

## MỤC LỤC

<b>CHƯƠNG 1: CÁC THÀNH PHẦN CƠ BẢN CỦA HTML</b>	2
I. Giới thiệu và thuật ngữ	2
II. Tổng quan về HTML	5
<b>CHƯƠNG 2: THAO TÁC VỚI PHẦN MỀM TẠO WEB</b>	14
I. Giới thiệu về IIS- Internet Information Server	14
II. Xây dựng Web Site bằng phần mềm Microsoft Visual InterDev 6.0	15
III. Thiết kế giao diện	24
IV. Tổ chức hệ thống tập tin thư mục của một Web site	25
<b>CHƯƠNG 3: SIÊU LIÊN KẾT</b>	26
I. Khái niệm về siêu liên kết	26
II. Tạo siêu liên kết	26
<b>CHƯƠNG 4: TẠO DANH SÁCH, THIẾT KẾ BẢNG, TẠO BIỂU MẪU</b>	31
<b>GỬI DỮ LIỆU QUA INTERNET</b>	31
I. Định nghĩa một danh sách	31
II. Thiết kế bảng	33
III. Tạo biểu mẫu	35
IV. Gửi dữ liệu biểu mẫu qua Internet	36
V. Tạo khung (Frame)	37
<b>CHƯƠNG 5: ĐA PHƯƠNG TIỆN</b>	39
I. Khái niệm về đa phương tiện (Multimedia)	39
II. Giới thiệu một số kỹ thuật tạo file âm thanh, hình ảnh	39
III. Nhúng hình ảnh, âm thanh vào trang Web	39
<b>CHƯƠNG 6: VB SCRIPT VÀ CÁC ỨNG DỤNG</b>	41
I. Khái niệm về ngôn ngữ VB Script	41
II. Đưa các đoạn Script vào trong trang Web	45
III. Thiết lập các thư viện procedure, function cho 1 ứng dụng WEB	45
IV. Một số bài tập ứng dụng mẫu	47
<b>CHƯƠNG 7: LIÊN KẾT TRANG WEB ĐẾN HỆ CSDL</b>	
<b>XÂY DỰNG CÁC TRANG WEB ĐỘNG</b>	49
I. Nhắc lại một số khái niệm về CSDL	49
II. Các lệnh liên kết CSDL	56
III. Một số bài tập ứng dụng mẫu	57
<b>BÀI TẬP</b>	60
<b>CHƯƠNG 8: JAVASCRIPT</b>	61
I. PHÁT BIỂU LỆNH CỦA JAVASCRIPT	61
II. CÁC PHƯƠNG THỨC CỦA JavaScript	68
III. ĐỐI TƯỢNG CỦA JAVASCRIPT	83

## CHƯƠNG 1: CÁC THÀNH PHẦN CƠ BẢN CỦA HTML

### **Mục tiêu:**

- Sử dụng được cấu trúc tổng thể của HTML, các thẻ HTML
- Nhận dạng tên tập tin, các địa chỉ URL

### **I. Giới thiệu và thuật ngữ**

#### **- HTML là gì?**

HTML (hay Hypertext Markup Language-Ngôn ngữ liên kết siêu văn bản) là ngôn ngữ đánh dấu chuẩn dùng lập trình các tài liệu World Wide Web, tài liệu là các tập tin văn bản đơn giản. Ngôn ngữ HTML dùng các tag hoặc các đoạn mã lệnh để chỉ cho các trình duyệt(Web browsers) cách hiển thị các thành phần của trang như text và graphics ,và đáp lại những thao tác của người dùng bởi các thao tác ấn phím và nhấp chuột. Hầu hết các Web browser, đặc biệt là Microsoft Internet Explorer và Netscape Navigator, nhận biết các tag của HTML vượt xa những chuẩn HTML đặt ra. Microsoft FrontPage chỉ dùng để đọc và viết các tập tin HTML mà không hiểu ngôn ngữ HTML yêu cầu phải làm gì.

- **Dynamic HTML (DHTML-HTML động)**: là ngôn ngữ HTML mở rộng làm tăng hiệu ứng trình bày văn bản và đối tượng khác. Trong FrontPage, bạn có thể sử dụng thanh công cụ **DHTML Effects** để làm tăng cường hiệu ứng cho các thành phần của mà không cần lập trình.

- **Active Data Objects(ADO)**: Các thành phần giúp các ứng dụng của người dùng (client applications) truy cập và chế tác dữ liệu của cơ sở dữ liệu trên server qua 1 nhà cung cấp.

- **Active Server Page(ASP)**: là 1 tài liệu chứa script nhúng trên server . Web servers tương thích ASP có thể chạy các script này. Trên máy trạm, 1 ASP là 1 tài liệu HTML chuẩn có thể được xem trên bất kỳ máy nào trên Web browser nào.

- **ActiveX**: 1 tập hợp các kỹ thuật cho phép các thành phần phần mềm tương tác với một thành phần khác trong môi trường mạng, bất chấp ngôn ngữ của thành phần được tạo ra. ActiveX được dùng làm chính yếu để phát triển nội dung tương tác của World Wide Web, mặc dù nó có thể sử dụng trong các ứng dụng người-máy và các chương trình khác.

- **URL(Uniform Resource Locator)**: địa chỉ tới một trạm Internet hay mạng nội bộ, là 1 chuỗi cung cấp địa chỉ Internet của 1 Web site tài nguyên trên World Wide Web, đi theo sau 1 nghi thức. URL thường dùng là http://, để chỉ định địa chỉ Web site trên Internet. Những URL khác là gopher://, ftp://, mailto://...

- **Bookmark**: 1 vị trí trên 1 trang web có thể là đích của 1 hyperlink. 1 bookmark có thể áp dụng cho 1 chuỗi ký tự tồn tại trên trang ngăn cách bởi bất kỳ ký tự nào. Bookmarks cho phép tác giả link đến 1 phần đã chỉ định trên trang. Trong 1 URL, 1 bookmark được đánh dấu phía trước bằng dấu thăng(#). Cũng được gọi là neo(anchor).

- **Web browser(Trình duyệt web):** Phần mềm phiên dịch đánh dấu của các file bằng HTML, định dạng chúng sang các trang Web, và thể hiện chúng cho người dùng. Vài browser có thể cho phép người dùng gửi nhận e-mail, đọc newsgroups, và thực hiện các file sound hoặc video đã được nhúng và trong tài liệu Web.
- **Script:** Một tập các chỉ lệnh dùng để báo cho chương trình biết cách thực hiện một thủ tục qui định, như đăng nhập vào hệ thống thư điện tử chẳng hạn. Các khả năng script được cài sẵn trong một số chương trình. Bạn phải tìm hiểu cách viết script đó bằng loại ngôn ngữ không khác gì ngôn ngữ lập trình mini. Có một số chương trình ghi script này một cách tự động bằng cách ghi lại những lần gõ phím và chọn dùng lệnh của bạn khi bạn tiến hành thủ tục này. Các script giống như các macro, trừ một điều là thuật ngữ macro được dành riêng để chỉ những script nào mà bạn có thể khởi đầu bằng cách ấn một tổ hợp phím do bạn tự quy định.
- **Structured Query Language - SQL:** Trong các hệ quản trị cơ sở dữ liệu, đây là ngôn ngữ vấn đáp do IBM soạn thảo được sử dụng rộng rãi trong máy tính lớn và hệ thống máy tính mini. SQL đang được trang bị trong các mạng khách/chủ như là một phương pháp làm cho các máy tính cá nhân có khả năng thâm nhập vào các tài nguyên của các cơ sở dữ liệu hợp tác. Đây là loại ngôn ngữ độc lập với dữ liệu; người sử dụng không phải bận tâm đến vấn đề dữ liệu sẽ được thâm nhập vào bằng cách nào về mặt vật lý. Theo lý thuyết, SQL cũng độc lập với thiết bị; có thể dùng cùng một ngôn ngữ vấn đáp để thâm nhập vào các cơ sở dữ liệu trên máy tính lớn, máy tính mini, và máy tính cá nhân. Tuy nhiên, hiện nay có một số phiên bản của SQL đang bị cạnh tranh. SQL là một ngôn ngữ vấn đáp súc tích chỉ với 30 lệnh. Bốn lệnh cơ bản ( SELECT, UPDATE, DELETE, và INSERT) đáp ứng cho bốn chức năng xử lý dữ liệu cơ bản (phục hồi, cải tiến, xoá, và chèn vào). Các câu hỏi của SQL gần giống cấu trúc của một câu hỏi tiếng Anh tự nhiên. Kết quả của câu hỏi sẽ được biểu hiện trong một bản dữ liệu bao gồm các cột (tương ứng với các trường dữ liệu) và các hàng (tương ứng với các bản ghi dữ liệu).
- **Hypertext Transfer Protocol-HTTP:** Giao thức truyền siêu văn bản
- **Hyperlink:** Siêu liên kết
- **Web site:** nghĩa hẹp: trang web, nghĩa rộng: chỗ chứa web.
- **HTTP address:** địa chỉ HTTP, ví dụ: <http://www.dalattech.edu.vn>
- **Database:** cơ sở dữ liệu, kho dữ liệu. Một bộ sưu tập các thông tin về một chủ đề, được tổ chức hợp lý để thành một cơ sở cho các thủ tục công việc như truy tìm thông tin, rút ra các kết luận, và thành lập các quyết định. Bất kỳ một sưu tập thông tin nào phục vụ cho các mục đích này đều được coi là một cơ sở dữ liệu, ngay cả trường hợp các thông tin đó không được lưu trữ trong máy tính. Thực ra, "tổ tiên" của các hệ cơ sở dữ liệu doanh thương phức tạp hiện nay là những tệp được giữ trên các tấm bìa, có mang mã số, và được xếp trong các phòng lưu trữ trước đây. Thông tin thường được chia ra thành nhiều bản ghi ( record) dữ liệu khác nhau, trên đó có một hay nhiều trường dữ liệu.
- **Table:** bảng trong chương trình quản lý cơ sở dữ liệu liên quan, đây là cấu trúc cơ sở của việc lưu trữ và hiển thị dữ liệu, trong đó các khoản mục dữ liệu được liên kết với nhau bởi các quan hệ hình thành do việc đặt chúng theo các hàng và các cột. Các hàng

ứng với các bản ghi dữ liệu của các chương trình quản lý dữ liệu hướng bảng, và các cột thì ứng với các trường dữ liệu.

- **Hypertext**: văn bản của một tài liệu truy tìm không theo tuần tự. Người đọc tự do đuổi theo các dấu vết liên quan qua suốt tài liệu đó bằng các mối liên kết xác định sẵn do người sử dụng tự lập nên.

Trong một môi trường ứng dụng heperext thực sự, bạn có thể trở vào (highlight) bất kỳ từ nào của tài liệu và sẽ tức khắc nhảy đến các tài liệu khác có văn bản liên quan đến nó. Cũng có những lệnh cho phép bạn tự tạo cho riêng mình những dấu vết kết hợp qua suốt tài liệu. Các trình ứng dụng hypertext rất hữu ích trong trường hợp phải làm với một số lượng văn bản lớn, như các bộ từ điển bách khoa và các bộ sách nhiều tập.

- **Multimedia**: đa phương tiện, đa môi trường, đa truyền thông

Một phương pháp giới thiệu thông tin bằng máy tính, sử dụng nhiều phương tiện truyền thông tin như văn bản, đồ hoạ và âm thanh, cùng với sự gây ấn tượng bằng tương tác.

- **World Wide Web - WWW**: Là một dịch vụ của Internet. Đây là một hệ thống dùng để truy tìm và phục hồi các tài liệu hypertext thực hiện.

- **File Transfer Protocol - FTP**: định ước truyền tệp, giao thức truyền tệp. Trong truyền thông không đồng bộ, đây là một tiêu chuẩn nhằm bảo đảm truyền dẫn không bị lỗi cho các tệp chương trình và dữ liệu thông qua hệ thống điện thoại. Chương trình FTP cũng được dùng để gọi tên cho định ước truyền tệp UNIX, một định ước hướng dẫn việc truyền dữ liệu.

- **Internet**: Một hệ thống gồm các mạng máy tính được liên kết với nhau trên phạm vi toàn thế giới, tạo điều kiện thuận lợi cho các dịch vụ truyền thông dữ liệu, như đăng nhập từ xa, truyền các tệp tin, thư tín điện tử, và các nhóm thông tin.

Internet là một phương pháp ghép nối các mạng máy tính hiện hành, phát triển một cách rộng rãi tầm hoạt động của từng hệ thống thành viên.

Nguồn gốc đầu tiên của Internet là hệ thống máy tính của Bộ Quốc Phòng Mỹ, gọi là mạng ARPAnet, một mạng thí nghiệm được thiết kế từ năm 1969 để tạo điều kiện thuận lợi cho việc hợp tác khoa học trong các công trình nghiên cứu quốc phòng. ARPAnet đã nêu cao triết lý truyền thông bình đẳng (peer-to-peer), trong đó mỗi máy tính của hệ thống đều có khả năng "nói chuyện" với bất kỳ máy tính thành viên nào khác.

Bất kỳ mạng máy tính nào dựa trên cơ sở thiết kế của ARPAnet đều được mô tả như một tập hợp các trung tâm điện toán tự quản, mang tính địa phương và tự điều hành, chúng được liên kết dưới dạng "vô chính phủ nhưng có điều tiết" .

Sự phát triển thiết kế của mạng ARPAnet đơn thuần chỉ do những yêu cầu về quân sự: Mạng này phải có khả năng chống lại một cuộc tấn công có thể vô hiệu hoá một số lớn các trạm thành viên của nó.

Tư tưởng này đã được chứng minh là đúng khi Mỹ và các đồng minh tham gia vào cuộc chiến tranh vùng Vịnh.

Sự chỉ huy và mạng kiểm soát của Irak, được tổ chức mô phỏng theo công nghệ ARPAnet, đã chống lại một cách thành công đối với các nỗ lực của lực lượng đồng minh nhằm tiêu diệt nó.

Đó là lý do tại sao công nghệ có nguồn gốc từ ARPAnet hiện nay đang được xuất cảng một cách rộng rãi.

Mạng Internet nguyên thủy được thiết kế nhằm mục đích phục vụ việc cung cấp thông tin cho giới khoa học, nên công nghệ của nó cho phép mọi hệ thống đều có thể liên kết với nó thông qua một cổng điện tử.

Theo cách đó, có hàng ngàn hệ máy tính hợp tác, cũng như nhiều hệ thống dịch vụ thư điện tử có thu phí, như MCI và Compuserve chẳng hạn, đã trở nên thành viên của Internet.

- **Password**: mật khẩu, mật lệnh. Một công cụ bảo vệ an toàn, dùng để xác định đúng người sử dụng được phép đối với một chương trình máy tính hoặc mạng máy tính, và để xác định các phạm vi quyền hạn của họ như chỉ đọc ra, được đọc và ghi, hoặc được sao chép các tệp.

## II. Tổng quan về HTML

### 1. Cấu trúc trang web

```
<HTML>
<HEAD>
<TITLE>Tiêu đề trang web</TITLE>
</HEAD>
<BODY>
  Thân trang web
</BODY>
</HTML>
```

### 2. Cú pháp tag tổng quát

Khi trình duyệt (web browser) hiển thị nội dung 1 trang web nó sẽ tìm kiếm trong đó những đoạn mã đặc biệt là các tag được đánh dấu bởi ký hiệu <và>, và căn cứ vào đó để thể hiện. Cú pháp tổng quát của 1 tag:

**<ten\_tag thuộc\_tính\_1=giá\_trị\_1 thuộc\_tính\_2=giá\_trị\_2...>...</ten\_tag>**

hoặc đơn giản nhất là: **<ten\_tag>Chuỗi văn bản</ten\_tag>**

ten\_tag gõ vào dạng chữ thường hoặc hoa đều được và các tag có thể lồng vào nhau.

Ví dụ:

- tag chữ đậm <B> cú pháp **<B>chuỗi văn bản in đậm</B>**
- tag xuống hàng <P> có cú pháp **<P ALIGN=align-type>một đoạn văn bản</P>**

HTML	Kết quả
Dòng chữ này thường	Dòng chữ này thường
<b>&lt;b&gt;Dòng chữ này đậm&lt;/b&gt;</b>	<b>Dòng chữ này đậm</b>

<code>&lt;i&gt;Dòng chữ này nghiêng&lt;/i&gt;</code>	Dòng chữ này nghiêng
<code>&lt;font color="#800000" face="Verdana"&gt;Đây là đoạn văn thứ nhất màu đỏ có phông là Verdana.&lt;/font&gt;&lt;p&gt;&lt;font color="Black" face="Verdana"&gt;Đây là đoạn văn thứ hai màu ô liu có phông cũng là Verdana.&lt;/font&gt;&lt;/p&gt;</code>	Đây là đoạn văn thứ nhất màu đỏ có phông là Verdana. Đây là đoạn văn thứ hai màu ô liu có phông cũng là Verdana.
<code>&lt;b&gt;&lt;i&gt;Dòng chữ này vừa đậm vừa nghiêng&lt;/i&gt;&lt;/b&gt;</code>	<b>Dòng chữ này vừa đậm vừa nghiêng</b>
<code>Đoạn này canh trái.&lt;/font&gt;&lt;p align="center"&gt;&lt;font face="Verdana"&gt;Đoạn này canh giữa.&lt;/font&gt;&lt;/p&gt;&lt;p align="right"&gt;&lt;font face="Verdana"&gt;Đoạn này canh phải.&lt;/font&gt;</code>	Đoạn này canh trái. Đoạn này canh giữa. Đoạn này canh phải.

**3. Tag đầu trang <HTML></HTML>** :Tag <HTML> ở đầu trang web tĩnh và </HTML> ở cuối trang. Trang web được bắt đầu và kết thúc bởi tag này.

**4. Tag bắt đầu <HEAD></HEAD>** :Tag <HEAD> đánh dấu điểm bắt đầu phần đầu heading của trang web. Tag </HEAD> có thể bỏ qua.

**5. Tag tiêu đề <TITLE></TITLE>** :Tag <TITLE> chỉ định tiêu đề của của trang web. Tiêu đề này sẽ xuất hiện trên thanh tiêu đề của trình duyệt(web browser: Internet Explorer hoặc Netscape Navigator). Tag này chỉ hợp lệ khi đưa vào bên trong phần <HEAD>

Ví dụ:

HTML	Kết quả
<pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;Web demo&lt;/title&gt; &lt;/head&gt; &lt;body&gt; Hello World! &lt;/body&gt; &lt;/html&gt;</pre>	<ul style="list-style-type: none"> <li>- Trên thanh tiêu đề xuất hiện: Web demo - Microsoft Internet Explorer</li> <li>- Nội dung trang web xuất hiện: Hello World!</li> </ul>

**6. Tag thân trang <BODY></BODY>**: dùng chỉ định bắt đầu và kết thúc phần nội dung trang web. Tag này được định nghĩa như sau:

`<BODY`

`BACKGROUND=url`

`BGCOLOR=color`

`BGPROPERTIES=FIXED`

LEFTMARGIN=n

LINK=color

TEXT=color

TOPMARGIN=n

VLINK=color&gt;

trong đó:

Thuộc tính	Diễn giải
url	Viết tắt của Uniform Resource Locators: là 1 chuỗi cung cấp địa chỉ Internet của 1 Web site hoặc tài nguyên trên World Wide Web, theo cùng là nghi thức protocol của site hoặc tài nguyên truy cập vào. Nghi thức thường dùng http://, cho các địa chỉ Internet của một trang web. Vài dạng URL khác là gopher://, cho các địa chỉ Internet của 1 thư mục dịch vụ gopher, và ftp://, cho các địa chỉ tài nguyên FTP. Trong tag <BODY> ta thường dùng dạng URL là file://, tên tập tin ảnh .GIF, .JPG, .BMP. Khi lưu trang web, bạn sẽ xác định thư mục mới sẽ lưu trữ tập tin ảnh.
color	Màu dạng #NNNNNN với N:0,1,...,F. Hoặc tên những màu đã quy định trước như: AQUA, BLACK, Black, FUCHSIA, GRAY, GREEN, LIME, MAROON, NAVY, OLIVE, PURPLE, #800000, SILVER, TEAL, WHITE, YELLOW.
BACKGROUND=url	Chỉ định ảnh nền cho trang web, url là địa chỉ và tên tập tin làm ảnh nền.
BGCOLOR=color	Chỉ định màu nền cho trang web
BGPROPERTIES=FIXED	Chỉ định ảnh nền trang web không cuộn khi cuộn thanh scrollbar.
LEFTMARGIN=n	Chỉ định lề trái trang web, n là 1 số nguyên dương tính theo pixel
LINK=color	Chỉ định màu cho các hyperlink khi chưa được "duyet" trong trang web.
TEXT=color	Chỉ định màu văn bản trong trang web
TOPMARGIN=n	Chỉ định lề trên trang web, n là 1 số nguyên dương tính theo pixel
VLINK=color	Chỉ định màu cho các hyperlink khi đã được "duyet" trong trang web.

Ví dụ:

HTML	Kết quả
<BODY BGCOLOR=YELLOW TEXT=#800000> Chào mừng thế kỷ mới	Chào mừng thế kỷ mới

</BODY>	
<pre>&lt;BODY BGCOLOR="#0000FF" TEXT="#FFFFFF"&gt; Chào mừng thể kỷ mới &lt;/BODY&gt;</pre>	<p>Chào mừng thể kỷ mới</p>
<pre>&lt;html&gt; &lt;head&gt; &lt;title&gt;Web demo&lt;/title&gt; &lt;/head&gt; &lt;BODY text=BLACK background="images/so01871_1.gif" topmargin="20" leftmargin="25" link=#800000 vlink=GREEN&gt; &lt;p align="center"&gt;&lt;b&gt;Danh mục công việc&lt;/b&gt;&lt;p&gt;&lt;a href="#Nội dung mục 1"&gt;Mục 1&lt;/a&gt;&lt;/p&gt; &lt;p&gt;&lt;a href="#Nội dung mục 2"&gt;Mục 2&lt;/a&gt;&lt;/p&gt; &lt;p&gt;&lt;a href="#Nội dung mục 3"&gt;Mục 3&lt;/a&gt;&lt;/p&gt; &lt;p&gt;&lt;/p&gt; &lt;p&gt;&lt;a name="Nội dung mục 1"&gt;Nội dung mục 1&lt;/a&gt;&lt;/p&gt; &lt;p&gt;&lt;/p&gt; &lt;p&gt;&lt;/p&gt; &lt;p&gt;&amp;nbsp;&lt;/p&gt; &lt;p&gt;&amp;nbsp;&lt;/p&gt; &lt;p&gt;&lt;a name="Nội dung mục 2"&gt;Nội dung mục 2&lt;/a&gt;&lt;/p&gt; &lt;p&gt;&lt;/p&gt; &lt;p&gt;&lt;/p&gt; &lt;p&gt;&amp;nbsp;&lt;/p&gt; &lt;p&gt;&amp;nbsp;&lt;/p&gt; &lt;p&gt;&amp;nbsp;&lt;/p&gt; &lt;p&gt;&amp;nbsp;&lt;/p&gt; &lt;p&gt;&amp;nbsp;&lt;/p&gt; &lt;p&gt;&lt;a name="Nội dung mục 3"&gt;Nội dung mục 3&lt;/a&gt;&lt;/p&gt; &lt;p&gt;&amp;nbsp;&lt;/p&gt; &lt;/BODY&gt; &lt;/html&gt;</pre>	<ul style="list-style-type: none"> <li>- file ảnh nền: so01871_1.gif</li> <li>- Vì không đưa vào thuộc tính <code>bgproperties="fixed"</code> nên ảnh nền bị cuộn</li> <li>- lề trái=25</li> <li>- lề trên=20</li> <li>- màu văn bản đen</li> <li>- màu hyperlink chưa duyệt là đỏ</li> <li>- màu hyperlink đã duyệt là xanh</li> </ul>



**7. Tag xuống dòng sang đoạn mới <P></P>:** ngắt văn bản sang đoạn (paragraph) mới. Thành phần </P> cuối đoạn không nhất thiết phải có. Tag này được định nghĩa như sau: <P ALIGN=align-type>, ALIGN=align-type dùng chỉ định canh đoạn mới. align-type=LEFT(canh trái), RIGHT(canh phải) hoặc CENTER(canh giữa).

