

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN

LÊ TẤN HÙNG
ĐINH THỊ PHƯƠNG THU
VŨ ĐỨC VƯỢNG
HUỖNH QUYẾT THẮNG (CHỦ BIÊN)

Lập trình ứng dụng
**Web
Internet
& MẠNG KHÔNG DÂY**

INTERNET AND WIRELESS PROGRAMMING

TẬP 1

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN

LÊ TẤN HÙNG
ĐINH THỊ PHƯƠNG THU
VŨ ĐỨC VƯỢNG
HUỲNH QUYẾT THẮNG (Chủ biên)

LẬP TRÌNH ỨNG DỤNG
WEB INTERNET
VÀ MẠNG KHÔNG DÂY

(INTERNET AND WIRELESS PROGRAMMING)

TẬP 1



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
HÀ NỘI

Chịu trách nhiệm xuất bản: PGS. TS. TÔ ĐĂNG HẢI
Biên tập và sửa bài: ThS. NGUYỄN HUY TIẾN
NAM SƠN
NGỌC LINH
Trình bày bìa: HƯƠNG LAN

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
70 Trần Hưng Đạo - Hà Nội

60 – 6T7.3 150 – 183.1 – 05
KHKT – 06

LỜI NÓI ĐẦU

Ngày nay việc sử dụng Internet và các dịch vụ trên Internet là một phần tất yếu của cuộc sống. Công nghệ Internet và các ứng dụng phát triển trên nền Web và Internet đã và đang làm cho đời sống của mỗi chúng ta thay đổi. Với các cán bộ viên chức, với các nhà khoa học, với các nhà kinh doanh cho đến người nội trợ bình thường, việc sử dụng Internet, sử dụng các dịch vụ trên Internet là một công việc thường xuyên. Internet là một nguồn thông tin quý giá phục vụ cho nghiên cứu, cho kinh doanh, cho giải trí. Vì vậy ngày càng có nhiều ứng dụng trên Internet, sử dụng giao diện Web. Các công nghệ phát triển trên Internet ngày một hoàn thiện.

Trên cơ sở đó, môn học Lập trình Web, Internet đã được đưa vào giảng dạy và nghiên cứu tại các khoa, ngành tin học, toán tin, thương mại điện tử, cơ khí hay các ngành có liên quan khác trong các trường đại học. Môn học này sẽ là tiền đề cung cấp cho sinh viên và các nhà nghiên cứu các lý thuyết từ cơ sở cho đến ứng dụng nhằm xây dựng và xử lý thông tin dưới dạng các trang Web, các định dạng và đặc tả theo các chuẩn thông tin trên trang Web, nhờ đó cho phép họ tiếp cận dễ dàng hơn với các kỹ thuật lập trình hiệu quả các ứng dụng trên nền Web và Internet.

Cuốn sách này được biên soạn dựa trên giáo trình giảng dạy các môn học **Thiết kế Web và Lập trình ứng dụng Web và Internet** tại Khoa Công nghệ thông tin Trường Đại học Bách Khoa - Hà Nội. Trong cuốn sách cũng đề cập đến công nghệ mạng không dây và ngôn ngữ đặc tả WML cùng với các kỹ thuật lập trình tiêu biểu. Mục đích của cuốn sách là giúp cho sinh viên và các bạn đọc nắm được các kiến thức cơ bản trong thiết kế trang Web, lập trình các ứng dụng giao diện Web trên Internet, nắm được các kiến thức về HTML, HMTML, DHTML, WML, công cụ lập trình Javascript, công cụ thiết kế trang Web Macromedia Dreamweaver. Trong cuốn sách cũng trình bày mô hình lập trình: lập trình hướng sự kiện với các kỹ thuật tiêu biểu của mô hình này. Phần cuối của cuốn sách trình bày về WML sử dụng để xây dựng các ứng dụng sử dụng mạng không dây. Cuốn sách có thể làm tài liệu tham khảo để sử dụng làm các bài tập lớn trong các môn học như Kỹ thuật lập trình, Lập trình hướng đối tượng, v.v.

Vì khuôn khổ cuốn sách có hạn mà lĩnh vực kỹ thuật lập trình Web và Internet là lĩnh vực rộng với các định hướng kỹ thuật và phát triển ứng dụng khác nhau cho nên ngoài phần đưa ra các lý thuyết cơ sở, nội dung viết chỉ đi sâu vào khai thác các kỹ thuật tiêu biểu.

Do những hạn chế của các tác giả cho nên cuốn sách không thể không có các thiếu sót nhất định. Chúng tôi chân thành mong nhận được các ý kiến đóng góp của các đồng nghiệp và các bạn đọc nhằm giúp cho chất lượng của cuốn sách được cao hơn ở các lần tái bản sau.

Ý kiến đóng góp xin gửi về:

Trường Đại học Bách Khoa Hà Nội

Khoa Công nghệ Thông tin

Tel: 04-8692463

Người nhận:

Huỳnh Quyết Thắng

Email: thanghq@it-hut.edu.vn

Vũ Đức Vượng

Email: vuongvd@it-hut.edu.vn

Đinh Thị Phương Thu

Email: thudtp@it-hut.edu.vn

Lê Tấn Hùng

Email: hunglt@it-hut.edu.vn

Trong quá trình biên soạn, chúng tôi nhận được rất nhiều những ý kiến đóng quý báu từ các giáo sư, bạn bè và các đồng nghiệp trong và ngoài Khoa, đặc biệt là các thầy Đỗ Văn Uy, Nguyễn Ngọc Bình, Lương Mạnh Bá.

Chúng tôi cũng xin chân thành bày tỏ sự biết ơn đến **Ban Chủ nhiệm Khoa Công nghệ Thông tin - Trường Đại học Bách Khoa Hà Nội** đã động viên và tạo mọi điều kiện để cuốn sách này sớm được ra mắt bạn đọc.

Các tác giả

PHẦN 1

MÁY TÍNH & INTERNET

NỘI DUNG

Chương 1 - Giới thiệu về máy tính và Internet

Chương 2 - Microsoft Internet Explorer 6

Chương 1

Giới thiệu về máy tính và Internet

Nội dung của chương:

- Giới thiệu về những khái niệm căn bản của thế giới máy tính
- Làm quen với các ngôn ngữ lập trình khác nhau
- Lịch sử tiến hoá của Internet và World Wide Web
- Vai trò của HTML, XHTML, JavaScript, Dynamic HTML trong việc phát triển các ứng dụng Khách/Chủ cho Internet và WWW.

1.1. Giới thiệu

Xin chào mừng bạn đến với thế giới lập trình Internet và World Wide Web. Chúng tôi đã làm việc thực sự nghiêm túc để đưa đến cho các bạn một cuốn sách đầy đủ thông tin, thú vị và tạo hứng thú cho việc học tập của các bạn.

Những công nghệ mà các bạn học trong cuốn sách này thích hợp cho những người mới bắt đầu cũng như các chuyên gia có kinh nghiệm đang xây dựng những hệ thống thông tin quan trọng. Làm thế nào để một cuốn sách có thể thích hợp với cả hai nhóm độc giả này: những người mới bắt đầu và những người đã giàu kinh nghiệm? Câu trả lời là lời của cuốn sách nhấn mạnh vào việc đạt được sự trong sáng của chương trình thông qua những kỹ thuật đã được khẳng định như *Lập trình có cấu trúc* và *Lập trình hướng đối tượng*. Những người mới bắt đầu sẽ được hướng dẫn lập trình đúng cách ngay từ đầu. Chúng tôi đã cố gắng viết theo một phong cách trong sáng và dễ hiểu.

Có lẽ là quan trọng nhất, cuốn sách mang đến cho bạn hàng trăm ví dụ cùng các kết quả minh họa khi chạy trên trình duyệt hoặc chạy trên máy tính. Chúng tôi cung cấp tất cả các khái niệm trong văn cảnh của những chương trình hoàn chỉnh và gọi đó là **tiếp cận theo kiểu Mã sống**.

Trong các chương đầu chúng tôi giới thiệu các khái niệm căn bản về máy tính, Internet và World Wide Web. Giới thiệu cách sử dụng phần mềm để duyệt Web, hướng dẫn

Chương 1

Giới thiệu về máy tính và Internet

các bạn theo từng bước một cách cẩn thận về việc lập trình Web với ngôn ngữ phổ dụng là JavaScript. Trong cuốn sách này, chúng tôi sẽ sử dụng từ *Scripting* (lập kịch bản) và được hiểu như là “lập trình” (*programming*). Những người mới bắt đầu sẽ tìm thấy trong phần giới thiệu về JavaScript những kiến thức cơ sở để có thể tạo các kịch bản trên rất nhiều ngôn ngữ như Perl, PHP, ASP.NET, ColdFusion, Python, VBScript và JSP,... Những lập trình viên kinh nghiệm sẽ đọc rất nhanh một số chương đầu và sẽ tìm thấy nhiều điều thú vị ở những chương sau.

Hầu hết mọi người đều quen thuộc với những gì mà máy tính làm. Sử dụng cuốn sách này, bạn sẽ biết cách làm chủ được nó để thực hiện những tác vụ cụ thể. **Phần mềm** (ví dụ như những dòng lệnh bạn viết để ra lệnh cho máy tính thực hiện các hành động hoặc quyết định) điều khiển máy tính (thường được gọi là **Phần cứng**). Ứng dụng của máy tính ngày càng tăng lên ở bất kỳ một lĩnh vực nào. Trong một kỷ nguyên của sự tăng giá ổn định như hiện nay, thì giá cả máy tính ngày càng giảm mạnh vì sự phát triển nhanh chóng của cả phần cứng lẫn phần mềm. Máy tính trước đây chứa trong cả một căn phòng lớn và giá hàng triệu đô la thì chỉ mới hai thập kỷ vừa qua bây giờ có thể thu gọn lại chỉ còn là trên bề mặt của một con Chip silicon nhỏ hơn móng tay, giá khoảng vài đô la. Silicon là một trong những vật liệu có nhiều trên trái đất bởi nó có trong cát. Các chip sử dụng công nghệ silicon đã làm cho máy tính được thương mại hoá và phổ biến. Hàng trăm triệu máy tính đang có trên thế giới phục vụ con người ở tất cả các lĩnh vực như thương mại, công nghiệp, quản lý, giáo dục và những công việc thường ngày.

Cho đến gần đây, sinh viên vẫn học về một kỹ thuật lập trình gọi là **Lập trình có cấu trúc**. Và khi học về rất nhiều những ngôn ngữ ở trong cuốn sách này, bạn sẽ học cả hai kỹ thuật đó là lập trình có cấu trúc và **dựa trên đối tượng**. Sau cuốn sách này, bạn sẽ sẵn sàng để học những ngôn ngữ ứng dụng rộng rãi như C++, Java, C# và Visual Basic.NET để học về một kỹ thuật lập trình mạnh mẽ hơn: **Lập trình hướng đối tượng**. Chúng tôi tin rằng lập trình hướng đối tượng sẽ là kỹ thuật quan trọng, ít ra là cho đến thập kỷ sau.

Người sử dụng ngày nay có thói quen với những ứng dụng có giao diện màn hình đồ hoạ (GUI). Người dùng muốn các ứng dụng sử dụng hết các khả năng đa phương tiện như hình ảnh, đồ hoạ, ảnh động, tiếng nói và video. Họ muốn các ứng dụng có thể chạy trên Internet và World Wide Web và giao tiếp với các ứng dụng khác. Người dùng muốn áp dụng cả hai kỹ thuật xử lý file và xử lý cơ sở dữ liệu. Họ muốn các ứng dụng không giới hạn vào máy để bàn hay các máy mạng cục bộ, mà là có thể tích hợp được các thành phần Internet và World Wide Web, và cơ sở dữ liệu từ xa. Các lập trình viên muốn sử dụng tất cả những khả năng đó trong một ứng dụng thực sự có khả năng linh động do vậy các ứng dụng sẽ chạy ở bất kỳ nền tảng nào (ví dụ các kiểu máy tính khác nhau chạy những hệ điều hành khác nhau). Trong cuốn sách này, chúng tôi mang lại cho bạn một nền tảng công nghệ phát triển phần mềm mạnh mẽ, cho phép bạn xây dựng rất nhiều hệ thống như vậy. Nội dung của cuốn sách tập trung vào việc sử dụng các công nghệ như *Extensible HyperText Markup Language (XHTML)*, *JavaScript*, *Dynamic HTML*, *Flash*, *Dreamweaver* và *Extensible*

Chương 1

Giới thiệu về máy tính và Internet

Markup Language (XML) để xây dựng các ứng dụng trên nền Web và chạy ở phía máy khách (*Client*). Các lập trình viên sử dụng những công nghệ đó để xây dựng ứng dụng phía máy chủ. Những ứng dụng này thường chạy trong những máy công suất lớn mà ở đó có những web site thương mại quan trọng của các tổ chức. Bạn đọc sau khi làm chủ được các công nghệ trong cuốn sách này sẽ có khả năng xây dựng được rất nhiều ứng dụng khác nhau trên nền web, khách/chủ, nhiều lớp.

