### 2.2. Request.ServerVariables

Để truy cập vào một số giá trị của các biến môi trường tại server, ta dùng collection **Request.ServerVariables**. Bảng sau liệt kê danh sách các tên biến môi trường tại server.

Variable	Description
ALL_HTTP	All HTTP headers sent by the client.
ALL_RAW	Retrieves all headers in raw form. The difference between ALL_RAW and ALL_HTTP is that ALL_HTTP places an HTTP_ prefix before the header name and the header name is always capitalized. In ALL_RAW the header name and values appear as they are sent by the client.
APPL_MD_PATH	Retrieves the metabase path for the Application for the ISAPI DLL.

APPL_PHYSICAL_PATH	Retrieves the physical path corresponding to the metabase path. IIS converts the APPL_MD_PATH to the physical (directory) path to return this value.
AUTH_PASSWORD	The value entered in the client's authentication dialog. This variable is available only if Basic authentication is used.
AUTH_TYPE	The authentication method that the server uses to validate users when they attempt to access a protected script.
AUTH_USER	Raw authenticated user name.
CERT_COOKIE	Unique ID for client certificate, returned as a string. Can be used as a signature for the whole client certificate.
CERT_FLAGS	bit0 is set to 1 if the client certificate is present.  bit1 is set to 1 if the certification authority of the client certificate is invalid (it is not in the list of recognized CAs on the server).
CERT_ISSUER	Issuer field of the client certificate (O=MS, OU=IAS, CN=user name, C=USA).
CERT_KEYSIZE	Number of bits in Secure Sockets Layer connection key size. For example, 128.

CERT_SECRETKEYSIZE	Number of bits in server certificate private key. For example, 1024.
CERT_SERIALNUMBER	Serial number field of the client certificate.
CERT_SERVER_ISSUER	Issuer field of the server certificate.
CERT_SERVER_SUBJECT	Subject field of the server certificate.
CERT_SUBJECT	Subject field of the client certificate.
CONTENT_LENGTH	The length of the content as given by the client.
CONTENT_TYPE	The data type of the content. Used with queries that have attached information, such as the HTTP queries GET, POST, and PUT.
GATEWAY_INTERFACE	The revision of the CGI specification used by the server. The format is CGI/revision.
HTTP_<HeaderName>	The value stored in the header <i>HeaderName</i> . Any header other than those listed in this table must be prefixed by HTTP_ in order for the <b>ServerVariables</b> collection to retrieve its value. <b>Note</b> The server interprets any underscore (_) characters in <i>HeaderName</i> as dashes in the actual header. For example if you specify HTTP_MY_HEADER, the server searches for a header sent as MY-HEADER.

HTTP_ACCEPT	Returns the value of the Accept header.
HTTP_ACCEPT_LANGUAGE	Returns a string describing the language to use for displaying content.
HTTP_USER_AGENT	Returns a string describing the browser that sent the request.
HTTP_COOKIE	Returns the cookie string that was included with the request.
HTTP_REFERER	Returns a string containing the URL of the page that referred the request to the current page, but does not include redirect requests.
HTTPS	Returns ON if the request came in through secure channel (SSL) or it returns OFF if the request is for a non-secure channel.
HTTPS_KEYSIZE	Number of bits in Secure Sockets Layer connection key size. For example, 128.
HTTPS_SECRETKEYSIZE	Number of bits in server certificate private key. For example, 1024.
HTTPS_SERVER_ISSUER	Issuer field of the server certificate.
HTTPS_SERVER_SUBJECT	Subject field of the server certificate.

INSTANCE_ID	The ID for the IIS instance in textual format. If the instance ID is 1, it appears as a string. You can use this variable to retrieve the ID of the Web-server instance (in the metabase) to which the request belongs.
INSTANCE_META_PATH	The metabase path for the instance of IIS that responds to the request.
LOCAL_ADDR	Returns the Server Address on which the request came in. This is important on multihomed computers where there can be multiple IP addresses bound to the computer and you want to find out which address the request used.
LOGON_USER	The Windows account that the user is logged into.
PATH_INFO	Extra path information as given by the client. You can access scripts by using their virtual path and the PATH_INFO server variable. If this information comes from a URL, it is decoded by the server before it is passed to the CGI script.
PATH_TRANSLATED	A translated version of PATH_INFO that takes the path and performs any necessary virtual-to-physical mapping.
QUERY_STRING	Query information stored in the string following the question mark (?) in the HTTP request.

REMOTE_ADDR	The IP address of the remote host making the request.
REMOTE_HOST	The name of the host making the request. If the server does not have this information, it will set REMOTE_ADDR and leave this empty.
REMOTE_USER	Unmapped user-name string sent in by the user. This is the name that is really sent by the user, as opposed to the names that are modified by any authentication filter installed on the server.
REQUEST_METHOD	The method used to make the request. For HTTP, this is GET, HEAD, POST, and so on.
SCRIPT_NAME	A virtual path to the script being executed. This is used for self-referencing URLs.
SERVER_NAME	The server's host name, DNS alias, or IP address as it would appear in self-referencing URLs.
SERVER_PORT	The port number to which the request was sent.
SERVER_PORT_SECURE	A string that contains either 0 or 1. If the request is being handled on the secure port, then this will be 1. Otherwise, it will be 0.
SERVER_PROTOCOL	The name and revision of the request information protocol. The format is <i>protocol/revision</i> .

SERVER_SOFTWARE	The name and version of the server software that answers the request and runs the gateway. The format is <i>name/version</i> .
URL	Gives the base portion of the URL.

Ví dụ sau minh họa việc hiển thị các giá trị của biến môi trường tại server.

```
<HTML>
<!-- This example displays the content of several ServerVariables. --
ALL_HTTP server variable =
<%= Request.ServerVariables("ALL_HTTP") %> <BR>
CONTENT_LENGTH server variable =
<%= Request.ServerVariables("CONTENT_LENGTH") %> <BR>
CONTENT_TYPE server variable =
<%= Request.ServerVariables("CONTENT_TYPE") %> <BR>
QUERY_STRING server variable =
<%= Request.ServerVariables("QUERY_STRING") %> <BR>
SERVER_SOFTWARE server variable =
<%= Request.ServerVariables("SERVER_SOFTWARE") %> <BR>
</HTML>
```

Ví dụ sau sử dụng vòng lặp **For...Each** để hiển thị các giá trị của các biến môi trường tại server.

```
<TABLE BORDER="1">
<TR><TD><B>Server
Variable</B></TD><TD><B>Value</B></TD></TR>
<% For Each strKey In Request.ServerVariables %>
<TR>
<TD><%= strKey %></TD>
<TD><%= Request.ServerVariables(strKey) %></TD>
</TR>
<% Next %>
</TABLE>
```

### 3. APPLICATION VÀ SESSION

Như chúng ta đã biết, nếu một biến được khai báo trong một trang ASP, nó sẽ bị hủy ngay một khi server xử lý xong trang này. Để có thể khai báo các biến có thể sử dụng chung cho nhiều trang, ta sử dụng hai đối tượng **Application** và **Session**.

### 3.1. Đối tượng Application

Đối tượng **Application** để lưu trữ các thông tin có thể được chia sẻ cho nhiều người dùng trong cùng một ứng dụng. Một ứng dụng ASP được định nghĩa như là các trang asp nằm trong cùng một thư mục ảo và các thư mục con của thư mục ảo này. Do các biến được lưu trong đối tượng Application được dùng chung cho nhiều người dùng, do đó một khi một người dùng muốn thay đổi một giá trị dùng chung này, các yêu cầu phải được đặt trong cặp lệnh **Lock** và **Unlock**. Điều này đảm bảo an toàn khi nhiều người cùng yêu cầu thay đổi của cùng một giá trị trong cùng một lúc.

Cần cẩn thận khi tạo ra các biến Application vì các biến này chiếm bộ nhớ và tồn tại suốt thời gian hoạt động của ứng dụng.

Để xóa toàn bộ các biến Application ra khỏi bộ nhớ ta dùng phương thức **RemoveAll**.

Sau đây là một số ví dụ minh họa việc tạo ra một biến và lưu giá trị của nó vào trong đối tượng Application, cũng như lấy giá trị của một biến đã được lưu trong đối tượng Application.

```
<%
Application("greeting") = "Welcome to My Web World!"
Application("num") = 25
Set Application("Obj1") = Server.CreateObject("MyComponent")

%>
...
<%
Set MyLocalObj1 = Application("Obj1")
MyLocalObj1.MyObjMethod
%>
```

```
<%
Application.Lock
Application("NumVisits") = Application("NumVisits") + 1
Application.Unlock
%>
This application page has been visited
<%= Application("NumVisits") %> times!
```

Ví dụ sau minh họa việc sử dụng đối tượng Application để thao tác trên các biến kiểu mảng.

#### ---file1.asp---

```
<%
'Creating and initializing the array.
dim MyArray()
Redim MyArray(5)
MyArray(0) = "hello"
MyArray(1) = "some other string"

'Storing the array in the Application object.
Application.Lock
Application("StoredArray") = MyArray
Application.Unlock
Server.Transfer("file2.asp")
%>
```

#### ---file2.asp---

```
<%
'Retrieving the array from the Application Object
'and modifying its second element.
LocalArray = Application("StoredArray")
LocalArray(1) = " there"

'Printing out the string "hello there."
Response.Write(LocalArray(0)&LocalArray(1))
'Re-storing the array in the Application object.
'This overwrites the values in StoredArray with the new one
Application.Lock
Application("StoredArray") = LocalArray
Application.Unlock
%>
```

### 3.2. Đối tượng Session

Đối tượng **Session** được dùng để lưu trữ các thông tin dùng cho một phiên làm việc của một người dùng. Các biến được lưu trong đối tượng này sẽ không bị hủy khi người dùng chuyển từ trang này sang trang khác.

Webserver sẽ tự động tạo ra một đối tượng Session khi người dùng yêu cầu một trang web từ một ứng dụng mà trước đó chưa có phiên làm việc nào. Webserver sẽ tự động hủy đối tượng Session này khi đối tượng này hết hạn.

Đối tượng Session thường được dùng để lưu các thông tin riêng biệt cho từng phiên làm việc của người dùng. Ví dụ như trong ứng dụng mua hàng trên mạng, mỗi khi người dùng duyệt qua các mặt

hàng và quyết định chọn nó, rồi lại tiếp tục qua các trang khác để chọn mặt hàng khác, ... Các thông tin liên quan đến các mặt hàng mà người dùng đã chọn từ nhiều trang khác nhau nên được lưu trong biến của đối tượng Session để sau này in ra phiếu mua hàng cho người dùng.

Sau đây là một số ví dụ minh họa việc tạo ra một biến và lưu giá trị của nó vào trong đối tượng Session, cũng như lấy giá trị của một biến đã được lưu trong đối tượng Session.

```
<%
Session("username") = "Janine"
Session("age") = 24
Set Session("Obj1") =
Server.CreateObject("MyComponent.class1")
%>
...
<%
Set MyLocalObj1 = Session("Obj1")
MyLocalObj1.MyObjMethod
%>
```

### Lưu trữ các mảng trong các biến Session

Ví dụ để tạo ra một biến Session có tên là Username, ta dùng cú pháp Session("UserName"). Để gán giá trị cho biến Session này, ta dùng: Session("UserName") = "Hoang Dung"

Để lấy giá trị của biến Session ra, ta dùng: vUserName = Session("UserName")

Riêng đối với biến mảng, phải có một chút thay đổi vì ta không thể can thiệp trực tiếp vào các phần tử trong mảng biến Session. Ví dụ, gán giá trị mảng cho Session:

```
<%
MyAr(0)=1
MyAr(1) = 10
Session("MyArray") = MyAr
%>
```

Nhưng khi lấy giá trị ra, ta không dùng được Session("MyArray")(0) hay Session("MyArray")(1) mà phải làm như sau:

```
<%
MyArTemp = Session("MyArray")
v0 = MyArTemp(0)
%>
```

Hay khi gán giá trị cho từng thành phần của mảng, không dùng được Session("MyArray")(0) = 4 mà phải làm như sau:

```
<%
MyArTemp = Session("MyArray")
MyArTemp(0) = 4
Session("MyArray") = MyArTemp
%>
```

Ví dụ sau minh họa việc sử dụng đối tượng Session để thao tác trên các biến kiểu mảng.

```
---file1.asp---
<%
'Creating and initializing the array
Dim MyArray()
Redim MyArray(5)
MyArray(0) = "hello"
MyArray(1) = "some other string"

'Storing the array in the Session object.
Session("StoredArray") = MyArray
```

```
Response.Redirect("file2.asp")
%>
```

```
---file2.asp---
<%
'Retrieving the array from the Session Object
'and modifying its second element.
LocalArray = Session("StoredArray")
LocalArray(1) = " there"

'Printing out the string "hello there."
Response.Write(LocalArray(0)&LocalArray(1))

'Re-storing the array in the Session object.
'This overwrites the values in StoredArray with the new one.
Session("StoredArray") = LocalArray
%>
```

Để đặt thời gian hết hạn của một đối tượng Session, ta đặt giá trị (tính bằng phút) cho thuộc tính **Session.TimeOut**. Để hủy Session hiện hành, ta dùng phương thức **Session.Abandon**. Phương thức này thường được dùng khi người dùng logout khỏi hệ thống

### 3.3. GLOBAL.ASA

Nếu ta muốn thực hiện một số thao tác ngay khi hệ thống tạo ra các đối tượng Application, Session, ta phải đặt các thao tác này trong thủ tục **Application\_OnStart**, **Session.On\_Start** tương ứng. Tương tự, thực hiện các thao tác trước khi hệ thống chuẩn bị hủy bỏ các đối tượng này, ta phải đặt các thao tác này trong thủ tục **Application\_OnEnd**, **Session.On\_End** tương ứng.

Tất cả bốn thủ tục này đều phải được đặt trong tập tin có tên gọi là **Global.asa**. Cấu trúc chung của một tập tin **Global.asa** là:

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
Sub Application_OnStart
    ...
End Sub

Sub Application_OnEnd
    ...
End Sub

Sub Session_OnStart
    ...
End Sub

Sub Session_OnEnd
    ...
End Sub

</SCRIPT>
```

## 4. COOKIES

Cookie là thông tin của ứng dụng web lưu trữ tại máy client. Thông thường các cookie lưu trữ các thông tin riêng của bạn khi sử dụng một website nào đó. Ví dụ, nếu bạn dùng Yahoo Mail, thông tin về tên đăng nhập của bạn có thể sẽ được lưu vào cookie máy bạn đang dùng để nhớ tên đăng nhập của bạn. Đó là lí do tại sao sau này khi quay lại trang Yahoo Mail, hệ thống sẽ tự động hiển thị tên đăng nhập của bạn trước đó và chỉ yêu cầu bạn nhập vào mật khẩu.

Với giao thức HTTP, mỗi yêu cầu cho một trang web không có mối liên quan nào với các yêu cầu khác. Chính vì vậy mà webserver không nhớ được trang nào đã được gửi đến cho người dùng trước đó hay bắt cứ những gì mà người dùng đã duyệt qua. Cookie là một

cơ chế cho phép server lưu trữ các thông tin của nó về người dùng trên chính máy của người dùng hiện tại.

Các thuộc tính chính của cookie là:

- **Domain:** Xác định tên miền mà cookie được gửi đi.
- **Expires:** Ngày hết hạn của cookie. Ta phải đặt ngày hết hạn để chỉ định thời gian lưu trữ cookie trên máy client sau khi một session kết thúc. Nếu ta không đặt thuộc tính này, cookie sẽ hết hạn ngay khi session kết thúc.
- **Path:** Xác định đường dẫn mà cookie được gửi đi.

Để gán giá trị cho cookie, ta dùng cú pháp sau:  
**Response.Cookies(cookie)[(key)].attribute = value**. Với:

- **cookie:** tên của cookie.
- **key:** tham số tùy chọn, dùng để đặt nhiều giá trị cho một cookie.
- **attribute:** thuộc tính của cookie

Nếu cookie chưa tồn tại, lệnh này sẽ tạo mới cookie. Ngược lại, nếu cookie đã tồn tại, lệnh này sẽ thay thế giá trị mới cho cookie và bỏ giá trị cũ đi.

Để lấy giá trị từ cookie, ta dùng cú pháp sau:  
**Request.Cookies(cookie)[(key)].attribute**.

Các ví dụ sau minh họa việc sử dụng cookie.