Ví dụ:

HTML	Kết quả
Có 2 trình duyệt hiện đang được sử dụng rộng rãi là: <p align=left>- Netscape Navigator <p align=center>- Và Internet Explorer	Có 2 trình duyệt hiện đang được sử dụng rộng rãi là: - Netscape Navigator - Và Internet Explorer

**8. Tag xuống dòng không sang đoạn mới <BR></BR>:** ngắt xuống dòng mới nhưng vẫn thuộc cùng một đoạn hiện hành.</BR> không nhất thiết phải có.

Ví dụ:

HTML	Kết quả
Prontpage hay tự thêm vào các chuỗi mã thay thế cho việc đưa các ký tự vào source trang web như: - chuỗi "&reg;" thay thế ký tự (R). - chuỗi "&lt;" thay thế dấu <.	Prontpage hay tự thêm vào các chuỗi mã thay thế cho việc đưa các ký tự vào source trang web như: - chuỗi "&reg;" thay thế ký tự (R). - chuỗi "&lt;" thay thế dấu <.

**9. Tag thêm đường thẳng nằm ngang <HR></HR>:** thêm đường thẳng nằm ngang trong trang web. Tag này được định nghĩa như sau:

<HR

ALIGN=align-type

COLOR=color

NOSHADE

SIZE=n



WIDTH=m>

trong đó:

- ALIGN=align-tpye: canh lề cho đường thẳng ngang, align-type có thể là LEFT, RIGHT, CENTER
- COLOR=color: màu đường thẳng ngang
- NOSHADE: không có bóng
- SIZE=n : độ đậm của đường thẳng ngang, n có giá trị là số nguyên tính theo đơn vị pixel.

Ví dụ:

HTML	Kết quả
------	---------

<pre>&lt;HR ALIGN=CENTER COLOR=#800000 NOSHADE SIZE=3 WIDTH=200&gt;</pre>	
<pre>&lt;HR ALIGN=LEFT COLOR=BLACK SIZE=2 WIDTH=150&gt;</pre>	

**10. Tag đậm <B></B>:** Hiển thị văn bản đậm theo physical type

**11. Tag nghiêng <I></I>:** Hiển thị văn bản nghiêng theo physical type

**12. Tag gạch dưới <U></U>:** Hiển thị văn bản gạch dưới theo physical type

**13. Tag gạch giữa <S></S>:** Hiển thị văn bản gạch giữa(strikeout) theo physical type

Ví dụ:

HTML	Kết quả
Dòng này thông thường	Dòng này thông thường
<b>&lt;b&gt;Dòng này đậm&lt;/b&gt;</b>	<b>Dòng này đậm</b>
<i>&lt;i&gt;Dòng này nghiêng&lt;/i&gt;</i>	<i>Dòng này nghiêng</i>
<u>&lt;u&gt;Dòng này gạch dưới&lt;/u&gt;</u>	<u>Dòng này gạch dưới</u>
<del>&lt;s&gt;Dòng này gạch giữa&lt;/s&gt;</del>	<del>Dòng này gạch giữa</del>
<b>&lt;b&gt;&lt;i&gt;Dòng này vừa đậm vừa nghiêng&lt;/i&gt;&lt;/b&gt;</b>	<b><i>Dòng này vừa đậm vừa nghiêng</i></b>
<u><i>&lt;i&gt;&lt;u&gt;Dòng này vừa nghiêng vừa gạch dưới&lt;/u&gt;&lt;/i&gt;</i></u>	<u><i>Dòng này vừa nghiêng vừa gạch dưới</i></u>

**14. Tag đậm logic type <STRONG></STRONG>:**Hiển thị văn bản đậm theo logic type

**15. Tag nghiêng logic type <EM></EM>:** Hiển thị văn bản nghiêng theo logic type

**16. Tag gạch ngang logic type <STRIKE></STRIKE>:** Hiển thị văn bản gạch dưới theo logic type

**17. Tag chỉ số logic type <SUB></SUB>:** Hiển thị văn bản dạng chỉ số theo logic type

**18. Tag mũ logic type <SUP></SUP>:** Hiển thị văn bản dạng mũ theo logic type

Ví dụ:

HTML	Kết quả
Dòng này thông thường	Dòng này thông thường
<code>&lt;strong&gt;Dòng này đậm&lt;/strong&gt;</code>	<b>Dòng này đậm</b>
<code>&lt;em&gt;Dòng này nghiêng&lt;/em&gt;</code>	<i>Dòng này nghiêng</i>
<code>&lt;strike&gt;Dòng này gạch giữa&lt;/strike&gt;</code>	<del>Dòng này gạch giữa</del>
<code>&lt;strong&gt;&lt;em&gt;Dòng này vừa đậm vừa nghiêng&lt;/em&gt;&lt;/strong&gt;</code>	<b><i>Dòng này vừa đậm vừa nghiêng</i></b>
<code>&lt;em&gt;&lt;u&gt;Dòng này vừa nghiêng vừa gạch dưới&lt;/u&gt;&lt;/em&gt;</code>	<i><u>Dòng này vừa nghiêng vừa gạch dưới</u></i>
<code>H&lt;sub&gt;2&lt;/sub&gt;SO&lt;sub&gt;4&lt;/sub&gt;</code>	H <sub>2</sub> SO <sub>4</sub>
<code>x&lt;sup&gt;y&lt;/sup&gt;</code>	x <sup>y</sup>

**19. Tag định font chữ cơ sở <BASEFONT></BASEFONT>**: chỉ định font, size, màu văn bản cho các văn bản không định dạng và dùng với tag

`<BASEFONT`

`SIZE=n`

`FACE=name`

`COLOR=color>`

Trong đó:

SIZE=n: chỉ định kích thước văn bản, n từ 1 đến 7. Giá trị mặc nhiên là 3.

COLOR=color: màu văn bản

FACE=name: tên font

**20. Tag font chữ <FONT></FONT>**: chỉ định font, size, màu văn bản. Định nghĩa như sau:

`<FONT`

`SIZE=n`

`FACE=name`

`COLOR=color>`

Trong đó:

SIZE=n: chỉ định kích thước văn bản, n từ 1 đến 7. Thêm dấu cộng + hoặc trừ - phía trước để chỉ định việc tăng hoặc giảm kích thước so với kích thước đã định trong BASEFONT.

COLOR=color: màu văn bản

FACE=name: tên font

Ví dụ: Văn bản thường

HTML	Kết quả
------	---------

<pre>&lt;basefont size=3&gt; &lt;font size=5&gt; Kích thước văn bản là 5&lt;/font&gt; &lt;p&gt;&lt;font size=+1&gt;Kích thước văn bản là 4&lt;/p&gt;&lt;/font&gt; &lt;p&gt;&lt;font size=+2&gt;Kích thước văn bản là 5&lt;/p&gt;&lt;/font&gt; &lt;p&gt;&lt;font size=-1&gt;Kích thước văn bản là 2&lt;/p&gt;&lt;/font&gt;</pre>	<p>Kích thước văn bản là 5</p> <p>Kích thước văn bản là 4</p> <p>Kích thước văn bản là 5</p> <p>Kích thước văn bản là 2</p>
---	---

**21. Tag nhạc nền <BGSOUND>:** định nhạc nền cho trang tư liệu. Tag này được đặt dưới tag </HEAD> nhưng trên tag <BODY>. Định nghĩa như sau:

```
<BGSOUND
```

```
    SCR=url
```

```
    LOOP=n>
```

Trong đó:

url: chỉ định tập tin nhạc. Các kiểu nhạc có thể là WAV, AU, MIDI.

n: chỉ số lần lặp lại bài nhạc. Nếu n=-1 hoặc infinite thì sẽ lặp đến khi nào đóng trang web.

Ví dụ:

```
<HTML>
```

```
<TITLE>Background Sound</TITLE>
```

```
<HEAD>
```

```
</HEAD>
```

```
<BGSOUND
```

```
    SCR="sound/bgs.wav"
```

```
    LOOP=-1>
```

```
<BODY>
```

```
    Enjoy my sound.
```

```
</BODY>
```

```
</HTML>
```

## 22. Tag 6 mức tiêu đề <Hn></Hn>

Dùng để tạo các chuỗi tiêu đề to nhỏ tùy ý tùy theo cấp tiêu đề đó trong trang web. Có 6 cấp tiêu đề khác nhau được định nghĩa như sau:

```
<Hn>Text xuất hiện trong tiêu đề</Hn>
```

với n là một số từ 1 đến 6

Ví dụ:

HTML	Kết quả
<pre>&lt;H1&gt;Tiêu đề cấp 1&lt;/H1&gt;</pre>	Tiêu đề cấp 1

<H2>Tiêu đề cấp 2</H2>	<b>Tiêu đề cấp 2</b>
<H3>Tiêu đề cấp 3</H3>	<b>Tiêu đề cấp 3</b>
<H4>Tiêu đề cấp 4</H4>	<b>Tiêu đề cấp 4</b>
<H5>Tiêu đề cấp 5</H5>	<b>Tiêu đề cấp 5</b>
<H6>Tiêu đề cấp 6</H6>	<b>Tiêu đề cấp 6</b>

## CHƯƠNG 2: THAO TÁC VỚI PHẦN MỀM TẠO WEB

### **Mục tiêu:**

- Thiết kế được một Web Site trên một phần mềm tạo Web đáp ứng yêu cầu đặt ra
- Tổ chức được hệ thống thư mục tập tin lưu trên đĩa.

### **I. Giới thiệu về IIS- Internet Information Server.**

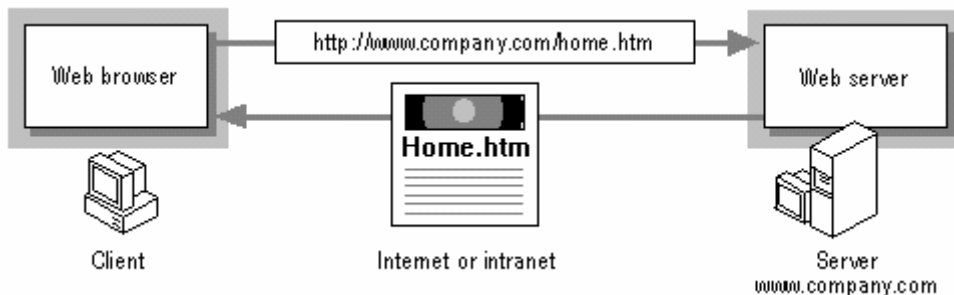
#### **a. IIS là gì?**

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

#### **b. IIS có thể làm được gì?**

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

#### **c. IIS hoạt động như thế nào ?**

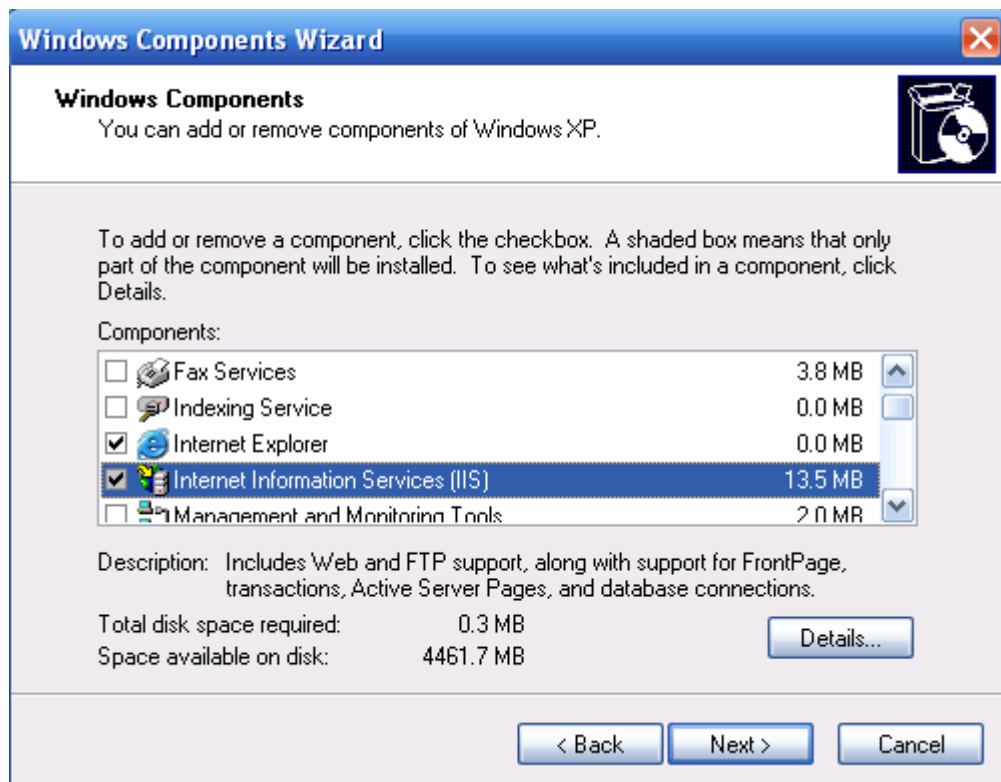


Web, về cơ bản thực sự là một hệ thống các yêu cầu (Request) và các đáp ứng (Response). IIS phản hồi lại các yêu cầu đòi thông tin của Web Browser. IIS lắng nghe các yêu cầu đó từ phía Users trên một mạng sử dụng WWW.

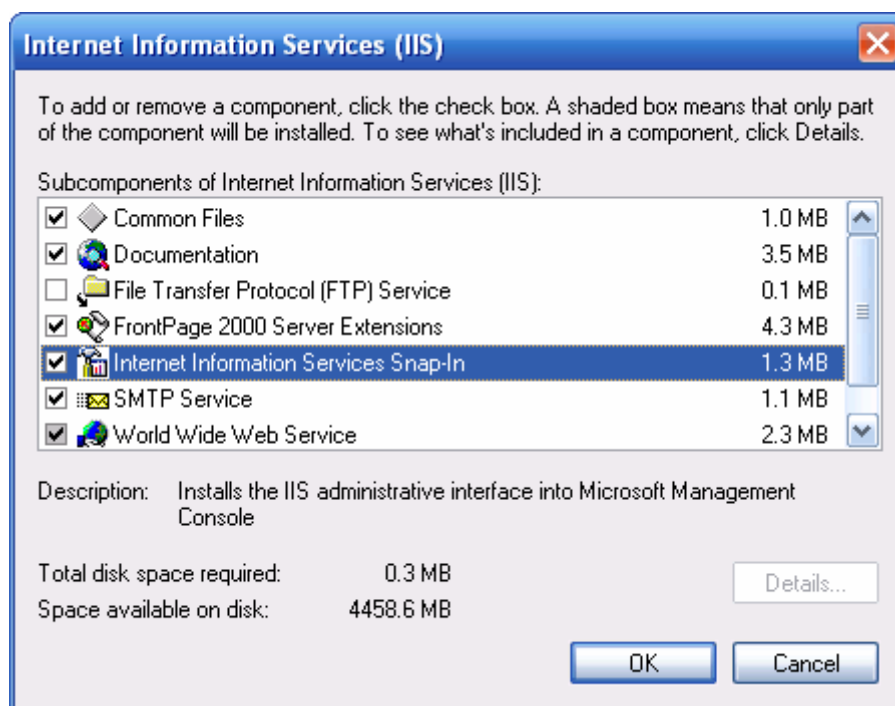
#### **d. Cài đặt IIS**

##### **Đối với Window XP:**

- Vào Start → Control Panel → Add or Remove Programs
- Chọn tab Add/Remove Windows Components → đánh dấu vào ô Internet Information Services(IIS).



- Nhấn nút Details..., cửa sổ hiện ra, đánh dấu mục Frontpage 2000 Server Extensions và World Wide Web Service, nhấn OK.



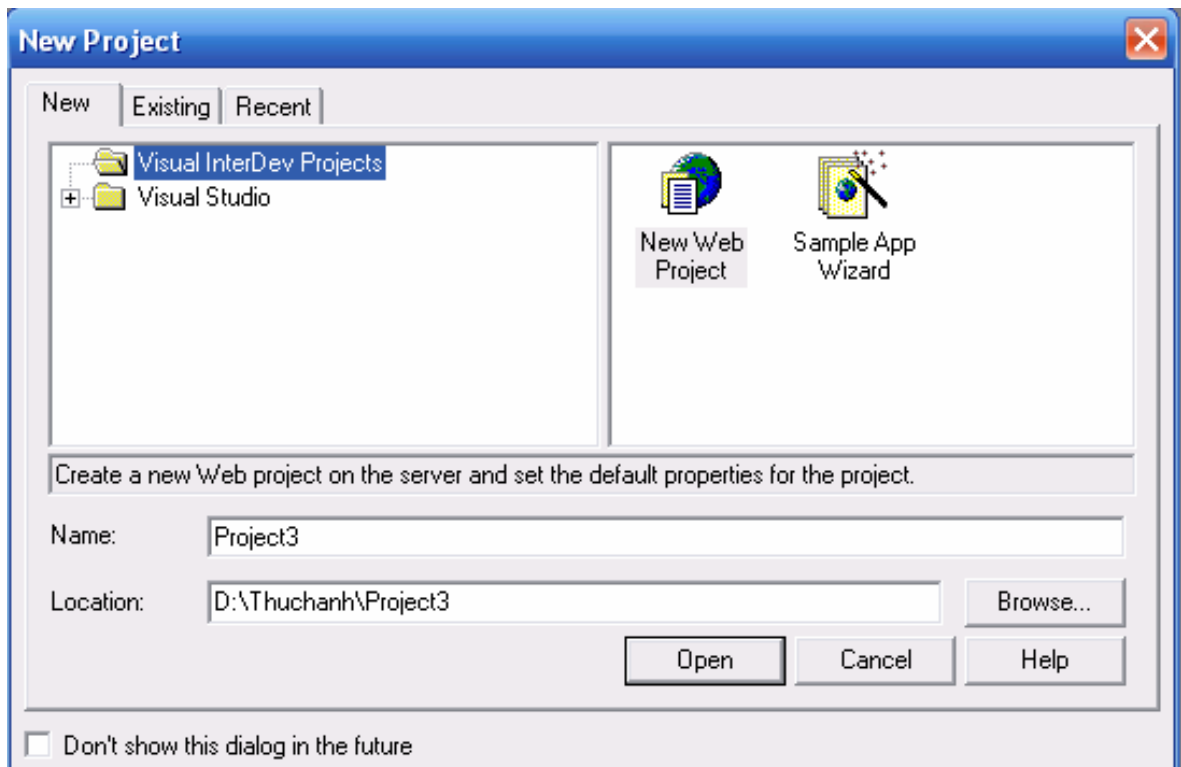
- Nhấn Next để cài đặt → Finish hoàn tất.

## II. Xây dựng Web Site bằng phần mềm Microsoft Visual InterDev 6.0

### 1. Khởi tạo một đề án Web

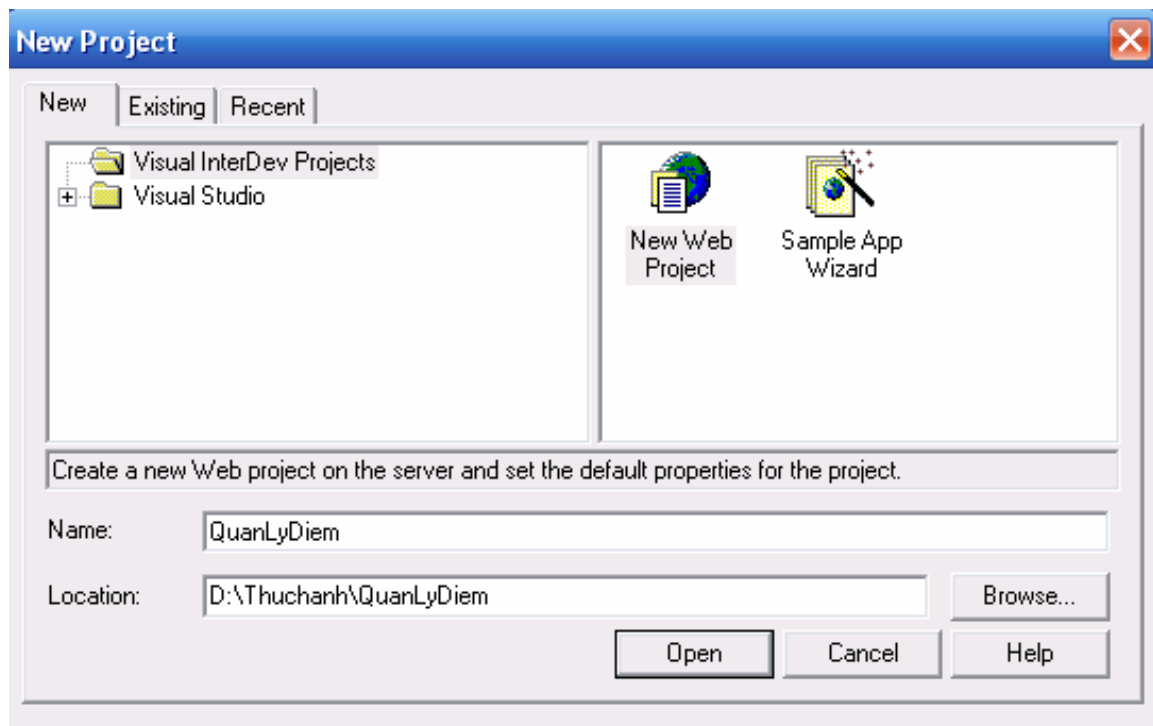
Để bắt đầu với việc xây dựng một Website chúng ta phải tiến hành tạo một đề án nhằm thuận tiện cho việc phát triển về sau, đối với phần mềm Microsoft Visual InterDev 6.0 công việc này thực hiện khá dễ dàng qua các thao tác thân thiện như sau:

1.1 Vào Start → Program files → Microsoft Visual Studio 6.0 → Microsoft Visual InterDev 6.0. Môi trường phát triển sẽ hiện ra, thường sẽ xuất hiện hộp thoại New Project như sau (nếu không có, hãy vào menu File → New Project...):

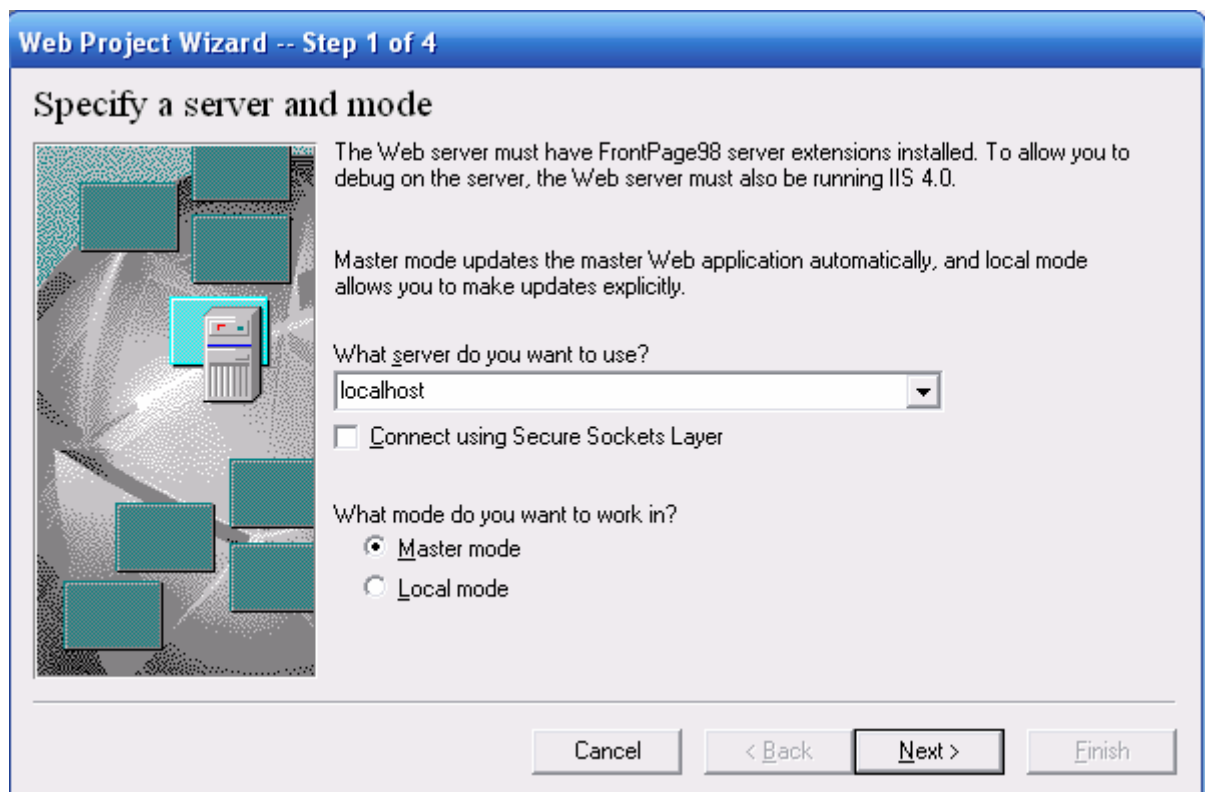


1.2 Nhập tên Project, như trong hình trên ứng dụng tự đặt tên là Project3 không trùng với những đề án đã có. Chúng ta có thể thay đổi lại tên đề án, nên đặt tên cho gợi nhớ phù hợp với ứng dụng mà chúng ta muốn phát triển.