Chúng ta bắt đầu bằng một cuộc thảo luận về phần cứng máy tính và cơ bản về phần mềm. Nếu đã quen thuộc với máy tính, bạn có thể bỏ qua một số phần của chương này.

1.2. Máy tính là gì?

Một chiếc máy tính là một thiết bị để thực hiện các thao tác tính toán và tạo ra các quyết định logic với tốc độ hàng triệu thậm chí đến hàng tỷ lần, nhanh hơn tốc độ mà con người có thể làm rất nhiều lần. Ví dụ, một người điều khiển một máy tính tay có thể phải mất cả đời để hoàn thành hàng trăm triệu phép tính trong khi một máy vi tính mạnh có thể thực hiện việc đó chỉ trong một giây. Ngày nay, siêu máy tính nhanh nhất có thể thực hiện được hàng trăm tỷ phép cộng mỗi giây.

Các máy tính xử lý dữ liệu dưới sự hướng dẫn của các tập lệnh gọi là chương trình máy tính. Các chương trình máy tính chỉ dẫn máy tính được tạo ra bởi các lập trình viên.

Những thiết bị khác nhau như bàn phím, màn hình, đĩa, bộ nhớ và bộ xử lý, các thành phần tạo nên một hệ thống máy tính được gọi là **Phần cứng**. Phần cứng có thể chia ra làm các phần logic khác nhau như sau:

- **Đơn vị nhập vào:** đây là nơi để máy tính nhận thông tin. Nó lấy thông tin (dữ liệu và chương trình) từ các nguồn vào khác nhau và làm cho các thông tin đó được sẵn sàng sử dụng bởi các thành phần khác.
- **Đơn vị ra:** đây là bộ phận chuyển thông tin ra khỏi máy tính. Nó lấy thông tin đã được xử lý bởi máy tính và gửi nó tới các thiết bị ra để làm cho những thông tin đó có thể sử dụng được ở ngoài máy tính.
- **Bộ nhớ:** là nơi có tốc độ truy cập cao, có thể nói như là một cái “kho” nhỏ của máy tính. Nó chứa tạm thời các thông tin được nhập vào và những thông tin đã xử lý.
- **Đơn vị xử lý toán học và logic:** đây là “nhà máy” của máy tính. Nó chịu trách nhiệm thực hiện các phép tính toán và ra quyết định.
- **Bộ xử lý trung tâm (CPU):** đây là bộ phận điều khiển hầu như toàn bộ hoạt động của hệ thống. CPU hoạt động như điều phối viên của máy tính. CPU chỉ dẫn cho các bộ phận khác biết khi nào thì phải làm gì, thực hiện như thế nào.
- **Bộ nhớ thứ cấp:** đây là một “kho” có dung lượng lớn, lưu trữ lâu dài. Các chương trình hoặc dữ liệu hiện thời không sử dụng bởi các đơn vị khác thường được để trong các thiết bị nhớ thứ cấp.

Chương 1

Giới thiệu về máy tính và Internet

1.3. Các kiểu ngôn ngữ lập trình

Các chương trình máy tính thường được gọi là phần mềm. Các lập trình viên viết các lệnh tạo thành phần mềm từ nhiều ngôn ngữ khác nhau, một số ngôn ngữ máy tính hiểu được ngay và một số khác thì đòi hỏi phải có các bước dịch. Có thể chia các ngôn ngữ lập trình đang được sử dụng ngày nay ra làm 3 loại:

- Ngôn ngữ máy
- Ngôn ngữ Assembly
- Các ngôn ngữ bậc cao

Ngôn ngữ máy

Bất kỳ máy tính nào cũng chỉ hiểu và có thể hiểu trực tiếp được ngôn ngữ máy. Ngôn ngữ máy là một “ngôn ngữ tự nhiên” đối với bất kỳ một máy tính nào và được định nghĩa bởi thiết kế phần cứng của nó, thường là một dãy các số số 1 và 0 có ý nghĩa chỉ dẫn máy tính thực hiện công việc tại một thời điểm. Ngôn ngữ máy phụ thuộc vào máy (ví dụ một ngôn ngữ cụ thể chỉ được sử dụng bởi một nền tảng, hoặc loại máy tính nào đó). Ngôn ngữ máy rất khó hiểu đối với con người, chúng ta sẽ thấy ví dụ để thực hiện phép cộng cần các lệnh sau đây :

+1300042774
+1400593419
+1200274027

Ngôn ngữ Assembly

Máy tính ngày càng phổ biến và rõ ràng là ngôn ngữ máy làm cho việc phát triển ứng dụng trở nên chậm chạp và mệt mỏi đối với lập trình viên. Thay vì sử dụng một dãy các số mà máy tính có thể trực tiếp hiểu được, lập trình viên bắt đầu sử dụng các ngôn ngữ tự nhiên (như tiếng Anh) viết tắt để biểu diễn các thao tác cơ bản của máy tính. Những chữ viết tắt đó tạo thành ngôn ngữ *Assembly*. Các chương trình dịch nó được gọi là *Assemblers*, chúng ta phát triển và chuyển ngôn ngữ *Assembly* sang ngôn ngữ máy.

Các lệnh thực hiện phép cộng như trên được viết như sau trong *Assembly*:

LOAD X
ADD Y
STORE Z

Các ngôn ngữ bậc cao

Mặc dù những mã như vậy là có thể hiểu được, nhưng nó không dễ dàng đối với máy tính cho đến khi được dịch ra thành ngôn ngữ máy. Việc sử dụng máy tính ngày càng tăng nhanh với ngôn ngữ *assembly*, nhưng việc lập trình vẫn còn đòi hỏi rất nhiều lệnh để thực hiện một nhiệm vụ đơn giản. Để tăng tốc độ lập trình, các ngôn ngữ bậc cao đã được phát

Chương 1

Giới thiệu về máy tính và Internet

triển, trong đó một câu lệnh cũng có thể được viết để thực hiện một tác vụ. Các chương trình dịch chuyển các chương trình được viết bằng ngôn ngữ bậc cao sang ngôn ngữ máy gọi là các trình dịch. Các ngôn ngữ bậc cao cho phép các lập trình viên viết ra các lệnh giống như các câu nói của tiếng Anh hàng ngày chứa các ký hiệu toán học. Chương trình trên được viết lại thành:

$$Z = X + Y$$

Từ đây, chúng ta dễ thấy tại sao các lập trình viên thấy các ngôn ngữ bậc cao tiện dụng hơn cả ngôn ngữ máy lẫn ngôn ngữ *assembly*. C++, Java, C# và Visual Basic.NET là một số ngôn ngữ bậc cao mạnh mẽ và hiện đang được sử dụng rộng rãi.

Quá trình dịch từ một chương trình ngôn ngữ bậc cao sang ngôn ngữ máy có thể tốn khá nhiều thời gian. Các chương trình thông dịch có thể chạy các lệnh của ngôn ngữ bậc cao một cách trực tiếp mà không phải biên dịch chúng thành ngôn ngữ máy. Mặc dù các chương trình được biên dịch chạy nhanh hơn so với chương trình được thông dịch, nhưng các chương trình thông dịch lại phổ biến trong việc phát triển ở các môi trường mà chương trình thường hay bị dịch lại như thêm vào một chức năng, hoặc lỗi vừa được sửa. Trong cuốn sách này chúng tôi nghiên cứu các ngôn ngữ quan trọng: HTML, JavaScript DHTML, XML, XHTML và WML. Ngôn ngữ kịch bản được xử lý bởi các trình thông dịch. Bạn sẽ thấy rằng các trình thông dịch đóng vai trò quan trọng đặc biệt trong việc giúp các ngôn ngữ kịch bản đạt được kết quả và khả chuyển giữa các nền tảng khác nhau.

1.4. Các ngôn ngữ bậc cao khác

Fortran

Trong số hàng trăm ngôn ngữ được phát triển chỉ một số ít các ngôn ngữ bậc cao đạt được sự chấp nhận rộng rãi. IBM phát triển **Fortran** (*FORmula TRANslator*) từ 1954 đến 1957 để phục vụ cho việc tính toán khoa học và các ứng dụng kỹ thuật đòi hỏi những phép toán phức tạp. Fortran hiện vẫn còn được sử dụng rộng rãi.

Cobol

Một nhóm các nhà sản xuất máy tính và chính phủ và giới công nghiệp đã phát triển ngôn ngữ **Cobol** (*Common Business Oriented Language*) năm 1959. Các ứng dụng thương mại thao tác trên một lượng lớn dữ liệu được lập trình trên Cobol. Ngày nay, gần như một nửa các chương trình thương mại đều sử dụng Cobol.

Basic

Basic đã được phát triển năm 1965 tại Dartmouth College là một ngôn ngữ đơn giản để giúp những người mới bắt đầu học lập trình. Bill Gates là người đã thể hiện Basic trên các máy cá nhân đầu tiên. Ngày nay, Microsoft – công ty mà Bill Gates lập ra – là công ty phần mềm hàng đầu thế giới.

Chương 1.

Giới thiệu về máy tính và Internet

1.5. Lập trình có cấu trúc

Giữa những năm 1960, rất nhiều nỗ lực để phát triển phần mềm lớn đều gặp khó khăn. Kế hoạch phát triển phần mềm thường hay bị quá thời gian cho phép, giá tăng cao vượt ngân sách và những sản phẩm được hoàn thành thì không đáng tin cậy. Người ta bắt đầu nhận ra rằng phát triển phần mềm là một công việc phức tạp hơn nhiều so với những gì họ tưởng tượng. Các hoạt động nghiên cứu vào những năm 1960 cho thấy sự tiến hoá của việc lập trình có cấu trúc - một tiếp cận có hệ thống để viết các chương trình trong sáng hơn những chương trình không có cấu trúc, để kiểm tra hơn và để thay đổi hơn. Một trong những kết quả rõ ràng nhất của nghiên cứu này là sự phát triển ngôn ngữ **Pascal** bởi Giáo sư Nicklaus Wirth năm 1971. Pascal, tên của một nhà toán học và triết học thế kỷ 17 Blaise Pascal, đã được phát triển cho mục đích dạy học lập trình có cấu trúc trong môi trường học viện và đã nhanh chóng trở thành ngôn ngữ được ưa thích trong các trường Đại học.

Ngôn ngữ lập trình **Ada** đã được phát triển dưới sự hỗ trợ của Bộ Quốc phòng Mỹ (*United States Department of Defense* (DoD)) vào những năm 1970 và đầu những năm 1980. Hàng trăm ngôn ngữ khác nhau đã được sử dụng để xây dựng hệ thống phần mềm ra lệnh điều khiển đồ sộ của DoD. DoD muốn một ngôn ngữ đơn nhất có thể đảm nhiệm hầu hết các nhu cầu của họ. Pascal được chọn là ngôn ngữ cơ sở, nhưng cuối cùng Ada lại rất khác Pascal. Ngôn ngữ này đã được đặt tên theo Augusta Ada Byron King, Lady Lovelace, con gái của nhà thơ Lord Byron. Lady Lovelace được coi là người đã tạo ra chương trình máy tính đầu tiên, vào đầu những năm 1800 (cho máy động cơ phân tích tính toán thiết kế bởi Charles Babbage). Một khả năng quan trọng của Ada được gọi là đa nhiệm, cho phép lập trình viên có nhiều tác vụ chạy cùng lúc với nhau. Ngôn ngữ lập trình **Java**, thông qua một kỹ thuật gọi là *multithreading*, cũng cho phép lập trình viên viết các chương trình với những tác vụ có thể thực hiện song song. Những ngôn ngữ bậc cao phổ dụng khác, như C và C++, thường chỉ có thể cho phép các chương trình chạy một tác vụ tại một thời điểm (mặc dù chúng có thể hỗ trợ đa nhiệm thông qua một thư viện đặc biệt).