```
---cookiesform.asp---
<%
    Response.Buffer=true
%>
<HTML>
<HEAD>
<TITLE>cookiesform.asp</TITLE>
<BODY bgcolor="#FFFFFF">
<%
    ln=Request.Cookies("thatperson")("lastname")
    fn=Request.Cookies("thatperson")("firstname")
    st=Request.Cookies("thatperson")("state")
%>
<FORM ACTION = "cookiesformrespond.asp">
    Form with Cookies<p>
```

```
Please enter your First Name<p>
<INPUT NAME="NameFirst" size ="40" value=<%=fn%>>
<p>
Please enter your Last Name<p>
<INPUT NAME="NameLast" size ="40" value=<%=ln%>>
<p>
Please enter your State abbreviation<p>
<INPUT NAME="State" MaxLength="2" value=<%=st%>>
<INPUT TYPE=submit>
</FORM>
</BODY>
</HTML>
```

**---cookiesformrespond.asp---**

```
<%
  Response.Buffer=true
%>
<HTML>
<HEAD>
<TITLE>cookiesformrespond.asp</TITLE>
<BODY bgcolor="#FFFFFF">
<%
l=request.QueryString("namelast")
f=request.QueryString("namefirst")
st=request.QueryString("state")
cookypath="/learn/test"
cookydomain=".www.activeserverpages.com"
cookydie=date+365
Response.Cookies("thatperson")("lastname") = l
Response.Cookies("thatperson")("firstname") = f
Response.Cookies("thatperson")("state") = st
Response.Cookies("thatperson").Expires = cookydie
Response.Cookies("thatperson").Domain = cookydomain
Response.Cookies("thatperson").Path = cookypath
Response.Write Request.Cookies("thatperson")("lastname") &
"<p>"
  Response.Write Request.Cookies("thatperson")("firstname") &
"<p>" 
  Response.Write Request.Cookies("thatperson")("state") & "<p>" 
%>
</BODY>
</HTML>
```

**The script below demonstrates how to remove a cookie.**

**---cookiesformforget.asp---**

```
<%
```

```
  Response.Buffer=true
%>
<HTML>
<HEAD>
<TITLE>cookiesformforget.asp</TITLE>
<BODY bgcolor="#FFFFFF">
<%
cookiepath="/learn/test"
cookiedomain=".www.activeserverpages.com"
cookiesdie=date-365
Response.Cookies("thatperson").Expires = cookiesdie
Response.Cookies("thatperson").Domain = cookiedomain
Response.Cookies("thatperson").Path = cookiepath
Response.Write "I will not remember you"
%>
</BODY>
</HTML>
```

**5. ĐỐI TƯỢNG SERVER**

Đối tượng **Server** cung cấp các thuộc tính và phương thức hỗ trợ cho các chức năng ở mức server.

Để chỉ định khoảng thời gian mà một đoạn mã script có thể thi hành trên server trước khi hết thời gian cho phép (time out), ta đặt giá trị (tính bằng giây) cho thuộc tính **Server.ScriptTimeOut**. Ví dụ:

```
<%Server.ScriptTimeOut=100%>
```

Để gọi thi hành một trang asp khác như nó là một phần của trang asp hiện hành, ta dùng phương thức **Server.Execute**. Ví dụ:

```
<%Server.Execute("Welcome.asp")%>
```

Để chuyển một chuỗi về dạng mã HTML, ta dùng phương thức **Server.HTMLEncode**. Ví dụ lệnh sau:

```
<%= Server.HTMLEncode("The paragraph tag: <P>") %>
```

sẽ sinh ra chuỗi kết quả là: The paragraph tag: &lt;P&gt;. Trên màn hình sẽ hiển thị: The paragraph tag: <P>.

Để ánh xạ đường dẫn ảo thành đường dẫn thực chỉ nơi lưu trữ vật lí thực sự trên máy, ta dùng phương thức **Server.MapPath(Path)**

Trong ví dụ sau, tập tin **data.txt** được lưu trữ tại thư mục C:\Inetpub\Wwwroot\Script cùng với tập tin **test.asp**. Thư mục C:\Inetpub\Wwwroot directory được coi như như là webroot (server's home directory).

Giả sử tập tin test.asp chứa đoạn mã sau. Đoạn mã này sử dụng biến server PATH\_INFO để cho biết ra thư mục vật lí của tập tin này.

```
<%= Server.MapPath(Request.ServerVariables  
("PATH_INFO"))%><BR>
```

Kết quả là c:\inetpub\wwwroot\script\test.asp<BR>

Các ví dụ sau minh họa việc ánh xạ sang thư mục vật lí cần cứ vào cách đặt đường dẫn là tương đối hay tuyệt đối. Một đường dẫn tương đối là đường dẫn không bắt đầu bằng dấu "/" hay "\".

- Ví dụ 1:

```
<%= Server.MapPath("data.txt")%><BR>  
<%= Server.MapPath("script\data.txt")%><BR>
```

Kết quả là: c:\inetpub\wwwroot\script\data.txt<BR>

- Ví dụ 2:

```
<%= Server.MapPath("/script\data.txt")%><BR>  
<%= Server.MapPath("\script")%><BR>
```

Kết quả là : c:\inetpub\wwwroot\script\data.txt<BR> và c:\inetpub\wwwroot\script<BR>

Để gửi các thông tin trạng thái hành của trang asp đang được xử lý tới một trang asp khác để tiếp tục xử lý, ta dùng phương thức **Server.Transfer**. Ví dụ:

#### ASP1.asp

```
<HTML>  
<BODY>  
<%  
Dim sessvar1 Response.Write Session.SessionID  
Response.Write ("<BR>")  
Response.Write("I am going to ASP2 <BR>")  
Server.Transfer("/Myasps/ASP2.asp")  
%>  
  
ASP2.asp  
<HTML>  
<BODY>  
<%  
Response.Write Session.SessionID  
%>  
</BODY>  
</HTML>
```

Để mã hóa các chuỗi thành qui tắc mã hóa URL, ta dùng phương thức **Server.URLEncode**. Ví dụ đoạn mã sau:  
<%  
Response.Write(Server.URLEncode("http://www.microsoft.com"))  
%>  
sẽ in ra chuỗi: http%3A%2F%2Fwww%2Emicrosoft%2Ecom

## 6. XỬ LÍ LỖI VỚI ĐỐI TƯỢNG ASPERROR

Bạn có thể sử dụng các thuộc tính của đối tượng **ASPError** để bẫy các lỗi xảy ra trong đoạn mã script của trang web và để từ đó đưa ra các thông báo và xử lý tương ứng.

Để biết được một thao tác có gặp lỗi hay không, ta dùng phương thức **Server.GetLastError**. Phương thức này sẽ trả về một thẻ hiện của đối tượng ASPError mô tả các thông tin liên quan đến lỗi nếu có.

Các thuộc tính chính của đối tượng ASPError là **ASPCODE**, **Number**, **Source**, **Category**, **File**, **Line**, **Column**, **Description**, **ASPDescription**.

Ví dụ sau minh họa việc sử dụng đối tượng này:

```
<%  
Response.Write objASPError.Category  
If objASPError.ASPCode > "" Then Response.Write ", " &  
objASPError.ASPCode  
Response.Write " (0x" & Hex(objASPError.Number) & ")" & "<br>"  
  
Response.Write "<b>" & objASPError.Description & "</b><br>"  
  
If objASPError.ASPDescription > "" Then Response.Write  
objASPError.ASPDescription & "<br>"  
  
bInErrorWritten = False  
  
' Only show the Source if it is available and the request is from the  
same machine as IIS  
If objASPError.Source > "" Then  
strServername =  
LCase(Request.ServerVariables("SERVER_NAME"))  
strServerIP = Request.ServerVariables("LOCAL_ADDR")  
strRemoteIP = Request.ServerVariables("REMOTE_ADDR")  
If (strServername = "localhost" Or strServerIP = strRemoteIP)  
And objASPError.File <> "?" Then
```

```

Response.Write objASPError.File
If objASPError.Line > 0 Then Response.Write ", line " &
objASPError.Line
If objASPError.Column > 0 Then Response.Write ", column " &
objASPError.Column
    Response.Write "<br>"
    Response.Write "<font style=""COLOR:000000; FONT: 8pt/11pt
courier new""><b>"
    Response.Write Server.HTMLEncode(objASPError.Source) &
"<br>"
    If objASPError.Column > 0 Then Response.Write
String((objASPError.Column - 1), "-") & "^<br>"
        Response.Write "</b></font>"
        blnErrorWritten = True
    End If
End If

If Not blnErrorWritten And objASPError.File <> "?" Then
    Response.Write "<b>" & objASPError.File
    If objASPError.Line > 0 Then Response.Write ", line " &
objASPError.Line
    If objASPError.Column > 0 Then Response.Write ", column " &
objASPError.Column
        Response.Write "</b><br>"
    End If
%>

```

## 7. ADO NÂNG CAO

### 7.1. Thực hiện store procedure

Để yêu cầu thực hiện store procedure, đơn giản ta chỉ cần thay thế chuỗi strSQL trong câu lệnh **Conn.Execute strSQL** bằng chuỗi chỉ tới tên của store procedure. Ví dụ nếu ta có store procedure có tên là **sp\_AddUser**, ta gọi thực hiện như sau: Conn.Execute "sp\_AddUser"

### 7.2. Xử lí lỗi của các thao tác liên quan đến cơ sở dữ liệu

Để kiểm tra các thao tác thực hiện trên cơ sở dữ liệu có xảy ra lỗi hay không, ta sử dụng collection Connection.Errors.

Để biết thao tác xảy ra có thể hay không ta căn cứ vào giá trị Connection.Errors.Count. Nếu giá trị này lớn hơn 0, có nghĩa là thao tác gặp lỗi. Nếu gặp lỗi Connection.Errors.Number sẽ trả về một con số chỉ mã lỗi, Connection.Errors.Description sẽ trả về chuỗi kí tự mô tả lỗi, Connection.Errors.Source sẽ trả về chuỗi kí tự mô tả tên của đối tượng hay ứng dụng gây ra lỗi, Connection.Errors.SQLState sẽ trả về mã lỗi 5 kí tự mô tả lỗi liên quan đến việc thực hiện câu lệnh SQL.

Ta phải sử dụng câu lệnh “On Error Resume Next” ở các nơi có tiềm ẩn khả năng xảy ra lỗi để yêu cầu hệ thống tiếp tục thực hiện các lệnh tiếp theo một khi có lỗi xảy ra. Nếu không, một khi xảy ra lỗi, thông báo lỗi của hệ thống sẽ hiện ra và chương trình bị dừng lại (lỗi run-time error)

Ví dụ sau minh họa việc xử lí lỗi trong khi thực hiện các thao tác liên quan đến cơ sở dữ liệu:

```

<!-- BeginErrorExampleVBS -->
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio
6.0">
<TITLE>Error Handling Example (VBScript)</TITLE>
</HEAD>
<BODY>

<h1>Error Handling Example (VBScript)</h1>

<%
Dim cnn1
Dim errLoop
Dim strError

On Error Resume Next

' Intentionally trigger an error.
Set cnn1 = Server.CreateObject("ADODB.Connection")
cnn1.Open "nothing"

```

```

If cnn1.Errors.Count > 0 Then
    ' Enumerate Errors collection and display
    ' properties of each Error object.
    For Each errLoop In cnn1.Errors
        strError = "Error #" & errLoop.Number & "<br>" & _
                    " " & errLoop.Description & "<br>" & _
                    " (Source: " & errLoop.Source & ")" & "<br>" & _
                    " (SQL State: " & errLoop.SQLState & ")" & "<br>" & _
                    " (NativeError: " & errLoop.NativeError & ")" & "<br>"
        If errLoop.HelpFile = "" Then
            strError = strError & _
                        " No Help file available" & _
                        "<br><br>"
        Else
            strError = strError & _
                        " (HelpFile: " & errLoop.HelpFile & ")" & "<br>" & _
                        " (HelpContext: " & errLoop.HelpContext & ")" & _
                        "<br><br>"
        End If

        Response.Write("<p>" & strError & "</p>")
    Next
End If

%>

</BODY>
</HTML>
<!-- EndErrorExampleVBS -->

```

### 7.3. Phân trang khi hiển thị kết quả

Trong trường hợp dữ liệu trả về quá lớn, người ta thường có xu hướng phân trang dữ liệu để tiện theo dõi. Để có thể phân trang được, ta cần phải lưu ý một số vấn đề sau:

- Kích thước của mỗi trang hay số lượng các mẫu tin cần hiển thị trên mỗi trang.
- Số thứ tự trang: Thông tin này dùng để xác định vị trí mẫu tin bắt đầu sẽ được hiển thị trong trang này. Ví dụ, nếu kích thước mỗi trang là nPageSize, thì nếu ta hiển thị trang thứ nPageNum (nPageNum=1, ..), vị trí của mẫu tin đầu tiên sẽ

được hiển thị trong trang này sẽ là: (nPageNum-1)\*nPageSize+1

- Tổng số mẫu tin được trả về. Thông tin này giúp ta tính được tổng số trang phải hiển thị. Nếu tổng số mẫu tin được trả về là nMaxRecords, kích thước mỗi trang là nPageSize, thì tổng số trang sẽ là: (nMaxRecords+nPageSize-1) div nPageSize

Tuy nhiên, khi sử dụng các thuộc tính của đối tượng Recordset, ta chỉ cần cung cấp các thông tin liên quan đến kích thước trang (Recordset.PageSize) và số thứ tự trang mà thôi (Recordset.AbsolutePage). Thông tin về tổng số mẫu tin được trả về lấy từ Recordset.RecordCount, số lượng các trang lấy từ Recordset.PageCount.

Ví dụ sau minh họa việc phân trang khi dữ liệu lớn:

```

<%
Title = Request("fmTITLE")
Author = Request("fmAUTHOR")
Publisher = Request("fmPUBLISHER")
YearPub = Request("fmYEARPUB")

PageNum = Request ("PageNum") ' Hien thi trang thu may
if PageNum = "" then
    PageNum = 1
else
    PageNum = Cint(PageNum)
end if
PageSize= Cint(Request("PageSize")) ' Kich thuoc cua trang
%>
<%
' ket noi voi CSDL
strDSN = "DRIVER={Microsoft Access Driver (*.mdb)}; " & "DBQ= "
& Server.MapPath("WebLibDB.mdb")
set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open strDSN

' thao tac tren CSDL
strSQL = "SELECT * FROM EDOC WHERE "
strSQL = strSQL & " TITLE LIKE " & "%" & Title & "%" 'like
'%activeX%'
strSQL = strSQL & " AND " & " AUTHOR LIKE " & "%" & Author &
"%"
strSQL = strSQL & " AND " & " PUBLISHER LIKE " & "%" & Publisher
& "%"

```

```

strSQL = strSQL & " AND " & " YEARPUB LIKE " & "%" & YearPub &
"%""

set rs = Server.CreateObject("ADODB.Recordset")
rs.CursorType = 3 '
rs.CursorLocation = 3 ' aduseclient
rs.ActiveConnection = Conn
rs.Open strSQL
rs.PageSize = PageSize
rs.Absolutepage = PageNum

TotalPage = Cint(rs.PageCount)
if rs.eof then
    Response.Write "No records is matched!"
    Response.End
end if
%>
Total Page: <%=rs.RecordCount%>
<p> Page </p>
<%
sURL = Request.ServerVariables("SCRIPT_NAME")
sURL= sURL & "?fmTITLE=" & Title
sURL= sURL & "&fmAUTHOR=" & Author
sURL= sURL & "&fmPUBLISHER=" & Publisher
sURL= sURL & "&fmYEARPUB=" & YearPub
sURL= sURL & "&PageSize=" & PageSize
for i=1 to TotalPage
    if i<10 then
        Response.Write "<a href="" &sURL &"&PageNo=" & i & "">" &
"0" & i & "</a>" & "
    else
        Response.Write "<a href="" &sURL &"&PageNo=" & i & "">" &
i & "</a>" & "
    end if
    next
%>
<TABLE border="1" cellpadding="0" cellspacing="0" width="1000"
>
<tr>
<td width="50" align="center"><b>No</b></td>
<td width="100" align="center"><b>ISBN</b></td>
<td width="300" align="center"><b>Title</b></td>
<td width="250" align="center"><b>Abstract</b></td>
<td width="150" align="center"><b>Authors</b></td>
<td width="100" align="center"><b>Publisher</b></td>
<td width="50" align="center"><b>Year</b></td>

```

```

</tr>
<%
i = 1
do while (not rs.eof) AND (i<=PageSize)
%>
<tr>
<td width="50"><%=i%>&nbsp;</td>
<td width="100"><%=rs("ISBN")%>&nbsp;</td>
<td width="300"><%=rs("TITLE")%>&nbsp;</td>
<td width="250"><%=rs("ABSTRACT")%>&nbsp;</td>
<td width="150" ><%=rs("AUTHOR")%>&nbsp;</td>
<td width="100" ><%=rs("PUBLISHER")%>&nbsp;</td>
<td width="50"><%=rs("YEARPUB")%>&nbsp;</td>
</tr>
<%
i = i+1
rs.movenext
loop
%>
</TABLE>
Total Page: <%=rs.RecordCount%>
<p> Page </p>
<%
sURL = Request.ServerVariables("SCRIPT_NAME")
sURL= sURL & "?fmTITLE=" & Title
sURL= sURL & "&fmAUTHOR=" & Author
sURL= sURL & "&fmPUBLISHER=" & Publisher
sURL= sURL & "&fmYEARPUB=" & YearPub
sURL= sURL & "&PageSize=" & PageSize
for i=1 to TotalPage
    if i<10 then
        Response.Write "<a href="" &sURL &"&PageNo=" & i & "">" &
"0" & i & "</a>" & "
    else
        Response.Write "<a href="" &sURL &"&PageNo=" & i & "">" &
i & "</a>" & "
    end if
    next
%>
<%
' giai phong tai nguyen
rs.Close
set rs = Nothing
Conn.Close
set Conn = Nothing
%>

```

## 8. ĐỐI TƯỢNG FILE SYSTEM OBJECT

Đối tượng **FileSystemObject** cung cấp các thuộc tính và phương thức để xử lý tập tin, thư mục, ổ đĩa, ...

Cũng giống như ADO, đối tượng FileSystemObject không phải là đối tượng được xây dựng sẵn của ASP. Do đó, để tạo một thể hiện của đối tượng FileSystemObject, ta dùng phương thức **Server.CreateObject("Scripting.FileSystemObject")**. Ví dụ:

```
set fso = Server.CreateObject("Scripting.FileSystemObject")
```

Các thuộc tính và phương thức dùng để truy xuất đến ổ đĩa là: **DriveLetter**, **DriveName**, **TotalSize**, **FreeSpace**, **Path**, **ShareName**, **VolumeName**, **GetDriver**, ... Ví dụ:

```
<%
Sub ShowDriveInfo(drvPath)
    Dim fso, drv, s
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set drv = fso.GetDrive(fso.GetDirectoryName(drvPath))
    s = "Drive " & UCASE(drvPath) & " - "
    s = s & drv.VolumeName & "<br>"
    s = s & "Total Space: " & FormatNumber(drv.TotalSize / 1024, 0)
    s = s & " Kb" & "<br>"
    s = s & "Free Space: " & FormatNumber(drv.FreeSpace / 1024, 0)
    s = s & " Kb" & "<br>"
    Response.Write s
End Sub
%>
```

Các thuộc tính và phương thức dùng để thao tác trên thư mục: **CreateFolder**, **DeleteFolder**, **CopyFolder**, **MoveFolder**, **GetFolder**, ... Ví dụ:

```
<%
Sub ShowFolderInfo()
    Dim fso, fldr, s
    ' Get instance of FileSystemObject.
    Set fso = CreateObject("Scripting.FileSystemObject")
    ' Get Drive object.
    Set fldr = fso.GetFolder("c:")
    ' Print parent folder name.
    Response.Write "Parent folder name is: " & fldr & "<br>"
    ' Print drive name.
    Response.Write "Contained on drive " & fldr.Drive & "<br>"
    ' Print root file name.
    If fldr.IsRootFolder = True Then
```

```
    Response.Write "This is the root folder." & ""<br><br>
Else
    Response.Write "This folder isn't a root folder." & "<br><br>
End If
' Create a new folder with the FileSystemObject object.
fso.CreateFolder ("C:\Bogus")
Response.Write "Created folder C:\Bogus" & "<br>
' Print the base name of the folder.
Response.Write "Basename = " & fso.GetBaseName("c:\bogus") &
"<br>
' Delete the newly created folder.
fso.DeleteFolder ("C:\Bogus")
Response.Write "Deleted folder C:\Bogus" & "<br>
End Sub
%>
```

Các thuộc tính và phương thức dùng để thao tác trên tập tin : **CreateTextFile**, **OpenTextFile**, **Write**, **WriteLine**, **WriteBlankLines**, **Read**, **ReadLine**, **ReadAll**, **MoveFile**, **CopyFile**, **DeleteFile**, ... Ví dụ:

```
<%
Sub CreateFile()
    Dim fso, tf
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set tf = fso.CreateTextFile("c:\testfile.txt", True)
    ' Write a line with a newline character.
    tf.WriteLine("Testing 1, 2, 3.")
    ' Write three newline characters to the file.
    tf.WriteBlankLines(3)
    ' Write a line.
    tf.Write ("This is a test.")
    tf.Close
End Sub
```

```
Sub ReadFiles
    Dim fso, f1, ts, s
    Const ForReading = 1
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set f1 = fso.CreateTextFile("c:\testfile.txt", True)
    ' Write a line.
    Response.Write "Writing file <br>
f1.WriteLine "Hello World"
f1.WriteBlankLines(1)
f1.Close
    ' Read the contents of the file.
    Response.Write "Reading file <br>
Set ts = fso.OpenTextFile("c:\testfile.txt", ForReading)
```

```

s = ts.ReadLine
Response.Write "File contents = " & s & ""
ts.Close
End Sub

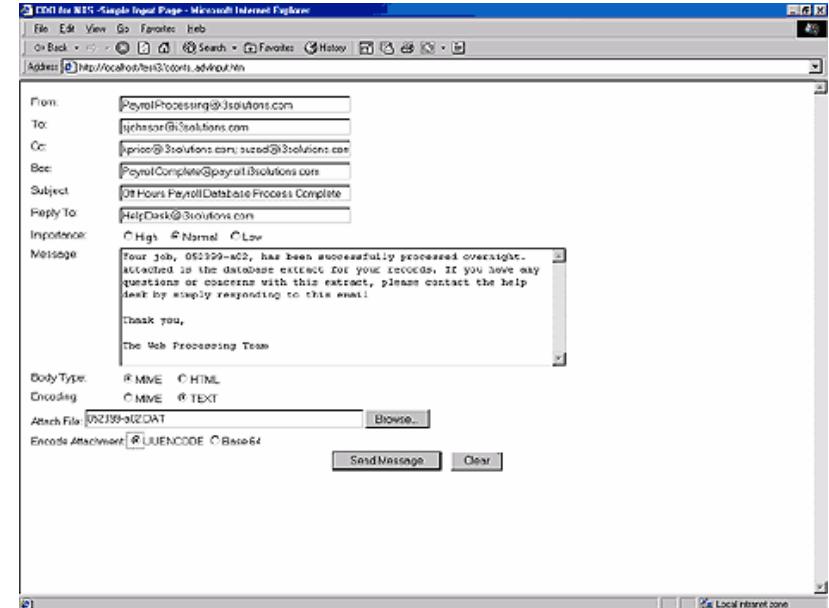
Sub ManipFiles
Dim fso, f1, f2, s
Set fso = CreateObject("Scripting.FileSystemObject")
Set f1 = fso.CreateTextFile("c:\testfile.txt", True)
Response.Write "Writing file <br>"
' Write a line.
f1.WriteLine ("This is a test.")
' Close the file to writing.
f1.Close
Response.Write "Moving file to c:\tmp <br>"
' Get a handle to the file in root of C:\.
Set f2 = fso.GetFile("c:\testfile.txt")
' Move the file to \tmp directory.
f2.Move ("c:\tmp\testfile.txt")
Response.Write "Copying file to c:\temp <br>"
' Copy the file to \temp.
f2.Copy ("c:\temp\testfile.txt")
Response.Write "Deleting files <br>"
' Get handles to files' current location.
Set f2 = fso.GetFile("c:\tmp\testfile.txt")
Set f3 = fso.GetFile("c:\temp\testfile.txt")
' Delete the files.
f2.Delete
f3.Delete
Response.Write "All done!"
End Sub
%>

```

## 9. GỬI EMAIL BẰNG CDONTS

Trong các ứng dụng web, ta có thể sử dụng CDONTS để gửi và nhận thư điện tử. Ví dụ sau minh họa thao tác gửi email từ ứng dụng web.

Trang **cdonts\_advInput.htm** sẽ hiển thị form cho giống như giao diện gửi email của các chương trình email thông thường khác:



```

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Language" CONTENT="en-us">
<META NAME="GENERATOR" CONTENT="Microsoft
FrontPage 4.0">
<META NAME="ProgId" CONTENT="FrontPage.
Editor.Document">
<TITLE>CDO for NTS -Simple Input Page</TITLE>
</HEAD>
<BODY>
<FORM METHOD="post" action="CDONTS_advInput.asp"
ID=FORM1 NAME="FrontPage_Form1"
ONSUBMIT="return FrontPage_Form1_Validator(this)">
<TABLE BORDER="0" WIDTH="100%">
<TR>
<TD VALIGN="top" ALIGN="left">From: </TD>
<TD VALIGN="top" ALIGN="left">
<P><!-- WEBBOT BOT="Validation"
STARTSPAN S-DISPLAY-NAME="From"
S-DATA-TYPE="String"
B-ALLOW-LETTERS="TRUE"
B-ALLOW-DIGITS="TRUE"
B-ALLOW-WHITESPACE="TRUE"

```

```

S-ALLOW-OTHER-CHARS="@."
B-VALUE-REQUIRED="TRUE"
I-MINIMUM-LENGTH="4" --><!-- WEBBOT
BOT="Validation" endspan-->
; <INPUT NAME="txtFrom" SIZE="45"></P>
</TD>
</TR>
<TR>
<TD VALIGN="top" ALIGN="left">To: </TD>
<TD VALIGN="top" ALIGN="left"><!--
WEBBOT BOT="Validation" STARTSPAN
S-DISPLAY-NAME="To" S-DATA-TYPE="String"
B-ALLOW-LETTERS="TRUE"
B-ALLOW-DIGITS="TRUE"
S-ALLOW-OTHER-CHARS="@." B-VALUE-REQUIRED="TRUE"
I-MINIMUM-LENGTH="7" --><!--
WEBBOT BOT="Validation" ENDSPAN --><INPUT
SIZE="45" ></TD>
</TR>
<TR>
<TD VALIGN="top" ALIGN="left">Cc: </TD>
<TD VALIGN="top" ALIGN="left">
<INPUT NAME="txtCc" SIZE="45"></TD>
</TR>
<TR>
<TD VALIGN="top" ALIGN="left">Bcc: </TD>
<TD VALIGN="top" ALIGN="left">
<INPUT NAME="txtBcc" SIZE="45"></TD>
</TR>
<TR>
<TD VALIGN="top" ALIGN="left">
Subject: </TD>
<TD VALIGN="top" ALIGN="left">
<INPUT NAME="txtSubject" SIZE="45"></TD>
</TR>
<TR>
<TD VALIGN="top" ALIGN="left">
Reply To: </TD>
<TD VALIGN="top" ALIGN="left">
<INPUT NAME="txtReplyTo" SIZE="45"></TD>
</TR>
<TR>
<TD VALIGN="top" ALIGN="left">
Importance: </TD>
<TD VALIGN="top" ALIGN="left">
<INPUT TYPE="radio" NAME="optImportance"

```

```

VALUE="2">
High&nbsp; &nbsp; &nbsp;
<INPUT TYPE="radio"
CHECKED NAME="optImportance"
VALUE="1">
Normal&nbsp; &nbsp; &nbsp;
<INPUT TYPE="radio"
NAME="optImportance" VALUE="0">Low</TD>
</TR>
<TR>
<TD VALIGN="top" ALIGN="left">
Message: </TD>
<TD VALIGN="top" ALIGN="left">
<TEXTAREA COLS=68 NAME="txtMessage" ROWS=9>
Type your message here in text
or HTML format

To use HTML in the body of your message, make sure
to select HTML Body Type and MIME
Encoding</TEXTAREA></TD>
</TR>
<TR>
<TD VALIGN="top" ALIGN="left">
Body Type: </TD>
<TD VALIGN="top" ALIGN="left">
<INPUT TYPE="radio" NAME="optMsgType" VALUE="1"
TABINDEX="1" checked>
MIME&nbsp; &nbsp; &nbsp; &nbsp;
<INPUT TYPE="radio"
NAME="optMsgType" VALUE="0" TABINDEX="2">HTML</TD>
</TR>
<TR>
<TD VALIGN="top" ALIGN="left">
Encoding: </TD>
<TD VALIGN="top" ALIGN="left">
<INPUT TYPE="radio" NAME="optMsgEncode"
VALUE="0">MIME&nbsp; &nbsp; &nbsp; &nbsp;
<INPUT TYPE="radio"
NAME="optMsgEncode" VALUE="1" checked>TEXT</TD>
</TR>
<TR>
<TD VALIGN="top" ALIGN="left"
COLSPAN="2"><FONT FACE="MS Sans Serif"
SIZE=1>Attach File: &nbsp;
<INPUT NAME="txtattfile" TYPE="file"
SIZE="55"></FONT></TD>

```

```

</TR>
<TR>
<TD VALIGN="top" ALIGN="left"
COLSPAN="2"><FONT FACE="MS Sans Serif"
size="1">Encode
Attachment: </FONT>
<INPUT TYPE="radio"
NAME="optAttEncode" VALUE="0" checked>UUENCODE
<INPUT TYPE="radio"
NAME="optAttEncode" VALUE="2">
<FONT FACE="MS Sans Serif"
size=1>Base 64</FONT></TD>
</TR>
<TR>
<TD VALIGN="top" ALIGN="left"
COLSPAN="2">
<P ALIGN="center">
<INPUT TYPE="submit" VALUE="Send Message" NAME="btnSend"
TABINDEX="1">&nbsp;&nbsp;&nbsp;
<INPUT TYPE="reset" VALUE=
Clear " NAME="btnClear" TABINDEX="2"></P></TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```

Trang **cdonts\_advInput.asp** sẽ hiển thị xử lí các dữ liệu nhập từ form rồi tạo email gửi đi:

```

<%@ LANGUAGE="VBSCRIPT" %>
<%
Option Explicit
On Error Resume Next

Sub WriteHTML(strInput)
    Response.Write(Server.HTMLEncode(strInput)
& "<BR>")
End Sub
%>

<HTML>
<HEAD>
<META NAME="GENERATOR"
Content="Microsoft FrontPage 4.0">
<TITLE>Send CDONTS -
Simple Input</TITLE>

```

```

</HEAD>
<BODY>

<%
Dim objMsg, strFrom, strTo,
strCc, strBcc, strReplyTo, strBody, _
strSubject, strFileName
Dim IngImportance, IngMsgFormat,
IngMsgEncode, IngAttEncode

strFrom = Trim(Request.Form("txtFrom"))
strTo = Trim(Request.Form("txtTo"))
strCc = Trim(Request.Form("txtCc"))
strBcc = Trim(Request.Form("txtBcc"))
strReplyTo = Trim(Request.Form("txtReplyTo"))
strSubject = Trim(Request.Form("txtSubject"))
strBody = Trim(Request.Form("txtMessage"))
IngImportance = Trim(Request("optImportance"))
IngMsgFormat = Trim(Request("optMsgType"))
IngMsgEncode = Trim(Request("optMsgEncode"))
IngAttEncode = Trim(Request("optAttEncode"))
strFileName = Trim(Request.Form("txtattfile"))

Set objMsg = Server.CreateObject("CDONTS.NewMail")
If Len(Trim(strReplyTo)) > 0 Then
    objMsg.Value("Reply-To")=strReplyTo
End If

objMsg.From = strFrom
objMsg.To = strTo
objMsg.Cc = strCc
ObjMsg.Bcc = strBcc
objMsg.Subject = strSubject
objMsg.Importance = IngImportance
objMsg.BodyFormat = IngMsgFormat
objMsg.MailFormat = IngMsgEncode
objMsg.Body = strBody
If Len(Trim(strFileName)) > 0 Then
    objMsg.AttachFile strFileName, , IngAttEncode
End If

objMsg.Send
Set objMsg = Nothing

WriteHTML("The following message was sent via
CDO for NTS:")

```

```

WriteHTML("From: " &strFrom)
WriteHTML("To: " &strTo)
WriteHTML("Cc: " &strCc)
WriteHTML("Bcc: " &strBcc)
WriteHTML("Reply To: " &strReplyTo)
WriteHTML("Subject: " &strSubject)
WriteHTML("Body: " &strBody)
WriteHTML("Importance: " &IntImportance)
WriteHTML("Message Format: " &IntMsgFormat)
WriteHTML("Message Encode: " &IntMsgEncode)
WriteHTML("File Attachment: " &strFileName)
WriteHTML("File Attachment Encode: " &IntAttEncode)

%>
<HR>
Send another message with <A HREF =
"cdonts_advInput.htm">
    advanced features</A><BR>
Send another message with <A HREF =
"cdonts_simpleInput.htm">
    basic features</A>

</BODY>
</HTML>

```

## 10. MỘT SỐ KĨ THUẬT KHÁC

### 10.1. Chèn tập tin

Để chèn một tập tin vào tập tin asp hiện hành, ta dùng từ dẫn hướng **#include**. Có hai cách để xác định đường dẫn đến tập tin cần chèn vào tập tin hiện hành:

- Sử dụng từ khóa **virtual** để chỉ ra đường dẫn đến tập tin cần chèn vào bắt đầu bằng thư mục ảo. Ví dụ, nếu một tập tin có tên là **Footer.inc** nằm trong thư mục ảo **/Myapp**, dòng lệnh sau sẽ chèn nội dung tập tin **Footer.inc** vào tập tin hiện hành:  
`<!-- #include virtual ="/myapp/footer.inc" -->`
- Sử dụng từ khóa **file** để chỉ ra đường dẫn tương đối đến tập tin cần chèn. Ví dụ, nếu tập tin **Header1.inc** nằm trong thư mục **Myapp\Headers**, thì tập tin nằm trong thư mục **Myapp** sẽ chứa dòng lệnh sau khi muốn chèn tập tin **Header1.inc** vào:  
`<!-- #include file ="headers\header1.inc" -->`

### 10.2. Bảo vệ mã asp

Để bảo vệ các đoạn mã được viết trong các trang asp, ta dùng một công cụ gọi là **ScriptEncoder** được cung cấp bởi Microsoft <http://msdn.microsoft.com/scripting/>.