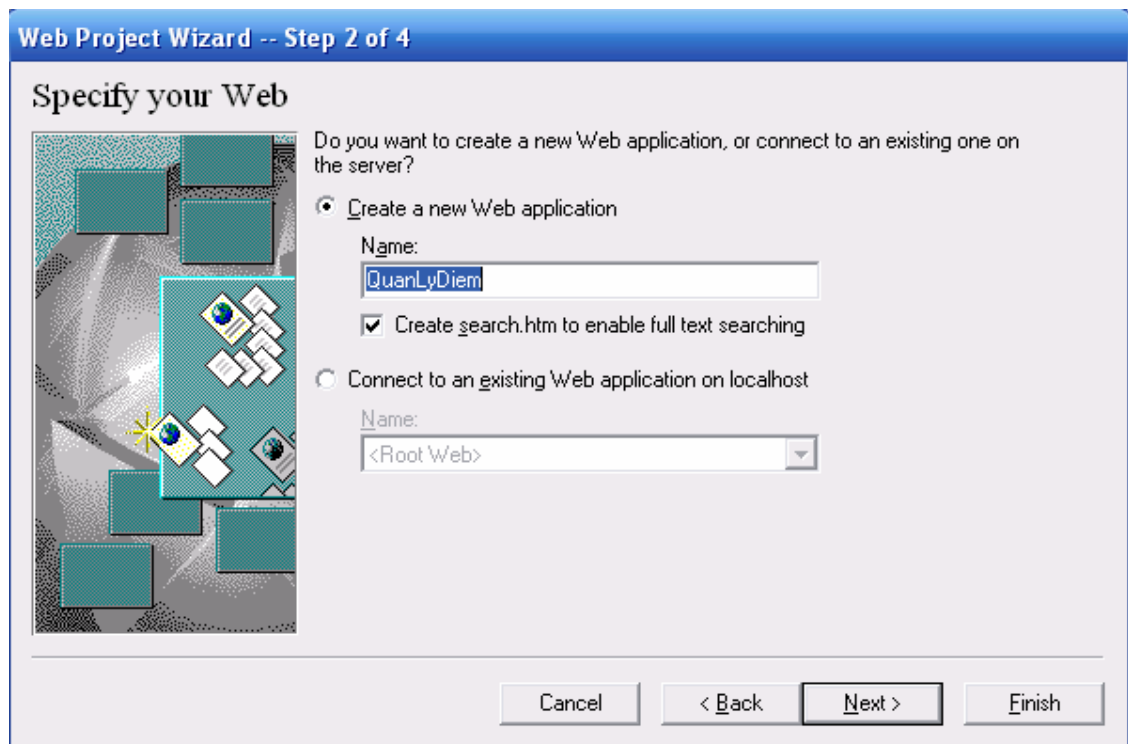




Nếu muốn thay đổi thư mục lưu trữ thì nhấn vào nút Browse... để chọn nơi lưu trữ. Nhấn nút Open để tiếp tục, cửa sổ tiếp theo sẽ hiện ra:

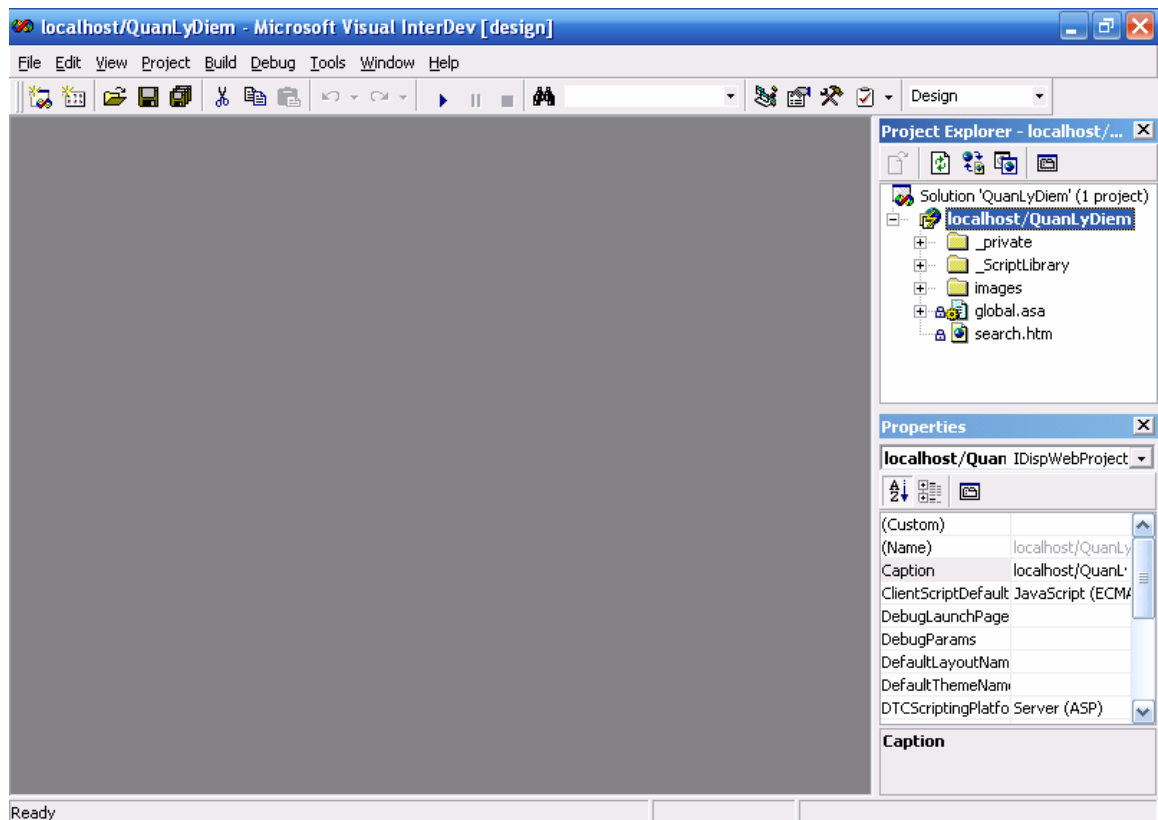


1.3 Nhập vào tên Server, nếu chúng ta đang phát triển đề án web trên IIS của máy cục bộ thì nhập vào tên server là **localhost** như hình trên. Nhấn Next để tiếp tục, cửa sổ tiếp theo sẽ hiện ra:



1.4 Nếu muốn chọn những mẫu giao diện có sẵn thì nhấn nút Next để tiếp tục, bằng không thì nhấn Finish để hoàn tất.

1.5 Khi nhấn nút Finish để hoàn thành, cửa sổ môi trường phát triển web sẽ xuất hiện có dạng như sau:

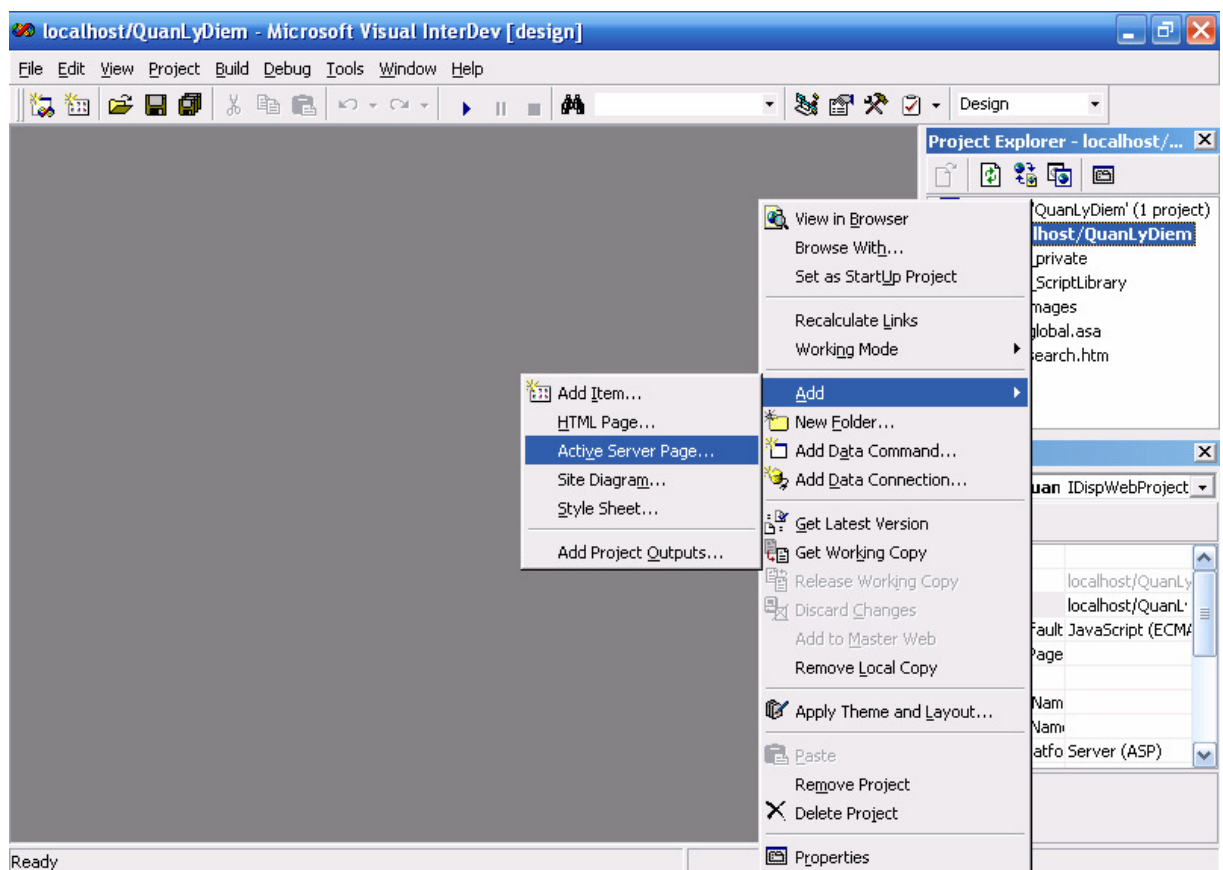


Mặc định, đề án sẽ tạo ra các thư mục và tập tin cần thiết để lưu trữ thông tin về đề án, một thư mục images và một tập tin global.asa (chúng ta sẽ tìm hiểu tập tin này ở chương sau).

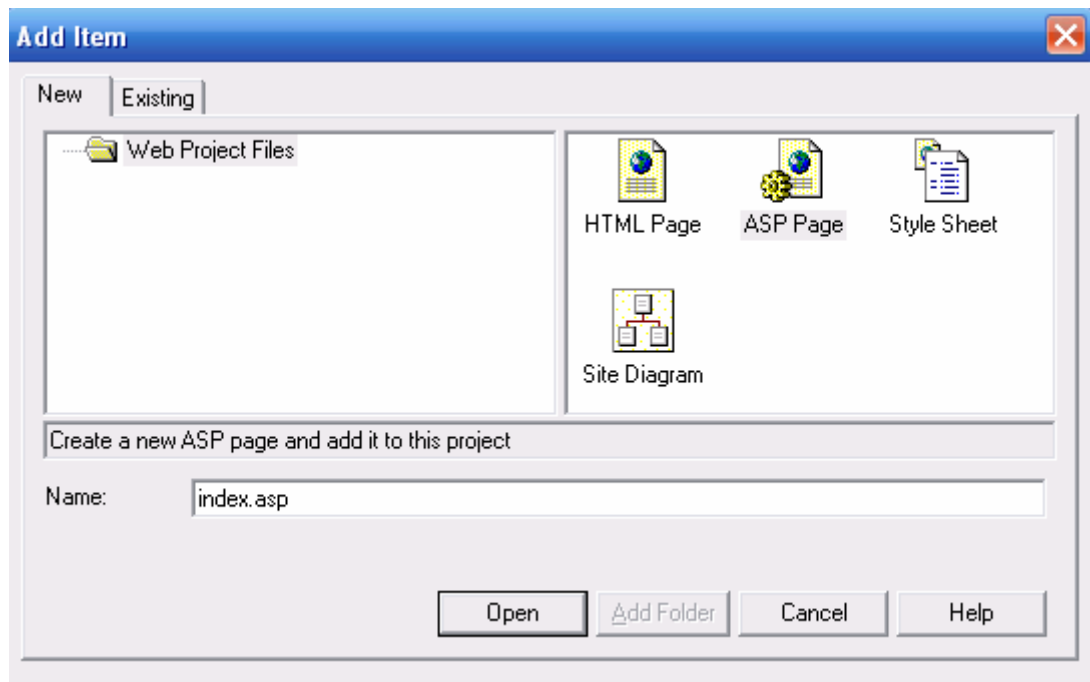
Bây giờ, công việc tiếp theo là làm thế nào để kiểm tra rằng đề án của chúng ta có thể phát triển và chạy được. Ta tiến hành các bước như sau:

a) Bổ sung vào một tập tin mới web mới, để ngôn ngữ kịch bản VBScript có thể chạy được, ta phải tạo tập tin Active Server Page (ASP) (đây cũng là loại tập tin mà chúng ta sẽ sử dụng trong hầu hết quá trình lập trình).

Nhấn chuột phải lên tên Project → Add → Active Server Page...



Cửa sổ tiếp theo sẽ hiện ra:



Đặt lại tên như hình trên, nhấn Open để tạo tập tin ASP mới.

Trang mới tạo chỉ gồm một số dòng như sau, ở đây chúng ta quan tâm nhất đến dòng đầu tiên, dòng đầu tiên để chỉ cho trình duyệt biết rằng ngôn ngữ kịch bản chính được sử dụng cho phía Server là VBScript.

```
<%@ Language=VBScript %>
```

```
<HTML>
```

```
<HEAD>
```

```
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
```

```
</HEAD>
```

```
<BODY>
```

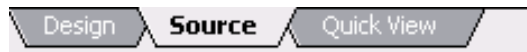
```
<P>&nbsp;</P>
```

```
</BODY>
```

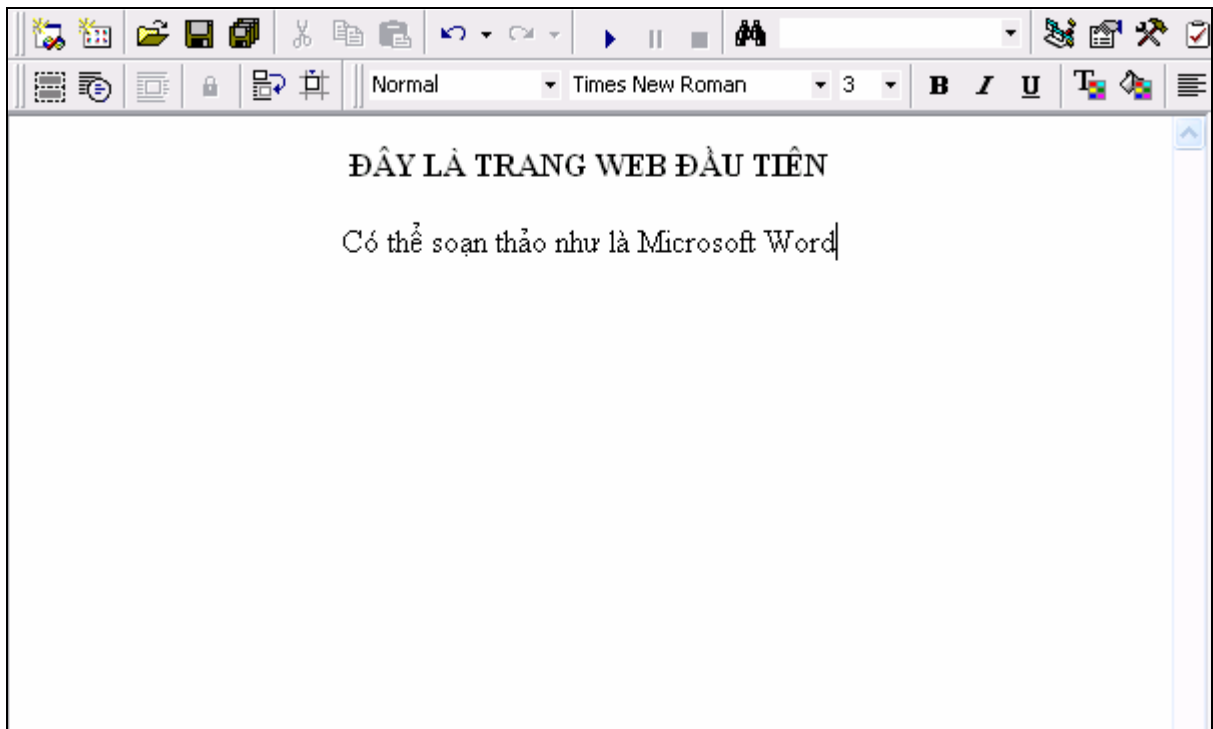
```
</HTML>
```

b) Đưa thông tin, viết lệnh cho trang web.

Ở phần dưới của trang chúng ta thấy có ba chức năng là Design, Source và Quick View, mặc định là Source:



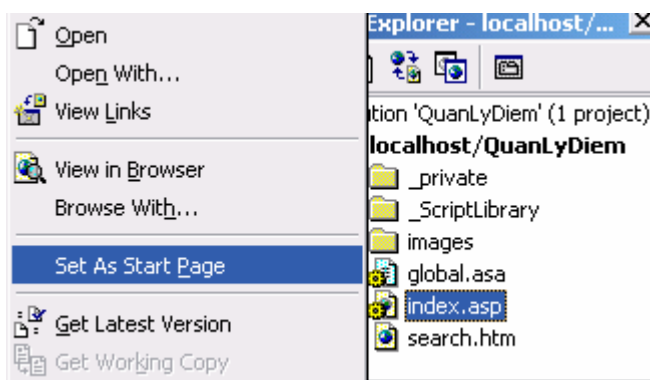
Ta có thể chuyển sang chế độ Design để nhập văn bản như là Microsoft Word:



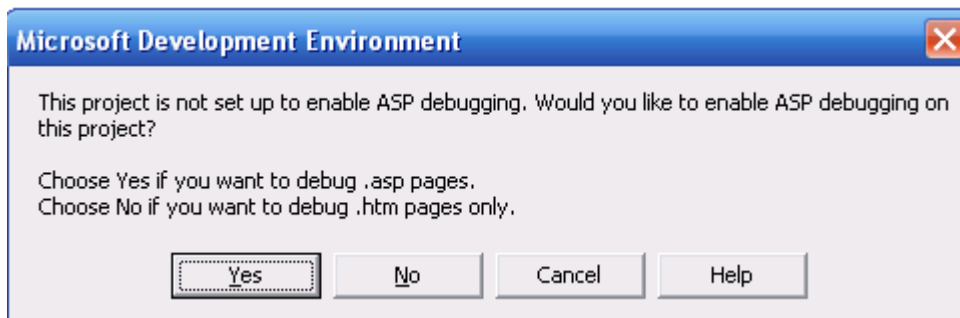
c) Lưu nội dung trang web bằng cách nhấn tổ hợp Ctrl+S hoặc nhấn nút Save

d) Thiết lập trang mặc định cho đề án:

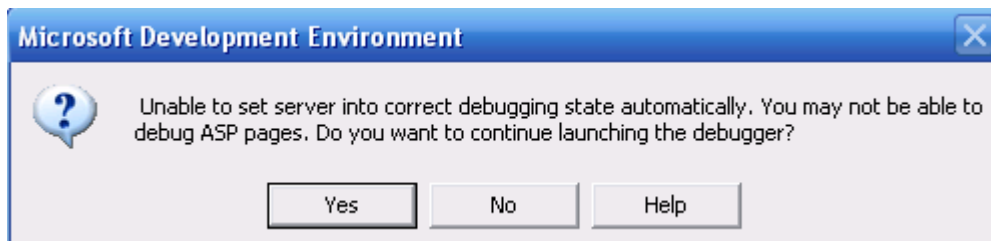
Nhấn chuột phải lên tên tập tin index.asp, chọn Set As Start Page:



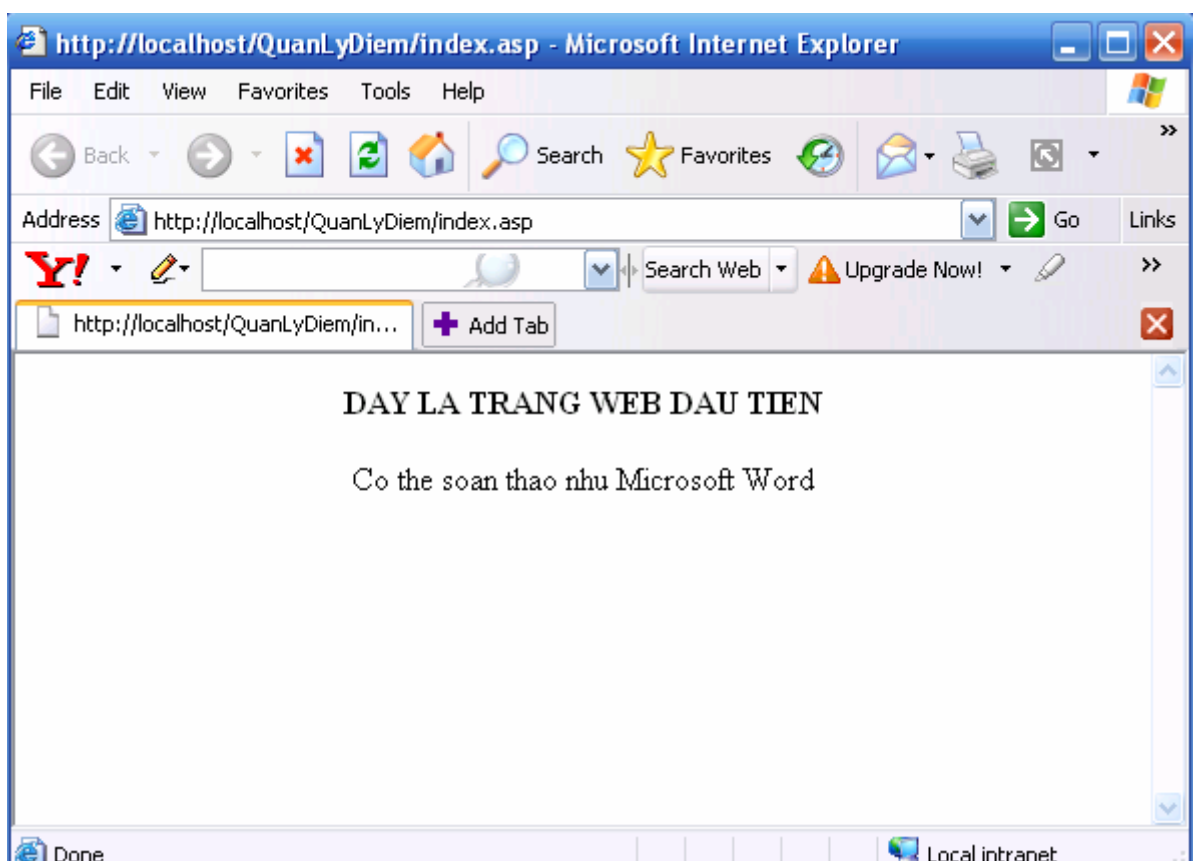
e) Click nút Start hoặc nhấn F5 để chạy đề án, nếu lần đầu tiên hộp thoại sau sẽ xuất hiện:



Chọn Yes để tiếp tục, cửa sổ tiếp theo hiện ra:



Chọn Yes một lần nữa, nếu trang web không có lỗi sẽ được hiển thị trong trình duyệt của Internet Explorer như sau:



Lưu ý: Trong quá trình thực hành, chúng ta chỉ cần gõ tiếng Việt không dấu!