1.6. Lịch sử của Internet

Vào cuối những năm 1960, một số nhà khoa học đã làm việc tại MIT trong một dự án được tài trợ bởi ARPA (*Advanced Research Projects Agency*) của Bộ Quốc phòng Mỹ. ARPA đã tổ chức một cuộc hội thảo tại đại học Illinois cho rất nhiều sinh viên được ARPA tài trợ để gặp gỡ và chia sẻ các ý tưởng. Trong khi thảo luận, ARPA đưa ra một kế hoạch chi tiết cho việc nối kết các hệ thống máy tính chính cho khoảng hơn một chục các trường đại học và viện nghiên cứu mà ARPA tài trợ. Chúng sẽ được kết nối với các đường truyền thông và hoạt động ở một tốc độ là 56Kbps (tức là 56.000 bit trên giây) trong khi lúc đó hầu hết mọi người được kết nối bằng đường điện thoại với tốc độ 110 bit trên giây. Trong cuộc hội thảo đó mọi người đã đạt được nhiều kết quả và đưa ra một viễn cảnh tốt đẹp. Ngay sau hội thảo, ARPA quyết định thực hiện ngay ARPANET, tiền thân của mạng Internet bây giờ. Mọi thứ diễn ra khác so với những gì trong kế hoạch còn hơn cả lợi ích của việc những nhà nghiên cứu chia sẻ máy tính của họ, nó nhanh chóng cho phép các nhà nghiên cứu đối thoại

Chương 1

Giới thiệu về máy tính và Internet

rất nhanh và dễ dàng thông qua **thư điện tử** (e-mail), đó chính là lợi ích chính to lớn của mạng ARPANET.

Một trong những lợi ích chính cho ARPANET là cũng cho phép nhiều người cùng gửi và nhận thông tin đồng thời thông qua một đường truyền chung (ví dụ đường điện thoại). Mạng hoạt động với một kỹ thuật gọi là kỹ thuật **chuyển mạch gói** (*packet switching*), trong đó các dữ liệu sẽ được gửi đi dưới dạng các bó nhỏ gọi là **gói** (*packet*). Các gói bao gồm địa chỉ, thông tin điều khiển lỗi và thứ tự. Thông tin địa chỉ cho phép các gói được định tuyến đến đích của nó. Các thông tin theo dòng cho phép tái hợp các gói (vì việc định tuyến khá phức tạp nên các gói có thể đến đích không theo thứ tự nào cả) theo thứ tự cũ của chúng. Các gói từ những người gửi khác nhau được trộn lại trong cùng một đường. Kỹ thuật chuyển mạch gói này đã giảm một cách rất đáng kể giá đường truyền, so sánh với giá của các đường thông tin dành riêng. Mạng đã được thiết kế để hoạt động mà không cần sự điều khiển trung tâm. Nếu một phần của mạng bị lỗi, Phần còn lại của mạng vẫn sẽ tiếp tục định tuyến các gói từ người gửi đến người nhận thông qua một đường thay thế. Các giao thức để đàm thoại trên ARPANET ngày nay được biết đến qua cái tên TCP-giao thức điều khiển truyền. TCP cho phép chắc chắn rằng thông báo được định tuyến đúng từ người gửi đến người nhận và không có lỗi xảy ra. Theo quá trình tiến hoá của Internet, các tổ chức trên thế giới triển khai mạng riêng của họ cho cả hai hình thức mạng cục bộ và liên mạng. Một loạt các thiết bị mạng và phần mềm mạng xuất hiện. Một thách thức là làm thế nào cho những mạng đó có thể làm việc được với nhau. ARPA đã hoàn thành nhiệm vụ này với sự phát triển IP- giao thức liên mạng – và đã thực sự tạo ra một “mạng của các mạng”. Kiến trúc bây giờ của Internet là sự kết hợp các giao thức này và gọi chung nó là TCP/IP.

Ban đầu việc sử dụng Internet chỉ giới hạn trong các trường Đại học và viện nghiên cứu; sau đó quân đội bắt đầu dùng Internet. Cuối cùng là chính phủ quyết định cho phép sử dụng Internet cho mục đích thương mại. Giới kinh doanh nhanh chóng nhận ra họ có thể điều chỉnh hoạt động của họ và đưa lại những dịch vụ mới và tốt hơn tới khách hàng, do vậy họ bắt đầu chi những món tiền lớn để phát triển Internet. Điều này đưa đến một cuộc cạnh tranh mạnh mẽ giữa các nhà cung cấp dịch vụ viễn thông, phần cứng và phần mềm để phù hợp với nhu cầu này. Kết quả là băng thông (*bandwidth*-khả năng mang thông tin) trong Internet đã tăng nhanh với tốc độ tăng không ngờ trong khi giá thành lại giảm rất đáng kể.

1.7. Máy tính cá nhân

Năm 1977, Apple Computer phổ biến Điện toán cá nhân (*personal computing*). Ban đầu, đó là một giấc mơ của những người có sở thích riêng, nhưng máy tính nhanh chóng trở nên kinh tế đối với ngày càng nhiều người và họ có thể sử dụng cho mục đích cá nhân.

Năm 1981, IBM, nhà cung cấp máy tính lớn nhất thế giới, giới thiệu máy tính cá nhân IBM, chính thức đưa điện toán đến cho giới thương mại, công nghiệp, và các tổ chức chính phủ. Tuy nhiên, những máy tính ấy chỉ là những cỗ máy “đứng cô lập”, người sử dụng

Chương 1

Giới thiệu về máy tính và Internet

làm công việc trên máy riêng của mình và sau đó lưu chuyển đĩa để chia sẻ thông tin. Mặc dù những máy tính cá nhân ban đầu không đủ mạnh để có nhiều người dùng, những chiếc máy đó có thể liên kết với nhau trong mạng máy tính, đôi khi là qua đường điện thoại, mạng cục bộ trong một tổ chức.

Điều này dẫn tới một khái niệm mới là điện toán phân tán (*distributed computing*), trong đó việc tính toán của một tổ chức thay vì được thực hiện tại một máy tính trung tâm thì được chia sẻ, phân tán qua mạng để cùng thực hiện. Điện toán cá nhân đã đủ mạnh để gánh vác những yêu cầu của người dùng đơn lẻ. Ngày nay một chiếc máy tính cá nhân mạnh cũng bằng một chiếc máy tính hàng triệu đô la cách đây hai thập kỷ. Máy tính để bàn-gọi là các trạm làm việc-và các máy tính xách tay gọi là *laptop* cung cấp cho chúng ta một khả năng vô hạn. Thông tin được chia sẻ dễ dàng thông qua các mạng máy tính như mạng Internet trong đó một số máy tính cung cấp nơi chứa chương trình và dữ liệu gọi là máy chủ (server) có thể sử dụng bởi các máy khách (client) phân tán khắp nơi trong mạng, cũng từ đây mà có thuật ngữ Client/Server.

Các hệ điều hành phổ biến ngày nay, như Unix, Linux, Mac OS X, Windows 2000 và Windows XP, đều cung cấp tất cả những khả năng đã nói đến trong phần này.

1.8. Lịch sử của World Wide Web

World Wide Web cho phép người dùng máy tính định vị và xem các tài liệu đa phương tiện (ví dụ như tài liệu văn bản, hình ảnh, hoạt hình, âm thanh và video) trong hầu hết các chủ đề. Mặc dù Internet đã được phát triển hàng thập kỷ qua, nhưng sự ra đời của World Wide Web chỉ mới gần đây. Năm 1990, Tim Berners-Lee của CERN (Tổ chức của châu Âu nghiên cứu hạt nhân) đã phát triển World Wide Web và một số giao thức đàm thoại hiện nay làm xương sống của web. Việc sử dụng từ "bùng nổ" vào năm 1993 nhằm ám chỉ đến việc sử dụng trình duyệt *Mosaic*, là một trình duyệt có giao diện đồ họa. Marc Andreessen, và đội của ông tại Trung tâm quốc gia về các ứng dụng siêu máy tính (NCSA) đã phát triển *Mosaic*, đã tiếp tục phát triển Netscape, công ty mà nhiều người cho rằng nó chính là ngòi nổ cho cuộc bùng nổ kinh tế Internet vào cuối những năm 1990.

Internet và World Wide Web chắc chắn sẽ được liệt kê trong danh sách những sáng tạo quan trọng và ảnh hưởng sâu nhất trong đời sống nhân loại. Trong quá khứ, hầu hết các ứng dụng máy tính chạy đơn lẻ (do chưa được kết nối với nhau). Ngày nay các ứng dụng có thể được viết để làm việc với hàng trăm, thậm chí là hàng triệu máy tính khác nhau. Internet trộn các công nghệ điện toán và truyền thông lại với nhau, nó làm cho thông tin được truy cập gần như ngay lập tức và tiện lợi khắp thế giới làm cho công việc của chúng ta dễ dàng hơn. Những người dùng đơn lẻ và các cơ sở kinh doanh nhỏ có thể nhận được sự quảng cáo rộng rãi trên Internet. Người ta có thể tìm giá tốt nhất cho một sản phẩm hoặc dịch vụ nào đó, thay đổi cách thức làm việc, thay đổi nhận thức và suy nghĩ. Mọi người có thể liên lạc với nhau, và các nhà nghiên cứu có thể nghiên cứu khoa học trên phạm vi toàn thế giới...

Chương 1

Giới thiệu về máy tính và Internet

1.9. World Wide Web Consortium (W3C)

Tháng 10 năm 1994, Tim Berners-Lee thành lập một tổ chức gọi là **World Wide Web Consortium (W3C)** - phục vụ cho mục đích phát triển các công nghệ tương tác phi lợi nhuận cho World Wide Web. W3C cũng là một tổ chức chuẩn hoá.

Một trong những mục tiêu chính của W3C là làm cho Web có thể được truy cập khắp thế giới - không quan tâm đến khả năng, ngôn ngữ hay văn hoá. Công nghệ Web được chuẩn hoá bởi W3C gọi là các **Khuyến cáo**. Khuyến cáo của W3C bao gồm ba ngôn ngữ đánh dấu siêu văn bản có thể mở rộng (*XHTML*), *Cascading Style Sheets (CSS)*, ngôn ngữ đánh dấu siêu văn bản (*HTML*) và *XML*. Một khuyến cáo không phải là một sản phẩm phần mềm thực sự, mà là một tài liệu mô tả vai trò của công nghệ, luật củ pháp... Các bạn có thể xem chi tiết thông tin về các khuyến cáo W3C tại địa chỉ:

www.w3.org/Consortium/Process/Process-19991111/process.html#Recs.

W3C gồm có 3 tổ chức chính Viện công nghệ Massachusetts (MIT), Tổ chức nghiên cứu Tin học và Toán học của châu Âu (ERCIM) và đại học Keio của Nhật Bản với hàng trăm thành viên. Các thành viên cung cấp tài chính và các định hướng chiến lược cho W3C Consortium.

Trang chủ W3C (<http://www.w3.org>) cung cấp rất nhiều tài nguyên về công nghệ Internet và Web. Với mỗi công nghệ, W3C cung cấp các mô tả về công nghệ và lợi ích của nó đối với những nhà thiết kế web, lịch sử của công nghệ và những mục tiêu sắp tới của W3C trong việc phát triển công nghệ.

1.10. Khuyhnh hướng phát triển phần cứng

Cộng đồng Internet phát triển mạnh nhờ sự cải tiến mạnh mẽ không ngừng về phần cứng, phần mềm và công nghệ truyền thông. Nói một cách tổng quát, người ta muốn trả tiền càng ít càng tốt cho bất kỳ dịch vụ hoặc hàng hoá nào. Trong nhiều thập kỷ, và cả trong tương lai, giá của phần cứng sẽ ngày càng giảm nhanh chóng. Đây là một hiện tượng trong công nghệ. Mỗi năm, hoặc hai năm, khả năng của máy tính lại tăng gấp đôi, đặc biệt là dung lượng bộ nhớ để chạy chương trình, độ lớn của các thiết bị nhớ thứ cấp (như đĩa từ) và tốc độ của bộ xử lý. Điều này cũng đúng trong lĩnh vực viễn thông, đặc biệt là trong những năm gần đây, với nhu cầu vô tận về băng thông viễn thông dẫn đến sự cạnh tranh mạnh mẽ. Có lẽ chẳng có lĩnh vực nào mà công nghệ lại chuyển biến nhanh và giá thành lại cũng hạ nhanh đến như vậy.

Đối với phần cứng việc phát minh ra công nghệ vi xử lý và việc cung cấp rộng rãi nó vào những năm 1970 và 1980 đã mang lại sự cải tiến năng suất đáng kể vào năm 1990. Đối với phần mềm sự cải tiến năng suất được mang lại từ công nghệ đối tượng, một công nghệ sử dụng rất nhiều trong cuốn sách này.