Ví dụ sau minh họa việc một đoạn mã được viết trong trang asp.

```

<HTML>
<HEAD>
<TITLE>Script Encoder Sample Page</TITLE>
<SCRIPT LANGUAGE="JScript">
<!--
//Copyright© 1998 Microsoft Corporation.
/**Start Encode**
function verifyCorrectBrowser(){
    if(navigator.appName == "Microsoft Internet Explorer")
        if (navigator.appVersion.indexOf ("5.") >= 0)
            return(true);
        else
            return(false);
}
function getAppropriatePage(){
    var str1 = "Had this been an actual Web site, a page compatible
    with ";
    var str2 = "browsers other than ";
    var str3 = "Microsoft Internet Explorer 5.0 ";
    var str4 = "would have been loaded.";
    if (verifyCorrectBrowser())
        document.write(str1 + str3 + str4);
    else
        document.write(str1 + str2 + str3 + str4);
}
//-->
</SCRIPT>
</HEAD>
<BODY onload="getAppropriatePage()">
</BODY>
</HTML>

```

Sau khi trang này được mã hóa bằng tiện ích ScriptEncoder, nó sẽ có nội dung như sau

```

HTML>
<HEAD>
<TITLE>Script Encoder Sample Page</TITLE>
<SCRIPT LANGUAGE="JScript.Encode">
<!--/

```

```
//Copyright© 1998 Microsoft Corporation.
/**Start Encode**#@~ ^QwIAAA==#@#&0; mDkWP7nD
b0zZKD.n1YAMGhk+Dvb`#@#&P,kW` UC7kLIDGDcI22gl:n~
{`~Jtr1DGKw6YP&xDnD+OPA62sKD+ME#@#&P, ~-
k6PxvC\rlmYGdcCwa.n.kkWU bx[+X66Pcr*cJ#, @
*{ ~!*P~P,P~. YEMU` DDE bIP,P,+s/n@#@&P~P,~PM+O;
Mx` WC^/n#/pn6EU1YbWx,o Obaw.WaDrCD+nmL+v#@#&@&
~P7IMPdY.q,'~J_CN,Y4rkP4nnPCx,C1Y;mV, + (PkrY ~ ~I,
wCL PmKhwmYK(snPSkDt~JI@#@&P~\m.PkY.+,
'PE8MWA/.kPGDt DPDtmUPri@#@&,P-CMP/D.&,
'Pr\rmMWkWWY~(YnDnY,2a2^WDn., * !,Ep@#@&,P7ID,/D.c,
'~JSW;s9Ptm-,4+ U~VKI9+[REI,Pr0,c\ DrWHZW.
. mOAMGS/nM` *#@#&P,~P9W^Es+UOchDbO+v/YMq~
_./DDfPQ~kY.c*IP,+sd@#@&~,P[W1;s+UDRSDKD+vdY
MF~/O.yP_,dYM&P3~dYMc*iNz&R @* ^#~@
//-->
</SCRIPT>
</HEAD>
<BODY onload="getAppropriatePage()">
</BODY>
</HTML>
```

## Bài tham khảo

# HIỂN THỊ HÌNH ẢNH TỪ CƠ SỞ DỮ LIỆU TRONG CÁC ỨNG DỤNG WEB

### 1. Giới thiệu

Việc lưu trữ và hiển thị hình ảnh từ CSDL trong các ứng dụng web là cần thiết. Ví dụ như các ứng dụng quản lý hồ sơ của nhân viên, ngoài thông tin bằng văn bản như Họ tên, địa chỉ, bằng cấp, ... sẽ rất cần thiết nếu có thêm dữ liệu về ảnh của nhân viên. Hay trong các ứng dụng bán hàng, bên cạnh thông tin về sản phẩm như Tên sản phẩm, phân loại sản phẩm, giá cả, ... hình ảnh trực quan về sản phẩm cũng rất cần thiết để giúp cho người dùng dễ dàng hơn khi chọn lựa, ...

Thông thường người ta dùng một trong hai cách để đạt được mục đích này. Cách thứ nhất là lưu trữ tập tin hình ảnh trên một thư mục riêng, trong CSDL ngoài dữ liệu văn bản chỉ lưu đường dẫn đến tập tin hình ảnh. Cách thứ hai là lưu trữ cả dữ liệu hình ảnh và văn bản trong cùng một bản ghi trong CSDL.

Cách tiếp cận thứ nhất thường được dùng hơn vì CSDL có kích thước nhỏ và thao tác hiển thị khá đơn giản thông qua tag IMG với thuộc tính SRC được gán bằng thông tin về đường dẫn đến tập tin hình ảnh đã có trong CSDL. Hạn chế của cách tiếp cận này là đòi hỏi dữ liệu ảnh phải được lưu trữ trên webserver hoặc ở một máy tính nào đó mà webserver có thể truy cập được. Trong các ứng dụng mà cơ sở dữ liệu có thể được sao lưu nhiều bản để phân tán, “rủi ro” sẽ xảy ra nếu các tập tin hình ảnh không được sao lưu theo đúng đường dẫn sẵn có.

Cách tiếp cận thứ hai do lưu trữ hình ảnh trong CSDL như là dữ liệu nhị phân nên sẽ dẫn đến hạn chế là làm cho kích thước của CSDL tăng lên đáng kể. Nhưng bù lại, dữ liệu hình ảnh và văn bản ở chung một nơi nên có thể dùng cho các CSDL được sao lưu nhiều nơi. Để lưu trữ dữ liệu hình ảnh trên các hệ quản trị CSDL ta phải dùng các kiểu dữ liệu dạng nhị phân cho nó. Ví dụ, trong MS SQL Server là kiểu dữ liệu image, trong MS Access là kiểu dữ liệu OLE Object.

## 2. Hiển thị hình ảnh có trong CSDL

Nếu dùng cách tiếp cận thứ nhất, ta sẽ tạo ra trong bảng dữ liệu một trường tên chẵng hạn là IMAGE\_URL để lưu trữ đường dẫn đến tập tin hình ảnh, ví dụ như là: images/id1234.gif. Để hiển thị hình ảnh này trong trang web, đơn giản chỉ cần dùng tag IMG với thuộc tính SRC được gán bằng dữ liệu trong trường ImageURL như: Response.Write("<IMG SRC="" & rs("IMAGE\_URL") & "">").

Nếu dùng cách tiếp cận thứ hai, ta sẽ tạo ra trong bảng dữ liệu một trường có tên chẵng hạn là APP\_IMGDATA để lưu trữ dữ liệu hình ảnh dưới dạng nhị phân. Sau đó, để hiển thị hình ảnh này lên, ta thực hiện tuần tự các bước sau:

- Đọc dữ liệu hình ảnh lưu trong ImgData vào recordset.
- Gán thuộc tính ContentType của đối tượng Response tương ứng với định dạng ảnh lưu trữ. Nếu lưu tập tin dưới dạng .gif, ta đặt Response.ContentType="image/gif".
- Sử dụng hàm Response.BinaryWrite để ghi nội dung dữ liệu của hình ảnh ra.

Giả sử ta dùng CSDL là MS Access với tập tin CSDL là DB\_USERS được lưu trong thư mục APP\_DB, bảng dữ liệu APP\_USERS được dùng để lưu thông tin của người dùng và hình ảnh tương ứng. Đoạn mã sau của tập tin showimage.asp minh họa các bước trên:

### showimage.asp

```
<%
' Kết nối với CSDL
strDSN = "DRIVER=Microsoft Access Driver (*.mdb);DBQ="
' đường dẫn tương đối đến tập tin CSDL
strDSN = strDSN & Server.MapPath("app_db/db_users.mdb")
set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open strDSN
' giả sử cần hiển thị hình ảnh lưu trong bản ghi
' có trường APP_USERID bằng vUserID
strSQL = "SELECT * FROM APP_USERS WHERE "
strSQL = strSQL & " APP_USERID = " & vUserID
' đọc dữ liệu hình ảnh vào recordset
Set rs = Conn.Execute(strSQL)
' gán thuộc tính ContentType là image/gif
Response.ContentType = "image/gif"
' ghi dữ liệu hình ảnh ra tập tin chuyển xuống client
```

```
Response.BinaryWrite(rs("APP_IMGDATA"))
rs.close
set rs = nothing
Conn.close
set Conn = nothing
%>
```

Với đoạn mã trên do ta đã đặt nội dung chuyển xuống cho trình duyệt là image/gif nên không thể vừa cùng hiển thị dữ liệu văn bản vừa hiển thị hình ảnh được.

## 3. Tải tập tin ảnh lên CSDL

Để có thể tải các tập tin dữ liệu lên server, thông thường ta dùng các component đã được viết sẵn để hỗ trợ cho việc này. Một trong các component cung cấp miễn phí là [aspSmartUpload](http://www.aspsmart.com/aspSmartUpload/) (<http://www.aspsmart.com/aspSmartUpload/>). Hai vấn đề cần lưu ý khi sử dụng các component dạng này là form trong trang dành cho người dùng nhập phải được đặt thuộc tính ENCTYPE là multipart/form-data và trang xử lý upload phải dùng đối tượng Form của các component để lấy dữ liệu về thay cho Request.Form. Với aspSmartUpload ta có thể tải đồng thời nhiều tập tin lên server, thậm chí có thể hạn chế kích thước tập tin, kiểu tập tin, ... sẽ được dùng để tải. Ví dụ sau minh họa việc tải dữ liệu lên CSDL bằng cách dùng tập tin upload.htm cho phép người dùng chỉ định tập tin cần upload, tập tin upload.asp dùng để lưu dữ liệu tập tin cần upload vào một trường trong bảng CSDL hoặc lưu thành một tập tin trong thư mục nào đó:

### upload.htm

```
<H1>aspSmartUpload : Sample </H1>
<HR>

<FORM method="POST" action="upload.asp"
enctype="multipart/form-data">
<input type="FILE" name ="FILE1" ><br>
<input type="submit" value="Upload">
</FORM>
```

```

upload.asp
<%
' tạo đối tượng aspSmartUpload
Set mySmartUpload = Server.CreateObject("aspSmartUpload.SmartUpload")
' upload tập tin
mySmartUpload.Upload
' kết nối với CSDL
strDSN = "DRIVER=Microsoft Access Driver (*.mdb);DBQ="
' đường dẫn tương đối đến tập tin CSDL
strDSN = strDSN & Server.MapPath("app_db/db_users.mdb")
set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open strDSN

set rs = Server.CreateObject("ADODB.Recordset")
set rs.ActiveConnection = Conn
rs.Source = "app_users"
rs.LockType = 3
rs.Open
rs.close

For each file In mySmartUpload.Files
    If NOT file.IsMissing then ' nếu tập tin tồn tại
        rs.AddNew
        ' lưu dưới dạng một field trong CSDL
        file.FileToField rs.Fields("APP_IMGDATA")
        ' lưu dưới dạng một tập tin trên thư mục của server
        ' với đường dẫn tuyệt đối
        file.SaveAs("c:\temp\" & file.FileName)
        ' lưu dưới dạng một tập tin trên thư mục của server
        ' với đường dẫn tương đối so với thư mục webroot
        file.SaveAs("dbimages/upload" &
                    file.FileName)
        '...
        rs.Update
    End if
Next
rs.close
set rs = nothing
Conn.close
set Conn = nothing
%>

```

#### 4. Kết luận

Bài viết vừa trình bày các hai thao tác cơ bản cho việc quản lý hình ảnh trong cơ sở dữ liệu của các ứng dụng web. Thao tác thứ nhất liên quan đến việc chọn hình thức lưu trữ dữ liệu ảnh, thao tác thứ hai liên quan đến việc tải các hình ảnh lên server. Lưu trữ dữ liệu ảnh dưới dạng là một trường cơ sở dữ liệu sẽ làm cho kích thước cơ sở dữ liệu lớn, việc xử lý khá phức tạp nhưng có thuận lợi trong trường hợp ứng dụng được thiết kế cho việc sử dụng phân tán. Đơn giản và thường được dùng hơn cả là lưu trữ các tập tin hình ảnh trong một thư mục định trước trên server và lưu trữ đường dẫn đến tập tin này trong cơ sở dữ liệu.

## Bài tham khảo

# PHƯƠNG PHÁP HỖ TRỢ GIỚI HẠN TRUY CẬP TRANG WEB

## 1. Giới thiệu

Thông thường, trong các ứng dụng web, người thiết kế muốn giới hạn sự truy cập đến một số trang web thông qua việc chứng thực người dùng (authentication) nhằm mục đích cho phép những người có quyền thực sự mới được phép truy cập và thực hiện một số trang web nào đó. Ví dụ các trang web dùng cho việc cập nhật CSDL từ xa chỉ cho phép người quản trị thực hiện hay trong các ứng dụng như diễn đàn thảo luận, thông thường các trang gửi bài mới chỉ cho phép những người đã đăng ký thực hiện mà thôi, ...

Để đạt được mục đích này, có hai cách tiếp cận:

- Dùng chức năng bảo mật của hệ thống: Cách này giới hạn quyền truy cập đến các trang web cần bảo vệ bằng quyền trên hệ thống tập tin NTFS. Ví dụ, nếu muốn giới hạn quyền truy cập đến tập tin admin.asp, ta xác lập quyền cho một người dùng nào đó được quyền đọc, thi hành mà thôi. Cách này có hạn chế là người dùng trang web phải có tài khoản trên server. Điều này sẽ thực sự khó khăn khi đa số các ứng dụng web thường được hosting tại các server của các ISP.
- Dùng các đoạn mã chương trình tự viết: Cách này sử dụng cookies (thông qua biến kiểu Session) kết hợp với CSDL về người dùng để làm việc này! Cách làm này cho phép đáp ứng khá hoàn hảo nhu cầu bảo mật các trang web và tương thích dễ dàng trong trường hợp hosting ở các server khác nhau.

## 2. Bảo vệ bằng các đoạn mã chương trình tự viết

Ý tưởng chính của cách làm này là ta sẽ dùng một biến Session có kiểu là boolean để lưu thông tin về người dùng đã được chứng thực hay chưa. Giả sử ta đặt tên cho biến này là blLoginOK, giá trị True sẽ tương ứng với người dùng đã được chứng thực và ngược lại.

Việc chứng thực người dùng sẽ được thông qua một trang đăng nhập (ví dụ là trang login.htm). Trang này sẽ yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu. Sau đó một đoạn mã (trang login.asp) sẽ được dùng để kiểm tra thông tin người dùng vừa nhập có trùng khớp với dữ liệu được lưu trữ trên CSDL hay không. Nếu thông tin trùng khớp, giá trị của biến blLoginOK sẽ được chuyển thành True (lưu ý ta phải thiết lập biến blLoginOK có giá trị mặc định là False).

Trong các trang cần hạn chế truy cập, ta chỉ cần kiểm tra giá trị của biến này là True hay False. Nếu giá trị là True, người dùng sẽ được phép thực hiện tiếp các đoạn mã tiếp theo của trang, còn ngược lại, ta sẽ thông báo yêu cầu người dùng chứng thực thông qua một trang đăng nhập trước khi tiếp tục.

Các bước thực hiện tuần tự như sau:

### Bước 1: Tạo cơ sở dữ liệu chứa thông tin về người dùng

Giả sử ta dùng MS Access để tạo cơ sở dữ liệu có tên là DB\_USERS.MDB, trong đó ta tạo một bảng dữ liệu có tên là APP\_USERS. Hai trường chính của bảng dữ liệu này là APP\_USERNAME và APP\_PASSWORD. Nếu đặt APP\_USERNAME như là khóa chính thì một người dùng sẽ được xác định bằng một tên đăng nhập duy nhất. Tất nhiên, ta có thể tạo thêm các trường khác để quản lý như Họ Tên, Địa chỉ Email (có thể sẽ cần để gửi email khi quên mật khẩu), Lần đăng nhập cuối cùng, Thời gian sử dụng hệ thống, ... Sau khi tạo xong, giả sử tập tin này được lưu tại thư mục APP\_DB.

### Bước 2: Tạo trang đăng nhập login.htm để yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu:

```
Login.htm
<FORM action="login.asp" method="POST">
Username: <input type="text" name="fmUserName" size="20"><br>
Password: <input type="password" name="fmPassword" size="20"><br>
<input type="submit" value="Submit" name="btnSubmit">
<input type="reset" value="Reset" name="btnReset">
</FORM>
```

**Bước 3:** Tạo trang login.asp để kiểm tra thông tin người dùng vừa nhập có trùng khớp với thông tin có sẵn trên CSDL hay không. Nếu trùng khớp, giá trị biến blLoginOK sẽ được chuyển thành True.

#### >Login.asp

```
<%
On Error Resume Next
vUserName = Request.Form("fmUserName")
' Thay thế dấu nháy đơn ' thành hai dấu nháy đơn để tránh lỗi SQL
injection
vUserName = Replace(vUserName, "'", "''")
vPassword = Request.Form("fmPassword")
vPassword = Replace(vPassword, "'", "''")

strDSN = "DRIVER=Microsoft Access Driver (*.mdb);DBQ="
' đường dẫn tương đối đến tập tin CSDL
strDSN = strDSN & Server.MapPath("app_db/db_users.mdb")
set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open strDSN

strSQL = "SELECT * FROM APP_USERS WHERE "
strSQL = strSQL & " APP_USERNAME = " & "''" & vUserName & "''"
strSQL = strSQL & " AND " & "APP_PASSWORD = " & "''" &
vPassword & "''"
Set rs = Conn.Execute(strSQL)

if rs.eof then ' người dùng không hợp lệ
    Response.Redirect("login.htm")
else
    Session("blLoginOK") = True
end if
set rs = nothing
set Conn = nothing
%>
```

**Bước 4:** Trong các trang web ví dụ như Admin.asp mà ta chỉ muốn những người đã được chứng thực mới được quyền sử dụng, đặt đoạn mã kiểm tra biến blLoginOK là True hay False ngay đầu trang:

```
Admin.asp
<%
if (Session("blLoginOK") <> True) then
    Response.Redirect("login.htm")
```

```
end if
%>
...
```

### 3. Kết luận

Nhu cầu hạn chế người dùng truy cập đến một số trang web nào đó trong ứng dụng là một nhu cầu thường xuyên khi xây dựng các ứng dụng. Bằng cách sử dụng biến Session và CSDL của người dùng cùng với các trang login.htm, login.asp, ta có thể đạt được mục đích trên một cách dễ dàng.