- Sau khi xem kết quả hiển thị trên trình duyệt, hãy nhấn nút Stop hoặc đóng cửa sổ trình duyệt lại, chuyển sang Tab HTML, bổ sung đoạn lệnh sau vào phần <body> của trang với nội dung:

```
<% If (Time >=#12:00:00 AM#) then%>
Good Morning!
<% Else %>
Hello !
<%End If %>
```

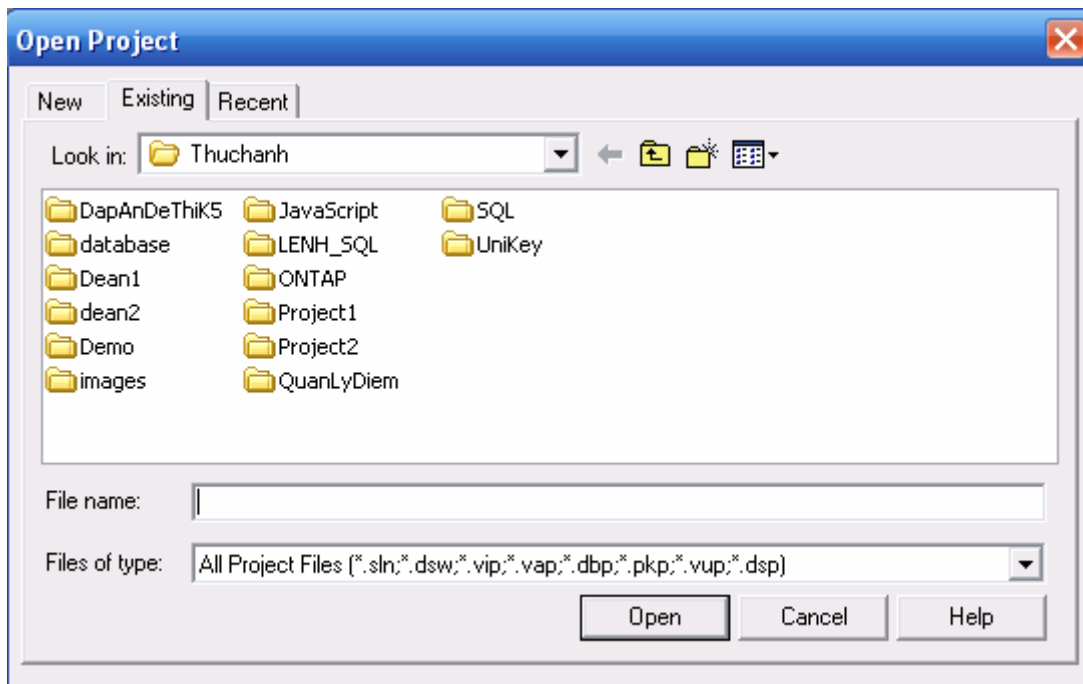
- Lưu trang web và chạy lại. Kết quả hiện ra thật bất ngờ! Những câu lệnh này có ý nghĩa gì? Chúng ta sẽ tìm hiểu trong chương 6.

## 2. Đóng một đề án Web

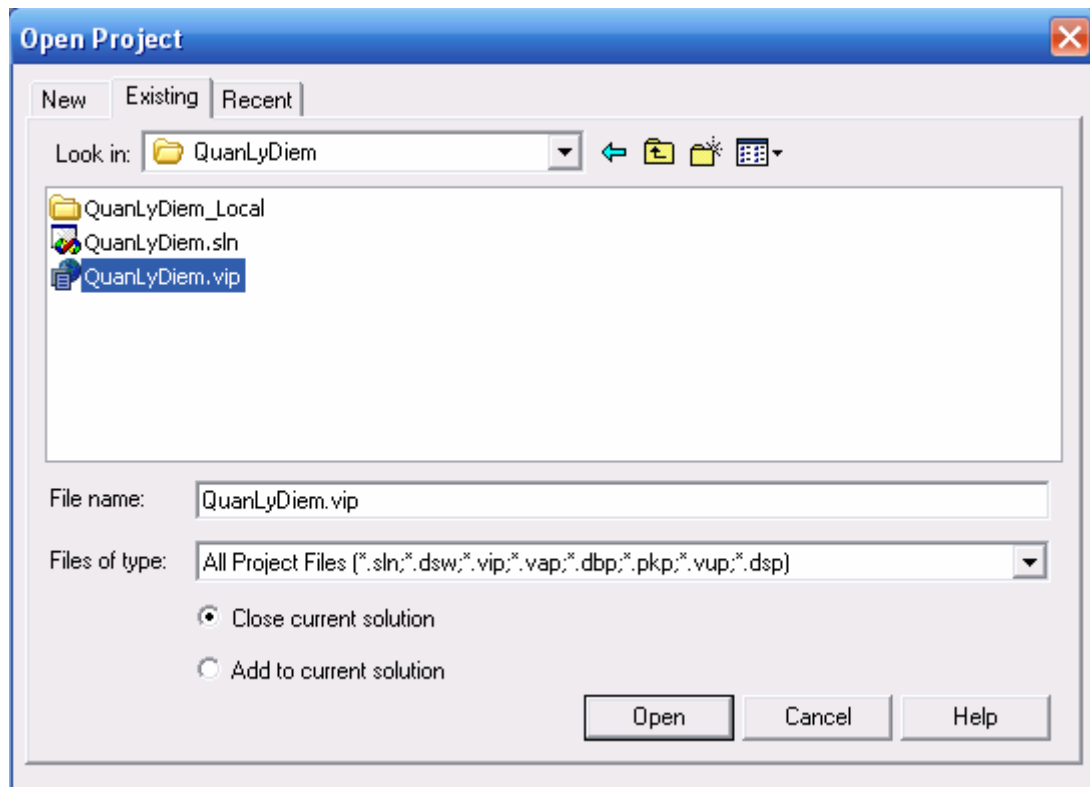
Vào menu File → Close All.

## 3. Mở một đề án đã có sẵn

Nhấn nút Open hoặc vào menu File → Open Project, chọn tab Existing như hình sau:

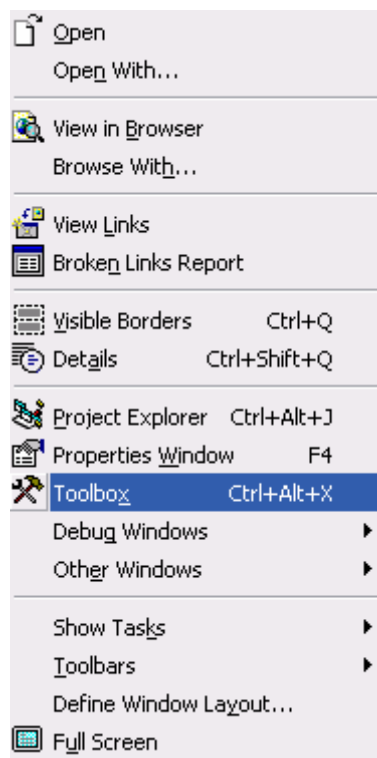


Tìm đến tên đề án cần mở, chọn tập tin có phần mở rộng **.sln** hoặc **.vip** trong đề án → Open.



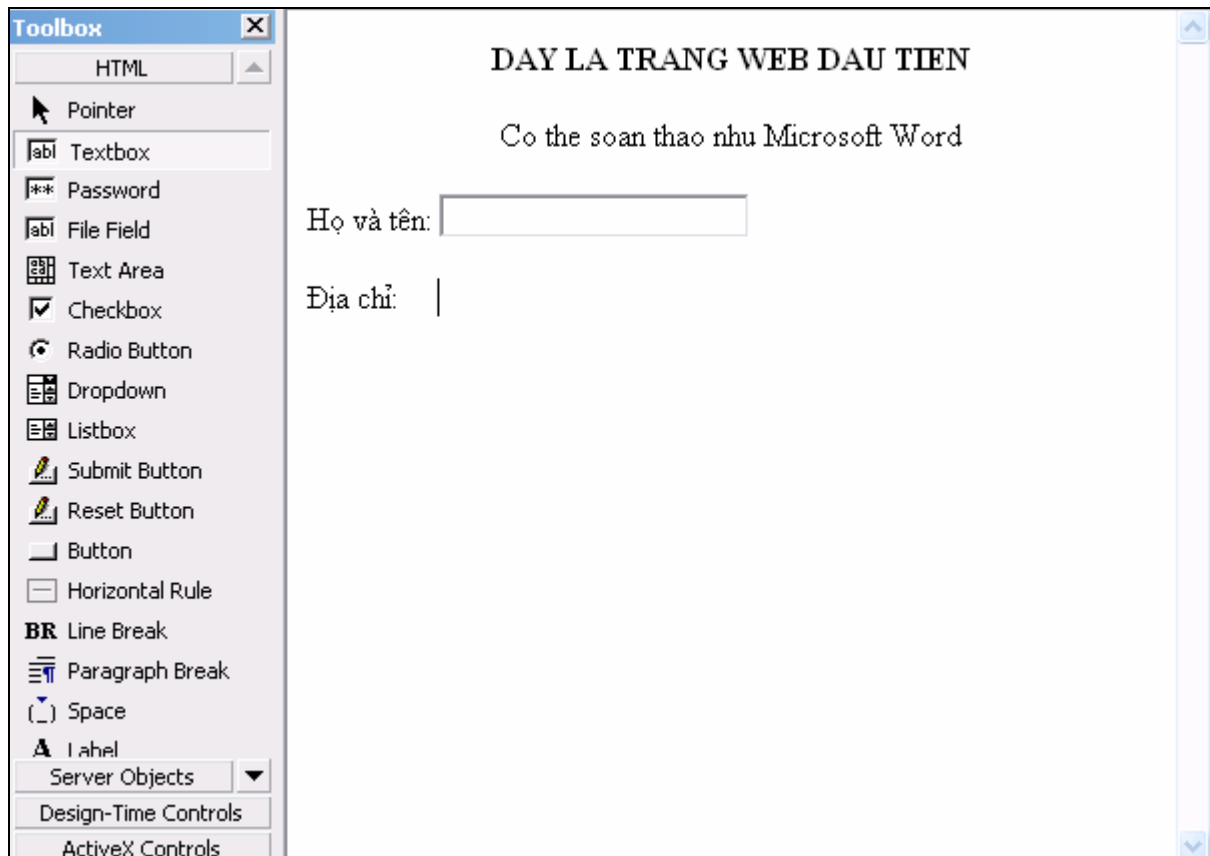
### III. Thiết kế giao diện

Để thuận tiện trong việc thiết kế giao diện, chúng ta cho hiện hộp ToolBox bằng cách vào menu View → ToolBox:



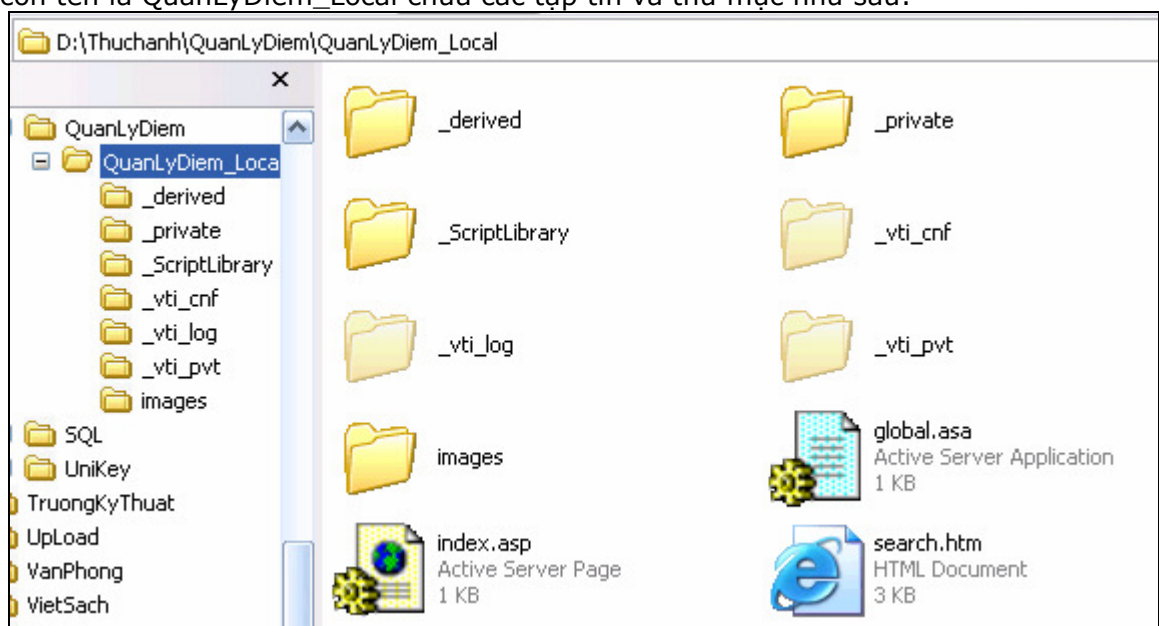


Cửa sổ hiện ra gồm các đối tượng HTML, để đưa các đối tượng này vào trang web, chúng ta click và rê chuột từ các đối tượng vào trang web:



#### IV. Tổ chức hệ thống tập tin thư mục của một Web site

Khi tạo một đề án web, ứng dụng luôn tạo ra một thư mục có tên **<tên\_đề\_án>\_\_Local**. Thư mục này sẽ chứa tất cả những tập tin, thư mục mà ta đã xây dựng trong đề án. Chẳng hạn, với đề án QuanLyDiem vừa tạo sẽ tự sinh ra một thư mục con tên là QuanLyDiem\_Local chứa các tập tin và thư mục như sau:



Người phát triển có thể thao tác Copy, Cut, Delete trực tiếp trong đề án Web.

## CHƯƠNG 3: SIÊU LIÊN KẾT

### **Mục tiêu:**

- Tạo được các mối liên kết từ một trang Web này đến một trang Web khác
- Tổ chức các mối liên kết

### **I. Khái niệm về siêu liên kết**

- Đặc trưng cơ bản của ngôn ngữ HTML chính là tính siêu liên kết. Tính siêu liên kết thể hiện ở tất cả mọi loại tài liệu mà người đọc tự do đuổi theo các dấu vết liên quan qua suốt tài liệu đó bằng các mối liên kết xác định sẵn do người sử dụng tự lập nên. Một khái niệm quan trọng được sử dụng trong liên kết là Hypertext.

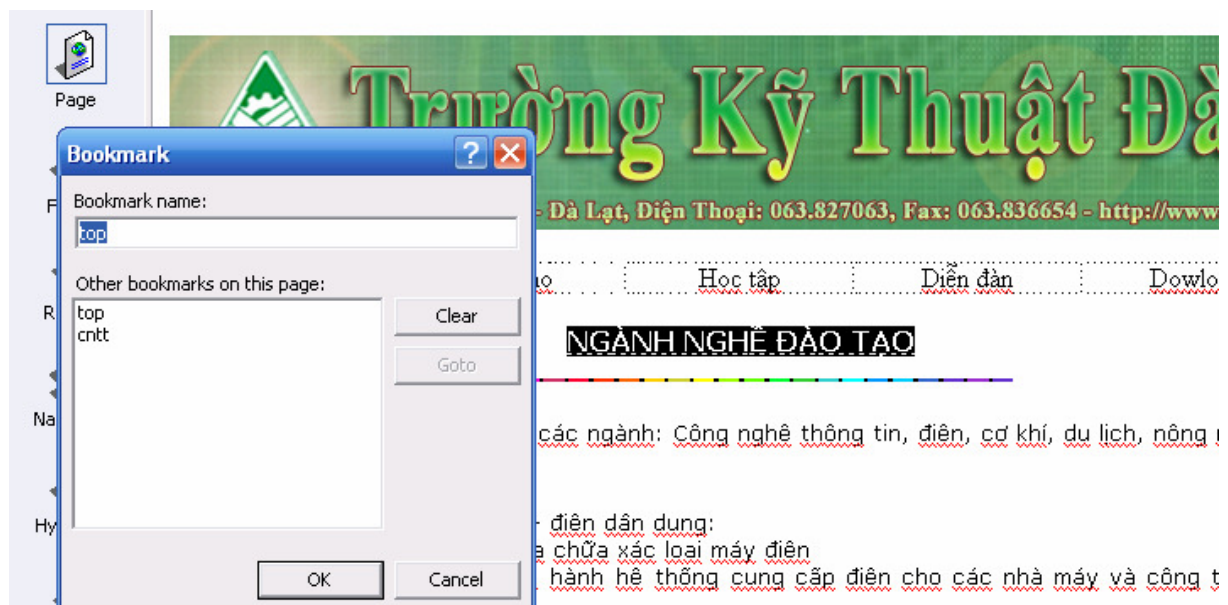
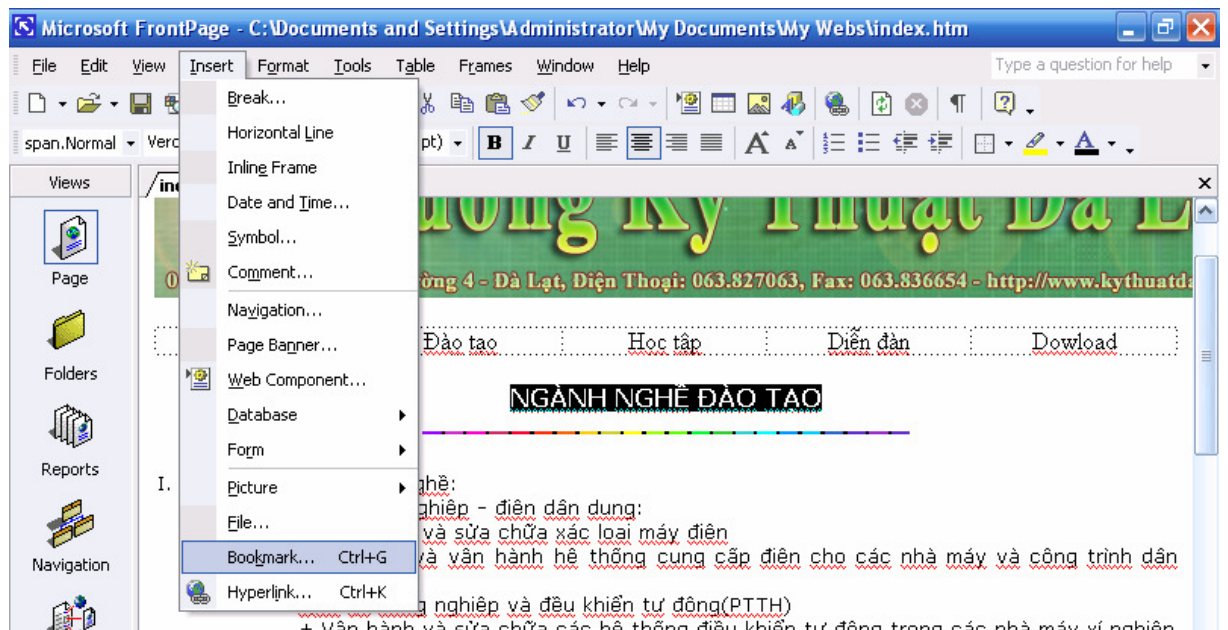
Trong một môi trường ứng dụng hypertext thực sự, bạn có thể trở vào ( highlight) bất kỳ từ nào của tài liệu và sẽ tức khắc nhảy đến các tài liệu khác có văn bản liên quan đến nó. Cũng có những lệnh cho phép bạn tự tạo cho riêng mình những dấu vết kết hợp qua suốt tài liệu. Các trình ứng dụng hypertext rất hữu ích trong trường hợp phải làm với một số lượng văn bản lớn, như các bộ từ điển bách khoa và các bộ sách nhiều tập.

Trong chương này chúng ta sẽ tìm hiểu liên kết qua môi trường ProntPage 2002 (ProntPage XP).

### **II. Tạo siêu liên kết**

#### **1. Tạo Bookmark**

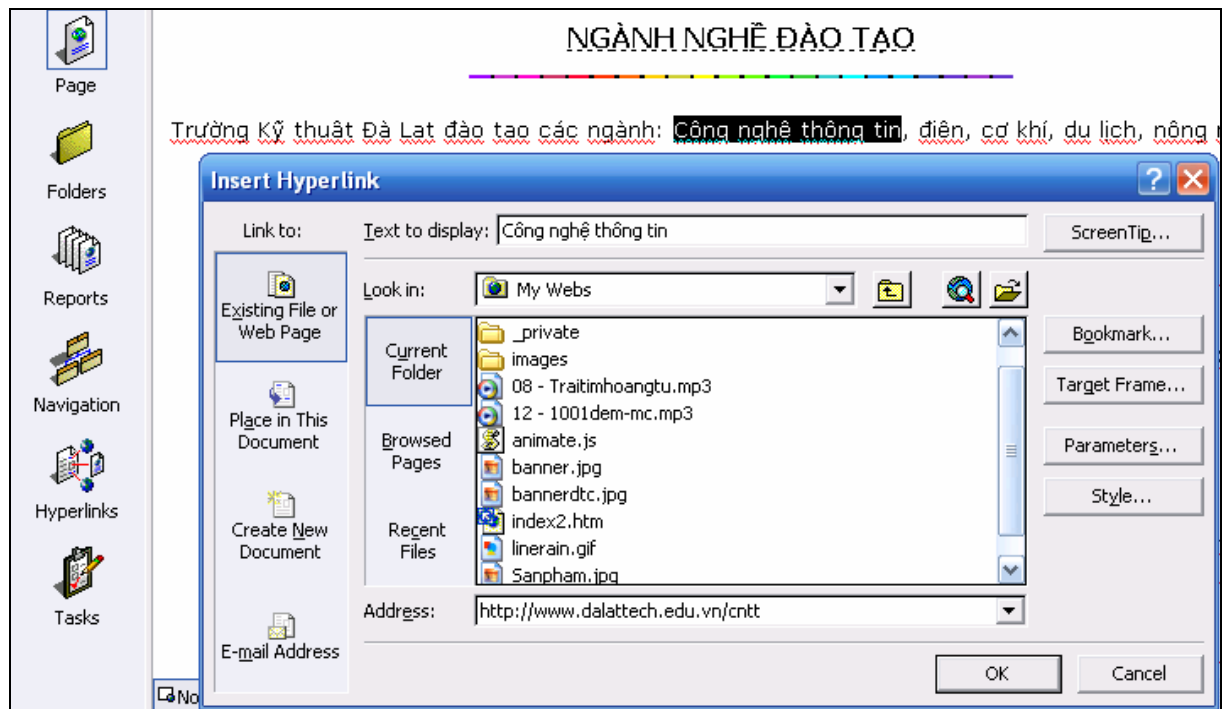
- Chọn đối tượng cần đặt tên (lưu ý, đối tượng ở đây có thể là đối tượng Text hoặc hình ảnh, ...).
- Vào menu Insert → Bookmark, trong phần Bookmark name nhập tên cho đối tượng.