Chương 1

Giới thiệu về máy tính và Internet

Gần đây, phần cứng đã dần dần chuyển theo hướng công nghệ di động, không dây. Các thiết bị nhô cầm tay ngày càng mạnh hơn những siêu máy tính đầu những năm 1970. Tính chất xách tay được trở thành tâm điểm của giới công nghiệp máy tính. Tốc độ truyền dữ liệu không dây đã nhanh đến mức một số người truy cập Internet chính thông qua mạng không dây cục bộ. Vài năm tới chúng ta chắc chắn sẽ chứng kiến những tiến bộ lớn lao trong công nghệ không dây dành cho cả người dùng đơn lẻ và các doanh nghiệp.

1.11. Xu hướng phần mềm chủ đạo - công nghệ hướng đối tượng

Trong lịch sử, vào những năm 1960, người ta đã thất bại rất nhiều trong phát triển những dự án phần mềm, đặc biệt là những dự án lớn. Trong khi giá của phần cứng đã giảm nhanh tạo ra triển vọng máy tính cá nhân có thể là một mặt hàng bình thường thì giá của việc phát triển phần mềm lại tăng và lập trình viên viết ra những ứng dụng mạnh, phức tạp nhưng lại không có một sự cải tiến đáng kể nào về công nghệ phát triển phần mềm. Đối tượng (*Objects*) về bản chất là các thành phần phần mềm có thể tái sử dụng mô hình hoá các thực thể trong thế giới thực. Các nhà phát triển phần mềm phát hiện ra rằng việc sử dụng thiết kế và tiếp cận triển khai kiểu modul, hướng đối tượng có thể làm cho các nhóm phát triển phần mềm trở nên năng suất hơn nhiều so với các kỹ thuật lập trình phổ biến như lập trình có cấu trúc.

Các chương trình hướng đối tượng thường dễ hiểu, dễ sửa chữa và thay đổi. Các cải tiến về công nghệ phần mềm bắt đầu xuất hiện và chứng tỏ hiệu năng (giống như việc lập trình có cấu trúc đã làm được những năm 1970). Không phải đợi đến khi kỹ thuật lập trình hướng đối tượng được sử dụng rộng rãi những năm 1980, và đặc biệt là những năm 1990, người ta mới cảm thấy họ có những công cụ mạnh để tạo nên những bước tiến dài trong quy trình phát triển phần mềm.

Trên thực tế, công nghệ đối tượng đã rất hứa hẹn từ khoảng giữa những năm 1960. Ngôn ngữ lập trình C++, do Bjarne Stroustrup phát triển tại AT&T vào đầu những năm 1980, dựa trên 2 ngôn ngữ là ngôn ngữ C, được bắt đầu phát triển tại AT&T để xây dựng hệ điều hành UNIX từ nửa đầu những năm 1970, và ngôn ngữ **Simula 67**, một ngôn ngữ lập trình mô phỏng phát triển ở châu Âu và được công bố năm 1967. C++ mang tất cả khả năng của C và thêm các khả năng của Simula để tạo ra và sử dụng các đối tượng.

Trước khi ngôn ngữ lập trình hướng đối tượng xuất hiện, các ngôn ngữ lập trình (như **Fortran**, **Pascal**, **Basic** và **C**) tập trung vào hành vi hơn là vào các sự vật hay đối tượng. Phong cách lập trình này được gọi là **lập trình hướng thủ tục**. Một trong những vấn đề chính của lập trình hướng thủ tục là các đơn vị chương trình do lập trình viên viết ra không phản ánh hiệu quả sự tồn tại của thế giới thực, vì vậy chúng không có tính tái sử dụng cao. Chúng ta sống trong một thế giới của các đối tượng. Hãy quan sát xung quanh bạn. Những chiếc ô tô, máy bay, những con người, những doanh nghiệp, động vật, các toà nhà. ... là những ví dụ về các đối tượng.

Với công nghệ đối tượng, những phần mềm được thiết kế đúng cách sẽ hướng tới

Chương 1

Giới thiệu về máy tính và Internet

khả năng tái sử dụng cao trong các dự án tiếp theo. Những thư viện các thành phần tái sử dụng như **Microsoft Foundation Classes (MFC)**, **Java Foundation Classes** của Sun Microsystems, Microsoft's **.NET Framework Class Library (FCL)** và những sản phẩm của các tổ chức phát triển phần mềm khác, có thể làm giảm đáng kể công sức để thực hiện một số loại chức năng của hệ thống. Các công ty cho rằng lập trình hướng đối tượng sẽ tiến tới tạo ra các phần mềm dễ hiểu hơn, có cấu trúc tốt hơn và dễ bảo trì hơn. Sự cải thiện này là rất quan trọng vì theo thống kê, khoảng 80% trị giá của phần mềm là ở công sức ban đầu để phát triển phần mềm, nhưng trong thực tế thì việc nâng cấp và bảo trì phần mềm được thực hiện trong suốt thời gian sử dụng của nó. Bất kỳ ai nhận thấy hiệu quả của hướng đối tượng cũng sẽ đồng ý rằng lập trình hướng đối tượng sẽ là phương pháp lập trình chính trong ít nhất là một hoặc hai thập kỷ tới.

1.12. JavaScript - kịch bản dựa vào đối tượng dành cho Web

JavaScript là một ngôn ngữ kịch bản dựa trên đối tượng với sự hỗ trợ mạnh mẽ cho công nghệ phần mềm. Các sinh viên có thể học cách tạo và sử dụng các đối tượng bắt đầu từ JavaScript. Trình xử lý JavaScript có sẵn trong hầu hết các trình duyệt Web thông dụng hiện nay. Chúng tôi nghĩ rằng các chương về JavaScript trong cuốn sách này không chỉ là một phần nhập môn về ngôn ngữ. Các chương này cũng giới thiệu về nền tảng lập trình máy tính, bao gồm cấu trúc điều khiển, hàm, mảng, cấu trúc đệ quy, xâu và đối tượng.

JavaScript do Netscape tạo ra. Phiên bản của Microsoft cho JavaScript được gọi là Jscript. Cả Netscape và Microsoft đều tuân theo chuẩn dành cho JavaScript/Jscript gọi là ECMAScript do tổ chức *ECMA International* đề ra. *ECMA International* (trước đây là Hiệp hội các nhà sản xuất máy tính châu Âu - *European Computer Manufacturers Association*) đã khuyến khích Netscape và Microsoft thêm vào những chức năng mới được xây dựng theo chuẩn *ECMAScript*. Ví dụ, Jscript cung cấp cách truy nhập đến một số đối tượng đặc biệt có thể tương tác với các công nghệ Microsoft khác. Tuy nhiên nếu bạn nhớ được tất cả các chức năng chính của ngôn ngữ kịch bản thì việc chuyển đổi giữa JavaScript và Jscript là rất dễ dàng.

1.13. Các trình duyệt Web

Để đảm bảo tạo ra giao diện nhất quán trên các trình duyệt cho máy khách là một trong những thách thức lớn của các chương trình ứng dụng phát triển cho Web. Trên thực tế không có một chuẩn chung nào để các nhà phát triển phần mềm tuân theo khi tạo ra các trình duyệt Web. Mặc dù các trình duyệt có chung một số chức năng, mỗi trình duyệt có thể hiển thị các trang khác nhau. Các trình duyệt có nhiều phiên bản khác nhau (1.0, 2.0, vv) và nhiều nền tảng khác nhau (UNIX, Microsoft Windows, Apple Macintosh, IBM OS/2, Linux, vv). Các nhà cung cấp mỗi khi thêm chức năng vào mỗi phiên bản của mình lại làm tăng lên sự không tương thích giữa các trình duyệt khác nhau. Rõ ràng điều đó là rất khó. Nếu không nói là không thể, để phát triển các trang Web có thể hiển thị đúng trên tất cả các phiên bản của mỗi trình duyệt.

Chương 1

Giới thiệu về máy tính và Internet

Nội dung quyển sách cố gắng giảm thiểu các vấn đề này bằng cách giới thiệu với các bạn về XHTML, ngôn ngữ hiện đang rất phổ biến như là một chuẩn chung trong các trình duyệt. Thêm vào đó chúng tôi cũng trình bày tập trung vào các công nghệ độc lập với nền tảng, như là XHTML, JavaScript, Cascading Style Sheets, XML. Tuy nhiên cũng có một số chủ đề là đặc trưng của Microsoft Windows, bao gồm trình duyệt Internet Explorer 6, HTML, Dynamic HTML.

1.14. C và C++

Trong nhiều năm, ngôn ngữ lập trình Pascal được ưa thích hơn trong các khóa học về nhập môn lập trình. Ngôn ngữ C phát triển từ ngôn ngữ B, được Dennis Ritchie của Bell Laboratories phát triển. C được cung cấp năm 1972 và nhanh chóng nổi tiếng là ngôn ngữ phát triển của hệ điều hành UNIX. Ngày nay, hầu như tất cả các hệ điều hành chính đều được viết bằng C và/hoặc C++.

Bjarne Stroustrup đã phát triển C++, một sự mở rộng của C, vào nửa đầu những năm 1980. C++ cung cấp một tính năng tương tự như C, nhưng quan trọng hơn, nó cung cấp những tính năng lập trình hướng đối tượng. C++ là một ngôn ngữ lai. Nó có thể lập trình theo phong cách của C (lập trình hướng thủ tục) hoặc theo phong cách hướng đối tượng hoặc cả hai. C và C++ có ảnh hưởng đến nhiều ngôn ngữ lập trình xuất hiện sau như Java, C#, JavaScript và Jscript, chúng có cú pháp tương tự như C và C++.

1.15. Java

Các thiết bị điện tử gia dụng thông minh có thể là khu vực chính tiếp theo chịu ảnh hưởng lớn của các bộ vi xử lý. Nhận thấy điều này, Sun Microsystems đã đầu tư một dự án nghiên cứu có tên là Green năm 1991. Kết quả của dự án là một ngôn ngữ lập trình hướng đối tượng dựa trên C và C++. Ban đầu, tác giả của nó, James Gosling, đặt tên là Oak vì có một cây sồi (Oak) bên ngoài cửa sổ văn phòng của ông. Sau đó họ phát hiện ra có một ngôn ngữ máy tính đã có tên là Oak. Khi một nhóm nhân viên của Sun đến một cửa hàng cà phê, cái tên Java đã được đề nghị và chấp nhận. Dự án Green tưởng chừng đã bị bỏ qua. Trong một cơ hội lớn, khi World Wide Web bùng nổ năm 1993, các thành viên của dự án Green nhìn thấy ngay khả năng sử dụng Java làm một ngôn ngữ lập trình Web. Điều đó đã thổi một luồng sinh khí mới cho dự án.

Java cho phép các lập trình viên tạo ra các trang Web chứa các nội dung động và tương tác với nhau. Các nhà phát triển cũng có thể dùng Java để tạo ra các ứng dụng phạm vi lớn, tăng cường chức năng cho Web servers (phần mềm cung cấp nội dung chúng ta nhìn thấy trong trình duyệt Web của mình), cung cấp ứng dụng cho các thiết bị gia dụng (ví dụ: điện thoại không dây, người giúp việc kỹ thuật số) và nhiều thứ khác.

Năm 1995, mọi thứ đã sẵn sàng để Sun phát triển Java. Tháng 10 năm 1995, tại cuộc hội thảo về Internet tổ chức ở Boston, đại diện của Sun đã mang đến một bài trình bày ấn tượng về Java. Bắt đầu từ đó, Java đã trở thành một trong những ngôn ngữ lập trình thông

Chương 1

Giới thiệu về máy tính và Internet

dụng nhất trên thế giới. Thêm vào đó là sự nổi bật trong việc xây dựng các ứng dụng dành cho Internet và Intranet, Java đã được lựa chọn cho các phần mềm dùng trong các thiết bị truyền thông mạng. Không có gì là ngạc nhiên cả khi dần âm thanh nổi mới của bạn và các thiết bị khác trong nhà được kết nối với nhau sử dụng công nghệ Java.