## Bài tham khảo

# BẢO VỆ CƠ SỞ DỮ LIỆU ACCESS TRONG CÁC ỨNG DỤNG WEB

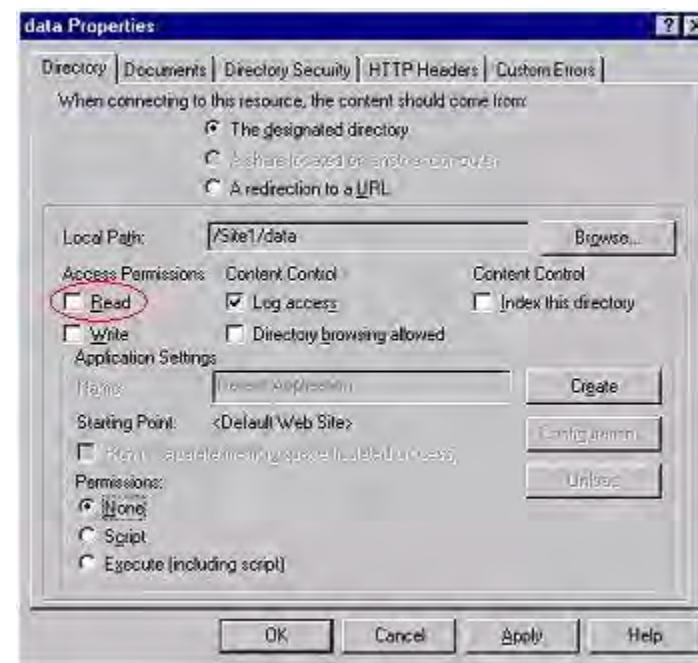
Các ứng dụng web sử dụng CSDL Access thường hay đặt tập tin CSDL .mdb vào một thư mục có thể truy cập được từ web, ví dụ như: D:\inetpub\wwwroot\myDB.mdb. Điều nguy hiểm nhất theo cách làm thông thường này là nếu người dùng biết được hay đoán được đường dẫn đến tập tin .mdb, họ có thể tải tập tin CSDL đó về và toàn bộ thông tin lưu trữ trên CSDL bị đánh cắp.

Để bảo vệ CSDL Access trong các ứng dụng web, nên kết hợp các phương án sau:

### 1. Phương án 1: Đặt tập tin CSDL .mdb vào thư mục được không được quyền truy cập từ Web.

Giả sử ta có website có thư mục webroot là D:\inetpub\wwwroot\. Thư mục chứa tập tin CSDL ví dụ là: D:\inetpub\wwwroot\Site1\data\myDB.mdb. Mặc định nếu người dùng đoán được đường dẫn này: http://www.yourserver.com/site1/data/myDB.mdb, họ có thể tải được tập tin CSDL này về bởi vì thông thường các tập tin trong thư mục này được thiết lập quyền Read.

Để hạn chế không cho phép người dùng tải tập tin CSDL về, ta sẽ bỏ quyền Read được thiết lập trong thư mục này bằng cách dùng tiện ích Internet Service Manager.



Thao tác này không ảnh hưởng gì đến việc các đoạn mã ASP truy cập đến CSDL do thiết lập này được đặt ở mức webserver chứ không phải ở mức hệ thống tập tin NTFS. Nghĩa là các đoạn mã ASP vẫn hoạt động bình thường như trước. Điểm khác duy nhất là người dùng không thể tải được tập tin CSDL dù biết đường dẫn đến nó mà thôi.

### 2. Phương án 2: Đặt tập tin CSDL .mdb tại nơi mà chỉ truy cập được ở mức server-side

Ý tưởng chính của phương án này là đặt tập tin CSDL trong một thư mục có cấp cao hơn thư mục webroot của webserver. Ví dụ, nếu thư mục D:\inetpub\wwwroot\ là webroot của webserver, ta có thể tạo một thư mục private đặt tại D:\inetpub\private và đặt tập tin CSDL vào đây. Bằng cách này, người dùng client không thể nào truy cập đến thư mục private này để tải CSDL về. Lúc này, đường dẫn đến tập tin CSDL trong chuỗi DSN sẽ được chỉnh lại như sau:

Nếu dùng đường dẫn tuyệt đối: sFileName = "D:\inetpub\private"

Nếu dùng đường dẫn tương đối:  
 sFileName = Server.MapPath("/") ' trả về giá trị  
 D:\inetpub\wwwroot  
 sFileName = Replace(sFileName, "wwwroot", "private")  
 sFileName = sFileName & "myDB.mdb"

## Bài tham khảo

# BẢO VỆ ỨNG DỤNG WEB CHỐNG TẤM CÔNG KIỂU SQL INJECTION

### 1. SQL Injection là gì?

Việc thiết kế và đưa vào hoạt động một website luôn đòi hỏi các nhà phát triển phải quan tâm đến các vấn đề về an toàn, bảo mật nhằm giảm thiểu tối đa khả năng bị tấn công từ các tin tặc. Tuy nhiên, thông thường các nhà phát triển đa số tập trung vào các vấn đề an toàn trong việc chọn hệ điều hành, hệ quản trị CSDL, webserver sẽ chạy ứng dụng, ... Ví dụ, người ta thường quan tâm nhiều đến các lỗ hổng về an toàn trên IIS hơn là quan tâm đến các đoạn mã của ứng dụng có tiềm ẩn các lỗ hổng nghiêm trọng hay không. Một trong số các lỗ hổng này đó là SQL injection attack.

SQL injection là một kĩ thuật cho phép những kẻ tấn công thi hành các câu lệnh truy vấn SQL bất hợp pháp (không được người phát triển lường trước) bằng cách lợi dụng lỗ hổng trong việc kiểm tra dữ liệu nhập trong các ứng dụng web. Hậu quả của nó rất tai hại vì nó cho phép những kẻ tấn công có thể thực hiện các thao tác xóa, hiệu chỉnh, ... do có toàn quyền trên cơ sở dữ liệu của ứng dụng. Lỗi này thường xảy ra trên các ứng dụng web có dữ liệu được quản lý bằng các hệ quản trị CSDL như SQL Server, Oracle, DB2, Sybase.

Xét một ví dụ điển hình, thông thường để cho phép người dùng truy cập vào các trang web được bảo mật, hệ thống thường xây dựng trang đăng nhập để yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu. Sau khi người dùng nhập thông tin vào, hệ thống sẽ kiểm tra tên đăng nhập và mật khẩu có hợp lệ hay không để quyết định cho phép hay từ chối thực hiện tiếp.

Trong trường hợp này, người ta có thể dùng 2 trang, một trang HTML để hiển thị form nhập liệu và một trang ASP dùng để xử lý thông tin nhập từ phía người dùng. Ví dụ:

#### Login.htm

```
<form action="ExecLogin.asp" method="post">
```

```
Username: <input type="text" name="txtUsername"><br>
Password: <input type="password" name="txtPassword"><br>
<input type="submit">
</form>
```

**ExecLogin.asp**

```
<%
Dim p_strUsername, p_strPassword, objRS, strSQL

p_strUsername = Request.Form("txtUsername")
p_strPassword = Request.Form("txtPassword")

strSQL = "SELECT * FROM tblUsers " & _
"WHERE Username=''' & p_strUsername & _ 
'' and Password=''' & p_strPassword & ''"

Set objRS = Server.CreateObject("ADODB.Recordset")
objRS.Open strSQL, "DSN=..."

If (objRS.EOF) Then
    Response.Write "Invalid login."
Else
    Response.Write "You are logged in as " & objRS("Username")
End If

Set objRS = Nothing
%>
```

Thoạt nhìn, đoạn mã trong trang ExecLogin.asp dường như không chứa bất cứ một lỗ hổng về an toàn nào. Người dùng không thể đăng nhập mà không có tên đăng nhập và mật khẩu hợp lệ. Tuy nhiên, đoạn mã này thực sự không an toàn và là tiền đề cho một SQL injection attack. Đặc biệt, chỗ sơ hở nằm ở chỗ dữ liệu nhập vào từ người dùng được dùng để xây dựng trực tiếp câu lệnh truy vấn SQL. Chính điều này cho phép những kẻ tấn công có thể điều khiển câu truy vấn sẽ được thực hiện.

Ví dụ, nếu người dùng nhập chuỗi sau vào trong cả 2 ô nhập liệu username/password của trang Login.htm: **' or ' = '**. Lúc này, câu truy vấn sẽ được gọi thực hiện là:

```
SELECT * FROM tblUsers WHERE Username=" or "=" and Password = " or "="
```

Câu truy vấn này là hợp lệ và sẽ trả về tất cả các bản ghi của tblUsers và đoạn mã tiếp theo xử lý người dùng đăng nhập bắt hợp pháp này như là người dùng đăng nhập hợp lệ.

Một ví dụ khác của SQL injection attack nữa là khi các trang web sử dụng dữ liệu nhập vào theo dạng querystring (bằng cách gõ cặp tham số và giá trị trực tiếp trên thanh địa chỉ hoặc dùng form với thuộc tính ACTION là GET). Ví dụ sau minh họa một trang ASP nhận dữ liệu cho biến ID thông qua querystring và phát sinh nội dung của trang đó dựa trên ID:

```
<%
Dim p_IngID, objRS, strSQL
p_IngID = Request("ID")

strSQL = "SELECT * FROM tblArticles WHERE ID=" & p_IngID

Set objRS = Server.CreateObject("ADODB.Recordset")
objRS.Open strSQL, "DSN=..."

If (Not objRS.EOF) Then Response.Write objRS("ArticleContent")

Set objRS = Nothing
%>
```

Trong các tình huống thông thường, đoạn mã này hiển thị nội dung của article có ID trùng với ID được chuyển đến cho nó dưới dạng querystring. Ví dụ, trang này có thể được gọi như sau:

<http://www.example.com/Article.asp?ID=1055>, để hiển thị nội dung của article có ID là 1055.

Giống như ví dụ đăng nhập ở trước, đoạn mã này để lộ sơ hở cho một SQL injection attack. Kẻ tấn công có thể thay thế một ID hợp lệ bằng cách gán ID cho một giá trị khác, để thực hiện một lệnh SQL bất hợp pháp, ví dụ như: **0 or 1=1** (nghĩa là, <http://www.example.com/Article.asp?ID=0 or 1=1>).

Câu truy vấn SQL lúc này sẽ trả về tất cả các article từ bảng dữ liệu vì nó sẽ thực hiện câu lệnh:

```
SELECT * FROM tblArticles WHERE ID=0 or 1=1
```

Tất nhiên ví dụ này dường như không có gì nguy hiểm, nhưng hãy thử tưởng tượng kẻ tấn công có thể xóa toàn bộ CSDL bằng cách chèn vào các đoạn lệnh nguy hiểm như lệnh DELETE. Tất cả chỉ là đơn giản thay đổi chuỗi gán dữ liệu cho ID, ví dụ như:

```
http://www.example.com/Article.asp?ID=1055; DELETE FROM
tblArticles.
```

## 2. Các tác hại và cách phòng tránh

Tác hại từ SQL Injection attack tùy thuộc vào môi trường và cách cấu hình hệ thống. Nếu ứng dụng sử dụng quyền dbo (quyền của người sở hữu CSDL - owner) khi thao tác dữ liệu, nó có thể xóa toàn bộ các bảng dữ liệu, tạo các bảng dữ liệu mới, ... Nếu ứng dụng sử dụng quyền sa (quyền quản trị hệ thống), nó có thể điều khiển toàn bộ hệ quản trị CSDL và với quyền hạn rộng lớn như vậy nó có thể tạo ra các tài khoản người dùng bất hợp pháp để điều khiển hệ thống của bạn.

Để phòng tránh các nguy cơ có thể xảy ra, hãy bảo vệ các câu truy vấn SQL là bằng cách kiểm soát chặt chẽ tất cả các dữ liệu nhập nhận được từ đối tượng Request (Request, Request.QueryString, Request.Form, Request.Cookies, and Request.ServerVariables).

- Trong trường hợp dữ liệu nhập vào là chuỗi, như trong ví dụ 1, lỗi xuất phát từ việc có dấu nháy đơn trong dữ liệu. Để tránh điều này, thay thế các dấu nháy đơn bằng hàm Replace để thay thế bằng 2 dấu nháy đơn:

```
p_strUsername = Replace(Request.Form("txtUsername"), "", "")  
p_strPassword = Replace(Request.Form("txtPassword"), "", "")
```

- Trong trường hợp dữ liệu nhập vào là số, như trong ví dụ 2, lỗi xuất phát từ việc thay thế một giá trị được tiên đoán là dữ liệu số bằng chuỗi chứa câu lệnh SQL bất hợp pháp. Để tránh điều này, đơn giản hãy kiểm tra dữ liệu có đúng kiểu hay không:

```
p_lngID = CLng(Request("ID"))
```

Như vậy, nếu người dùng truyền vào một chuỗi, hàm này sẽ trả về lỗi ngay lập tức.

Ngoài ra để tránh các nguy cơ từ SQL Injection attack, nên chú ý loại bỏ bất kì thông tin kĩ thuật nào chứa trong thông điệp chuyển xuống cho người dùng khi ứng dụng có lỗi. Các thông báo lỗi thông thường tiết lộ các chi tiết kĩ thuật có thể cho phép kẻ tấn công biết được điểm yếu của hệ thống.

Cuối cùng, để giới hạn mức độ của SQL Injection attack, nên kiểm soát chặt chẽ và giới hạn quyền xử lý dữ liệu đến tài khoản người dùng mà ứng dụng web đang sử dụng. Các ứng dụng thông

thường nên tránh dùng đến các quyền như dbo hay sa. Quyền càng bị hạn chế, thiệt hại càng ít.

### Các tài liệu tham khảo

SQL Injection FAQ:

<http://www.sqlsecurity.com/DesktopDefault.aspx?tabindex=2&tabid=3>

Advanced SQL Injection :

[http://www.nextgenss.com/papers/advanced\\_sql\\_injection.pdf](http://www.nextgenss.com/papers/advanced_sql_injection.pdf)

Preventing SQL Injection:

[http://www.owasp.org/asac/input\\_validation/sql.shtml](http://www.owasp.org/asac/input_validation/sql.shtml)

Biên dịch từ:

<http://www.4guysfromrolla.com/webtech/061902-1.shtml>

## Chương 7

# ỨNG DỤNG MINH HỌA CỦA HÀNG SÁCH TRỰC TUYẾN

### 1. Giới thiệu về ứng dụng

Trong phần này, chúng ta sẽ cùng thực hành để tạo ra một ứng dụng cửa hàng sách trực tuyến trên mạng. Người dùng có thể xem các cuốn sách có trong cơ sở dữ liệu, chọn mua, đặt hàng, ...

### 2. Tổ chức website ứng dụng

- Tạo thư mục C:\MyWebSite\BookStore trong C:\MyWebSite. Đây là thư mục chứa tất cả các mã nguồn của ứng dụng
- Tạo thư mục C:\MyWebSite\BookStore\BookPic trong C:\MyWebSite\BookStore, sau đó chép các tập tin lưu ảnh bìa của các sách vào thư mục này. Thư mục này dùng để lưu các ảnh bìa của các cuốn sách trong cơ sở dữ liệu.
- Tạo thư mục C:\MyWebSite\BookStore\DB trong C:\MyWebSite\BookStore, sau đó tạo tập tin BookStoreDB.mdb trong thư mục này

### 3. Thiết kế dữ liệu

Tập tin CSDL BookStoreDB.mdb bao gồm 3 bảng: T\_BOOK, T\_CATEGORY, T\_PUB lưu trữ thông tin về danh mục sách, chủ đề và nhà xuất bản như sau:

#### T\_BOOK

TT	Thuộc tính	Kiểu DL	Mô tả
1	BOOK_ID	AutoNumber	Định danh cho mỗi cuốn sách. Primary Key
2	BOOK_TITLE	Char(255)	Tên sách
3	BOOK_DESC	Memo	Tóm tắt nội dung sách

TT	Thuộc tính	Kiểu DL	Mô tả
4	BOOK_CATID	Number	Mã chủ đề mà sách thuộc vào. Qui ước: Mỗi sách chỉ thuộc về một chủ đề duy nhất
5	BOOK_AUTHOR	Char(128)	Danh sách tên tác giả
6	BOOK_PUBID	Number	Mã nhà xuất bản, nơi xuất bản cuốn sách này
7	BOOK_YEAR	Number	Năm xuất bản cuốn sách
8	BOOK_PIC	Char(255)	Đường dẫn đến tập tin lưu ảnh bìa của cuốn sách
9	BOOK_PRICE	Number	Giá tiền
10	BOOK_RATE	Number	Đánh giá về sách

#### T\_CATEGORY

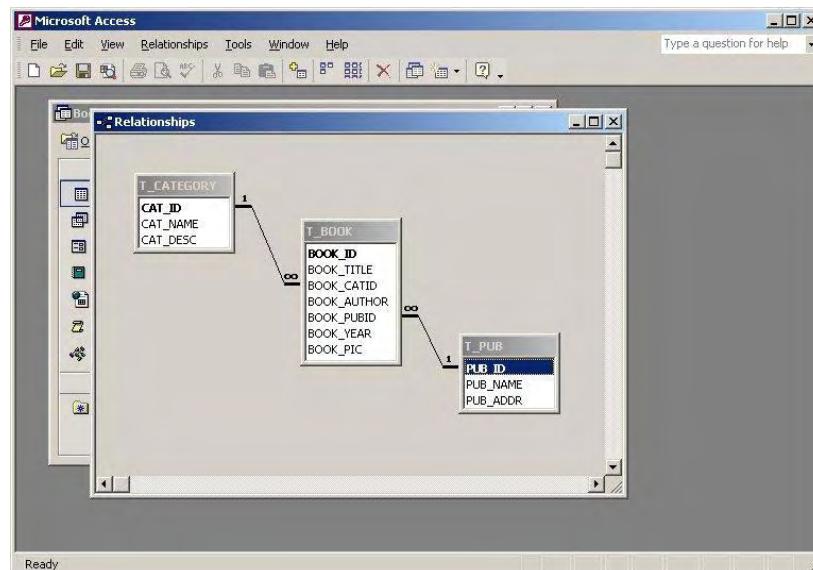
TT	Thuộc tính	Kiểu DL	Mô tả
1	CAT_ID	AutoNumber	Định danh cho chủ đề. Primary Key
2	CAT_NAME	char(64)	Tên chủ đề
3	CAT_DESC	char(255)	Mô tả chủ đề

#### T\_PUB

TT	Thuộc tính	Kiểu DL	Mô tả
1	PUB_ID	AutoNumber	Định danh cho nhà xuất bản. Primary Key
2	PUB_NAME	char(64)	Tên nhà xuất bản

TT	Thuộc tính	Kiểu DL	Mô tả
3	PUB_ADDR	char(255)	Thông tin về địa chỉ NXB

Tạo relationship cho các table và nhập liệu cho các bảng dữ liệu trên



Hình 1 – Relationship giữa các bảng

#### 4. Kết nối với cơ sở dữ liệu (CSDL)

Do thao tác kết nối với CSDL luôn được dùng trong các tập tin có nhu cầu thao tác trên CSDL nên ta sẽ tổ chức riêng thành một tập tin gọi là dbconn.asp chứa đoạn mã kết nối với CSDL. Sau đó trong các tập tin cần dùng ta sẽ include tập tin này vào.