Đặt tên cho đối tượng Bookmark

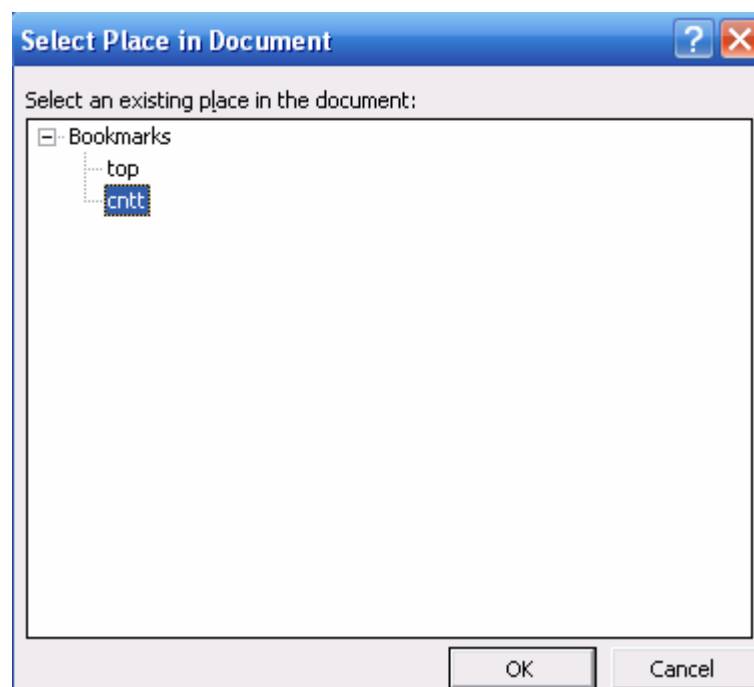
## 2. Tạo liên kết (Hyperlink)

- Chọn đối tượng cần liên kết
- Vào Insert → Hyperlink (hoặc nhấn Ctrl+K hoặc nhấn vào nút lệnh có hình quả cầu)
- Tại phần Address, nhập vào địa chỉ URL hoặc chọn tập tin cần liên kết.



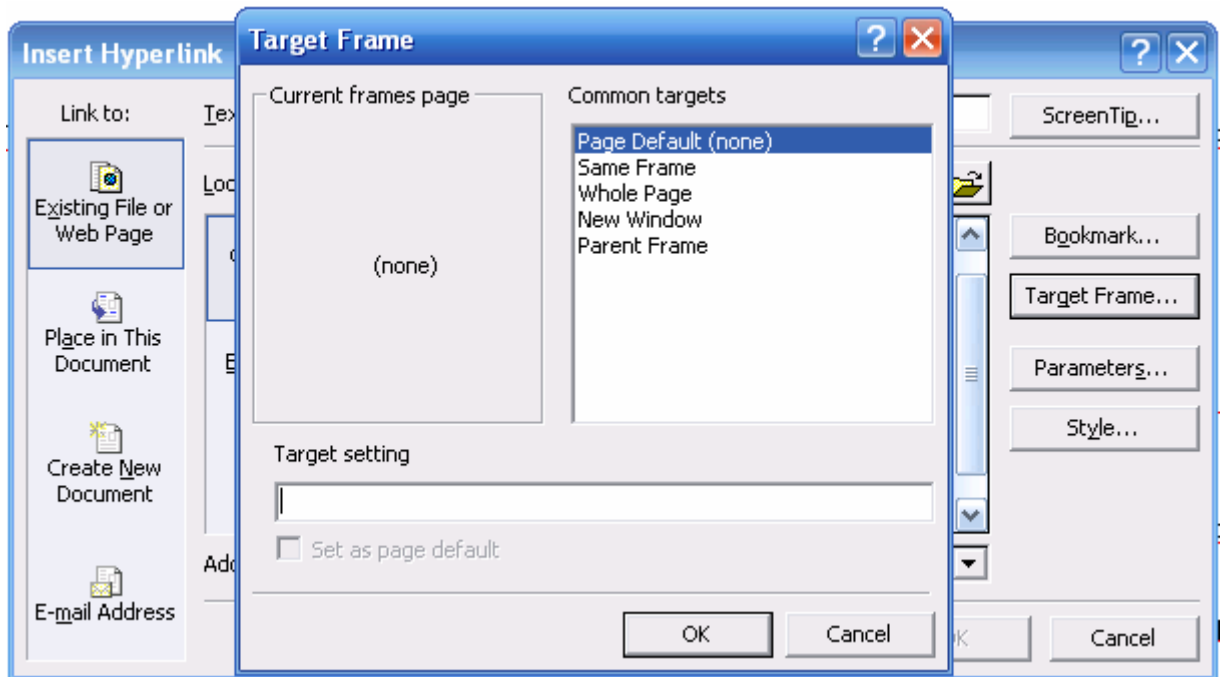
### Liên kết tới một địa chỉ URL

- Nếu muốn liên kết tới một Bookmark trong trang web vừa tạo thì nhấn vào nút Bookmark... → chọn tên Bookmark:



### Liên kết tới một Bookmark

- Nếu trang có sử dụng Frame thì bạn có thể chỉ định cho liên kết được hiển thị trong phần Frame nào bằng cách nhấn vào nút Target frame, cửa sổ hiện ra cho phép ta chọn các tham số:



Chọn Frame hiển thị

- + Same Frame: trang hiển thị đúng với frame chứa trang có chứa đường link
- + Whole Page: trang hiển thị toàn bộ cửa sổ (không thuộc một frame nào)
- + New Window: trang hiển thị ở một cửa sổ mới (không thuộc frame nào)
- + Parent Frame: trang hiển thị ở frame cha (bao frame có chứa đường link)

### 3. Xem mã HTML của siêu liên kết

Bây giờ chúng ta hãy xem HTML đã sinh ra những thẻ gì trong những công việc ta vừa thực hiện, từ đó các bạn có thể nắm bắt được những đoạn mã chủ yếu tạo nên sự siêu liên kết thú vị này.

Như ở trên, chúng ta đã tạo ra một bookmark tên là **cntt** (ở nội dung Công nghệ thông tin) và liên kết đoạn văn bản **Công nghệ thông tin** tới bookmark này, HTML đã sinh ra đoạn mã với thẻ `<a> ...</a>` sau:

**`<a target="_self" href="#cntt">Công nghệ thông tin</a>`**

- Nếu muốn liên kết đến một địa chỉ URL, các bạn chỉ cần thay đổi thuộc tính **href=<địa chỉ URL>**.

- Nếu muốn thay đổi frame hiển thị, chúng ta thay đổi thuộc tính target.

- Đôi khi nội dung trang web dài và chiếm nhiều trang màn hình, nhưng nội dung lại có mối liên hệ chặt chẽ, hay vì 1 lý do nào đó mà ta không muốn toàn bộ nội dung này được chứa trong cùng 1 trang web và chia ra nhiều trang. Khi đó ta cần đặt các hyperlink ngay đầu trang web để khi cần người dùng có thể click vào đó để đi đến một đoạn tư liệu họ quan tâm mà không cần phải kéo scrollbar đến đoạn đó.

- **Định nghĩa tag `<A></A>` như sau:**

**a) Định nghĩa 1 bookmark và hyperlink trong cùng trang web:** bookmark là kỹ thuật chỉ định 1 định danh (đích) để hyperlink nhảy đến đầu đoạn văn bản.

`<A name="tên">chuỗi ký tự đầu đoạn văn bản</A>`

Ví dụ:

`<A name="Nd phần 1">Nội dung phần 1</A>` bao gồm các nội dung chi tiết sau...

Khi đó trong phần khai báo hyperlink bạn khai báo theo cú pháp sau:

`<a href="#tên_của_bookmark_đã_khai_báo">Chuỗi hướng dẫn nhảy đến đầu đoạn có bookmark="tên_của_bookmark_đã_khai_báo"</a>`

Ví dụ:

`<a href="#Nd phần 1">Xem nội dung phần 1</a>`

**b) Hyperlink đến một trang web khác:**

Cú pháp:

`<A HREF="địa_chỉ_trang_web_sẽ_nhảy_tới"> Chuỗi diễn giải hyperlink </A>`

Ví dụ:

`<A HREF="http://www.dalattech.edu.vn">Link đến trang web chủ của hãng máy tính Microsoft.</A>`

`<A HREF="trang_chu.htm">Link đến trang web trang_chu.htm trong cùng thư mục.</A>`

`<A TARGET="viewer" HREF="sample.htm">Click vào đây để mở trang web sample.htm trong cửa sổ "viewer".</A>`

`<A HREF="http://www.dalattech.edu.vn"><IMG SRC="images/bullet.gif">Click vào ảnh này để link đến trang web của Microsoft</A>`

`<A HREF="mailto:admin@kythuatdalat.edu.vn"> admin@kythuatdalat.edu.vn </A><br>`

## CHƯƠNG 4: TẠO DANH SÁCH, THIẾT KẾ BẢNG, TẠO BIỂU MẪU GỬI DỮ LIỆU QUA INTERNET

### Mục tiêu:

- Tạo các danh sách, theo thứ tự, không theo thứ tự, danh sách lồng nhau
- Thiết kế và định dạng các bảng, thiết kế và định dạng khung
- Tạo mẫu biểu, tạo các hộp nhập dữ liệu, tạo các nút, gửi dữ liệu mẫu biểu thông qua e-mail

### I. Định nghĩa một danh sách

#### - Thẻ `<DL>` `</DL>`

Cú pháp:

```
<DL> Đề mục danh sách
    <DT> Danh sách mức 1 </DT>
    <DT> Danh sách mức 1 </DT>
    .....
        <DD> Danh sách mức 2 </DD>
        <DD> Danh sách mức 2 </DD>
    .....
    <DT> Danh sách mức 1 </DT>
        <DD> Danh sách mức 2 </DD>
</DL>
```

Ví dụ: Trong phần HTML chúng ta gõ vào đoạn lệnh sau:

```
<html>
<head>
<title>Vi du ve danh sach</title>
</head>
<body>
<b><DL> Đề mục danh sách</b>
    <DT> Danh sách mức 1 </DT>
    <DT> Danh sách mức 1 </DT>
        <DD> Danh sách mức 2 </DD>
        <DD> Danh sách mức 2 </DD>
    <DT> Danh sách mức 1 </DT>
        <DD> Danh sách mức 2 </DD>
</DL>
</body>
</html>
```

→ Kết quả hiển thị trong trình duyệt:

### Đề mục danh sách

Danh sách mức 1

Danh sách mức 1

Danh sách mức 2

Danh sách mức 2

Danh sách mức 1

Danh sách mức 2

- **Thẻ <UL><LI></UL>, <OL><LI></OL>**: Danh sách sắp xếp (Ordered List), danh sách không sắp xếp (Unordered List) và danh sách các định nghĩa (Definition List). Danh sách sắp xếp bắt đầu bằng tag <OL>, danh sách không sắp xếp bắt đầu bằng 1 trong các tag sau: <UL>, <DIR> và <MENU>, danh sách các định nghĩa bắt đầu bằng tag <DL>. Để thể hiện mỗi một mẫu trong danh sách sắp xếp và không sắp xếp dùng tag <LI>.

UL, DIR, MENU: danh sách không sắp xếp kiểu bullet, mỗi mẫu bắt đầu bằng tag LI.

Ví dụ: danh sách kiểu bullet (không sắp xếp).

<UL>

<LI>Đây là mục 1 trong danh sách.

<LI>Đây là mục 2 trong danh sách.

</UL>

Kết quả trình duyệt như sau:

- Đây là mục 1 trong danh sách.
- Đây là mục 2 trong danh sách.

OL: danh sách sắp xếp. Mỗi mục trong danh sách cũng bắt đầu bằng <LI>.

### - Định nghĩa OL:

<OL START=n TYPE=order-type> trong đó:

START=n: chỉ định chỉ số bắt đầu

TYPE=order-type: chỉ định kiểu chỉ số. Có thể là một trong các giá trị sau:

A: sử dụng ký tự lớn A,B,C,...

a: sử dụng ký tự nhỏ a,b,c,...

I: sử dụng số La Mã lớn I,II,III,...

i: sử dụng số La Mã nhỏ i,ii,iii,...

1: sử dụng số 1,2,3,...

Ví dụ:

<OL TYPE=I>

<LI>Nghị thức HTTP

<LI>Nghị thức FTP

<LI>Nghị thức SMTP

</OL>



Kết quả trình duyệt như sau:

- I. Nghi thức HTTP
- II. Nghi thức FTP
- III. Nghi thức SMTP

## II. Thiết kế bảng

- **Thẻ <TABLE><TR><TD></TABLE>**: sử dụng tag <TABLE> để bắt đầu và kết thúc 1 bảng. Sử dụng kèm theo với tag <TR></TR> để thêm 1 dòng trong bảng và tag <TD></TD> để thêm vào 1 ô trên dòng.

Ví dụ:

```
<TABLE>
<TR><TD>Ô 11<TD>Ô 12
<TR><TD>Ô 21<TD>Ô 22
</TABLE>
```

Kết quả trình duyệt như sau:

Ô 11   Ô 12  
Ô 21   Ô 22

Để đóng khung cho bảng, sử dụng thuộc tính BORDER

Ví dụ:

```
<TABLE BORDER=1>
<TR><TD>Ô 11<TD>Ô 12
<TR><TD>Ô 21<TD>Ô 22
</TABLE>
```

Kết quả trình duyệt như sau:

Ô 11	Ô 12
Ô 21	Ô 22

Chiều rộng mặc định của bảng sẽ là chiều rộng lớn nhất của dòng rộng nhất trong bảng. Để chỉ định chiều rộng các bảng, sử dụng thuộc tính WIDTH=n% để quy định chiều rộng.

Ví dụ:

```
<TABLE BORDER=1 WIDTH=100%>
<TR><TD>Ô 11<TD>Ô 12
<TR><TD>Ô 21<TD>Ô 22
</TABLE>
```

Kết quả trình duyệt như sau:

Ô 11	Ô 12
Ô 21	Ô 22

Để thêm tiêu đề cho bảng, sử dụng tag <CAPTION></CAPTION>. Mặc định tiêu đề của bảng nằm ở trên và canh lề giữa. Tuy nhiên có thể thay đổi bằng cách dùng thuộc tính ALIGN bên trong CAPTION.

Ví dụ:

```
<TABLE BORDER=1 WIDTH=100%>
<CAPTION> Tiêu đề của bảng</CAPTION>
<TR><TD>Ô 11<TD>Ô 12
<TR><TD>Ô 21<TD>Ô 22
</TABLE>
```

Kết quả trình duyệt như sau:

Tiêu đề của bảng

Ô 11	Ô 12
Ô 21	Ô 22

Để thêm vào các ô trải dài trên nhiều cột, dòng khác, dùng thuộc tính COLSPAN=n và ROWSPAN=n.

Ví dụ:

```
<TABLE BORDER=1 WIDTH=100%>
<CAPTION> Tiêu đề của bảng</CAPTION>
<TR><TD>Ô 11<TD COLSPAN=2>Ô 12 trải dài trên 2 ô
<TR><TD>Ô 21<TD>Ô 22<TD>Ô 23
</TABLE>
```

Kết quả trình duyệt như sau:

Ô 11	Ô 12 trải dài trên 2 ô	
Ô 21	Ô 22	Ô 23

#### - Định nghĩa TABLE(bảng):

```
<TABLE
ALIGN=align-type
BACKGROUND=url
BGCOLOR=color-type
BORDER=n
CELLPADDING=n
WIDTH=n%>
```

Trong đó:

ALIGN=align-type: lề của bảng, có giá trị là LEFT hoặc RIGHT

BACKGROUND=url: chỉ định ảnh nền của bảng

BGCOLOR=color-type: màu nền của bảng

BORDER=n: đường viền bảng, n tính bằng pixel

CELLPADDING=n: khoảng cách từ ô tới nội dung của ô, đơn vị pixel

WIDTH=n: độ rộng của bảng, n tính theo % (phải có dấu % theo sau)

**- Định nghĩa TR(dòng):**

<TR

ALIGN=align-type

BACKGROUND=url

BGCOLOR=color-type

VALIGN=v-align-type>

Trong đó:

ALIGN=align-type: lề của các ô trong dòng, có giá trị là LEFT, RIGHT hoặc CENTER

BACKGROUND=url: chỉ định ảnh nền của dòng

BGCOLOR=color-type: màu nền của dòng

VALIGN=v-align-type: lề theo chiều dọc cho văn bản trong các ô trên dòng. Giá trị có thể là: TOP, BOTTOM, MIDDLE.

**- Định nghĩa TD (ô):**

<TD

ALIGN=align-type

BACKGROUND=url

BGCOLOR=color-type

COLSPAN=n

ROWSPAN=n

VALIGN=v-align-type>

Trong đó:

ALIGN=align-type: lề văn bản trong ô, có giá trị là LEFT, RIGHT hoặc CENTER

BACKGROUND=url: chỉ định ảnh nền cho ô

BGCOLOR=color-type: màu nền của ô

COLSPAN=n: ô trải rộng trên n cột

ROWSPAN=n: ô trải dài trên n hàng

VALIGN=v-align-type: lề theo chiều dọc cho văn bản trong các ô. Giá trị có thể là: TOP, BOTTOM, MIDDLE.

**TH:** tạo ô tiêu đề. Tương tự như TD chỉ khác là văn bản được thể hiện ở dạng nghiêng.

### III. Tạo biểu mẫu

- **Thẻ <form> ...</form>:** là công cụ chủ yếu tạo ra các trang web có tính chất tương tác.

- Biểu mẫu cho phép người sử dụng nhập thông tin hoặc tùy chọn các khả năng có sẵn.

- Thẻ <form> thường gắn với một chương trình thực thi, thường gọi là "script", chương trình này sẽ thực hiện một công việc nào đó khi người sử dụng nhấn vào nút **Submit**.

- Cú pháp:

**<FORM METHOD=<Phương thức> ACTION=<Đường dẫn đến chương trình>**

[nội dung trong FORM có thể là văn bản hoặc các thẻ FORM]

**<input type=submit value = "Nút Submit">**

**</FORM>**

- Phương thức: gồm phương thức POST và GET.

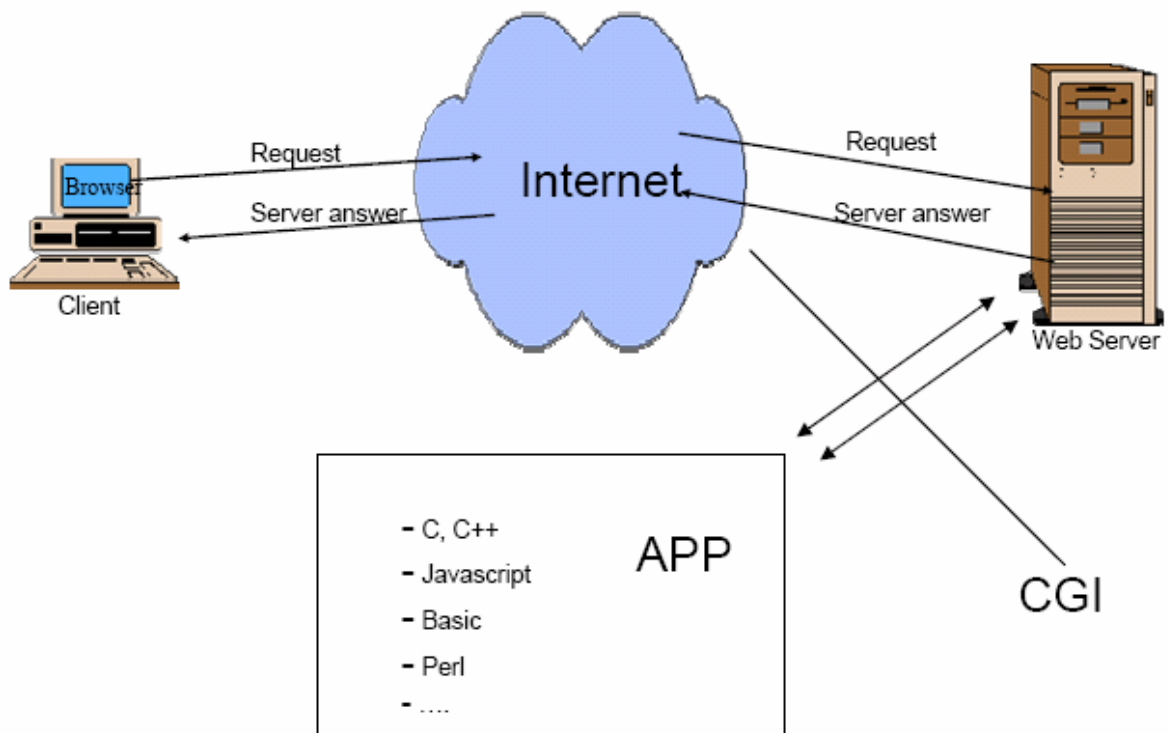
+ Khi dùng phương thức POST, dữ liệu gửi đi sẽ *không được hiển thị* trên địa chỉ URL.

+ Khi dùng phương thức GET, dữ liệu gửi đi *được hiển thị* trên địa chỉ URL.

- Các thẻ Form: có thể là thẻ <input>, <TextArea>, <Button>, <option>, ...

#### IV. Gửi dữ liệu biểu mẫu qua Internet

- Khi người sử dụng nhấn nút Submit, thông tin trong form sẽ được gửi đi lên Server để yêu cầu (Request) Server trả lời (Server Answer). Trên Server đã được cài đặt các chương trình ứng dụng (có thể bằng một ngôn ngữ Web C++, ASP, Java,...), các chương trình ứng dụng này sẽ tự động xử lý thông tin gửi đi từ người sử dụng (Client) và phản hồi thông tin (Response). Phương thức này đôi khi người ta còn gọi là Yêu cầu – Đáp ứng, tức là nên Client thì Request còn trên Server sẽ Response. Các bạn có thể hình dung tiến trình gửi thông tin qua Internet với sơ đồ sau:



Cách thức trao đổi thông tin qua chương trình

## V. Tạo khung (Frame)

### - Thẻ Frame: <FRAMESET><FRAME></FRAMESET>:

Sử dụng tag <FRAMESET> và </FRAMESET> để chia cửa sổ trình duyệt thành nhiều cửa sổ con frame. Tag <FRAMESET> được dùng kèm với tag <FRAME> để định nghĩa 1 frame.