Bạn đọc nếu muốn tìm hiểu thêm về Java có thể tìm đọc quyển *Java How to Program, Fifth Edition* and *Advanced Java 2 Platform..*

1.16. Microsoft .NET

Tháng 6 năm 2000, Microsoft giới thiệu công nghệ **.NET**, một cách nhìn mới rộng hơn cho việc tích hợp Internet và Web vào việc phát triển, phân phối và sử dụng của phần mềm. Sẽ tốt hơn nếu tập trung các nhà phát triển vào việc sử dụng một ngôn ngữ lập trình duy nhất, công nghệ **.NET** cho phép họ tạo ra các ứng dụng trên bất cứ ngôn ngữ nào tương thích với **.NET** (ví dụ: Jscript.NET, C#, Visual Basic.NET, Visual C++ .NET và nhiều ngôn ngữ khác). Một phần của công nghệ này là ASP.NET, cho phép lập trình viên tạo ra các ứng dụng theo mô hình khách/chủ, chuyên về cơ sở dữ liệu dành cho Web. Công nghệ **.NET** mở rộng ý tưởng về phần mềm tái sử dụng cho Internet bằng cách cho phép lập trình viên tập trung vào công việc của họ mà không cần phải thực hiện tất cả các thành phần của mỗi ứng dụng. Thay vào đó, các công ty có thể mua các dịch vụ Web (**Web services**), là các chương trình dành cho Web mà các tổ chức có thể tham gia phát triển để tăng tốc độ cho tiến trình phát triển ứng dụng. Chú ý rằng công nghệ **.NET** của Microsoft không phải là cách tiếp cận duy nhất đến các dịch vụ Web. Cũng có một vài hệ thống dựa trên Java để phát triển các dịch vụ Web.

Microsoft **.NET Framework** là hạt nhân của công nghệ **.NET**. Bộ khung này thực thi các ứng dụng và các dịch vụ Web, bao gồm một thư viện lớp (được gọi là *Framework Class Library* hay *FCL*) và cung cấp nhiều tính năng lập trình khác được dùng để xây dựng các ứng dụng **.NET**.

1.17. HTML động (Dynamic HTML)

HTML động là công cụ dùng để phát triển các ứng dụng hiệu năng cao dành cho Web mà phần lớn là thực hiện trên máy khách. HTML động làm cho trang Web trở nên sống động hơn bằng cách cung cấp những hiệu ứng đa phương tiện tuyệt vời bao gồm hoạt hình, âm nhạc và phim ảnh. Vậy chính xác HTML động là gì? Đó là một câu hỏi rất đáng chú ý, bởi vì nếu bạn dạo qua các cửa hàng máy tính hoặc các cửa hàng phần mềm trực tuyến, bạn sẽ không tìm thấy một sản phẩm nào có tên như thế được bán.

HTML động, có 2 phiên bản – của Microsoft và của Netscape – bao gồm một số công nghệ có sẵn và được gọi theo nhiều tên khác nhau. Ví dụ,

- *Microsoft Dynamic HTML* bao gồm *XHTML*, *JavaScript*, *Cascading Style Sheets*, mô hình sự kiện và đối tượng *HTML* động và các điều khiển *ActiveX* và các công nghệ liên quan khác.

Chương 1

Giới thiệu về máy tính và Internet

- *Netscape Dynamic HTML* cũng cung cấp những chức năng tương tự.

1.18. Tài nguyên trên Web (Web Resources)

Các bạn có thể tham khảo một số nguồn tài nguyên trên Web khác như:

- www.deitel.com

Hãy vào trang Web này để cập nhật và lấy tài nguyên của tất cả các thông tin được công bố của Deitel & Associates.

- netforbeginners.about.com:

Phản hướng dẫn Internet cho người mới bắt đầu, cung cấp các tài nguyên có giá trị cho việc tìm hiểu sâu về lịch sử và cách làm việc của Internet và World Wide Web.

- www.learnthenet.com/english/index.html

Learn the Net là một trang Web trình bày tổng quan về Internet, World Wide Web và các công nghệ bên dưới. Trang có nhiều thông tin thích hợp cho người mới học.

- www.w3.org

Trang Web của World Wide Web Consortium (W3C) cung cấp một sự mô tả toàn diện về công nghệ mạng. Với mỗi công nghệ Internet cùng với cách mà W3C xử lý, trang Web này cung cấp mô tả về công nghệ, lợi ích của người thiết kế, lịch sử của công nghệ và mục tiêu trong tương lai của W3C trong phát triển công nghệ.

- www.ukans.edu/cwis/units/coms2/class/intro/index.htm

Trang Web của Đại học Kansas cho một cái nhìn toàn diện về Internet và World Wide Web, với tài liệu trình chiếu cho mỗi chủ đề.

- members.tripod.com/~teachers/index.html

Trang Web này giới thiệu cho người mới bắt đầu về Internet và World Wide Web, nhằm cung cấp cho những người dùng sẽ lướt Web tại lớp học.

- www.ed.gov/pubs/OR/ConsumerGuides/internet.html

Trang hướng dẫn của Bộ giáo dục Mỹ cung cấp một bản hướng dẫn ngắn gọn sáng sủa về cấu trúc, nội dung, tính tương thích của Internet và World Wide Web.

1.19. Kết chương

Trong chương này chúng ta đã có một cái nhìn tổng quan về thế giới máy tính và Internet. Bạn đọc có thể tham khảo thêm các tài liệu trên Internet. Trong chương sau chúng ta sẽ đi sâu vào tìm hiểu về trình duyệt Web Internet Explorer 6 và các chức năng cơ bản của nó.

Chương 2

Microsoft Internet Explorer 6

Nội dung của chương:

- Làm quen với các chức năng của trình duyệt Web Internet Explorer 6 (IE6).
- Sử dụng IE6 để tìm kiếm thông tin trên World Wide Web.
- Sử dụng Internet như một công cụ tin học.
- Làm quen với e-mail.
- Tin nhắn nhanh.
- Giới thiệu một số trình duyệt Web khác.

2.1. Giới thiệu về trình duyệt Web Internet Explorer 6

Internet là một môi trường giao tiếp và truyền thông trên toàn thế giới. Nhu cầu phổ biến và trao đổi thông tin đã dẫn tới việc thúc đẩy sự phát triển của Web và các công nghệ đi kèm. *Trình duyệt Web* là chương trình phần mềm cho phép người dùng truy cập vào các nội dung đa phương tiện phong phú của Web. Ngày nay hàng triệu người sử dụng trình duyệt Web để truy cập vào khối lượng thông tin khổng lồ được cung cấp trên mạng. Trong đó hai trình duyệt Web thông dụng nhất là *Internet Explorer* và *Nescape*.

Nội dung chương này sẽ tập trung giới thiệu các chức năng cơ bản của *Internet Explorer 6* để xem, trao đổi và truyền các thông tin như hình ảnh, tin nhắn và văn bản qua Internet.

2.2. Kết nối Internet

Một máy tính độc lập thôi thì sẽ không đủ để truy cập Internet. Ngoài trình duyệt Web, máy tính cần các phần cứng đặc biệt và một kết nối đến một nhà cung cấp dịch vụ Internet (*Internet Service Provider - ISP*) để có thể truy cập đến các trang Web trên Internet. Trong mục này chúng ta sẽ mô tả các thành phần cần thiết để có thể truy cập Internet.

Chương 2

Microsoft Internet Explorer 6

- Đầu tiên, máy tính *phải* có một **modem** hoặc một **card mạng** (*network card*). *Modem* là một phần cứng cho phép máy tính kết nối mạng thực hiện chuyển dữ liệu sang tín hiệu điện thoại và truyền qua đường điện thoại. *Card* mạng, còn gọi là *card* giao tiếp mạng (*network interface card - NIC*) là phần cứng cho phép một máy tính kết nối Internet thông qua một mạng hoặc một kết nối Internet tốc độ cao như mạng cục bộ (*local area network - LAN*), *cable modem* hoặc *Digital Subscriber Line (DSL)*.
- Sau khi chắc chắn rằng máy tính có một modem hoặc một card mạng (hầu hết máy tính có một hoặc hai thiết bị đó), bước tiếp theo là đăng ký với một nhà cung cấp dịch vụ Internet (ISP). Các nhà cung cấp dịch vụ Internet sẽ kết nối máy tính của bạn với mạng Internet. Các máy tính kết nối với nhà cung cấp dịch vụ Internet qua modem, đường điện thoại hoặc thông qua NIC sử dụng LAN, DSL hoặc *cable modem*.

Độ rộng băng thông và giá cả là hai vấn đề cần quan tâm khi quyết định lựa chọn nhà cung cấp dịch vụ Internet. Độ rộng băng thông là lượng dữ liệu có thể truyền qua đường truyền trong một khoảng thời gian nhất định. Sự khác nhau giữa các nhà cung cấp dịch vụ Internet là ở tốc độ kết nối cao, gọi là kết nối băng thông rộng, bao gồm DSL, *cable modem*, mạng tích hợp dịch vụ số (*Integrated Services Digital Network - ISDN*) và kết nối qua điện thoại (*dial-up*) với tốc độ thấp hơn, mỗi loại có một mức giá và độ rộng băng thông khác nhau đối với người dùng. Băng thông rộng là một loại dịch vụ Internet có độ rộng băng thông lớn, hầu hết được cung cấp cho người dùng tại nhà bởi các công ty điện thoại và truyền hình.

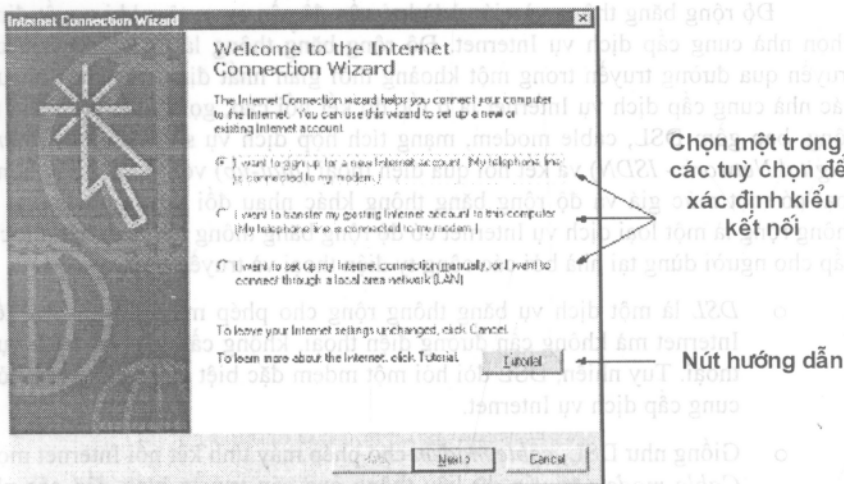
- DSL là một dịch vụ băng thông rộng cho phép máy tính có thể kết nối Internet mà không cần đường điện thoại, không cần gán với dịch vụ điện thoại. Tuy nhiên, DSL đòi hỏi một modem đặc biệt được cung cấp bởi nhà cung cấp dịch vụ Internet.
- Giống như DSL, *cable modem* cho phép máy tính kết nối Internet mọi lúc. *Cable modem* truyền dữ liệu thông qua cáp truyền hình đến các nhà và công ty. Không giống như DSL, độ rộng băng thông đường truyền được chia sẻ cho mọi người dùng. Sự chia sẻ này làm giảm độ rộng băng thông cần thiết cho mỗi người khi có nhiều người sử dụng hệ thống cùng lúc.
- ISDN cung cấp dịch vụ Internet qua đường điện thoại chuẩn hoặc số. ISDN đòi hỏi phần cứng đặc biệt gọi là *terminal adaptor (TA)* để thu từ các nhà cung cấp dịch vụ Internet. Dịch vụ quay số sử dụng đường điện thoại có sẵn. Nếu máy tính kết nối Internet, thông thường người dùng không thể nhận các cuộc gọi. Nếu cuộc gọi được nhận, kết nối Internet sẽ bị ngắt. Để ngăn chặn điều này, người dùng thường cài đặt thêm một đường điện thoại chuyên dùng cho dịch vụ Internet.

Chương 2

Microsoft Internet Explorer 6

■ Khi một kết nối mạng được thiết lập, chương trình *Internet Connection Wizard* (ICW) trên Windows 98 và 2000 hoặc *New Connection Wizard* trên Windows XP được dùng để thiết lập cấu hình cho máy tính để kết nối Internet. Bạn có thể khởi động các chương trình này từ menu *Start*.