Để mô tả đường dẫn đến tập tin CSDL BookStoreDB đã tạo ở trên, ta dùng hàm Server.MapPath.

Tạo tập tin dbconn.asp với đoạn mã như sau:

```
<%
'On Error Resume Next
```

```
strDSN = "Driver={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.MapPath("DB\BookStoreDB.mdb")
```

```
'create a connection object
Set Conn = Server.CreateObject("ADODB.Connection")
```

```
'open a connection using ODBC connection string
Conn.Open strDSN
%>
```

Lưu ý rằng, hàm Server.MapPath(path) được dùng để ánh xạ đường dẫn path thành đường dẫn đến thư mục vật lý trên server. Ví dụ ta có thư mục Home Directory là C:\MyWebSite, thư mục vật lý hiện hành của tập tin dùng hàm Server.MapPath là C:\MyWebSite\BookStore thì:

- Server.MapPath("BookStoreDB.mdb") sẽ trả về C:\MyWebSite\BookStore\BookStoreDB.mdb
- Server.MapPath("DB\BookStoreDB.mdb") sẽ trả về C:\MyWebSite\BookStore\DB\BookStoreDB.mdb
- Server.MapPath("/BookStoreDB.mdb") sẽ trả về C:\MyWebSite\BookStoreDB.mdb
- Server.MapPath("../BookStoreDB.mdb") sẽ trả về C:\BookStoreDB.mdb

#### 5. Thao tác truy vấn trên cơ sở dữ liệu

Tạo tập tin listallbook.asp dùng để liệt kê các cuốn sách có trong CSDL

Các thao tác cơ bản

- Câu lệnh SQL dùng để thực thi là: SELECT \* FROM T\_BOOK
- Kết quả trả về của câu lệnh SELECT sẽ là tập các mẫu tin chứa trong biến Recordset. Ta sẽ dùng một vòng lặp để liệt kê các mẫu tin này theo dạng bảng

Đoạn mã của tập tin listallbook.asp như sau:

```
<%@CODEPAGE=65001%>
<%Response.Charset = "utf-8"%>
<!--#include file="dbconn.asp"-->
<%
```

```

sPicPath = "BookPic/"
strSQL = "SELECT * FROM T_BOOK"

Set rs = Conn.Execute(strSQL)
%>
<table width="600" border="1" cellpadding="5" cellspacing="0"
style="border-collapse: collapse" bordercolor="#111111">
|
|  |

```

- Do trong mã chương trình trên có tiếng Việt Unicode nên tập tin cần được lưu dưới dạng Unicode (UTF-8)

- Hai dòng đầu tiên đảm bảo cho việc hiển thị tiếng Việt Unicode (xem thêm bài tham khảo Lập trình web ASP với tiếng Việt Unicode)
- Do các tập tin ảnh bìa tương ứng với các cuốn sách được lưu trong thư mục BookPic (lưu trong biến sPicPath) nên đường dẫn đến các tập tin này phải là : sPicPath & rs("BOOK\_PIC")
- Mở rộng đoạn mã trên sao cho trong bảng kết quả trả về, có thêm thông tin của nhà xuất bản, chủ đề mà cuốn sách thuộc về (xem bài giải ở tập tin listallbookext.asp)

## 6. Thao tác thêm một mẫu tin vào cơ sở dữ liệu

Giả sử muốn thêm một mẫu tin mới vào table T\_BOOK với các thông tin: Tên sách là "**Thiết kế và lập trình ứng dụng web bằng ASP**", chủ đề của sách là Tin học có ID là 1, tác giả là "**Lê Đình Duy**", nhà xuất bản là Thống kê có ID là 3, năm xuất bản **2001**, tập tin ảnh bìa tên là **Itwebasp.jpg**.

Ta tạo tập tin **addbook.asp** như sau:

```

<%@CODEPAGE=65001%>
<%Response.Charset = "utf-8"%>
<!--#include file="dbconn.asp"-->
<%
' get input data

vBookTitle = "Thiết kế và lập trình ứng dụng web bằng ASP"
vBookCatID = 1
vBookAuthor = "Lê Đình Duy"
vBookPubID = 3
vBookYear = 2001
vBookPrice = 16000
vBookPic = "Itwebasp.jpg"

' build SQL statement
strSQL = "INSERT INTO T_BOOK(BOOK_TITLE, BOOK_CATID,
BOOK_AUTHOR, BOOK_PUBID, BOOK_YEAR, BOOK_PRICE,
BOOK_PIC)"
strSQL = strSQL & "VALUES("
strSQL = strSQL & "" & vBookTitle & "" & ", "
strSQL = strSQL & vBookCatID & ", "
strSQL = strSQL & "" & vBookAuthor & "" & ", "
strSQL = strSQL & vBookPubID & ", "
strSQL = strSQL & vBookYear & ", "

```

```

strSQL = strSQL & vBookPrice & ", "
strSQL = strSQL & "" & vBookPic & """
strSQL = strSQL & ")"
Response.Write "Thực hiện thao tác tạo mới với câu lệnh SQL: " &
strSQL

' execute SQL statement
Conn.Execute strSQL
%>

<%
Conn.Close
Set Conn = NOTHING
%>

```

- Thi hành tập tin **addbook.asp**, sau đó thi hành tập tin **listallbookext.asp** để xem lại thông tin của mẫu tin mới được thêm vào.
- Lưu ý rằng trong câu lệnh INSERT, các dữ liệu kiểu chuỗi phải được rào quanh bởi cặp dấu nháy đơn (').
- Mở rộng đoạn mã trên bằng cách tạo một form nhập liệu để nhập các thông tin trên. Trong đó, chương trình sẽ liệt kê các chủ đề, nhà xuất bản có trong CSDL trong một dropdown listbox để người dùng lựa chọn (xem bài giải ở tập tin **addbookform.asp** và **addbookext.asp**)

## 7. Thao tác cập nhật một mẫu tin trên cơ sở dữ liệu

Để cập nhật một cuốn sách trong CSDL, cần phải trao vào mã sách BOOK\_ID. Giả sử muốn cập nhật mẫu tin trong table T\_BOOK có **BOOK\_ID=1** với các thông tin: Tên sách là "**Thiết kế và lập trình ứng dụng web bằng ASP**", chủ đề của sách là Tin học có ID là 1, tác giả là "**Lê Đình Duy**", nhà xuất bản là Thông kê có ID là 3, năm xuất bản **2001**, tập tin ảnh bìa tên là **ltwebasp.jpg**.

Ta tạo tập tin **updatebook.asp** như sau:

```

<%@CODEPAGE=65001%>
<%Response.Charset = "utf-8"%>
<!--#include file="dbconn.asp"-->
<%
' get input data

vBookID = 1

```

```

vBookTitle = "Thiết kế và lập trình ứng dụng web bằng ASP"
vBookCatID = 1
vBookAuthor = "Lê Đình Duy"
vBookPubID = 3
vBookYear = 2001
vBookPrice = 16000
vBookPic = "ltwebasp.jpg"

' build SQL statement
strSQL = "UPDATE T_BOOK SET "
strSQL = strSQL & "BOOK_TITLE=" & "" & vBookTitle & "" & ", "
strSQL = strSQL & "BOOK_CATID=" & vBookCatID & ", "
strSQL = strSQL & "BOOK_AUTHOR=" & "" & vBookAuthor & "" &
", "
strSQL = strSQL & "BOOK_PUBID=" & vBookPubID & ", "
strSQL = strSQL & "BOOK_YEAR=" & vBookYear & ", "
strSQL = strSQL & "BOOK_PRICE=" & vBookPrice & ", "
strSQL = strSQL & "BOOK_PIC=" & "" & vBookPic & ""
strSQL = strSQL & " WHERE BOOK_ID=" & vBookID
Response.Write "Thực hiện thao tác cập nhật với câu lệnh SQL: " &
strSQL

' execute SQL statement
Conn.Execute strSQL
%>

<%
Conn.Close
Set Conn = NOTHING
%>

```

- Thi hành tập tin **updatebook.asp**, sau đó thi hành tập tin **listallbook.asp** để xem mẫu tin mới được cập nhật.
- Mở rộng đoạn mã trên bằng cách tạo một form cho phép người dùng hiệu chỉnh các thông tin của cuốn sách có **BOOK\_ID=1**. Sau khi người dùng sửa xong, nhấn nút Submit, các thông tin vừa sửa sẽ được cập nhật lên CSDL. (xem bài giải ở tập tin **updatebookform.asp** và tập tin **updatebookext.asp**).
- Lưu ý rằng, với trang **updatebookform.asp** chứa form hiển thị các dữ liệu của cuốn sách cần cập nhật thông tin, ta có thể sử dụng lại form đã dùng trong trang **addbookform.asp**.

Điểm khác nhau là thêm thuộc tính value cho các control với các giá trị đọc từ CSDL của cuốn sách cần cập nhật.

- Để lưu thông tin BOOK\_ID của cuốn sách cần cập nhật trong form của trang **updatebookform.asp** ta sử dụng một control có type là hidden (`<input type="hidden" name="fBOOK_ID" value="<%Response.Write rsBook("BOOK_ID")%>">`).
- Trong trang **updatebookform.asp** ta lấy dữ liệu của fBOOK\_ID từ query string bằng cách gõ vào cặp tên control và giá trị tương ứng khi dùng tập tin này. Ví dụ: [http://..../updatebookform.asp?fBOOK\\_ID=1](http://..../updatebookform.asp?fBOOK_ID=1)
- Trong trang **updatebookext.asp** dùng để xử lý các dữ liệu từ form của trang **updatebookform.asp**, ta thực hiện tương tự như trang **addbookext.asp**. Điểm khác nhau ở chỗ thay câu lệnh INSERT bằng câu lệnh UPDATE.

Tạo một form tìm kiếm sách, cho phép người dùng nhập vào các thông tin như là tên sách, tên tác giả, chủ đề, nhà xuất bản, năm xuất bản. Chương trình sẽ liệt kê các cuốn sách tìm được theo yêu cầu của người dùng. Trong mỗi cuốn sách được liệt kê, tạo một hyperlink sao cho khi nhấp vào đó, chương trình sẽ hiển thị form cập nhật dữ liệu của cuốn sách trên (xem bài giải ở tập tin **searchbookform.asp** và tập tin **searchbook.asp**).

- Trang **searchbookform.asp** chứa form hiển thị các dữ liệu mà người dùng cần nhập vào để tìm ra cuốn sách cần cập nhật thông tin, ta có thể sử dụng lại form đã dùng trong trang **addbookform.asp**. Lưu ý thêm rằng, đối với các dropdown listbox ta cần phải có thêm mục chọn Tất cả
- Để tìm gần đúng các chuỗi 'tin' trong một trường cho trước ví dụ **BOOK\_TITLE**, ta dùng kí tự đại diện %. Ví dụ **SELECT \* FROM T\_BOOK WHERE BOOK\_TITLE LIKE "%tin%"**
- Trong tập tin **searchbook.asp**, sau khi hiển thị các kết quả tìm được theo dạng bảng, ta thêm một hyperlink ngay số thứ tự của mỗi cuốn sách. Mỗi hyperlink này sẽ chứa địa chỉ của trang cập nhật sách dạng: [updatebookform.asp?fBOOK\\_ID=<book\\_id>](updatebookform.asp?fBOOK_ID=<book_id>)

## 8. Thao tác xóa một mẫu tin trên cơ sở dữ liệu

Giả sử muốn xóa một mẫu tin trong table T\_BOOK có BOOK\_ID=1.

Ta tạo tập tin **delbook.asp** như sau:

```
<%@CODEPAGE=65001%>
<%Response.Charset = "utf-8"%>
<!--#include file="dbconn.asp"-->
<%
' get input data

vBookID = 1

' build SQL statement
strSQL = "DELETE FROM T_BOOK "
strSQL = strSQL & "WHERE BOOK_ID=" & vBookID
Response.Write "Thực hiện thao tác xóa với câu lệnh SQL: " &
strSQL

' execute SQL statement
Conn.Execute strSQL
%>

<%
Conn.Close
Set Conn = NOTHING
%>
```

- Thi hành tập tin **delbook.asp**, sau đó thi hành tập tin **listallbook.asp** để kiểm tra mẫu tin đã được xóa hay chưa
- Mở rộng tập tin **delbookext.asp** để nhận thông tin của BOOK\_ID cần xóa.

Tương tự như trang cập nhật sách, ta có thể truyền BOOK\_ID cho trang xóa sách bằng cách thay đổi hyperlink của trong trang **searchbook.asp** từ **updatebookform.asp** thành **delbookext.asp**.

## 9. Sử dụng Cookies

### 9.1. Nhắc lại khái niệm về Session, Application, Cookies

Nếu một biến được khai báo trong một trang ASP, nó sẽ bị hủy ngay khi server xử lý xong trang này. Để có thể khai báo các biến có thể sử dụng chung cho nhiều trang, ta sử dụng hai đối tượng Application và Session.

Đối tượng Application để lưu trữ các thông tin có thể được chia sẻ cho nhiều người dùng trong cùng một ứng dụng. Một ứng dụng ASP được định nghĩa như là các trang asp nằm trong cùng một thư mục ảo và các thư mục con của thư mục ảo này. Do các biến được lưu trong đối tượng Application được dùng chung cho nhiều người dùng, do đó một khi một người dùng muốn thay đổi một giá trị dùng chung này, các yêu cầu phải được đặt trong cặp lệnh Lock và Unlock. Điều này đảm bảo an toàn khi nhiều người cùng yêu cầu thay đổi của cùng một giá trị trong cùng một lúc.

Đối tượng Session được dùng để lưu trữ các thông tin dùng cho một phiên làm việc của một người dùng. Các biến được lưu trong đối tượng này sẽ không bị hủy khi người dùng chuyển từ trang này sang trang khác.

Webserver sẽ tự động tạo ra một đối tượng Session khi người dùng yêu cầu một trang web từ một ứng dụng mà trước đó chưa có phiên làm việc nào. Webserver sẽ tự động hủy đối tượng Session này khi đối tượng này hết hạn.

Đối tượng Session thường được dùng để lưu các thông tin riêng biệt cho từng phiên làm việc của người dùng. Ví dụ như trong ứng dụng mua hàng trên mạng, mỗi khi người dùng duyệt qua các mặt hàng và quyết định chọn nó, rồi lại tiếp tục qua các trang khác để chọn mặt hàng khác, ... Các thông tin liên quan đến các mặt hàng mà người dùng đã chọn từ nhiều trang khác nhau nên được lưu trong biến của đối tượng Session để sau này in ra phiếu mua hàng cho người dùng.

Để đặt thời gian hết hạn của một đối tượng Session, ta đặt giá trị (tính bằng phút) cho thuộc tính Session.TimeOut. Để hủy Session hiện hành, ta dùng phương thức Session.Abandon

Cookie là thông tin của ứng dụng web lưu trữ tại máy client. Thông thường các cookie lưu trữ các thông tin riêng của bạn khi sử dụng một website nào đó. Ví dụ, nếu bạn dùng Yahoo Mail, thông tin về tên đăng nhập của bạn có thể sẽ được lưu vào cookie máy bạn đang dùng để nhớ tên đăng nhập của bạn. Đó là lí do tại sao sau này khi quay lại trang Yahoo Mail, hệ thống sẽ tự động hiển thị tên đăng nhập của bạn trước đó và chỉ yêu cầu bạn nhập vào mật khẩu.

Với giao thức HTTP, mỗi yêu cầu cho một trang web không có mối liên quan nào với các yêu cầu khác. Chính vì vậy mà webserver không nhớ được trang nào đã được gửi đến cho người dùng trước

đó hay bắt cứ những gì mà người dùng đã duyệt qua. Cookies là một cơ chế cho phép server lưu trữ các thông tin của nó về người dùng trên chính máy của người dùng hiện tại.