Ví dụ: tạo trang web có tên là 2frame.htm chứa 2 frame, frame 1 bên trái chứa nội dung trang web page\_1.htm và frame 2 bên phải chứa nội dung trang web page\_2.htm.

```
<html>
<head>
<title>Trang web có 2 khung</title>
</head>
<frameset cols="150,*">
  <frame name="muc_luc" target="noi_dung"
src="http://localhost/html/page_1.htm">
  <frame name="noi_dung" src="http://localhost/html/page_2.htm">
</frameset>
</html>
```

Mỗi frame có 1 tên, ví dụ frame trái có `name="muc_luc"` và frame phải có `name="noi_dung"`. Để tạo liên kết từ frame muc\_luc với frame noi\_dung ta chỉ định trong frame muc\_luc như sau: `target="noi_dung"`. `src=file:` để chỉ định ra trang web thể hiện trên khung. Các trang web page\_1.htm và page\_2.htm cần tạo ra trước khi tạo trang web 2frame.htm chứa chúng. Nội dung của trang web page\_1.htm trong tag <HEAD></HEAD> có khai báo sau:

```
<head>
<base target="noi_dung">
</head>
```

Để không xuất hiện scrollbar và border của frame, khi đó bạn thêm thuộc tính `scrolling="no"` và `frameborder=0` vào tag <FRAME>, ví dụ:

```
<frame name="muc_luc" target="noi_dung" src="http://localhost/html/page_1.htm"
scrolling="no">
```

Để tạo các liên kết hyperlink giữa trang page\_1.htm với các trang web khác như page\_2.htm, page\_3.htm,... để thể hiện trên khung bên phải thì trong nội dung trang page\_1.htm có nội dung như sau:

```
<html>
<head>
<title>Trang mục lục</title>
<base target="noi_dung">
</head>
<body>
<p>Mục lục công việc</p>
<p><a href="page_2.htm">page 2</a></p>
```

```
<p><a href="page_3.htm">page 3</a></p>  
</body>  
</html>
```

Khi đó trong trình duyệt bạn click vào các hyperlink thì các trang web page\_2.htm và page\_3.htm sẽ mở ra tương ứng ở frame bên phải(frame noi\_dung) nhờ khai báo `target="noi_dung"`.

## CHƯƠNG 5: ĐA PHƯƠNG TIỆN

### Mục tiêu:

- Đưa (Nhúng âm thanh, hình ảnh ...) vào trong một trang Web

### **I. Khái niệm về đa phương tiện (Multimedia)**

- **Multimedia**: đa phương tiện, đa môi trường, đa truyền thông.

Một phương pháp giới thiệu thông tin bằng máy tính, sử dụng nhiều phương tiện truyền thông tin như văn bản, đồ họa và âm thanh, cùng với sự gây ấn tượng bằng tương tác.

### **II. Giới thiệu một số kỹ thuật tạo file âm thanh, hình ảnh**

- Hiện nay trên thị trường xuất hiện rất nhiều phần mềm tạo âm thanh và hình ảnh với các mức chất lượng khác nhau, việc lựa chọn phần mềm sử dụng cho bản thân và doanh nghiệp, công ty phụ thuộc vào túi tiền, nhu cầu và nhân lực. Đối với môi trường học tập, các bạn có thể sử dụng một số phần mềm như Photoshop, Corel Draw, ... Hay để tạo nhanh các hình ảnh thiết kế Web mà không cần nhiều kỹ năng, bạn có thể sử dụng phần mềm tạo banner, nút lệnh... như WebStyle, Cool3D,... phần mềm tạo Video chất lượng cao như Ulead Video, phần mềm thu nhạc với định dạng MP3 như Super MP3 Recorder.

- Trong chương này, chúng tôi xin giới thiệu tới các bạn một số kỹ thuật tạo hình ảnh động bằng phần mềm **Photoshop**.

### **III. Nhúng hình ảnh, âm thanh vào trang Web**

- **Thẻ <IMG>**: Thêm hình ảnh hoặc phim vào trang web. Định nghĩa

<IMG

ALIGN=align-type

ALT=text

SRC=url

BORDER=n

HEIGHT=n

WIDTH=n

HSPACE=n

VSPACE=n>

Trong đó:

- ALIGN=align-type: lề cho ảnh hay cho văn bản bao quanh ảnh. Giá trị có thể là TOP, MIDDLE, BOTTOM, LEFT, RIGHT.

- ALT=text: văn bản hiển thị hay thay thế cho hình ảnh khi chức "Show Picture" trong trình duyệt bị tắt.

- SRC=url: chỉ định ảnh

- BORDER=n: đường viền ảnh. Nếu ảnh được dùng cho hyperlink thì đường viền có màu trùng với màu hyperlink. Nếu ảnh không dùng cho hyperlink thì đường viền không hiển thị.

- HEIGHT=n, WIDTH=n: chỉ độ cao và độ rộng của ảnh. Đơn vị bằng pixel.

- HSPACE=n, VSPACE=n: chỉ định khoảng cách từ ảnh đến văn bản quanh nó theo chiều ngang và dọc.

Ví dụ:

```
<A HREF="index.htm"><IMG SRC="images/btoc.gif" ALT="[các nội dung]"  
ALIGN=MIDDLE HSACE=5 VSPACE=5>Nội dung</A></P>
```

- Thẻ **<EMBED>**: thêm đối tượng âm thanh vào trang web.

Cú pháp:

```
<EMBED  
ALIGN=align-type  
SRC=url  
HEIGHT=n  
WIDTH=n
```

...

```
>
```

- Các tham số thuộc tính tương tự thẻ <IMG>

Ví dụ:

```
<EMBED SRC="nhac/muatrenphohue.mp3" ALIGN=MIDDLE HSACE=5  
VSPACE=5>Mưa trên phố Huế
```



## CHƯƠNG 6: VB SCRIPT VÀ CÁC ỨNG DỤNG

### Mục tiêu:

- Lập trình trên trang WEB để tính toán phục vụ một mục tiêu cụ thể bằng ngôn ngữ VB Script .

### **I. Khái niệm về ngôn ngữ VB Script**

#### **I.1 Giới thiệu về ASP**

ASP (Active Server Pages) là một môi trường lập trình cung cấp cho việc kết hợp HTML, ngôn ngữ kịch bản (Scripting) như VBScript, JavaScript, và các thành phần được viết trong các ngôn ngữ nhằm tạo ra một ứng dụng Internet mạnh mẽ và hoàn chỉnh.

#### **- ASP file là gì?**

File được tạo với phần mở rộng .ASP. Trong file này chứa các thẻ HTML, Các kịch bản Scripting như VBScript, JavaScript hay các lời gọi đến các components(Như DLL và ActiveX control).

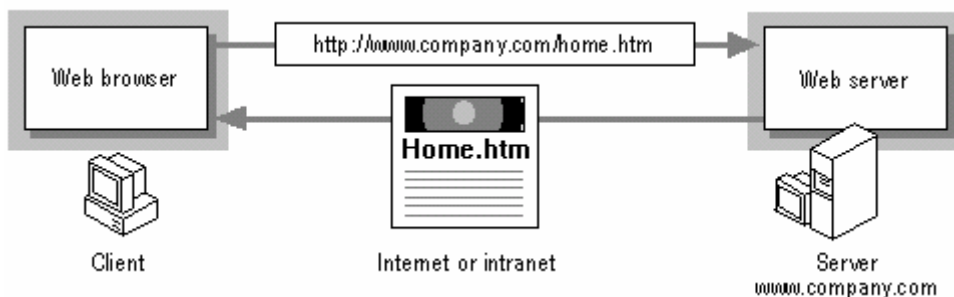
Các script của ASP được nằm trong cặp thẻ `<% %>`

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

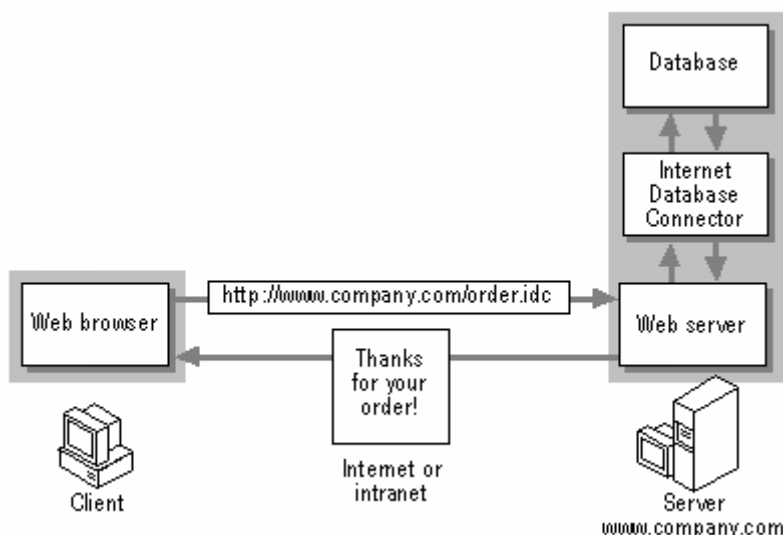
Công nghệ ASP được xây dựng trực tiếp bên trong ; IIS(WinNT, 2000), Personal Webserver.

#### **- ASP làm việc như thế nào ?**

##### **Trang HTML tĩnh:**



##### **Trang web động (Dynamic):**



## I.2 Lập trình với ASP

### 1. Khai báo biến.

- Không bắt buộc nhưng nên khai báo để kiểm soát và bắt lỗi.
- Cú pháp: Dim biến 1, biến 2...
- Để khai báo mảng:
  - a) Dim a(10) : chỉ số chạy từ 0->10 do đó có 11 phần tử.
  - b) Khi khai báo nên sử dụng các tiền tố:

Boolean	Bln
Byte	Byt
Double	Dbl
Integer	Int
Long	Lng
Object	Obj
String	Str
ADO command	Cmd
ADO connection	Cnn

### 2. Khai báo hằng số.

CONST tên hằng= giá trị

### 3. Các đối tượng xây dựng sẵn trong ASP.

Trong ASP có 5 đối tượng phục vụ cho việc sử dụng toàn cục.

- + **Request:** Lấy thông tin từ User
- + **Response:** Gửi thông tin từ Server tới User.
- + **Server:** Dùng để điều khiển IIS
- + **Session:** Dùng để lưu trữ các thông tin như các cài đặt, thay đổi cho một phiên làm việc hiện thời của User.
- + **Application:** Dùng để chia sẻ các thông tin cấp ứng dụng và điều khiển các thiết lập cho toàn bộ quá trình chạy ứng dụng

### 4. Cấu trúc điều khiển lệnh và Hàm.

#### 4.1 Các cấu trúc điều khiển.

##### 1) Cấu trúc điều kiện IF..THEN..ELSE..IF

Cú pháp:

<%

If <Điều kiện 1> then

<Các câu lệnh>

Else

If <Điều kiện 2> then

<Các câu lệnh>

End if

End if

%>

Ngoài ra có thể sử dụng cấu trúc IF.. THEN. . . ELSEIF. . . END IF

<%

If <Điều kiện 1> then

<Các câu lệnh 1>

ElseIf <Điều kiện 2> then

<Các câu lệnh 2>

Else

<Các câu lệnh 3>

End if

%>

## 2) Cấu trúc lựa chọn select case...end select

### Cú pháp:

<%

Select Case <tenbien>

Case <gia tri 1>

<Nhóm lệnh 1>

Case <gia tri 2>

<Nhóm lệnh 2>

Case <gia tri n>

<Nhóm lệnh n>

Case Else

<Nhóm lệnh n+1>

%>

**Ví dụ: Hiện ngày giờ trên máy chủ và cho biết hôm nay là thứ mấy?**

## 3) Cấu trúc lặp tuần tự for...next.

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

FOR.

### Cú pháp:

<%

**FOR** <tenbien>=<giatridau> **TO** <Giatribien> **STEP** <Buoc nhay>

**NEXT**

%>

## 4) Các Cấu trúc lặp .

### a) DO WHILE....LOOP.

**Cú pháp:**

```
<%  
Do while <Dieukien>  
<Các cau lenh>  
Loop  
%>
```

**b) WHILE....wend.****Cú pháp:**

```
<%  
While <Dieukien>  
<Các cau lenh>  
Wend  
%>
```

**c) DO....Loop until****Cú pháp:**

```
<%  
DO  
<Các cau lenh>  
Loop until <Điều kiện>  
%>
```

**4.2 Một số hàm hữu dụng trong ASP.****1) Các Hàm xử lý văn bản**

**i. TRIM(xâu as string):** Bỏ khoảng trắng hai đầu kí tự

**ii. LEFT(Xâu as string, n as interger):** Lấy bên trái xâu n kí tự.

**iii. RIGHT(Xâu as string, n as interger):** Lấy bên phải xâu n kí tự.

**iv. LCASE(Xâu as string) :** Chuyển xâu về chữ thường

**v. UCASE(Xâu as string) :** Chuyển xâu về chữ hoa

**vi. MID(xâu as string, n1, n2):** Lấy n2 kí tự trong xâu bắt đầu từ vị trí n1.

**vii. CSTR(Biến):** Hàm chuyển đổi biến thành kiểu string

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

**Ví dụ:**

```
<%  
x="Hà nội;Hải phòng; TPHCM"  
y=split(x,";")  
Response.write y(0)  
'y(0)="Hà nội"  
%>
```

**2) Các hàm xử lý số****i. SQR(n):** Căn bậc hai của n**ii. INT(n) :** Lấy phần nguyên n**iii. MOD :** Phép đồng dư**iv. Toán tử \ :** Chia nguyên**v. Round(số, n) :** Làm tròn số với n chữ số thập phân**vi. RND():** Trả về số ngẫu nhiên bất kì trong khoảng [0,1]**II. Đưa các đoạn Script vào trong trang Web**

➤ Trong ASP ,mặc định ngôn ngữ script là Vbscript

➤ Nếu muốn dùng Javascript ta phải chỉ định tags script language là Javascript.

- Đối với ngôn ngữ mặc định, bạn có thể chỉ dùng cặp dấu <% và %> để chứa đoạn lệnh Script.

- Ví dụ về một đoạn lệnh VBScript:

```
<% If (Time >=#12:00:00 AM#) then%>
Good Morning!
<% Else %>
Good Afternoon!
<%End If %>
```

- Hoặc bạn có thể chỉ ra cụ thể đang sử dụng ngôn ngữ nào bằng cách sử dụng thẻ:

```
<Script language= "tên_ngôn_ngữ">
```

[lệnh viết ở đây]

```
</Script>
```

- Ở đây tên ngôn ngữ có thể là VBScript hoặc JavaScript (Jscript).

Ví dụ:

```
<Script language= "VBScript">
Response.Write("Xin chào các bạn!")
</Script>
```

**III. Thiết lập các thư viện procedure, function cho 1 ứng dụng WEB**

Thủ tục và hàm trong ASP có thể chèn tại vị trí bất kỳ của trang Web, tuy nhiên đối với một số trường hợp, một số hàm và thủ tục có thể được dùng đi dùng lại nhiều lần trong nhiều trang web thì việc định nghĩa lại các thủ tục và hàm này sẽ mất nhiều thời gian và công sức. Vì vậy bạn nên xây dựng một thư viện hàm và thủ tục cho riêng mình để khi cần thì sẽ dùng đến. Chẳng hạn, khi bạn muốn xây dựng một đề án web có 10 trang, mỗi trang đều phải có thủ tục để kiểm tra mật khẩu đăng nhập, như vậy bạn nên định nghĩa thủ tục này trong một thư viện sẽ thuận lợi rất nhiều cho việc phát triển và chỉnh sửa.

### 1. Định nghĩa thủ tục

Cú pháp:

**Sub** <tên thủ tục>

[Nội dung thủ tục]

**End Sub**

Ví dụ:

Sub HienThi(hoten)

    Response.Write(hoten)

End Sub

### 2. Định nghĩa hàm

Cú pháp:

**Funtion** <tên thủ tục>

[xử lý]

<kết quả trả về>

**End Funtion**

Ví dụ:

Funtion TongDaySo(n)

    Dim i, S

    S = 0

    For i=1 to n

        S = S + i

    Next

**TongDaySo = S**

End Funtion

### 3. Sử dụng thư viện hàm, thủ tục

Sau khi định nghĩa thư viện hàm và thủ tục của riêng bạn (lưu ý là tập tin thư viện phải có dạng Text có phần mở rộng bất kỳ), để sử dụng được thư viện hàm và thủ tục này, bạn dùng thẻ:

**<!--#include file=tên\_tập\_tin\_thư\_viện -->**

để chèn thư viện tại vị trí nào trong trang web mà bạn muốn.

#### IV. Một số bài tập ứng dụng mẫu

**Ví dụ 1:** Giải hệ phương trình bậc nhất  $Ax + B = 0$  (trong đó các hệ số A, B được lấy ngẫu nhiên). Xuất kết quả ra trình duyệt.

```
%@ Language=VBScript %>
<HTML>

<BODY>
<%
Dim A, B
Randomize
A = Round(RND() * 10) 'Tạo số nguyên ngẫu nhiên trong khoản [0,10]
B = Round(RND() * 10) 'Tạo số nguyên ngẫu nhiên trong khoản [0,10]
IF (A=0) Then
  IF (B=0) Then
    Response.Write("Phương trình có vô số nghiệm!")
  ELSE
    Response.Write("Phương trình vô nghiệm!")
  End IF
ELSE
  Response.Write("Phương trình có 1 nghiệm là:" & B/A)
End IF
</BODY>
</HTML>
```

**Ví dụ 2:** Thư viện hàm và thủ tục

- Trước tiên, bạn tạo một tập tin tên là **thuvien.inc** có nội dung như sau:

```
Sub HienThi(hoten)
  Response.Write(hoten)
End Sub

Function TongDaySo(n)
```

```
Dim i, S
S = 0
For i=1 to n
    S = S + i
Next
TongDaySo = S
End Funtion
```

- Bây giờ trong tập tin thứ 2 tên là **ungdung.asp**, ta đưa vào dòng lệnh sau:

```
%@ Language=VBScript %>
<HTML>

<BODY>

<STRONG>DAY LA TRANG WEB SU DUNG THU VIEN HAM - THU TUC
TIEN</STRONG><br>
<!--#include file="thuvien.inc" -->
<%
Dim ten, n
ten = "Nguyen Van A"
Call HienThi(ten & "<br>")
n = 10
Response.Write("Tổng dãy số:" & TongDaySo(n))
</BODY>
</HTML>
```

## BÀI TẬP

Tạo một Web Project (đặt tên là DieuKhen)

1) Thêm vào project một trang **Bai1.asp** có một đoạn lệnh để thực hiện công việc gán một giá trị ngẫu nhiên cho biến nguyên n. Nếu n là chẵn thì xuất ra trình duyệt giá trị  $2*n$ , nếu n lẻ thì xuất ra  $n*n$ .

2) Thêm vào project trang **Bai2.asp** có một đoạn lệnh để thực hiện gán một dãy số ngẫu nhiên cho một mảng A gồm 10 phần tử, sau đó xuất giá trị của các phần tử của mảng ra trình duyệt.



## CHƯƠNG 7: LIÊN KẾT TRANG WEB ĐẾN HỆ CSDL XÂY DỰNG CÁC TRANG WEB ĐỘNG

### Mục tiêu:

- Nhúng được các form có trong trang Web đến một hệ CSDL
- Chuyển thông tin từ giao diện nhập đến hệ CSDL đã được thiết kế
- Chuyển tải thông tin từ CSDL đến trang Web

### I. Nhắc lại một số khái niệm về CSDL

#### 1. Database Device

##### 1.1 Tạo 1 Database Device

Databases và các logs giao dịch(transaction logs) được lưu trong các files gọi là database devices. Trước khi có thể tạo 1 database, trước tiên bạn phải tạo 1 device để lưu database. 1 device có thể lưu nhiều database, và 1 database có thể lưu trong vài device.

##### - Để thêm 1 Database Device:

1. Trong window Server Manager, chọn server để thêm 1 device vào.
2. Từ menu Manage, chọn Database Devices.  
window Manage Database Devices xuất hiện.
3. Click nút New Device.  
Hộp thoại New Database Device xuất hiện.
4. Trong hộp Name, gõ 1 name cho device.
5. Để chỉ định device là device mặc nhiên, chọn hộp kiểm tra Default Device.
6. Trong hộp Location, chọn 1 ổ đĩa đặt device vào.
7. Trong hộp Size (MB), gõ vào size tính bằng đơn vị megabytes cho device.
8. Click Create Now.

##### 1.2 Điều chỉnh 1 Database Device

1. Trong window Server Manager, double-click trên tên device muốn điều chỉnh.  
Hộp thoại Edit Database Device xuất hiện.
2. Để mở rộng cỡ database device, gõ vào cỡ lớn hơn trong hộp Size(MB) . Lưu ý rằng cỡ device có tăng, nhưng không giảm.
3. Để chỉ định device là default device, chọn check box Default Device.
4. Để điều chỉnh device mirroring, chọn Mirroring.  
Hộp thoại Mirror Database Device xuất hiện. Điều chỉnh sao cho thích hợp.
5. Để điều chỉnh device tức thì, trong hộp thoại Edit Database Device , click Change Now.  
Để lập lịch điều chỉnh device, click Schedule.

### 1.3 Bỏ 1 Database Device

Bỏ 1 database device giải phóng khoảng trống lưu trữ trên server. Khi bạn xoá 1 device, tất cả databases trên nó đều bị xoá thêm. Sau khi xoá, các file liên quan bị xoá hẳn trên đĩa và bạn không cần shutting down SQL Server. Bạn có thể tạo lại ngay device trùng tên device vừa xoá.

#### - Để bỏ (drop) 1 database device:

1. Trong window Server Manager, chọn server để gỡ bỏ device trên đó.
2. Từ menu Manage menu, chọn Devices.  
Window Manage Database Devices xuất hiện.
3. Từ đồ thị, chọn database device để bỏ.
4. Click nút Delete Device và xác nhận xoá.

## 2. Database

### 2.1 Tạo 1 Database

Khi bạn tạo 1 database, bạn cấp phát khoảng dung lượng lưu trữ cho database trên device. Bạn phải tạo 1 device vào nơi để lưu trữ database trước khi bạn tạo database.

Khi bạn tạo 1 database, 1 file log giao dịch cũng được tạo ra để ghi lại tất cả những hoạt động giao dịch trên database. Lưu file log giao dịch trên một device khác với device của database để dễ dàng phục hồi và nâng cao hiệu quả thực hiện. Nếu bạn không chỉ định một device khác, thì file log giao dịch được tạo như một phần của database.

#### - Để tạo 1 database:

1. Trong cửa sổ Server Manager, chọn server để tạo database.
2. Từ thực đơn Manage, chọn Databases.  
Cửa sổ Manage Databases window xuất hiện.
3. Click nút New Database.  
Hộp thoại New Database xuất hiện.
4. Trong hộp Name, gõ vào tên cho database.
5. Trong hộp Data Device, chọn device lưu trữ database vào đó.

Bạn có thể tạo 1 device cho database đã chọn bằng cách chọn <new> và hoàn tất thông tin trong hộp thoại New Database Device.

6. Trong hộp Size (MB), gõ vào dung lượng megabytes (MB), để cấp phát cho device của database.

7. Trong hộp Log Device, chọn device chứa the file log giao dịch.

Bạn có thể tạo 1 device cho file log giao dịch bằng cách chọn <new> và hoàn tất thông tin trong hộp thoại New Database Device.

8. Trong hộp Size (MB), gõ vào dung lượng megabytes (MB), để cấp phát cho device của log giao dịch.

Đồ thị Available Space on Database Devices chỉ tất cả device đang có dung lượng cấp phát cho từng device.