- Đầu tiên chọn *Accessories* trên menu *Programs* (hoặc *All Programs* trên Windows XP), sau đó chọn *Communications* và *Internet Connection Wizard* (hoặc *New Connection Wizard* trong Windows XP).
- Nhập các thông tin kết nối do nhà cung cấp dịch vụ Internet cung cấp theo những chỉ dẫn của hộp thoại *Internet Connection Wizard* (hình 2.1).
- Kích vào *Tutorial* (chỉ với Windows 2000) để tìm hiểu thêm về Internet và các chức năng của nó. Khi *Internet Connection Wizard* hoàn thành, máy tính của bạn có thể kết nối với Internet.



Hình 2.1. Internet Connection Wizard trợ giúp cài đặt cấu hình truy cập Internet.

2.3. Đặc điểm của Internet Explorer 6

Một trình duyệt Web là một phần mềm cho phép người dùng xem các loại file của Internet trong một môi trường tương tác. Hình 2.2, hiển thị trang chủ của Deitel trên trình duyệt Internet Explorer 6.

Mỗi trang Web trên Internet có một URL duy nhất. URL mô tả địa chỉ của trang Web đang hiển thị trên màn hình trình duyệt. Các URL thường được bắt đầu bằng *http://*, viết tắt của *Hypertext Transfer Protocol* (HTTP) – giao thức truyền siêu văn bản. Giao thức

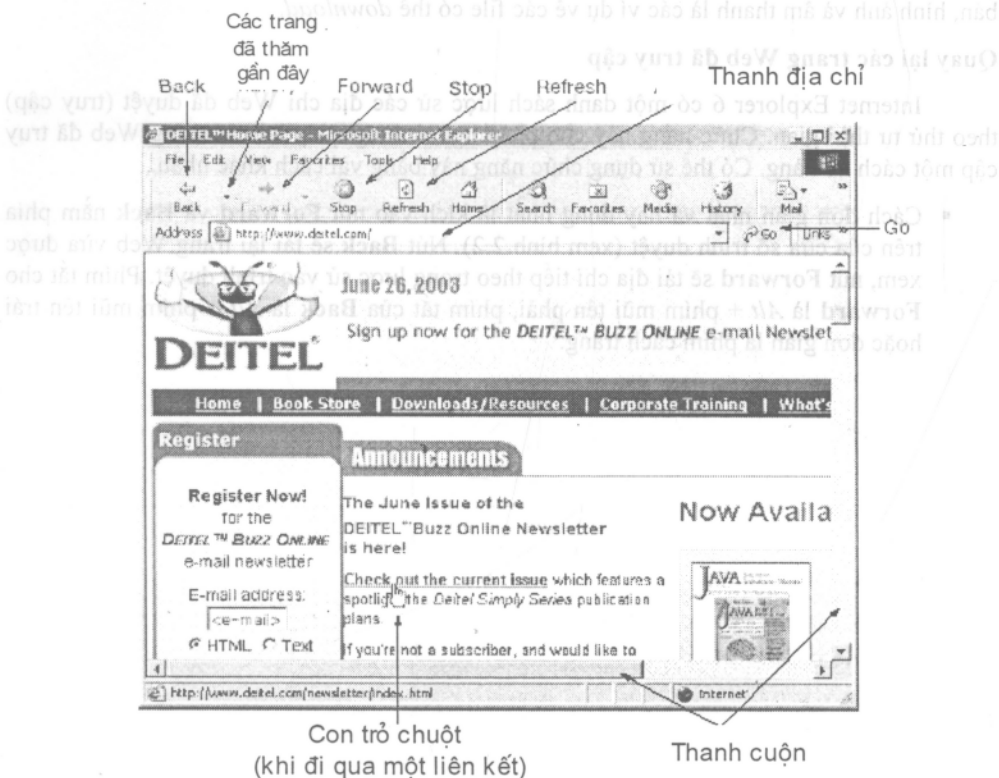
Chương 2

Microsoft Internet Explorer 6

này là chuẩn (một tập các quy tắc truyền thông) công nghiệp cho việc truyền các văn bản Web trên Internet. URL của các trang Web xử lý các thông tin cá nhân như mã thẻ tín dụng thường được bắt đầu bằng `https://`, viết tắt của *Secure Hypertext Transfer Protocol (HTTPS)*, chuẩn truyền dữ liệu mã hoá trên Internet. Trong hình vẽ URL của trang chủ Deitel là `http://www.deitel.com/` nằm trên thanh địa chỉ.

Mở một trang Web

Có một vài cách để mở các địa chỉ URL khác nhau. Bạn có thể kích và trường địa chỉ và gõ URL của trang Web, sau đó nhấn *Enter* hoặc kích vào nút *Go* để đọc trang Web tại địa chỉ đó. Ví dụ để vào trang Web của Yahoo!, bạn gõ `www.yahoo.com` vào thanh địa chỉ và nhấn *Enter*. Internet Explorer 6 sẽ tự động thêm tiền tố `http://` vào tên của trang Web vì HTTP là giao thức mặc định cho Web.



Hình 2.2. Trang Web Deitel.

Chương 2

Microsoft Internet Explorer 6

Một cách khác để vào trang Web là thông qua các thành phần đặc biệt của trang Web gọi là các siêu liên kết (*hyperlink*), khi kích vào liên kết trình duyệt sẽ tải trang Web về. Cả biểu tượng và văn bản đều có thể là siêu liên kết. Khi con trỏ chuột di chuyển qua một siêu liên kết, con trỏ mũi tên mặc định sẽ chuyển thành hình một bàn tay với ngón tay trỏ hướng lên trên. Dòng văn bản siêu liên kết thường xuất hiện gạch chân hoặc đổi màu khác so với các phần văn bản bình thường. Ban đầu được sử dụng như một công cụ phổ biến cho nghiên cứu khoa học, siêu liên kết tạo ra hiệu ứng của “mạng nhện” (Web). Siêu liên kết có thể tham chiếu đến các trang Web, địa chỉ e-mail và các file khác. Nếu đó là liên kết đến một địa chỉ e-mail, kích vào liên kết để chạy chương trình e-mail mặc định và mở cửa sổ viết thư với địa chỉ của e-mail đó trong mục địa chỉ người nhận thư. Nếu siêu liên kết tham chiếu đến một file mà trình duyệt không thể hiển thị nó, trình duyệt sẽ chuẩn bị tải (*download*) file này xuống bằng cách hiển thị thông báo cho người dùng. Khi một file đã được *download*, nó sẽ được copy vào máy tính của người dùng. Các file chương trình, văn bản, hình ảnh và âm thanh là các ví dụ về các file có thể *download*.

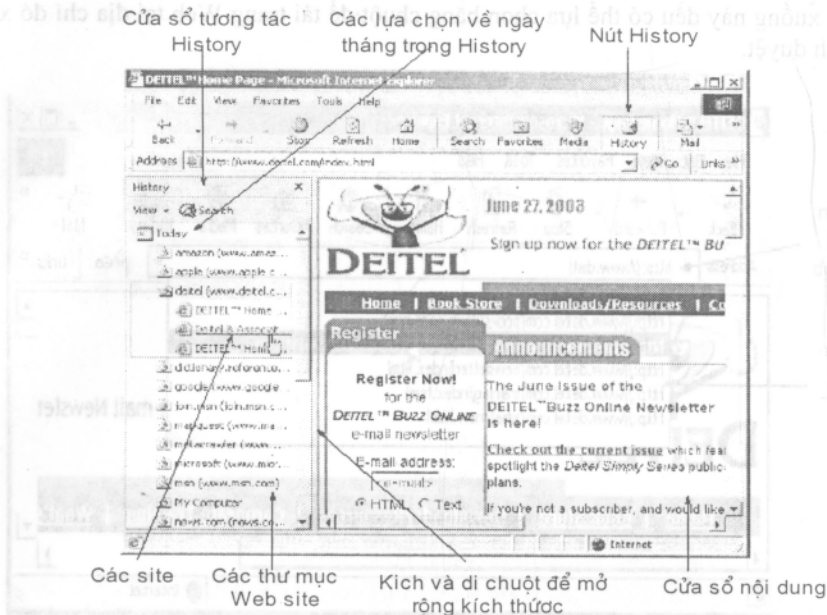
Quay lại các trang Web đã truy cập

Internet Explorer 6 có một danh sách lược sử các địa chỉ Web đã duyệt (truy cập) theo thứ tự thời gian. Chức năng này cho phép người dùng quay lại các trang Web đã truy cập một cách dễ dàng. Có thể sử dụng chức năng này bằng vài cách khác nhau.

- Cách đơn giản nhất và hay dùng nhất là kích vào nút **Forward** và **Back** nằm phía trên của cửa sổ trình duyệt (xem hình 2.2). Nút **Back** sẽ tải lại trang Web vừa được xem, nút **Forward** sẽ tải địa chỉ tiếp theo trong lược sử vào trình duyệt. Phím tắt cho **Forward** là **Alt +** phím mũi tên phải, phím tắt của **Back** là **Alt +** phím mũi tên trái hoặc đơn giản là phím cách trắng.

Chương 2

Microsoft Internet Explorer 6



Hình 2.3. Danh sách các trang Web trong History.

- Khi người dùng muốn xem các trang Web được cập nhật thường xuyên có thể kích vào nút **Refresh** để tải phiên bản mới nhất hiện tại. Nếu một trang Web tải không đúng hoặc chậm, kích vào nút **Stop** hoặc nhấn **Esc** để dừng việc tải trang Web. Người dùng có thể xem 9 trang Web vừa được duyệt bằng cách kích vào nút mũi tên trỏ xuống ngay bên phải của các nút **Back** và **Forward**.
- Người dùng cũng có thể duyệt một trong các trang Web đã truy cập bằng cách kích vào tiêu đề của trang trong danh sách trải xuống. Kích vào nút **History** (hình 2.3) để tách cửa sổ trình duyệt thành hai phần. Cửa sổ **History** (ở bên trái) và nội dung của nó. Cửa sổ **History** mặc định liệt kê các URL đã duyệt trong 20 ngày gần nhất.

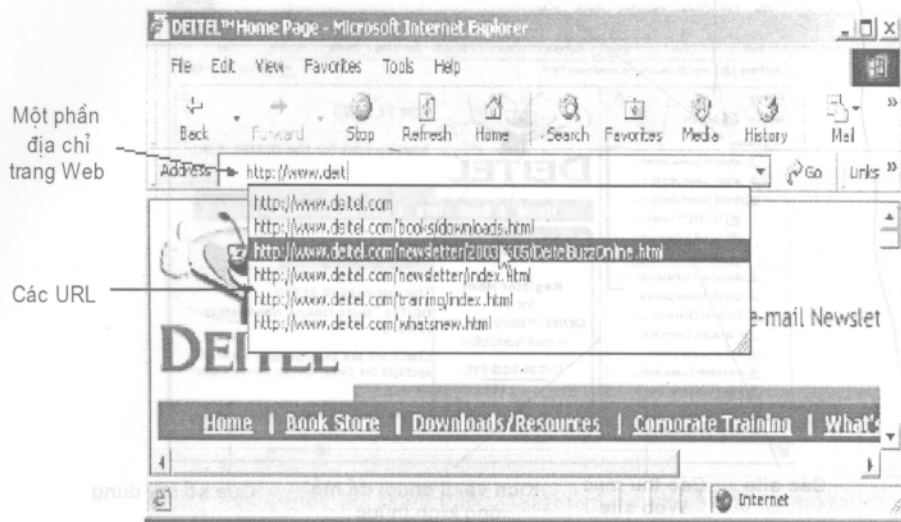
Cửa sổ **History** chứa các mức tiêu đề theo thứ tự. Với mỗi khung tiêu đề thời gian (ví dụ *Today*) là các tên trang Web theo thứ tự ABC. Cửa sổ này thường được dùng để tìm các trang Web đã đến thăm mà không nhớ chính xác địa chỉ. Chọn một URL trong cửa sổ **History** để tải trang Web vào cửa sổ nội dung.

Cửa sổ **History** có thể thay đổi kích thước bằng cách giữ chuột và kéo thanh thẳng đứng chia nó với cửa sổ nội dung. Các địa chỉ Web đã truy cập có thể hiển thị trong một danh sách trải xuống khi người dùng nhập địa chỉ URL vào thanh địa chỉ. Chức năng này gọi là **AutoComplete** (hình 2.4). Bất kỳ địa chỉ nào trong danh sách

Chương 2

Microsoft Internet Explorer 6

trải xuống này đều có thể lựa chọn bằng chuột để tải trang Web tại địa chỉ đó xuống trình duyệt.



Hình 2.4. Chức năng AutoComplete.