## 9.2. Phân quyền truy cập cho người dùng

Xem chi tiết trong bài tham khảo Phương pháp hỗ trợ giới hạn truy cập trang web

Ta sẽ tạo thêm một bảng dữ liệu lưu thông tin về người dùng của hệ thống để phục vụ cho việc phân quyền. Bảng này có tên là T\_USER với các trường như sau:

TT	Thuộc tính	Kiểu DL	Mô tả
1	USER_ID	Char(10)	Tên đăng nhập của người dùng. Primary Key
2	USER_PASSWORD	Char(10)	Mật khẩu đăng nhập
3	USER_NAME	Char(50)	Tên đầy đủ của người dùng
4	USER_EMAIL	Char(30)	Địa chỉ email
5	USER_PHONE	Char(30)	Điện thoại liên lạc
6	USER_ADDR	Char(255)	Địa chỉ liên lạc

Tạo trang login.htm hiển thị form đăng nhập để yêu cầu người dùng nhập vào tên đăng nhập và mật khẩu. Trong trang này, khai báo action của form là login.asp, là trang sẽ xử lý các dữ liệu do người dùng nhập vào

```
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
<title>Đăng nhập</title>
<style>
```

```

<!--
.txtText { font-family: Times New Roman; font-size: 12pt; color: #0000FF; font-weight: bold }
.txtControl { font-family: Arial; font-size: 10pt; color: #0000FF }
-->
</style>
</head>
<body>
<center>
<form action="login.asp" method="POST">
<table border="0" cellpadding="5" cellspacing="0" style="border-collapse: collapse" bordercolor="#111111" width="400"
id="AutoNumber1">
<tr>
<td width="600" colspan="2" bgcolor="#0000FF">
<p align="center"><b><font face="Tahoma" size="5"
color="#FFFFFF">Đăng nhập</font></b></td>
</tr>
<tr>
<td width="190" align="right" bgcolor="#CCCCCC"
class="txtText">Tên đăng nhập</td>
<td width="410" bgcolor="#CCCCCC">
<input type="text" size="20" name="fUSER_ID"
class="txtControl"></td>
</tr>
<tr>
<td width="190" align="right" bgcolor="#CCCCCC"
class="txtText">Mật khẩu</td>
<td width="320" bgcolor="#CCCCCC">
<input type="text" size="20" name="fUSER_PASSWORD"
class="txtControl"></td>
</tr>
<tr>
<td width="600" bgcolor="#CCCCCC" colspan="2" align="center">
<input type="submit" value="Đăng nhập" name="fb1"
class="txtControl">&ampnbsp&ampnbsp;
<input type="reset" value=" Làm lại " name="fb2"
class="txtControl"></td>
</tr>
</table>
</form>
</center>
</body>
</html>

```

Tạo trang login.asp để kiểm tra tên đăng nhập và mật khẩu nhập vào có hợp lệ hay không. Nếu hợp lệ, ngoài thông báo thành công, trang này sẽ gán giá trị cho các biến Session lưu thông tin về đăng nhập hợp lệ của người dùng.

```

<%@CODEPAGE=65001%>
<%Response.Charset = "utf-8"%>
<html>

<head>
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
<title>Đăng nhập</title>
<style>
<!--
.txtText { font-family: Times New Roman; font-size: 12pt; color: #0000FF; font-weight: bold }
.txtControl { font-family: Arial; font-size: 10pt; color: #0000FF }
-->
</style>
</head>

<body>
<!--#include file="dbconn.asp"-->
<%
' get input data

vUserID = Request("fUSER_ID")
vUserPassword = Request("fUSER_PASSWORD")

' build SQL statement
strSQL = "SELECT USER_ID, USER_PASSWORD FROM T_USER
WHERE USER_ID = " & "" & vUserID & ""
set rs = Conn.Execute(strSQL)

if rs.EOF then
    Response.Write "<p class=txtText>Không tồn tại tên đăng nhập.
Hãy <a href='login.htm'>đăng nhập lại</a>"
else
    if rs("USER_PASSWORD") <> vUserPassword then
        Response.Write "<p class=txtText>Mật khẩu nhập vào sai. Hãy
<a href='login.htm'>đăng nhập lại</a>"
```

```

else
    Response.Write "<p class=txtText>Đăng nhập thành công. Hãy
    <a href='>làm tiếp</a>"
    Session("LoginOK") = 1
end if
end if
%>

<%
rs.Close
Set rs = NOTHING
Conn.Close
Set Conn = NOTHING
%>
</body>
</html>

```

Trong bất kì trang web nào (giả sử là **authenpage.asp**) mà bạn chỉ muốn người dùng được phép truy cập một khi đã đăng nhập thành công, đặt dòng lệnh kiểm tra sau ngay đầu trang web như sau:

```

<%@CODEPAGE=65001%>
<%Response.Charset = "utf-8"%>

<html>

<head>
<meta http-equiv="Content-Language" content="en-us">
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Nhập mới sách</title>
<style>
<!--
.txtText { font-family: Times New Roman; font-size: 12pt; color: #0000FF; font-weight: bold }
.txtControl { font-family: Arial; font-size: 10pt; color: #0000FF }
-->
</style>
</head>

<body>
<%
if Session("LoginOK") <> 1 then

```

```

    Response.Write "<p class=txtText>Bạn không có quyền thực
    hiện thao tác này. Hãy <a href='login.htm'>đăng nhập</a>"
    Response.End
end if
%>
<!--Phần còn lại của trang đặt từ đây trở đi-->
</body>
</html>

```

Mở rộng các trang trên sao cho khi người dùng truy cập vào trang **authenpageext.asp** mà chưa đăng nhập thì sẽ được yêu cầu đăng nhập. Sau khi đăng nhập xong, cho phép quay trở lại trang **authenpageext.asp** để thực hiện tiếp. Ngoài ra, sử dụng thêm cookies để khi người dùng quay trở lại trang **loginform.asp** lần sau, hệ thống sẽ tự động nhớ lại tên người dùng đã đăng nhập trước đó (xem bài giải ở trang **authenpageext.asp**, trang **loginform.asp** và **loginext.asp**).

### 9.3. Tạo và quản lí giỏ đi chợ (shopping cart)

Ta sẽ tạo ra trang **booklist.asp** (ké thừa từ trang **searchbook.asp**) để liệt kê các cuốn sách cho người dùng chọn. Trên mỗi dòng, ta sẽ thêm vào một button **Add to Cart** để cho phép người dùng thêm cuốn sách đã chọn vào giỏ đi chợ. Để xem các cuốn sách đã được chọn mua, ta dùng trang **viewcart.asp**.

```

<%@CODEPAGE=65001%>
<%Response.Charset = "utf-8"%>
<!--#include file="dbconn.asp"-->
<%
vBookTitle = Request("fBOOK_TITLE")
vBookCatID = Request("fBOOK_CATID")
vBookAuthor = Request("fBOOK_AUTHOR")
vBookPubID = Request("fBOOK_PUBID")
vBookYear = Request("fBOOK_YEAR")

sPicPath = "BookPic/"
strSQL = "SELECT * FROM T_BOOK WHERE "
strSQL = strSQL & "BOOK_TITLE LIKE" & "%" & vBookTitle & "%"
strSQL = strSQL & " AND BOOK_AUTHOR LIKE" & "%" &
vBookAuthor & "%"

if vBookCatID <> "" then
    strSQL = strSQL & " AND BOOK_CATID =" & vBookCatID
end if

```

```

if vBookPubID <> "" then
    strSQL = strSQL & " AND BOOK_PUBID =" & vBookPubID
end if

if vBookYear <> "" then
    strSQL = strSQL & " AND BOOK_YEAR =" & vBookYear
end if
Set rs = Conn.Execute(strSQL)
%>

<table width="600" border="1" cellpadding="5" cellspacing="0"
style="border-collapse: collapse" bordercolor="#111111">
<tr align="center">
<td>TT</td>
<td>Tên sách</td>
<td>Tên tác giả</td>
<td>Năm XB</td>
<td>Giá tiền</td>
<td>Ảnh bìa</td>
<td>Mua hàng</td>
</tr>
<%
i=1
do while not rs.EOF
%>
<tr>
<td>
<form action="addtocart.asp" name="myform<%Response.Write i%>" method="post">
<a href="updatebookform.asp?fBOOK_ID=<%Response.Write rs("BOOK_ID")%>">>
<%Response.Write i%>
</a>
<input type="hidden" name="fBOOK_ID"
value="<%Response.Write rs("BOOK_ID")%> ">&nbsp;
</td>
<td><%Response.Write rs("BOOK_TITLE")%>&nbsp;</td>
<td><%Response.Write rs("BOOK_AUTHOR")%>&nbsp;</td>
<td><%Response.Write rs("BOOK_YEAR")%>&nbsp;</td>
<td><%Response.Write rs("BOOK_PRICE")%>&nbsp;</td>
<td><img src='<%Response.Write sPicPath &
rs("BOOK_PIC")%>'></td>
<td><input type="submit" value="Add to cart"
name="bAdd<%Response.Write i%> ">&nbsp;</td>

```

```

</form>
</tr>

<%
rs.movenext
i=i+1
loop
%><%
rs.Close
Set rs = NOTHING
Conn.Close
Set Conn = NOTHING
%>

```

Button Add to Cart này là button kiểu Submit liên kết với form có một control hidden chứa thông tin BOOK\_ID và action chỉ đến trang **addtocart.asp** để quản lý giỏ đi chợ

Để quản lý giỏ đi chợ, ta dùng một biến mảng 2 chiều lưu trong đối tượng Session. Biến mảng này một cột lưu mã sách BOOK\_ID, một cột lưu số lượng cần mua tương ứng với sách mà người dùng đã chọn. Ngoài ra dùng thêm biến NumBook để quản lý số lượng các cuốn sách đã được chọn

Lưu ý rằng, để thao tác trên biến mảng lưu trong Session, ta không thao tác trực tiếp mà phải chép dữ liệu từ biến Session ra mảng phụ trước khi dùng và cập nhật lại sau khi dùng xong. Kịch bản như sau:

```

arBook = Session("arBook") 'chép dữ liệu ra
// Thao tác trên arBookSelected ở đây
...
Session("arBook") = arBook 'cập nhật vào biến Session

```

Trang **addtocart.asp** sẽ làm nhiệm vụ lấy BOOK\_ID và cập nhật vào mảng arBook và NumBook

```

<%@CODEPAGE=65001%>
<%Response.Charset = "utf-8"%>
<html>

<head>
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

```

```

<title>Thêm sách vào giỏ hàng</title>
<style>
<!--
.txtText { font-family: Times New Roman; font-size: 12pt; color: #0000FF; font-weight: bold }
.txtControl { font-family: Arial; font-size: 10pt; color: #0000FF } 
-->
</style>
</head>

<body>

<%
' get input data

vBookID = Request("fBOOK_ID")
NumBook = Session("NumBook")
arBook = Session("arBook")

arBook(0, NumBook) = vBookID
arBook(1, NumBook) = 1 ' số lượng sách mua mặc định là 1
NumBook = NumBook + 1

Session("NumBook") = NumBook
Session("arBook") = arBook
%>

<%
%>
</body>

</html>

```

Trang **viewcart.asp** cho phép xem thông tin giỏ đi chợ về các mặt hàng đã chọn. Trang này sẽ lấy dữ liệu từ mảng các BOOK\_ID lưu trong biến Session, kết hợp với việc đọc CSDL để cho biết thông tin của từng mặt hàng.

```

<%@CODEPAGE=65001%>
<%Response.Charset = "utf-8"%>
<html>

<head>
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=utf-

```

```

8">
<title>Xem thông tin giỏ hàng</title>
<style>
<!--
.txtText { font-family: Times New Roman; font-size: 12pt; color: #0000FF; font-weight: bold }
.txtControl { font-family: Arial; font-size: 10pt; color: #0000FF } 
-->
</style>
</head>

<body>
<!--#include file="dbconn.asp"-->
<table width="600" border="1" cellpadding="5" cellspacing="0"
style="border-collapse: collapse" bordercolor="#111111">
<tr align="center">
<td>TT</td>
<td>Mã sách</td>
<td>Tên sách</td>
<td>Tên tác giả</td>
<td>Năm XB</td>
<td>Giá tiền</td>
<td>Số lượng</td>
</tr>
<%
NumBook = Session("NumBook")
arBook = Session("arBook")
Sum = 0
for i=0 to NumBook-1
%>
<tr>
<td>
<%Response.Write i%> &nbsp; </td>
<td><%Response.Write arBook(0, i)%>&nbsp; </td>
<td>
<%
strSQL = "SELECT * FROM T_BOOK WHERE BOOK_ID=" & arBook(0, i)
set rs = Conn.Execute (strSQL)
Response.Write rs("BOOK_TITLE")
%> &nbsp; </td>
<td><% Response.Write rs("BOOK_AUTHOR")%>&nbsp; </td>
<td><% Response.Write rs("BOOK_YEAR")%>&nbsp; </td>
<td><% Response.Write rs("BOOK_PRICE")%>&nbsp; </td>
<td><%Response.Write arBook(1, i)
```

```

Sum = Sum + rs("BOOK_PRICE")*arBook(1, i)
%>&nbsp;</td>
</tr>
<%
rs.Close
set rs = NOTHING
next
%>
<tr align="center">
<td></td>
<td></td>
<td></td>
<td></td>
<td>Tổng số tiền</td>
<td><%Response.Write Sum%>&nbsp;</td>
<td></td>
</tr>
</table>
<%
Conn.Close
set Conn = NOTHING
%>
</body>

</html>
</html>

```

Để khởi tạo các biến Session, ta sử dụng đoạn mã khởi tạo trong tập tin **global.asa**

```

<SCRIPT LANGUAGE=VBScript RUNAT=Server>

Sub Application_OnStart
End Sub

Sub Application_OnEnd
End Sub

Sub Session_OnStart
Session("LoginOK") = 0
Session("NumBook") = 0
Dim arBook(2, 10)
Session("arBook") = arBook
End Sub
Sub Session_OnEnd

```

```

End Sub
</SCRIPT>

```

#### 9.4. Một số mở rộng khác

- Mở rộng các trang trên cho phép khi người dùng chọn trùng một cuốn sách đã có sẵn trong giỏ đi chợ, hệ thống sẽ tự động tăng số lượng tương ứng với cuốn sách đó lên.
- Mở rộng bằng cách cho phép người dùng tăng số lượng sách có thể mua hơn số lượng khởi tạo biến mảng arBook ban đầu bằng cách ReDim lại mảng khi số lượng sách lớn hơn số lượng phần tử mảng khai báo ban đầu.
- Thay vì dùng biến Session để lưu trữ giỏ mua hàng của người dùng, hãy dùng một bảng tạm T\_CART để lưu các thông tin này.
- Viết thêm các chức năng cho phép người dùng cập nhật nội dung giỏ đi chợ như thay đổi số lượng đã chọn, loại bỏ sách đã chọn, ...
- Viết thêm chức năng in đơn đặt hàng khi người dùng dừng việc chọn mua hàng.

#### 10. Những câu hỏi thường gặp

**Hỏi:** Dữ liệu tiếng Việt nhập trực tiếp vào CSDL nhưng không hiển thị được lúc thực thi chương trình

**Trả lời:** Dữ liệu lưu trong Access là dạng UCS-2, do đó phải thiết lập Codepage=65001 và charset=utf-8 ở đầu trang asp để server chuyển dữ liệu từ UCS-2 sang UTF-8 trước khi gửi xuống cho trình duyệt của người dùng. (Xem thêm bài tham khảo Lập trình Web ASP với tiếng Việt Unicode)

**Hỏi:** Trong ứng dụng trên, ảnh của các cuốn sách được lưu trong một thư mục riêng. Vậy muốn lưu trực tiếp ảnh trong CSDL thì phải làm sao?

**Trả lời:** Xem thêm bài tham khảo Hiển thị hình ảnh từ CSDL trong các ứng dụng Web.

**Hỏi:** Khi thực hiện trang thao tác trên CSDL thì gấp thông báo lỗi:

Error Type:  
Microsoft OLE DB Provider for ODBC Drivers (0x80004005)

[Microsoft][ODBC Driver Manager] Data source name not found and no default driver specified  
**/BookStore/dbconn.asp, line 9**

**Trả lời:** Lỗi xảy ra do đường dẫn khai báo data source name không chính xác, hãy kiểm tra lại chuỗi khai báo data source name ở các mô tả tên driver, các thông số về cấu hình, ...

**Hỏi:** Khi thực hiện trang thao tác trên CSDL thì gặp thông báo lỗi:

Error Type:  
 Microsoft OLE DB Provider for ODBC Drivers (0x80004005)  
 [Microsoft][ODBC Driver Manager] Data source name not found and no default driver specified  
**/BookStore/dbconn.asp, line 9**

**Trả lời:** Lỗi xảy ra do đường dẫn khai báo data source name không chính xác, hãy kiểm tra lại chuỗi khai báo data source name ở các mô tả tên driver, các thông số về cấu hình, ...

**Hỏi:** Khi thực hiện trang thao tác trên CSDL thì gặp thông báo lỗi:

Error Type:  
 Microsoft OLE DB Provider for ODBC Drivers (0x80004005)  
 [Microsoft][ODBC Microsoft Access Driver] Operation must use an updateable query.  
**/BookStore/addbook.asp, line 27**

**Trả lời:** Lỗi xảy ra do tập tin CSDL chưa được đặt quyền ghi. Hãy vào Explorer để chỉnh lại Security cho phép Users có quyền trên tập tin này là Write

**Hỏi:** Khi thực hiện trang thao tác trên CSDL thì gặp thông báo lỗi

Error Type:  
 Microsoft OLE DB Provider for ODBC Drivers (0x80004005)  
 [Microsoft][ODBC Microsoft Access Driver]General error Unable to open registry key 'Temporary (volatile) Jet DSN for process 0x690 Thread 0xb50 DBC 0x7a9d8c Jet'.  
**/BookStore/dbconn.asp, line 9**

**Trả lời:** Có thể bạn đang thao tác trên tập tin CSDL đang được mở bởi ứng dụng khác, ví dụ như Access. Để khắc phục hãy đóng ứng dụng này lại và thực hiện lại.

Hoặc cũng có thể là bạn khai báo đường dẫn đến tập tin .mdb chưa chính xác. Hãy kiểm tra lại trong chuỗi khai báo bắt đầu từ Server.MapPath

## THỰC HÀNH

Hoàn chỉnh ứng dụng trên với các bổ sung:

1. Xây dựng trang chủ với các mục chọn chính của hệ thống
2. Bổ sung phần trình bày cho các bảng biểu kết xuất và các form nhập liệu
3. Thiết kế giỏ đi chợ không dùng biến Session mà dùng Cookies hoặc cập nhật trực tiếp trên cơ sở dữ liệu.
4. Thêm các bảng mới để quản lý các đơn đặt hàng của người dùng, các phiếu giao hàng.
5. Thêm các bảng mới để quản lý việc cung cấp hàng của các nhà cung cấp.
6. Thêm chức năng mới về quản lý doanh số bán hàng, báo cáo tồn kho, ...
7. Đăng ký một website hỗ trợ ASP miễn phí trên mạng và upload toàn bộ ứng dụng lên. Kiểm tra các vấn đề liên quan đến an toàn.