9. Nếu bạn dự định nạp 1 bản backup vào database đã chọn trước khi dùng nó, hãy chọn hộp kiểm tra Create for Load, nó sẽ tạo database nhanh hơn. Với tùy chọn này, các trang không khởi tạo rỗng (zero-initialized) khi database được tạo ra. Tuy nhiên, users ngoài DBO không thể truy cập database cho đến khi backup được nạp vào database và tùy chọn database DBO Use Only bị xoá. (Trong hội thoại Edit Database trong tab Options, xoá hội kiểm tra DBO Use Only).

10. Để tạo database tức thì, click Create Now. Để lập lịch trình tạo database, click Schedule.

## 2.2 Điều chỉnh Database

1. Trong window Server Manager, double-click database muốn điều chỉnh.

Hộp thoại Edit Database xuất hiện.

2. Trong tab Database, click nút Expand.

Hộp thoại Expand Database xuất hiện.

3. Để làm thay đổi tức thì, hoàn tất hộp thoại và click Expand Now.

Để thiết lập các tùy chọn database

Trong hộp thoại Edit Database, trong tab Options, chỉ định các tùy chọn và click OK.

## 2.3 Bỏ 1 Database

Chỉ có DBO SA có quyền bỏ(drop) databases. Quyền này không thể chuyển cho các User khác.

### - Để gỡ bỏ 1 database:

1. Từ menu Manage, chọn Databases.

Window Manage Databases xuất hiện.

2. Từ đồ thị, chọn database để bỏ.

3. Click Delete Database.

## 3. SQL Server Login

### 3.1 Tạo 1 SQL Server Login

Trước khi 1 user có thể truy cập 1 SQL Server, người quản trị hệ thống (System Administrator(SA)) phải thêm nhận dạng user(user's login ID-tên login) vào server. The SA cũng có ấn định cho user 1 password, 1 username, 1 database mặc nhiên, và 1 ngôn ngữ mặc nhiên. Database chủ là database mặc nhiên. Nếu user không được ấn định 1 username trong database mặc nhiên, thì user's login ID được dùng là username. Chỉ có SA có thể thêm các login của SQL Server.

Khi thêm 1 SQL Server login, bạn cũng có thể chọn databases cho phép user truy cập vào.

### - Để thêm SQL Server login:

1. Trong cửa sổ Server Manager, chọn server để thêm logins vào.

2. Từ thực đơn Manage, chọn Logins.

Hộp thoại Manage Logins xuất hiện.

3. Hoàn tất hộp thoại click Add.

### 3.2 Điều chỉnh 1 Login

1. Trong cửa sổ Server Manager, chọn server để điều chỉnh login.
  2. Từ thực đơn Manage, chọn Logins.
- Hộp thoại Manage Logins xuất hiện.
3. Hoàn tất hộp thoại và click Modify.

### 3.3 Bỏ 1 Login

1. Trong cửa sổ Server Manager, chọn server để bỏ login ID.
  2. Từ thực đơn the Manage, chọn Logins.
- Hộp thoại Manage Logins xuất hiện.
3. Hoàn tất hộp thoại và click Drop.

## 4. KHAI BÁO ODBC

ODBC(Open Database Connectivity), là 1 Driver quản lý và 1 tập hợp driver ODBC giúp cho các ứng dụng truy cập dữ liệu dùng SQL như 1 ngôn ngữ chuẩn.

DNS(Data Source Name), chứa dữ liệu và thông tin dữ liệu cần thiết để user lấy dữ liệu như SQL server Database.

### 4.1 Khai báo System DNS theo các bước:

- 1) Mở Control panel, nhấp đúp biểu tượng 'ODBC DATA SOURCES'
- 2) Chọn tab "System DNS", Nhấp nút 'ADD'
- 3) Chọn 'Sql server', nhấp nút 'FINISH'
- 4) khai báo name, description và server, nhấp nút 'NEXT>'
- 5) Chọn "With SQL server authentication..", đánh dấu hộp kiểm tra "Connect to sql server..", khai báo username và password của user Sql server. nhấp nút 'NEXT>'
- 6) Nhấp nút 'NEXT>'
- 7) Nhấp nút 'FINISH':
- 8) Nhấp nút 'TEST DATA SOURCE'. kiểm tra thành CÔNG chưa.
- 9) Nhấp nút các nút 'OK' kết thúc cài đặt ODBC.

### 4.2 Khai báo User DNS theo các bước:

- 1) Mở Control panel, nhấp đúp biểu tượng 'ODBC DATA SOURCES'
- 2) Chọn tab "User DNS", Nhấp nút 'ADD'
- 3) Chọn 'Sql server', nhấp nút 'FINISH'
- 4) khai báo name, description và server, nhấp nút 'NEXT>'
- 5) Chọn "With SQL server authentication..", đánh dấu hộp kiểm tra "Connect to sql server..", khai báo username và password của user Sql server. nhấp nút 'NEXT>'
- 6) Nhấp nút 'NEXT>'
- 7) Nhấp nút 'FINISH':

8) Nhấp nút 'TEST DATA SOURCE'. kiểm tra thành CÔNG chưa.

9) Nhấp nút các nút 'OK' kết thúc cài đặt ODBC.

## 5. CÁC LỆNH SQL

**5.1 CREATE TABLE:** tạo table mới

```
CREATE TABLE TableName (FieldName1 FieldType [(nFieldWidth [, nPrecision])]
[NULL | NOT NULL] [, FieldName2 FieldType [(nFieldWidth [, nPrecision])] [NULL | NOT
NULL]...)
```

**TableName:** chuỗi khoảng 30 ký tự.

**FieldName1 , FieldName2:** tên cột khoảng 30 ký tự.

**FieldType:** Kiểu cột.

Tên kiểu( FieldType)	Diễn giải
numeric	số gồm phần nguyên và phần thập phân
char	kiểu ký tự có độ dài tối đa 255 ký tự
datetime	kiểu ngày giờ
int	kiểu số nguyên
real	kiểu số thực
text	kiểu memo, text
image	kiểu hình ảnh
float	kiểu float

**NULL:** Là 1 mục vào không có giá trị rõ ràng. NULL không phải là 0 hoặc trống.

Ví dụ: **CREATE TABLE Vidu (**

**so numeric(10, 2) NULL ,**

**chuoai char (255) NULL ,**

**ngay datetime NULL ,**

**nguyen int NULL ,**

**thuc real NULL ,**

**vanban text NULL ,**

**anh image NULL ,**

**so float float NULL**

**)**

**5.2 SELECT:** truy vấn dữ liệu các table

```
SELECT [Alias.] Select_Item [AS Column_Name] [, [Alias.] Select_Item [AS
Column_Name] ...]
```

```
FROM [DatabaseName!]Table [INNER JOIN DatabaseName!]Table [ON JoinCondition
...]
```

[WHERE JoinCondition [AND JoinCondition ...] [AND | OR FilterCondition [AND | OR FilterCondition ...]]]  
 [GROUP BY GroupColumn [, GroupColumn ...]]  
 [HAVING FilterCondition]  
 [UNION [ALL] SELECTCommand]  
 [ORDER BY Order\_Item [ASC | DESC] [, Order\_Item [ASC | DESC] ...]]

Tham số:

**SELECT** chỉ định trường, hằng và biểu thức xuất hiện trong kết quả truy vấn.

**Alias**. Bí danh table.

**Select\_Item** chỉ định 1 mục được bao gồm trong kết quả truy vấn. Mục có thể là:

- Tên trường của table trong mệnh đề FROM.
- 1 hằng chỉ định cùng 1 giá trị cho tất cả các hàng trong kết quả truy vấn.
- 1 biểu thức có thể là hàm tự tạo.
- **AVG(Select\_Item)**, tính trung bình Select\_Item.
- **COUNT(Select\_Item)**, đếm số lần xuất hiện. **COUNT(\*)** đếm số hàng trong kết quả truy vấn.

- **MIN(Select\_Item)**, tính giá trị nhỏ nhất.

- **MAX(Select\_Item)**, tính giá trị lớn nhất.

- **SUM(Select\_Item)**, tính tổng các số.

**AS Column\_Name** Chỉ định tiêu đề của 1 cột trong table kết quả truy vấn.

**FROM** Liệt kê các table chứa dữ liệu cần truy vấn.

**DatabaseName!** chỉ định tên của CSDL chứa table.

**INNER JOIN** chỉ định kết quả truy vấn chỉ chứa các hàng từ 1 table trùng với 1 hoặc nhiều trong table kia.

**ON JoinCondition** chỉ định các điều kiện kể các table nối lại

**WHERE** Chỉ các hàng có thỏa mãn điều kiện đi liền sau WHERE mới được thực hiện truy vấn.

Bạn dùng toán tử **AND/OR** để kết nối cùng lúc nhiều điều kiện khác nhau.

Các toán tử so sánh:

Ký hiệu	Tương đương
=	Bằng
==	Chính xác bằng
LIKE	SQL LIKE
<>, !=, #	Không bằng
>	Lớn hơn
>=	Lớn hơn hoặc bằng
<	Nhỏ hơn

<=	Nhỏ hơn hoặc bằng
----	-------------------

**GROUP BY GroupColumn [, GroupColumn ...]** Các nhóm hàng trong câu truy vấn dựa trên giá trị trên 1 hoặc nhiều cột. GroupColumn dự trên thứ tự liệt kê ra.

**HAVING FilterCondition** Chỉ định điều kiện lọc nhóm trong kết quả truy vấn.

**[UNION [ALL] SELECTCommand]** Kết hợp kết quả cuối cùng của 1 SELECT với 1 SELECT khác.

**ALL** ngăn UNION ghép nối 2 lần các hàng từ kết quả UNION.

Ví dụ:

```
SELECT customer.company, orders.order_id, orders.emp_id ;
FROM customer, orders ;
WHERE customer.cust_id = orders.cust_id ;
UNION ;
SELECT customer.company, " ", " " ;
FROM customer ;
WHERE customer.cust_id NOT IN ;
(SELECT orders.cust_id FROM orders)
```

**ORDER BY Order\_Item** Sắp xếp kết quả truy vấn theo thứ tự Order\_Item.

**ASC** sắp xếp tăng dần

**DESC** sắp xếp giảm dần.

**5.3 INSERT:** thêm dữ liệu(hàng) vào table

**INSERT INTO dbf\_name [(fname1 [, fname2, ...])] VALUES (eExpression1 [, eExpression2, ...])**

**dbf\_name:** tên table mà dữ liệu sẽ thêm vào, lệnh này tự thêm 1 hàng mới và gán giá trị vào.

**fname1, fname2:** tên các cột trong table

**eExpression1 , eExpression2:** biểu thức sau khi tính thành giá trị sẽ đưa vào từng cột tương ứng đã chỉ định.

Ví dụ:

**R\_Don\_Vi="Cty ĐB-ĐT"**

**R\_Ho\_Ten="Lê Thị Thu Hoài"**

**R\_Username="thuhoai"**

**R\_Password="thuhoai"**

```
INSERT INTO DHTT_Login (Don_Vi, Ho_Ten, Username, Password) VALUES
(R_Don_Vi, R_Ho_Ten, R_Username, R_Password)
```

**5.4 UPDATE:** Gán giá trị mới cho 1 hoặc nhiều hàng của table.

```
UPDATE [DatabaseName!]TableName SET Column_Name1 = eExpression1 [,
Column_Name2 = eExpression2 ...] WHERE FilterCondition1 [AND | OR FilterCondition2
...]]
```

[DatabaseName!]TableName: tên CSDL và Table thuộc CSDL.

SET Column\_Name1= eExpression1,...: gán giá trị eExpression1 cho Column\_Name1,...

WHERE FilterCondition1 [AND | OR FilterCondition2 ...]: chỉ định các hàng thoả mãn điều kiện FilterCondition1 và/hoặc FilterCondition2 mới được gán giá trị mới.

Ví dụ:

```
UPDATE DHTT.DHTT_Login SET Username =R_Username_Moi, Password =
R_Password_Moi WHERE Username=R_Username_Cu AND Password =
R_Password_Cu
```

### 5.5 DELETE:

```
DELETE FROM [DatabaseName!]TableName [WHERE FilterCondition1 [AND | OR
FilterCondition2 ...]]
```

[DatabaseName!]TableName: tên CSDL và Table thuộc CSDL.

WHERE FilterCondition1 [AND | OR FilterCondition2 ...]: chỉ định các hàng thoả mãn điều kiện FilterCondition1 và/hoặc FilterCondition2 mới bị xoá.

Ví dụ:

```
DELETE FROM DHTT.DHTT_Login WHERE Don_Vi=R_Don_Vi_Xoa AND
Ho_Ten=R_Ho_Ten_Xoa
```

## II. Các lệnh liên kết CSDL

### 1. Kết nối cơ sở dữ liệu DSN(Data Source Name) qua đối tượng Recordset

#### + Hiển thị thông tin ra trình duyệt

```
Dim rs
```

```
Set rs=Server.CreateObject("ADODB.Recordset")
```

```
Rs.Open "SINHVIEN", "DSN=QLSV"
```

```
Rs.MoveFirst
```

```
Do While Not rs.EOF
```

```
Response.Write(rs("HoTen"))
```

```
Response.Write(.....)
```

```
Rs.MoveNext
```

```
Loop
```

```
Rs.Close
```



```
Set rs = Nothing
```

**+ Tìm kiếm để cập nhật các bản ghi trong cơ sở dữ liệu**

```
CONST adOpenDynamic = 2
```

```
CONST adLockPessimistic = 2
```

```
Dim rs
```

```
Set rs=Server.CreateObject("ADODB.Recordset")
```

```
Rs.Open "SINHVIEN", "DSN=QLTV", adOpenDynamic, adLockPessimistic
```

```
Rs.Find <điều kiện>
```

```
If Not rs.EOF then
```

```
    <cập nhật thông tin>
```

```
Else
```

```
    <thông báo không tìm thấy>
```

```
End If
```

```
Rs.Close
```

```
Set rs = Nothing
```

**2. Kết nối trực tiếp đến cơ sở dữ liệu bằng chuỗi kết nối qua đối tượng**

**Connection:**

**\* Với Microsoft Access:**

```
Dim conn
```

```
Set rs=Server.CreateObject("ADODB.Connection")
```

```
conn.Open "Provider=Microsoft.Jet.OleDB.4.0;Data source=c:\thuchanh\qltv.mdb;"
```

**\* Với Microsoft SQL Server:**

```
Dim conn
```

```
Set rs=Server.CreateObject("ADODB.Connection")
```

```
conn.Open "Provider=SQLOLEDB;Data source=servername_or_address; Initial  
Catalog=database_name;User ID=username;password=password"
```

Các công việc còn lại giống như khi kết nối qua đối tượng DSN.

**III. Một số bài tập ứng dụng mẫu**

**Ví dụ 1:** Mở cơ sở dữ liệu Quản lý Sinh viên và bổ sung vào bảng SINHVIEN một bản ghi, sử dụng đối tượng Recordset:

```
<%@ Language=VBScript %>

<HTML>

<BODY>

<%

const adOpenDynamic=2

const adLockPessimistic=2

dim rs

set rs=server.CreateObject("ADODB.Recordset")

rs.Open "SINHVIEN","DSN=QLSV",adOpenDynamic,adLockPessimistic

rs.AddNew

rs("MaSV")="12"

rs("TenSV")="Le Thanh Thuy"

rs.Update

rs.Close

Set rs = Nothing

%>

</BODY>

</HTML>
```

**Ví dụ 2:** Mở cơ sở dữ liệu Quản lý Học viên và bổ sung vào bảng HOCVIEN một bản ghi, sử dụng đối tượng Connection:

```
<%@ Language=VBScript %>

<HTML>

<BODY>

<%dim conn

set conn=server.createobject("adodb.connection")

conn.Open "Provider=SQLOLEDB;Data source=localhost; Initial catalog=QLHV;

User ID=sa;password="
```

```
dim strSQL
strSQL="INSERT into HOCVIEN(maHV, HotenHV, MaLop)"
strSQL = strSQL & " Values( '006','Nguyen Van Trinh','T01')"
```

conn.execute(strSQL)

response.write("Ban da them mot hoc vien !")

conn.close

set conn=nothing

%>

</BODY>

</HTML>

**Ví dụ 3:** Hiển thị tất cả học viên có tên là Trang.

```
<%@ Language=VBScript %>
<HTML>
<BODY>
<%dim conn
set conn=server.createobject("adodb.connection")
conn.Open "Provider=SQLOLEDB;Data source=localhost; Initial catalog=QLHV;
          User ID=sa;password="

dim strSQL
strSQL="SELECT * FROM HOCVIEN WHERE HotenHV Like '%Trang' "
```

Dim rs

Rs = conn.execute(strSQL)

response.write("Danh sách học viên tên Trang !<br>")

Rs.MoveFirst

Do While Not rs.EOF

    Response.Write("Mã học viên:" & rs("MaHV") & "<br>")

    Response.Write("Họ tên HV: " & rs("HotenHV") & "<br>")

Rs.MoveNext

```
Loop  
Rs.Close  
Set rs = Nothing  
conn.close  
set conn=nothing  
%>  
</BODY>  
</HTML>
```

**BÀI TẬP:**

- 1) Thiết kế biểu mẫu cho phép nhập thông tin thành viên vào cơ sở dữ liệu gồm Họ và tên, ngày sinh, địa chỉ, tên sử dụng, mật khẩu.
- 2) Thiết kế biểu mẫu cho phép nhập họ và tên học viên để tìm kiếm và hiển thị thông tin tìm thấy ra trình duyệt.

## CHƯƠNG 8: JAVASCRIPT

### Mục tiêu:

- Khai báo và sử dụng được biến, hàm trong JavaScript
- Viết được các sự kiện cho nút lệnh bằng ngôn ngữ JavaScript

### I. PHÁT BIỂU LỆNH CỦA JAVASCRIPT

#### 1. break

Ngưng vòng lặp **switch,while, for, for...in**, hoặc **do...while**.

Ví dụ:

```
function BreakTest(breakpoint)
{
    var i = 0;
    while (i < 100)
    {
        if (i == breakpoint)
            break;
        i++;
    }
    return(i);
}
```

Comment

Làm cho các ghi chú sẽ được bỏ qua trong quá trình biên dịch và chạy.

Cú pháp 1:

Chú thích 1 dòng:

```
// chú thích
```

Cú pháp 2

Chú thích nhiều dòng:

```
/*
```

```
chú thích
```

```
*/
```

**Ví dụ:**

```
function myfunction(arg1, arg2)
{
    /* This is a multiline comment that
       can span as many lines as necessary. */
    var r;
    // This is a single line comment.
```

```

r = arg1 + arg2; // Sum the two arguments.
return(r);
}

```

## 2. continue

Dừng lặp hiện tại và bắt đầu lặp mới. Dùng **continue** trong **while**, **do...while**, **for**, hoặc **for...in**

### Ví dụ:

```

function skip5()
{
var s = "", i=0;
while (i < 10)
{
i++;
// Skip 5
if (i==5)
{
continue;
}
s += i;
}
return(s);
}

```

## 3. do...while

Thực hiện 1 nhóm lệnh lặp đi lặp lại cho đến trong khi điều kiện **true**.

Cú pháp

**do**

*statement*

**while** (*expression*) ;

Phần	Diễn giải
<i>statement</i>	Nhóm lệnh được chạy khi điều kiện <i>expression</i> là đúng ( <b>true</b> ).
<i>expression</i>	Là biểu thức Boolean <b>true</b> hoặc <b>false</b> . Nếu <i>expression</i> = <b>true</b> , thì lặp lại lần nữa. Nếu <i>expression</i> = <b>false</b> , thì vòng lặp dừng.

### Ví dụ:

```

function GetDriveList()
{

```

```

var fso, s, n, e, x;
fso = new ActiveXObject("Scripting.FileSystemObject");
e = new Enumerator(fso.Drives);
s = "";
do
{
    x = e.item();
    s = s + x.DriveLetter;
    s += " - ";
    if (x.DriveType == 3)
        n = x.ShareName;
    else if (x.IsReady)
        n = x.VolumeName;
    else
        n = "[Drive not ready]";
    s += n + "<br>";
    e.moveNext();
}
while (!e.atEnd());
return(s);
}

```

#### 4. for

Thực hiện 1 khối lệnh trong khi điều kiện kiểm tra còn đúng(=true).

Cú pháp

**for** (*initialization; test; increment*)

*statement*

Phần	Diễn giải
<i>initialization</i>	Là 1 biểu thức. Biểu thức này chỉ chạy 1 lần trước khi vòng lặp được thực hiện.
<i>test</i>	Là biểu thức Boolean. Nếu <i>test</i> = <b>true</b> , <i>statement</i> được thực hiện. Nếu <i>test</i> = <b>false</b> , vòng lặp sẽ dừng.
<i>increment</i>	Là biểu thức. Biểu thức increment được thực hiện ở cuối mỗi vòng lặp.
<i>statement</i>	Đoạn lệnh được thực hiện khi <i>test</i> = <b>true</b> .

#### Ví dụ:

```

/* i is set to 0 at start, and is incremented by 1 at the end
of each iteration. Loop terminates when i is not less

```

```
than 10 before a loop iteration. */
var myarray = new Array();
for (i = 0; i < 10; i++)
{
    myarray[i] = i;
}
```

### 5. for..in

Thực hiện 1 hoặc nhiều câu lệnh cho với từng thuộc tính của 1 object, hoặc từng thành phần của 1 mảng.

Cú pháp

```
for (variable in [object | array])
statement
```

Phần	Diễn giải
<i>variable</i>	Là 1 biến có thể là thuộc tính của <i>object</i> hoặc thành phần của mảng.
<i>object, array</i>	Một object hoặc array qua mỗi lần lặp.
<i>statement</i>	Nhóm lệnh sẽ được thực hiện với từng thành phần của đối tượng hoặc mảng.

#### Ví dụ:

```
function ForInDemo()
{
    // Create some variables.
    var a, key, s = "";
    // Initialize object.
    a = {"a" : "Athens" , "b" : "Belgrade", "c" : "Cairo"}
    // Iterate the properties.
    for (key in a)
    {
        s += a[key] + "<BR>";
    }
    return(s);
}
```

### 6. function

Khai báo 1 function mới.

Cú pháp



```

function functionname([argument1 [, argument2 [, ...argumentn]]])
{
    statements
}

```

Phần	Diễn giải
<i>functionname</i>	Tên function
<i>argument1...argumentn</i>	Danh sách đối số của hàm
<i>statements</i>	1 hoặc nhiều lệnh JScript.

**Ví dụ:**

```

function myfunction(arg1, arg2)
{
    var r;
    r = arg1 * arg2;
    return(r);
}

```

**7. if...else**

Thực hiện 1 nhóm lệnh có điều kiện, phục thuộc vào giá trị của 1 biểu thức.

Cú pháp

```
if (condition)
```

```
    statement1
```

```
[else
```

```
    statement2]
```

Phần	Diễn giải
<i>condition</i>	Là biểu thức boolean. Nếu <i>condition</i> là null hoặc undefined , <i>condition</i> được gán là <b>false</b> .
<i>statement1</i>	Nhóm lệnh được thực hiện nếu <i>condition</i> = <b>true</b> .
<i>statement2</i>	Nhóm lệnh được thực hiện nếu <i>condition</i> = <b>false</b> .

**Ví dụ:**

```

if (x == 5)
if (y == 6)
    z = 17;
else
    z = 20;

```

**Ví dụ:**

```
    if (x == 5)
    {
        if (y == 6)
            z = 17;
    }
else
    z = 20;
Ví dụ:
    if (x == 5)
        z = 7;
        q = 42;
else
    z = 19;
```

### 8. return

Thoát khỏi function hiện tại và trả về 1 giá trị từ function đó.

Cú pháp

**return** *[expression]*;

Đối số *expression* là 1 giá trị trả về từ function. Nếu bỏ qua thì hàm không trả về giá trị.