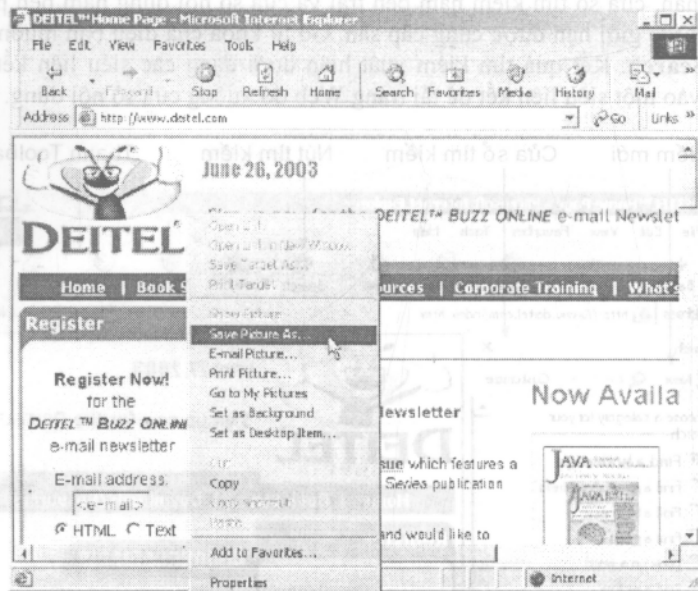
Lưu một trang Web

Đối với một số người dùng, ví dụ những người dùng kết nối quay số, việc duy trì một kết nối trong một thời gian dài là không có lợi. Vì lý do này, các trang Web có thể được lưu lại trực tiếp vào ổ cứng máy tính để xem ở chế độ off-line (duyệt Web trong khi không kết nối với Internet).

Chọn **Save As...** từ menu **File** trên phần đầu của cửa sổ trình duyệt để lưu một trang Web và các thành phần của nó. Các hình ảnh riêng lẻ trong trang Web cũng có thể lưu lại bằng cách kích chuột phải vào hình ảnh và chọn **Save Picture As...** từ menu ngữ cảnh hiện ra (hình 2.5).

Chương 2

Microsoft Internet Explorer 6



Hình 2.5. Lưu lại một hình ảnh từ trang Web.

2.4. Tìm kiếm trên Internet

Internet cung cấp một số lượng lớn thông tin về hầu hết các chủ đề. Để giúp người dùng tìm kiếm thông tin, nhiều trang Web cung cấp các công cụ tìm kiếm (**search engines**) để tìm kiếm trên Internet và tạo ra các chỉ dẫn chứa thông tin về nội dung trang Web tìm được.

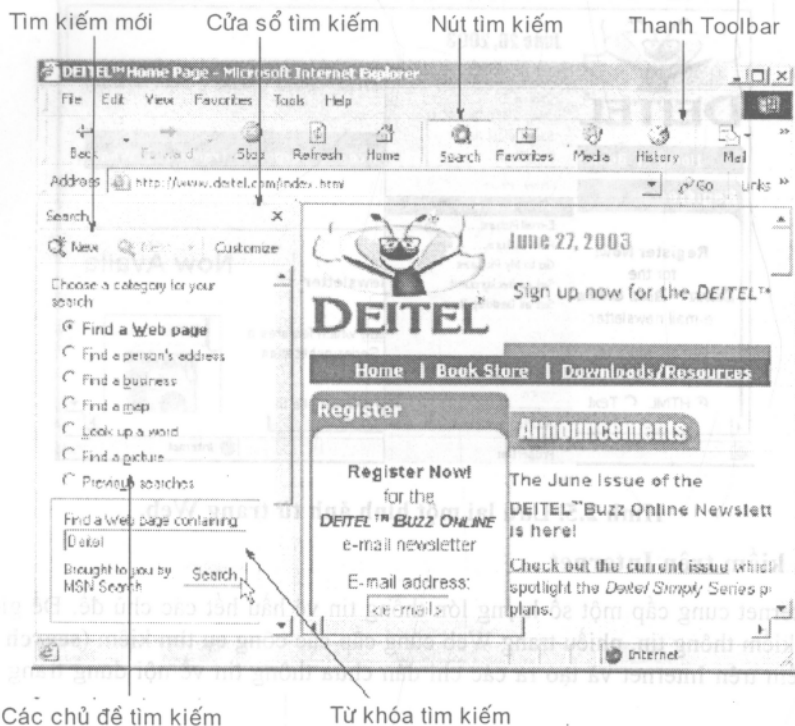
Các công cụ tìm kiếm như Google (www.google.com), Yahoo! (www.yahoo.com), AlltheWeb (www.alltheweb.com), AltaVista (www.altavista.com) và Lycos (www.lycos.com) chứa thông tin trong các kho dữ liệu gọi là cơ sở dữ liệu (**databases**) nhằm tìm kiếm thuận lợi hơn. Khi người dùng nhập một từ hoặc cụm từ, công cụ tìm kiếm sẽ trả lại một danh sách các siêu liên kết đến các trang thỏa mãn tiêu chuẩn tìm kiếm. Mỗi trang công cụ tìm kiếm có tiêu chuẩn khác nhau để hạn chế vùng tìm kiếm, như ngày xuất bản, ngôn ngữ. Sử dụng nhiều công cụ tìm kiếm có thể cung cấp kết quả tốt nhất và nhanh nhất.

Các trang như MSN (www.msn.com) và MetaCrawler (www.metacrawler.com) sử dụng công cụ **metasearch** không cần duy trì cơ sở dữ liệu. Thay vào đó chúng gửi tiêu chuẩn tìm kiếm đến các công cụ tìm kiếm khác và tập hợp kết quả trả về. Internet Explorer 6 được cài đặt sẵn công cụ **metasearch** và có thể sử dụng bằng cách kích vào nút **Search** trên thanh công cụ (Hình 2.6). Giống như chức năng lược sử, cửa sổ trình duyệt sẽ được

Chương 2

Microsoft Internet Explorer 6

chia thành 2 phần, cửa sổ tìm kiếm nằm bên trái và cửa sổ nội dung nằm bên phải. Một vài loại tìm kiếm được giới hạn được cung cấp sẵn. Gõ từ khóa của điều bạn muốn tìm kiếm và kích vào nút **Search**. Kết quả tìm kiếm xuất hiện dưới dạng các siêu liên kết trên cửa sổ **Search**. Kích vào một siêu liên kết để tải trang Web đó xuống cửa sổ nội dung.



Hình 2.6. Tìm kiếm trên Internet với Internet Explorer 6.

2.5. Trợ giúp và hướng dẫn trực tuyến

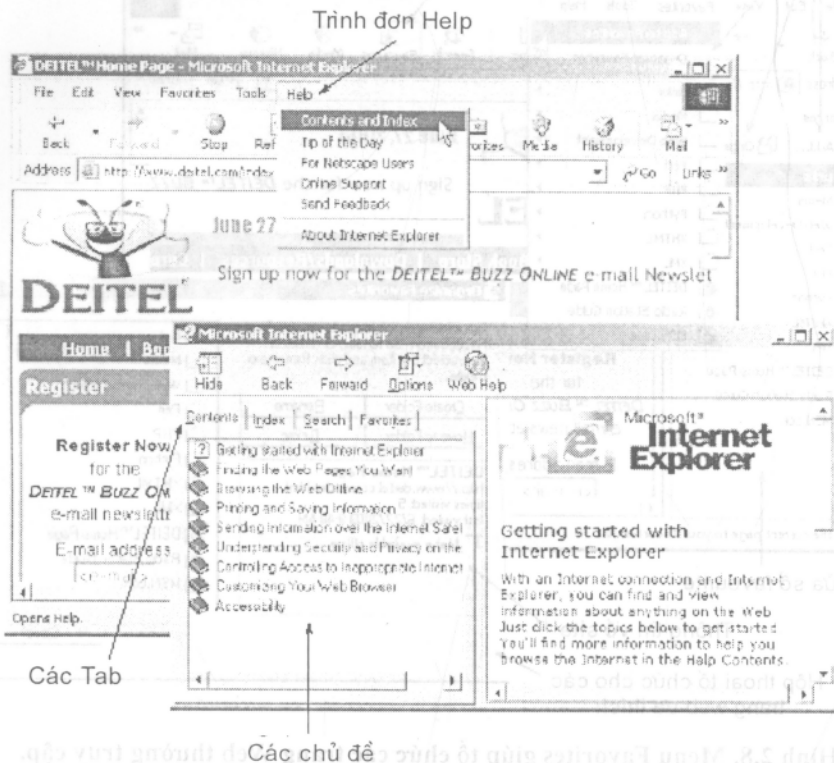
Các trình duyệt Web là một phần phức tạp của phần mềm với nhiều chức năng. Mặc dù những nhà thiết kế trình duyệt đã cố gắng hết sức để tạo ra phần mềm thân thiện với người dùng, nhưng người dùng vẫn cần thời gian để tự làm quen với mỗi trình duyệt và các đặc điểm của nó. Câu trả lời cho các vấn đề thường được hỏi về việc sử dụng trình duyệt Web đã được kết hợp trong Internet Explorer 6. Các thông tin này có thể truy cập thông qua chức năng giúp đỡ có sẵn trên menu **Help** (Hình 2.7).

Nguồn thông tin tốt nhất để tìm kiếm sự giúp đỡ về một chức năng đặc biệt là mục **Contents and Index** truy cập thông qua menu **Help**. Khi **Contents and Index** được chọn, hộp thoại giúp đỡ của Microsoft Internet Explorer sẽ hiển thị. Trên tab **Contents** tổ chức

Chương 2

Microsoft Internet Explorer 6

các chủ đề trợ giúp theo loại, tab **Index** chứa danh sách theo thứ tự ABC các chủ đề. Tab **Search** cung cấp khả năng tìm kiếm các tài liệu trợ giúp. Tab **Favorites** cho phép người dùng lưu lại danh sách các chủ đề trợ giúp thường dùng.



Hình 2.7. Hộp thoại Help của Internet Explorer 6.

2.6. Lưu giữ địa chỉ những trang Web yêu thích của bạn

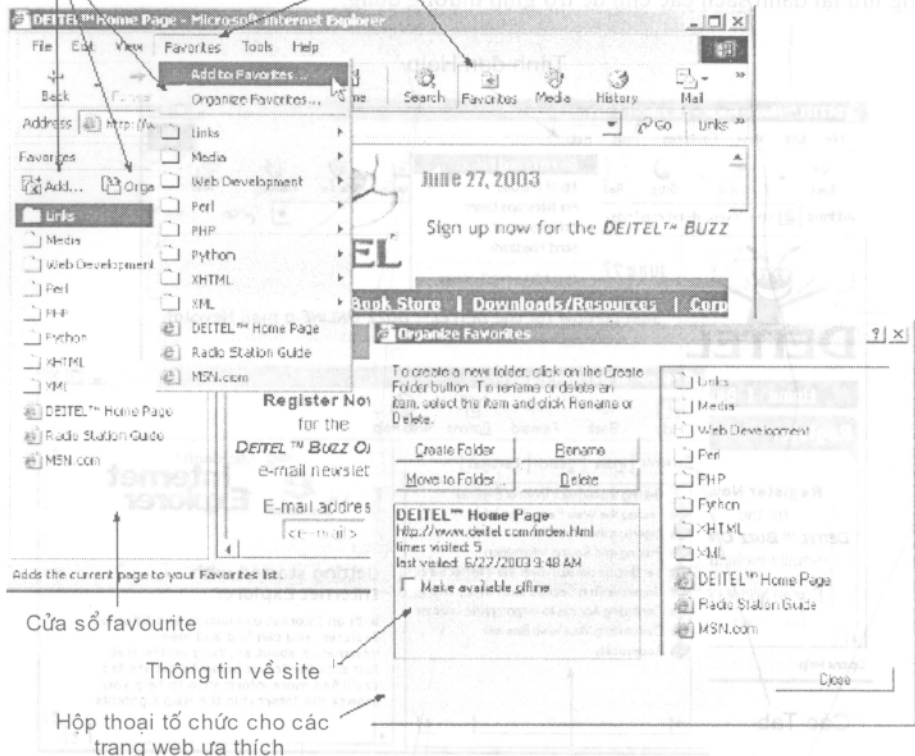
Những người dùng Web thường xuyên thăm một số trang nhất định và có thể muốn lưu lại URL và tiêu đề của chúng. Internet Explorer cung cấp một chức năng gọi là **Favorites** để đánh dấu địa chỉ của những trang đó (Hình 2.8).