# ĐỒ ÁN

Xây dựng các đồ án sau:

- Website siêu thị tương tự như Amazon (<http://www.amazon.com>),  
Yahoo Shopping (<http://shopping.yahoo.com>),  
Minh Khai Pub (<http://www.minhkhainv.com>),  
NXB Giáo Dục (<http://www.nxbgd.com.vn>),  
VDC Siêu thị (<http://vdcsieuthi.vnn.vn>),  
ConfidantVCD (<http://www.confidantvcd.com>),  
Nhà sách VN (<http://www.nhasachvn.com>)
- Website cung cấp dịch vụ thông tin dạng báo điện tử như VnExpress (<http://vnexpress.net>),  
VietnamNet (<http://www.vnn.vn>),  
VDC Media (<http://media.vdc.com.vn>)
- Website cung cấp dịch vụ điểm tuyển sinh ĐH như NetSoft (<http://kgthidaihoc.netcenter-vn.net>),  
VDC-Moet (<http://tintuc.vnn.vn/tuyensinh>),
- Website quản lý danh bạ các địa chỉ web như FPTNet (<http://website.fptnet.com>),  
ContactVietnam (<http://www.contactvietnam.com>),  
BanCanBiet (<http://www.bancanbiet.com>)
- Website quản lý danh bạ tương tự như dịch vụ Những trang vàng (<http://www.yp.com.vn>),  
Danh bạ trực tuyến VDC (<http://danhba.vdc.com.vn>)
- Website về nhà đất như Nhã Đạt (<http://ww.nhadat.com>),
- Website về dịch vụ việc làm như:  
VASC (<http://laodongvieclam.vasc.com.vn>),  
Vietnam Works (<http://www.vietnamworks.com>)

## Bài tham khảo

### WEB HOSTING - DỊCH VỤ THUÊ CHỖ WEBSITE TRÊN INTERNET

Nếu công ty của bạn chuẩn bị thiết lập một website phục vụ cho việc quảng bá và kinh doanh trên mạng, việc đầu tiên là phải đăng ký một tên miền, dạng như mycompany.com. Sau khi đăng ký tên miền xong, để có thể đưa website lên Internet, bạn phải lựa chọn dịch vụ thuê chỗ website. Bài viết này sẽ cung cấp một số khái niệm cơ bản liên quan đến dịch vụ này, đồng thời đề xuất các bước cho việc lựa chọn dịch vụ thuê chỗ phù hợp với nhu cầu của công ty bạn

#### 1. Một số khái niệm cơ bản

Dịch vụ thuê chỗ website (web hosting) là khái niệm dùng để chỉ quá trình làm một website có thể truy cập được từ Internet. Để một website có thể truy cập được từ Internet, website phải được đặt trên một máy có cấu hình mạnh và có kết nối Internet thường xuyên. Vì chi phí để có được một máy như vậy không nhỏ nên mới xuất hiện các công ty cung cấp dịch vụ thuê chỗ phục vụ cho các công ty có nhu cầu đưa website của mình lên Internet.

Có hai hình thức thuê chỗ là miễn phí và phải trả tiền. Nếu sử dụng dịch vụ miễn phí, bạn sẽ phải chịu quảng cáo của người khác chen vào, đồng thời không thể sử dụng địa chỉ truy cập với tên miền đã đăng ký. Ví dụ, nếu bạn đăng ký dịch vụ thuê chỗ miễn phí của công ty Brinkster ([www.brinkster.com](http://www.brinkster.com)) thì địa chỉ truy cập website của bạn có dạng: <http://wwwxx.brinkster.com/myname>. Dịch vụ thuê chỗ miễn phí chỉ phù hợp cho các website của các cá nhân với mục đích thử nghiệm, không thể dùng cho các công ty muốn kinh doanh trên Internet.

Dịch vụ thuê chỗ có trả tiền cho phép bạn sử dụng tên miền đã đăng ký để truy cập vào website của mình, không có quảng cáo, có độ tin cậy cao và được bảo trì nâng cấp thường xuyên. Các website chuyên nghiệp đều tìm đến các dịch vụ hosting có chất lượng cao để đặt website của mình.

Có hai dạng thuê chỗ, một dạng là gọi là shared (virtual) hosting trong đó nhiều website cùng chia sẻ tài nguyên một server. Một dạng

khác bao gồm dedicated hosting và colocated hosting là thuê nguyên một máy tính và có toàn quyền thực hiện các thao tác trên máy này.

Hiện nay, thông dụng nhất chiếm đến 99% vẫn là shared hosting vì nó phù hợp cho hầu hết các công ty vừa và nhỏ với chi phí hợp lý.

Khi chọn dịch vụ webhosting cần lưu ý đến các tính năng mà các nhà cung cấp đưa ra:

- Dung lượng đĩa cứng (Space) là dung lượng tối đa cho phép dữ liệu website của bạn. Đơn vị tính được liệt kê trong các bảng thông tin dịch vụ thường là megabytes. Các tập tin chiếm dung lượng đĩa nhiều thông thường là các tập tin nhạc, video như .mp3, .avi, .mpeg, ...
- Băng thông (Transfer/Bandwidth) là tổng số những gì mà người dùng đã tải từ website của bạn khi truy cập. Thông thường nếu vượt quá giới hạn cho phép, bạn phải trả thêm tiền cho nhà cung cấp dịch vụ. Ví dụ: nếu một khách vào website của bạn xem 3 trang HTML mỗi trang có kích thước trung bình khoảng 20KB và 8 tấm hình nhúng trong đó mỗi tấm hình 10 KB thì với 500 khách truy cập một ngày, bạn sẽ cần ít nhất  $(3 * 20 + 8 * 10) * 500 * 30 = 2,100,000 \text{ KB} = 2.1 \text{ GB}$  băng thông mỗi tháng.
- Uptime là phần trăm thời gian một server làm việc. Ví dụ nếu bạn đặt website tại một server có uptime trung bình là 99.86%, điều này có nghĩa là website của bạn sẽ bị down khoảng 1h mỗi tháng. Hầu như không có dịch vụ thuê chổ dạng shared hosting nào có thể đảm bảo uptime 100%.

Sau khi trả tiền cho dịch vụ thuê chổ xong, các thao tác mà bạn thường phải làm đó là:

- Upload website lên server
- Quản lí các thông tin về truy cập website

Với các nhà cung cấp chất lượng cao, bạn sẽ được hỗ trợ các thao tác này hết sức thuận tiện thông qua giao tiếp trên web cũng như bằng FTP mà không cần đòi hỏi nhiều kiến thức về lĩnh vực này.

## 2. Các bước cơ bản để lựa chọn phương án thuê chỗ thích hợp

### 2.1. Lựa chọn hệ điều hành mà server sử dụng

Việc lựa chọn hệ điều hành tuỳ thuộc vào những nhu cầu của bạn. Nếu bạn đang sử dụng IIS, ASP, VBScript, Windows Media, Microsoft Access, Microsoft SQL Server, or Visual InterDev và không có nhiều thời gian cho việc tìm hiểu về các giải pháp dựa trên Unix, Linux, hãy chọn Windows NT hoặc Windows 2000. Tuy nhiên bạn phải trả giá cho sự lựa chọn này vì các phần mềm của Microsoft đều đòi hỏi phải trả tiền bản quyền. Nếu bạn muốn sử dụng SQL server, bạn phải trả thêm khá nhiều tiền trong đó một phần là tiền bản quyền.

Bạn cũng nên biết rằng hiện nay hệ điều hành Linux và Apache Web servers được sử dụng rất nhiều trong các công ty cung cấp dịch vụ thuê chổ. Sở dĩ có như vậy vì Apache' cung cấp rất nhiều tính năng hỗ trợ dạng thuê chổ này, đồng thời rất ổn định và hiệu suất thi hành cao. Ngoài ra, Linux và Apache là miễn phí.

### 2.2. Nhu cầu về băng thông

Bạn nên chi tiết hóa nhu cầu về băng thông của bạn trước khi đăng kí. Cần ước lượng kích thước của dữ liệu website thông qua việc lượng giá kích thước trung bình của trang, số trang trong website, và số lần xem các trang này trong một tháng.

Một số nhà cung cấp webhosting thường quảng cáo có thể hỗ trợ không giới hạn băng thông (unlimited bandwidth). Tuy nhiên điều này thường là thủ thuật mà thôi bởi vì nên nhớ rằng họ cũng phải trả tiền thuê băng thông. Sở dĩ có quảng cáo trên bởi những nhà cung cấp ước lượng rằng băng thông của từng người dùng thay đổi khác nhau và có thể bù qua sót lại mà bạn không biết. Hơn nữa, đa số các nhà cung cấp dạng này đều yêu cầu bạn chấp nhận chính sách của họ trước khi sử dụng (Bạn rất ít khi đọc kỹ chính sách này) mà trong đó cũng nêu rõ điều kiện sẽ yêu cầu bạn trả thêm tiền hoặc ra đi nếu lạm dụng tài nguyên.

### 2.3. Nhu cầu về không gian đĩa

Tiếp đến là ước lượng kích thước không gian đĩa cần cho website. Với các website nhiều ảnh đồ họa, kích thước cần nhiều hơn các website chỉ có văn bản. Tuy nhiên với sự tiến bộ của phần

cứng, hầu hết các dung lượng đĩa mà nhà cung cấp hỗ trợ vượt quá nhu cầu thông thường, ví dụ như 5GB, 10GB, 100GB.

## 2.4. Các ngôn ngữ lập trình được hỗ trợ

Với các website có lập trình, ngôn ngữ lập trình server-side cũng cần phải được xem xét đến một cách thận trọng. Hiện nay thông dụng nhất trên web vẫn là các ngôn ngữ PHP, JSP, ASP. Sử dụng ASP đòi hỏi hệ điều hành phải là Windows và thông thường là chi phí cao hơn so với PHP do PHP được tích hợp sẵn trong Apache và Linux.

## 2.5. Hệ quản trị cơ sở dữ liệu hỗ trợ

Một ứng dụng web thật sự đòi hỏi phải có sự ước lượng giữa chi phí bỏ ra và nhu cầu cần có khi bàn đến việc chọn hệ quản trị CSDL nào để hỗ trợ. Nếu dùng Microsoft thì có Microsoft Access hay Microsoft SQL Server, còn nếu dùng Unix hay Linux thì có mSQL, MySQL, hay PostgreSQL. MS SQL Server thì rất mạnh hơn nhiều so với Access cũng như MySQL và PostgreSQL so với mSQL. Tuy nhiên cần chú ý đến vấn đề giá cả, ví dụ như nếu dùng MS. Access thì được cung cấp miễn phí, trong khi dùng MS. SQL Server thì phải trả tiền bản quyền, bảo trì hệ thống, ...

## 2.6. Hỗ trợ kỹ thuật

Vấn đề này cũng rất quan trọng vì nếu trực trặc trong lúc vận hành hệ thống của bạn không được sửa chữa kịp thời, có thể sẽ dẫn đến mất nhiều cơ hội kinh doanh. Các hình thức hỗ trợ kỹ thuật được cung cấp thường là email, chat hay điện thoại trực tiếp tư vấn. Lưu ý rằng chi phí sẽ tỉ lệ thuận với chất lượng và hình thức tư vấn.,

## 2.7. Các tính năng khác

Ngoài ra cũng cần xem xét đến các tính năng khác như:

- Hỗ trợ SSL cho các giao tác an toàn trên mạng như trả tiền, chuyển tiền hay không.
- Hỗ trợ việc truy xuất đến các dữ liệu thống kê cho phép tổng hợp và đánh giá các khách hàng tham quan website
- Hỗ trợ nghe nhạc và xem video trực tiếp trên mạng mà không cần tải toàn bộ về.
- Số email accounts cung cấp và hỗ trợ mailing lists.

- Giá cả cho việc sử dụng vượt quá băng thông và dung lượng cho phép
- Hỗ trợ giao tiếp (control panel for graphical access) thuận tiện trong việc quản lý website.
- Hỗ trợ và tư vấn an ninh của hệ thống
- Các component được hỗ trợ sẵn

## 2.8. Các nhà cung cấp dịch vụ hosting hàng đầu thế giới hiện nay

- [Lunarpages.com](http://www.lunarpages.com)
- [iPowerWeb.com](http://www.ipowerweb.com)
- [ICDSoft](http://www.icdsoft.com)
- [Globat.com](http://www.globat.com)

### Tham khảo tại

- <http://www.tophostinfo.com/review1.htm>
- <http://www.webhosting-reviews.com/comm1.htm>
- <http://www.practicallynetworked.com/serving/webhost/>
- <http://www.webhostingratings.com/>
- <http://www.findmyhosting.com/>

# TÀI LIỆU THAM KHẢO

8. Active Server Page Unleashed, Stephan Walther
9. Xây dựng trang web động với ASP, Nhóm tác giả ELICOM
10. ASP Database, Nhóm tác giả SAIGONBOOK
11. Tự học Lập trình Thương mại Điện tử với ASP trong 21 ngày, Nhóm tác giả SAIGONBOOK
12. <http://www.learnasp.com>
13. <http://www.4guysfromrolla.com>
14. <http://www.15seconds.com>
15. <http://asp.supereexpert.com>
16. <http://www.fit.hcmuns.edu.vn/~ldduy/WebCourse>

# MỤC LỤC

<b>CHƯƠNG 1: GIỚI THIỆU MẠNG INTERNET .....</b>	<b>3</b>
1. Sơ lược về mạng Internet.....	3
2. Các dịch vụ cơ bản của mạng Internet.....	8
3. Khai thác các tài nguyên của mạng Internet .....	11
4. Một số thuật ngữ thông dụng .....	20
5. Một số câu hỏi thường gặp .....	21
Câu hỏi .....	28
Thực hành .....	29
<b>CHƯƠNG 2: THIẾT LẬP WEBSITE.....</b>	<b>37</b>
1. Cài đặt webserver.....	37
2. Cấu hình webserver.....	43
3. Một số vấn đề liên quan đến an toàn .....	49
4. Một số câu hỏi thường gặp .....	51
Câu hỏi .....	54
Thực hành .....	55
<b>CHƯƠNG 3: THIẾT KẾ TRANG WEB .....</b>	<b>61</b>
1. Giới thiệu về HTML.....	61
2. Một số tag cơ bản của HTML .....	63
3. Định dạng bảng biểu.....	69
4. Form và các thành phần của form.....	71
5. Frame .....	77
6. Cascading Style Sheet .....	79
Thực hành .....	83

**CHƯƠNG 4: CÁC NGÔN NGỮ SCRIPT HỖ TRỢ LẬP TRÌNH WEB**  
.....  
**129**

1. Giới thiệu về VBScript và JavaScript.....	129
2. Sử dụng VBScript.....	130
3. Sử dụng JavaScript .....	139
4. Sử dụng VBScript và JavaScript trong các trang web .....	144
5. Dấu mã VBScript/JavaScript .....	147
6. Một số ví dụ minh họa .....	148
Thực hành .....	157

**CHƯƠNG 5: NHẬP MÔN ASP**.....  
**159**

1. Giới thiệu về ASP .....	159
2. Các khái niệm cơ bản về ASP.....	160
3. Xử lý dữ liệu nhập từ người dùng .....	163
4. Xử lý các kết xuất dữ liệu .....	165
5. Kết nối với cơ sở dữ liệu .....	172
6. Những câu hỏi thường gặp .....	180
Thực hành .....	184

Bài tham khảo: Lập trình Web ASP với tiếng Việt Unicode .....

Bài tham khảo: Kết xuất dữ liệu ra dạng Excel từ trang ASP .....

**CHƯƠNG 6: ASP NÂNG CAO**.....  
**199**

1. Đối tượng Response .....	199
2. Đối tượng Request .....	201
3. Application và Session .....	208
4. Cookies .....	213
5. Đối tượng Server .....	216
6. Xử lý lỗi với đối tượng ASPError .....	218
7. ADO nâng cao .....	219
8. Đối tượng FileSystemObject .....	225

9. Gửi email bằng CDONTS.....  
**227**10. Một số kĩ thuật khác.....  
**233**

Bài tham khảo: Hiển thị hình ảnh từ cơ sở dữ liệu trong các ứng dụng web .....

Bài tham khảo: Phương pháp hỗ trợ giới hạn truy cập trang web....  
**241**

Bài tham khảo: Bảo vệ cơ sở dữ liệu Access trong các ứng dụng web .....

Bài tham khảo: Bảo vệ ứng dụng web chống tấn công kiểu SQL Injection.....  
**248****CHƯƠNG 7: ỨNG DỤNG MINH HỌA CỦA HÀNG SÁCH  
TRỰC TUYẾN**.....  
**253**

1. Giới thiệu về ứng dụng .....	253
2. Tổ chức website ứng dụng .....	253
3. Thiết kế cơ sở dữ liệu.....	253
4. Kết nối với cơ sở dữ liệu .....	255
5. Thao tác truy vấn trên cơ sở dữ liệu .....	256
6. Thao tác thêm một mẫu tin vào cơ sở dữ liệu.....	258
7. Thao tác cập nhật một mẫu tin trên cơ sở dữ liệu .....	259
8. Thao tác xóa một mẫu tin trên cơ sở dữ liệu .....	262
9. Sử dụng Cookies .....	262
10. Những câu hỏi thường gặp .....	274

Thực hành .....

Đồ án .....

Bài tham khảo: Web Hosting – Dịch vụ thuê chổ website trên Internet .....

**TÀI LIỆU THAM KHẢO**.....  
**283****MỤC LỤC**.....  
**284**