**Ví dụ:**

```
function myfunction(arg1, arg2)
{
    var r;
    r = arg1 * arg2;
    return(r);
}
```

### 9. switch

Thực hiện 1 hoặc nhiều lệnh khi giá trị biểu thức chỉ định trùng với 1 nhãn.

Cú pháp

**switch** (*expression*) {

**case** *label* :

*statementlist*

**case** *label* :

*statementlist*

...

**default** :

*statementlist*

}

Phần	Diễn giải
<i>expression</i>	Biểu thức được tính toán so sánh với các nhãn.
<i>label</i>	Là định danh trùng với <i>expression</i> . Nếu <i>label</i> === <i>expression</i> , thì thực hiện <i>statementlist</i> sau dấu :, và tiếp tục đến khi gặp lệnh <b>break</b> ( lệnh break có hay không là tùy ý) hoặc cuối lệnh <b>switch</b> .
<i>statementlist</i>	1 hoặc nhiều lệnh.

Dùng **default** để chỉ ra rằng nếu *expression* không trùng với bất kỳ *label* nào đã chỉ định thì chạy đoạn lệnh *statementlist* trong default.

**Ví dụ:**

```
function MyObject() {
...}
switch (object.constructor){
case Date:
...
case Number:
...
case String:
...
case MyObject:
...
default:
...
}
```

## 10. this

Tham khảo đến object hiện tại.

Cú pháp

**this**.*property*

**Ví dụ:**

```
function Car(color, make, model)
{
  this.color = color;
  this.make = make;
  this.model = model;
}
```

## 11. var

Khai báo biến.

Cú pháp

**var** *variable* [ = *value* ] [, *variable2* [ = *value2*], ...]

Phần	Diễn giải
<i>variable, variable2</i>	Tên các biến được khai báo.
<i>value, value2</i>	Các giá trị gán cho biến.

**Ví dụ:**

**var** index;

**var** name = "Thomas Jefferson";

**var** answer = 42, counter, numpages = 10

## 12. while

Thực hiện đoạn lệnh lặp lại cho đến trong khi điều kiện chỉ định = **false**.

Cú pháp

**while** (*expression*)

*statement*

Phần	Diễn giải
<i>expression</i>	Biểu thức Boolean kiểm tra trước mỗi khi thực hiện đoạn lệnh. Nếu <i>expression</i> = <b>true</b> , vòng lặp được thực hiện. Nếu <i>expression</i> = <b>false</b> , vòng lặp dừng.
<i>statement</i>	Nhóm lệnh được thực hiện nếu <i>expression</i> = <b>true</b> .

**Ví dụ:**

function BreakTest(breakpoint)

```
{
  var i = 0;
  while (i < 100)
  {
    if (i == breakpoint)
      break;
    i++;
  }
  return(i);
}
```

## II. CÁC PHƯƠNG THỨC CỦA JavaScript

**Bảng toán tử:**

Toán tử	Ký hiệu
Cộng	+
Gán giá trị	=
AND	&
NOT	~
OR	
XOR	^
Chia	/
So sánh bằng	==
S/s lớn hơn	>
S/s lớn hơn hoặc bằng	>=
Giảm 1 đơn vị	--
Tăng 1 đơn vị	++
So sánh không bằng	!=
S/s nhỏ hơn	<
S/s nhỏ hơn hoặc bằng	<=
AND	&&
NOT	!
OR	
Chia lấy phần dư	%
Nhân	*
Tạo object mới	new
Trừ	-

Lưu ý: Giá trị logic trả về:

Đúng	<b>true</b>
Sai	<b>false</b>

### 1. substr

`stringvar.substr(start [, length ])`: trả về 1 chuỗi con từ chuỗi stringvar gồm length ký tự bắt đầu từ vị trí start

Ví dụ

```
function SubstrDemo()
{
    var s, ss;
```

```
var s = "The quick brown fox jumped over the lazy yellow dog.";
ss = s.substr(16, 3);
// Trả về "fox".
return(ss);
}
```

## 2. abs

**Math.abs(number)**: trả về trị tuyệt đối của number

Ví dụ:

```
function ComparePosNegVal(n)
{
    var s;
    var v1 = Math.abs(n);
    var v2 = Math.abs(-n);
    if (v1 == v2)
        s = "The absolute values of " + n + " and "
        s += -n + " are identical.";
    return(s);
}
```

## 3. atEnd

**myEnum.atEnd()**: trả về giá trị **true** nếu thành phần của tập hợp myEnum là cuối cùng hoặc rỗng, ngược lại cho giá trị **false**

Ví dụ:

```
function ShowDriveList()
{
    var fso, s, n, e, x;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    e = new Enumerator(fso.Drives);
    s = "";
    for (; !e.atEnd(); e.moveNext())
    {
        x = e.item();
        s = s + x.DriveLetter;
        s += " - ";
        if (x.DriveType == 3)
            n = x.ShareName;
        else if (x.IsReady)
            n = x.VolumeName;
        else
            n = "[Drive not ready]";
    }
}
```

```
s += n + "<br>";  
}  
return(s);  
}
```

#### 4. charAt

**strVariable.charAt(index):** trả về ký tự vị trí thứ index trong biến chuỗi strVariable. Ký tự đầu tiên của chuỗi có thứ tự là 0, ký tự thứ hai là 1...

Ví dụ:

```
function charAtTest(n)  
{  
    var str = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
    var s;  
    s = str.charAt(n - 1);  
    return(s);  
}
```

#### 5. dimensions

**array.dimensions():** trả về số chiều của mảng trong VBScript

Ví dụ:

```
<HEAD>  
<SCRIPT LANGUAGE="VBScript">  
<!--  
Function CreateVBAArray()  
    Dim i, j, k  
    Dim a(2, 2)  
    k = 1  
    For i = 0 To 2  
        For j = 0 To 2  
            a(j, i) = k  
            k = k + 1  
        Next  
    Next  
    CreateVBAArray = a  
End Function  
-->  
</SCRIPT>  
<SCRIPT LANGUAGE="JScript">  
    <!-- function VBAArrayTest(vba)  
    { var i, s;  
      var a = new VBAArray(vba);
```

```

for (i = 1; i <= a.dimensions(); i++)
{ s = "The upper bound of dimension ";
  s += i + " is ";
  s += a.ubound(i) + "<BR>"; } return(s);
} -->
</SCRIPT>
</HEAD>
<BODY> <SCRIPT language="jscript">
document.write(VBArrayTest(CreateVBArray())); </SCRIPT> </BODY>

```

## 6. eval

**eval(*codestring*)**: thực hiện lệnh *codestring*, *codestring* viết ở dạng chuỗi

Ví dụ:

```

eval("var mydate = new Date();");
//cho kết quả mydate là 1 ngày

```

## 7. getDate

**objDate.getDate()**: trả về ngày trong tháng. Ngày lấy từ ngày hiện tại của hệ thống.

Ví dụ:

```

function DateDemo()
{
var d, s = "Today's date is: ";
d = new Date();
s += (d.getMonth() + 1) + "/";
s += d.getDate() + "/";
s += d.getFullYear();
return(s);
}

```

## 8. getItem

**safeArray.getItem(*dimension1* [, *dimension2*, ...], *dimensionn*)**: trả về giá trị trong mảng được chỉ định bởi thứ tự chiều của phần tử.

Ví dụ:

```

<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Function CreateVBArray()
Dim i, j, k
Dim a(2, 2)
k = 1
For i = 0 To 2
  For j = 0 To 2

```



```

        a(i, j) = k
        document.writeln(k)
        k = k + 1
    Next
    document.writeln("<BR>")
Next
CreateVBAArray = a
End Function
-->
</SCRIPT>
<SCRIPT LANGUAGE="JScript">
<!--
function GetItemTest(vbarray)
{
    var i, j;
    var a = new VBAArray(vbarray);
    for (i = 0; i <= 2; i++)
    {
        for (j = 0; j <= 2; j++)
        {
            document.writeln(a.getItem(i, j));
        }
    }
}
}-->
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JScript">
<!--
    GetItemTest(CreateVBAArray());
-->
</SCRIPT>
</BODY>

```

## 9. getTime

**objDate.getTime():** trả về giờ hệ thống.

Ví dụ:

```

function GetTimeTest()
{
    var d, s, t;
    var MinMilli = 1000 * 60;

```

```
var HrMilli = MinMilli * 60;
var DyMilli = HrMilli * 24;
d = new Date();
t = d.getTime();
s = "It's been "
s += Math.round(t / DyMilli) + " days since 1/1/70";
return(s);
}
```

### 10. **getDate**

**objDate.getDate():** Trả về thứ trong tuần với:

0 = Sunday  
1 = Monday  
2 = Tuesday  
3 = Wednesday  
4 = Thursday  
5 = Friday  
6 = Saturday

Ví dụ:

```
function DateDemo()
{
var d, day, x, s = "Today is: ";
var x = new Array("Sunday", "Monday", "Tuesday");
var x = x.concat("Wednesday", "Thursday", "Friday");
var x = x.concat("Saturday");
d = new Date();
day = d.getDate();
return(s += x[day]);
}
```

### 11. **join**

**arrayobj.join(separator):** trả về 1 chuỗi gồm các giá trị trong mảng arrayobj cách nhau bằng separator.

Ví dụ:

```
function JoinDemo()
{
var a, b;
a = new Array(0,1,2,3,4);
b = a.join("-");
return(b);
}
```

**12. lbound**

**`safeArray.lbound(dimension)`**: trả về giá trị thứ tự nhỏ nhất của mảng `safeArray` với chiều `dimension` chỉ định.

Ví dụ:

```
<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Function CreateVBAArray()
    Dim i, j, k
    Dim a(2, 2)
    k = 1
    For i = 0 To 2
        For j = 0 To 2
            a(j, i) = k
            k = k + 1
        Next
    Next
    CreateVBAArray = a
End Function
-->
</SCRIPT>
<SCRIPT LANGUAGE="JScript">
<!--
function VBAArrayTest(vba)
{
    var i, s;
    var a = new VBAArray(vba);
    for (i = 1; i <= a.dimensions(); i++)
    {
        s = "The lower bound of dimension ";
        s += i + " is ";
        s += a.lbound(i)+ "<BR>";
        return(s);
    }
}
-->
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT language="jscript">
```

```
document.write(VBArrayTest(CreateVBArray()));  
</SCRIPT>  
</BODY>
```

### 13. match

*stringObj*.**match**(*rgExp*): trả về 1 mảng kết quả tìm kiếm *rgExp* trong *stringObj*.

Ví dụ:

```
function MatchDemo()  
{  
  var r, re;  
  var s = "The quick brown fox jumped over the lazy yellow dog.";  
  re = /fox/i;  
  r = s.match(re);  
  return(r);  
}
```

### 14. max

*retVal* = **Math.max**(*number1*, *number2*): trả về số lớn nhất trong 2 số *number1* và *number2*

Ví dụ:

```
function MaxNum(n1,n2)  
{  
  var nmax = Math.max(n1,n2)  
  return(nmax);  
}
```

### 15. min

*retVal* = **Math.min**(*number1*, *number2*): trả về số bé nhất trong 2 số *number1* và *number2*

Ví dụ:

```
function MinNum(n1,n2)  
{  
  var nmin = Math.min(n1,n2)  
  return(nmin);  
}
```

### 16. moveFirst

*myEnum*.**moveFirst**( ): Thiết lập thành phần hiện tại trong tập hợp là phần tử đầu tiên.

Ví dụ:

```
function ShowFirstAvailableDrive()
{
    var fso, s, e, x;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    e = new Enumerator(fso.Drives);
    e.moveFirst();
    s = "";
    do
    {
        x = e.item();
        if (x.IsReady)
        {
            s = x.DriveLetter + ":";
            break;
        }
        else
            if (e.atEnd())
            {
                s = "No drives are available";
                break;
            }
        e.moveNext();
    }
    while (!e.atEnd());
    return(s);
}
```

### 17. moveNext

*myEnum.moveNext()*: Dời thành phần hiện tại trong tập hợp sang phần tử kế tiếp.

Ví dụ:

```
function ShowDriveList()
{
    var fso, s, n, e, x;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    e = new Enumerator(fso.Drives);
    s = "";
    for (; !e.atEnd(); e.moveNext())
    {
        x = e.item();
        s = s + x.DriveLetter;
        s += " - ";
    }
}
```

```
if (x.DriveType == 3)
    n = x.ShareName;
else if (x.IsReady)
    n = x.VolumeName;
else
    n = "[Drive not ready]";
s += n + "<br>";
}
return(s);
}
```

### 18. replace

***stringObj.replace(rgExp, replaceText)***: Trả về 1 chuỗi từ chuỗi *stringObj* sau khi thay thế chuỗi *replaceText* vào vị trí xuất hiện chuỗi *rgExp*.

Ví dụ:

```
function ReplaceDemo()
{
    var r, re;
    var s = "The quick brown fox jumped over the lazy yellow dog.";
    re = /fox/i;
    r = s.replace(re, "pig");
    return(r);
}
```

### 19. reverse

***arrayobj.reverse()***: trả về mảng đảo ngược thứ tự của mảng *arrayobj*

Ví dụ:

```
function ReverseDemo()
{
    var a, l;
    a = new Array(0,1,2,3,4);
    l = a.reverse();
    return(l);
}
```

### 20. sort

***arrayobj.sort(sortfunction)***: trả về 1 mảng từ mảng *arrayobj* sau khi đã sắp xếp giá trị theo hàm *sortfunction* chỉ định. Nếu không chỉ định hàm thì sắp theo thứ tự tăng dần.

Ví dụ:

```
function SortDemo()
{
```

```
var a, l;  
a = new Array("X", "y", "d", "Z", "v", "m", "r");  
l = a.sort();  
return(l);  
}
```

## 21. split

**stringObj.split(str):** Trả về 1 mảng chứa các đoạn ký tự trong chuỗi *stringObj* phân cách bởi chuỗi phân cách *str* chỉ định.

Ví dụ:

```
function SplitDemo()  
{  
  var s, ss;  
  var s = "The quick brown fox jumped over the lazy yellow dog.";  
  // Split at each space character.  
  ss = s.split(" ");  
  return(ss);  
}
```

## 22. substring

**strVariable.substring(start, end):** trả về 1 chuỗi con cắt ra từ chuỗi mẹ *strVariable* bắt đầu từ ký tự có vị trí *start* và kết thúc ở ký tự có vị trí là *end*.

Ví dụ:

```
function SubstringDemo()  
{  
  var s, ss;  
  var s = "The quick brown fox jumped over the lazy yellow dog.";  
  ss = s.substring(16, 19);  
  return(ss);  
}
```

## 23. test

**rgexp.test(str):** trả về giá trị **true** nếu *str* chứa trong *rgexp* và **false** nếu không có.

Ví dụ:

```
function TestDemo(re, s)  
{  
  var s1;  
  // Test string for existence of regular expression.  
  if (re.test(s))  
    s1 = " contains ";  
  else
```

```

    s1 = " does not contain ";
    // Get text of the regular expression itself.
    return(s + s1 + re.source);
}

```

## 24. toArray

**safeArray.toArray( )**: Trả về mảng 1 chiều JavaScript từ mảng *safeArray* của VBScript.

Ví dụ:

```

<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Function CreateVBAArray()
    Dim i, j, k
    Dim a(2, 2)
    k = 1
    For i = 0 To 2
        For j = 0 To 2
            a(j, i) = k
            document.writeln(k)
            k = k + 1
        Next
        document.writeln("<BR>")
    Next
    CreateVBAArray = a
End Function
-->
</SCRIPT>
<SCRIPT LANGUAGE="JScript">
<!--
function VBAArrayTest(vbarray)
{
    var a = new VBAArray(vbarray);
    var b = a.toArray();
    var i;
    for (i = 0; i < 9; i++)
    {
        document.writeln(b[i]);
    }
}
-->

```



```

</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JScript">
<!--
    VBAArrayTest(CreateVBAArray());
-->
</SCRIPT>
</BODY>

```

## 25. toLowerCase

*strVariable*.**toLowerCase**( ): Trả về chuỗi thường từ chuỗi *strVariable*.

Ví dụ:

```

var strVariable = "This is a STRING object";
strVariable = strVariable.toLowerCase( );

```

## 26. toUpperCase

*strVariable*.**toUpperCase**( ): Trả về chuỗi thường từ chuỗi *strVariable*.

Ví dụ:

```

var strVariable = "This is a STRING object";
strVariable = strVariable.toUpperCase( );

```

## 27. ubound

*safeArray*.**ubound**(*dimension*): trả về giá trị thứ tự lớn nhất của mảng *safeArray* với chiều *dimension* chỉ định.

Ví dụ:

```

<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Function CreateVBAArray()
    Dim i, j, k
    Dim a(2, 2)
    k = 1
    For i = 0 To 2
        For j = 0 To 2
            a(j, i) = k
            k = k + 1
        Next
    Next
    CreateVBAArray = a
End Function

```

```
-->
</SCRIPT>
    <SCRIPT LANGUAGE="JScript">
<!--
function VBAArrayTest(vba)
{
    var i, s;
    var a = new VBAArray(vba);
    for (i = 1; i <= a.dimensions(); i++)
    {
        s = "The upper bound of dimension ";
        s += i + " is ";
        s += a.ubound(i) + "<BR>";
        return(s);
    }
}
-->
</SCRIPT>
</HEAD>
    <BODY>
<SCRIPT language="jscript">
    document.write(VBAArrayTest(CreateVBAArray()));
</SCRIPT>
</BODY>
```

## 28. item

**myEnum.item( )**: Trả về giá trị thành phần hiện tại của tập hợp *myEnum*.

Ví dụ:

```
function ShowDriveList()
{
    var fso, s, n, e, x;
    fso = new ActiveXObject("Scripting.FileSystemObject");
    e = new Enumerator(fso.Drives);
    s = "";
    for (; !e.atEnd(); e.moveNext())
    {
        x = e.item();
        s = s + x.DriveLetter;
        s += " - ";
        if (x.DriveType == 3)
            n = x.ShareName;
```

```

else if (x.IsReady)
    n = x.VolumeName;
else
    n = "[Drive not ready]";
s += n + "<br>";
}
return(s);
}

```

### 29. isNaN

Trả về 1 trị Boolean xác định 1 giá trị có phải là số hay không.

Cú pháp

**isNaN**(*numvalue*)

Trong đó *numvalue* là giá trị cần kiểm tra. Hàm **isNaN** trả về giá trị **true** nếu *numvalue* không phải là số, và **false** nếu là số.

Ví dụ:

```

function checkinput(value)
{
    if (isNaN(value)==true)
    {
        alert("Làm ơn nhập vào kiểu số");
        document.forms[0].o_so.focus();
        return;
    }
    document.forms[0].submit();
}

```

## III. ĐỐI TƯỢNG CỦA JAVASCRIPT

### 1. ActiveXObject

Cho phép và trả về một sự tham khảo đến một đối tượng Automation .

Cú pháp

var newObject = new **ActiveXObject**(*servername.typename*[, *location*])

Phần	Mô tả
<i>servername</i>	Phải có. Tên của ứng dụng cung cấp đối tượng.
<i>typename</i>	Phải có. Kiểu hoặc lớp của đối tượng để tạo.
<i>location</i>	Tùy chọn. Tên của server nơi đối tượng được tạo.

Ví dụ:

Để tạo 1 Automation object, tạo **ActiveXObject** mới cho 1 biến đối tượng:

```

var ExcelSheet;
ExcelSheet = new ActiveXObject("Excel.Sheet");
Ví dụ:
// Make Excel visible through the Application object.
ExcelSheet.Application.Visible = true;
// Place some text in the first cell of the sheet.
ExcelSheet.ActiveSheet.Cells(1,1).Value = "This is column A, row 1";
// Save the sheet.
ExcelSheet.SaveAs("C:\\TEST.XLS");
// Close Excel with the Quit method on the Application object.
ExcelSheet.Application.Quit();
// Release the object variable.
ExcelSheet = "";

```

Ví dụ:

Trả về version của Excel chạy từ PC từ xa của mạng có computer name là "myserver":

```

Function GetVersion {
    var XLApp = CreateObject("Excel.Application", "MyServer");
    return(XLApp.Version);
}

```

## 2. Array

Hỗ trợ tạo các mảng với nhiều kiểu dữ liệu.

### Cú pháp

**new Array()**

**new Array(size)**

**new Array(element0, element1, ..., elementn)**

Phần	Diễn giải
<i>size</i>	Cỡ của mảng. Các thành phần của mảng có chỉ số đầu tiên là 0 đến <i>size</i> - 1.
<i>element0, ..., elementn</i>	Các thành phần đặt trong mảng. Có $n + 1$ thành phần với thành phần là các giá trị gán trực tiếp cho mảng.

**Ví dụ:**

```

var my_array = new Array();
for (i = 0; i < 10; i++)
{
    my_array[i] = i;
}

```

```
x = my_array[4];
```

### 3. Date

Định khoảng lưu trữ và cho 1 ngày.

#### Cú pháp

```
var newDateObj = new Date()
```

```
var newDateObj = new Date(year, month, date[, hours[, minutes[, seconds[,ms]]]])
```

### 4. Function

Tạo một function mới.

#### Cú pháp

```
function functionname( [argname1 [, ... argnameN]] )
```

```
{
  body
}
```

Part	Description
<i>functionname</i>	Tên của function.
<i>argname1...argnameN</i>	Tuỳ chọn. Danh sách đối số của hàm.
<i>body</i>	Chuỗi chứa đựng các mã lệnh JScript sẽ chạy khi function được gọi.

#### Ví dụ:

```
function add(x, y)
{
  return(x + y);
}
```

**ví dụ:** gọi hàm add như sau:

```
add(2, 3);
```

### 5. Math

Một đối tượng bên trong cung cấp hằng hàm toán học.

#### Cú pháp

```
Math[.{property | method}]
```

#### Các thuộc tính(property):

LN2

LN10

LOG2E

LOG10E

PI

SQRT1\_2

SQRT2

**Ví dụ:**

**var numVar**

*numVar* = **Math.E** //Trả về hằng Euler's. **E** tương đương 2.718

**var numVar**

*numVar* = **Math.PI** // trả về giá trị PI tương đương 3.141592653589793

**var numVar**

*numVar* = **Math.LN2** // trả về giá trị LN2 tương đương 0.693

## 6. Number

Đối tượng biểu diễn kiểu dữ liệu số, khai báo rõ ràng.

**Cú pháp**

**new Number( value )**

đối số *value* đòi hỏi giá trị số.

Ví dụ:

```
so=new Number(12);
```

## 7. VBArray

Hỗ trợ truy xuất mảng kiểu Visual Basic.

**Cú pháp**

**new VBArray(safeArray)**

*safeArray* là 1 giá trị **VBArray**.

**Ví dụ:**

```
<HEAD>
```

```
<SCRIPT LANGUAGE="VBScript">
```

```
<!--
```

```
Function CreateVBArray()
```

```
Dim i, j, k
```

```
Dim a(2, 2)
```

```
k = 1
```

```
For i = 0 To 2
```

```
For j = 0 To 2
```

```
    a(j, i) = k
```

```
    document.writeln(k)
```

```
    k = k + 1
```

```
Next
```

```
document.writeln("<BR>")
```

```
Next
```

```
CreateVBArray = a
```

```
End Function
```

```
-->
</SCRIPT>
<SCRIPT LANGUAGE="JScript">
<!--
function VBAArrayTest(vbarray)
{
    var a = new VBAArray(vbarray);
    var b = a.toArray();
    var i;
    for (i = 0; i < 9; i++)
    {
        document.writeln(b[i]);
    }
}
-->
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JScript">
<!--
    VBAArrayTest(CreateVBAArray());
-->
</SCRIPT>
</BODY>
```