Chương 2

Microsoft Internet Explorer 6

Thêm/Tổ chức lưu địa chỉ các trang web ưa thích

Chức năng favourite



Hình 2.8. Menu Favorites giúp tổ chức các trang Web thường truy cập.

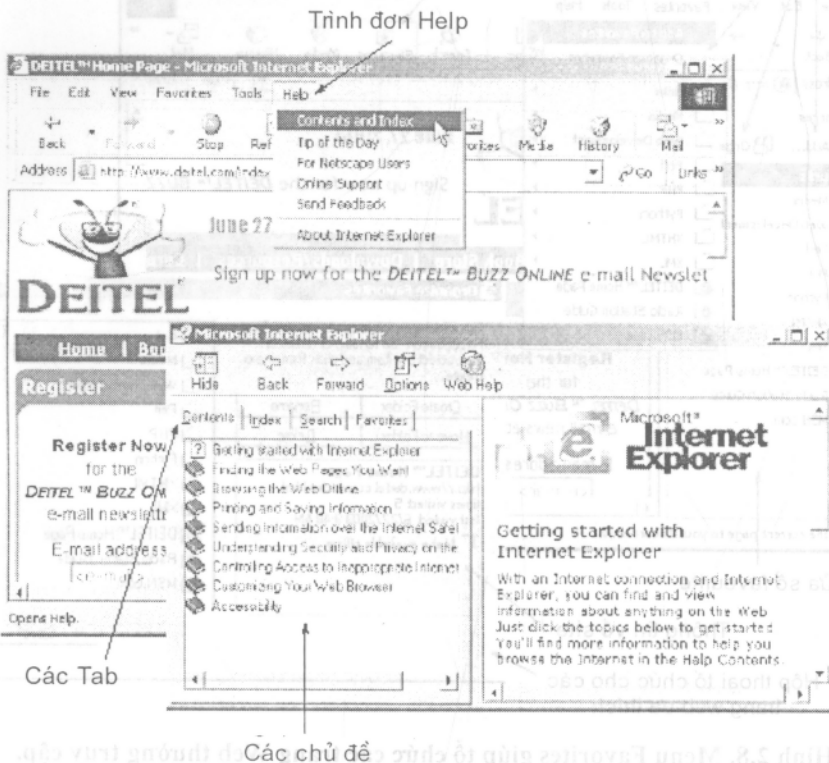
Địa chỉ của bất kỳ trang Web nào cũng có thể thêm vào danh sách trang Web yêu thích của bạn bằng cách sử dụng lệnh **Add to Favorites** của menu **Favorites**. Một cửa sổ **Favorites** cũng có thể truy cập bằng cách kích vào biểu tượng **Favorites** trên thanh công cụ. Bạn có thể truy cập trang Web yêu thích của mình vào bất kỳ khi nào bằng cách chọn chúng bằng chuột từ menu **Favorites**.

Các trang Web yêu thích có thể sắp xếp và nhóm lại vào các thư mục trong hộp thoại **Organize Favorites** (hiển thị khi **Organize Favorites** được chọn từ menu **Favorites**). Các thư mục này xuất hiện như một menu con trong menu **Favorites**. Hộp thoại **Organize Favorites** cũng cho phép người dùng đổi tên, xóa và di chuyển các địa chỉ giữa các thư mục. Với mỗi mục yêu thích, hộp thoại **Organize Favorites** hiển thị thông tin về tần suất

Chương 2

Microsoft Internet Explorer 6

các chủ đề trợ giúp theo loại, tab **Index** chứa danh sách theo thứ tự ABC các chủ đề. Tab **Search** cung cấp khả năng tìm kiếm các tài liệu trợ giúp. Tab **Favorites** cho phép người dùng lưu lại danh sách các chủ đề trợ giúp thường dùng.



Hình 2.7. Hộp thoại Help của Internet Explorer 6.

2.6. Lưu giữ địa chỉ những trang Web yêu thích của bạn

Những người dùng Web thường xuyên thăm một số trang nhất định và có thể muốn lưu lại URL và tiêu đề của chúng. Internet Explorer cung cấp một chức năng gọi là **Favorites** để đánh dấu địa chỉ của những trang đó (Hình 2.8).

Chương 2

Microsoft Internet Explorer 6

truy cập trang Web đó. Các mục yêu thích cũng có thể được lưu lại để dùng *off-line* bằng cách đánh dấu chọn vào mục **Make available offline** trong hộp thoại **Organize Favorites**.

2.7. Giao thức truyền file (File Transfer Protocol - FTP)

Như đã đề cập, các tệp trên Internet có thể được sao chép vào ổ cứng của máy tính qua một tiến trình gọi là tải xuống (**download**). Mục này sẽ giới thiệu về các kiểu văn bản thường được tải về từ Internet và kỹ thuật để tải chúng.

Loại file thường được tải từ mạng về nhất là các ứng dụng (tức là các phần mềm thực hiện các chức năng đặc biệt như xử lý văn bản) và các **plug-ins**. *Plug-ins* là các phần chuyên dụng của phần mềm nhằm mở rộng ứng dụng bằng cách cung cấp thêm các chức năng mới. Một ví dụ về một plug-in của Internet Explorer 6 là *Acrobat Reader* của *Adobe* (www.adobe.com) cho phép người dùng xem các tài liệu PDF (*Portable Document Format*), nếu không thì các tài liệu đó không thể hiển thị trên trình duyệt. Một *plug-in* thông dụng khác cho phép trình duyệt hiển thị nội dung *Mucromedia Shockwave* để thêm các hiệu ứng âm thanh, video và hoạt hình cho trang Web. Để xem các trang có *Shockwave*, hãy tải *plug-in* để chơi *Shockwave* tại địa chỉ <http://sdc.shockwave.com/shockwave/download/download.cgi>. Thông thường các trình duyệt sẽ thông báo cho người dùng để tải *plug-in* về khi cần. Các *plug-in* cũng có thể tải về từ CNET (www.download.com). Trang này có một lượng dữ liệu lớn, được đánh chỉ mục về các chương trình *plug-in* có thể tải về.

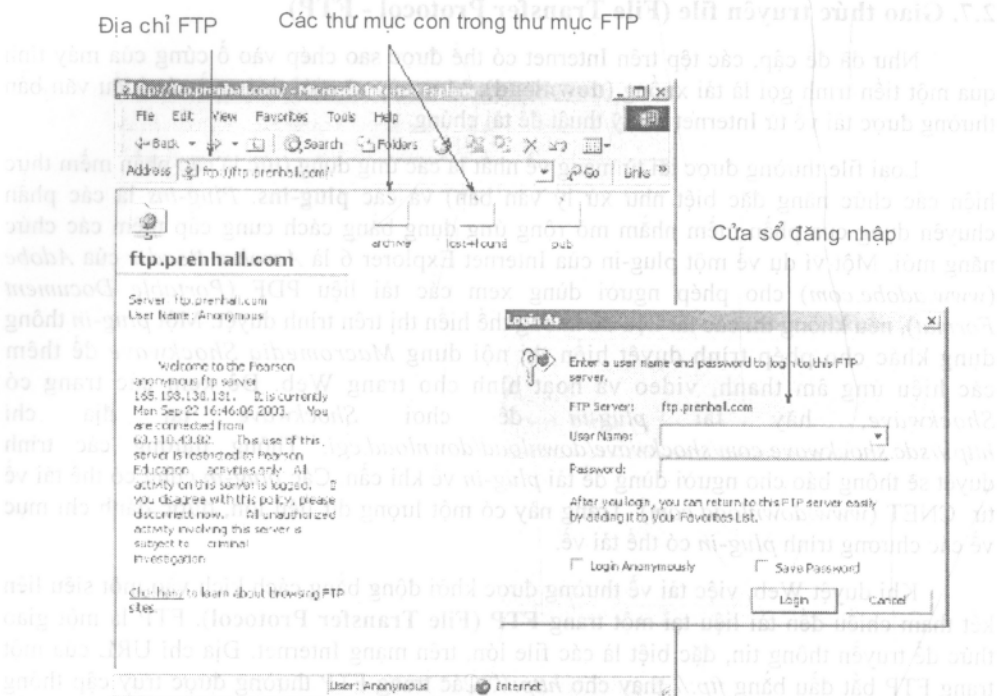
Khi duyệt Web, việc tải về thường được khởi động bằng cách kích vào một siêu liên kết tham chiếu đến tài liệu tại một trang **FTP (File Transfer Protocol)**. FTP là một giao thức để truyền thông tin, đặc biệt là các file lớn, trên mạng Internet. Địa chỉ URL của một trang FTP bắt đầu bằng *fip://* thay cho *http://*. Các trang FTP thường được truy cập thông qua các siêu liên kết, nhưng cũng có thể truy cập bằng một phần mềm hỗ trợ FTP. Phần mềm này có thể có hoặc không dùng một trình duyệt. **Ipswitch WS_FTP™ LE** là một ứng dụng khách FTP độc lập thông dụng, với các chức năng nằm ngoài trình duyệt. Nó miễn phí cho người dùng và có thể tải về tại địa chỉ www.fipplanet.com. Khi Internet Explorer 6 trở đến địa chỉ URL của một trang FTP, nội dung của thư mục trên trang này xuất hiện phía bên phải của cửa sổ và thông tin FTP nằm bên trái (Hình 2.9).

Hai kiểu biểu tượng có thể xuất hiện trong một thư mục: các file và các thư mục. Các file được tải về bằng cách kích chuột phải vào biểu tượng, chọn **Copy to Folder...** và xác định nơi sẽ lưu file đó. Khi một người dùng truy cập một trang FTP, Internet Explorer 6 gửi địa chỉ e-mail và tên của người dùng (mặc định được đặt là *anonymous*) đến trang này. Quy trình này xảy ra cho phép truy cập FTP giấu tên trên trang FTP và tại các phạm vi cho phép mọi người dùng đều được truy cập. Nhiều trang FTP trên Internet chứa các thư mục giới hạn truy cập, chỉ những người dùng có tên và mật khẩu cho phép là được phép truy cập các thư mục này. Khi một người dùng thử truy cập vào một thư mục FTP bị giới hạn, một hộp

Chương 2

Microsoft Internet Explorer 6

thoại **Login As** như trong hình 2.9 sẽ hiển thị, nhắc nhở người dùng đăng nhập (log-in) vào hệ thống.



Hình 2.9. Truy cập trang FTP trong Internet Explorer 6.

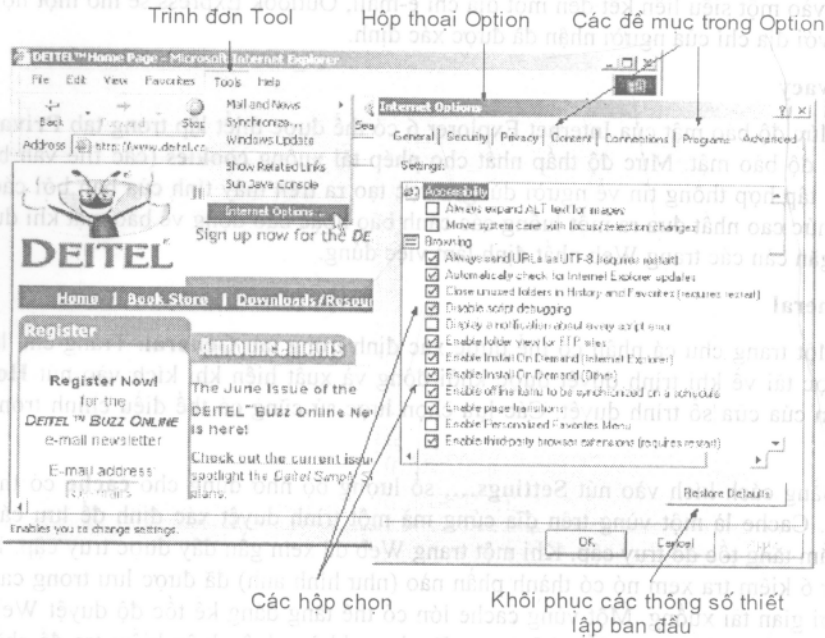
Quá trình truyền một file từ máy cục bộ đến một nơi khác trên Internet được gọi là tải lên (**uploading**) và có thể thực hiện sử dụng giao thức FTP. Để đặt thông tin vào một trang Web, các file phải được tải lên một máy chủ FTP bị giới hạn truy cập đặc biệt (được các nhà cung cấp dịch vụ Internet cung cấp). Tiến trình bao gồm cả tải file lên một thư mục trên trang FTP được thực hiện trên mạng.

2.8. Tuỳ biến các thiết lập cho trình duyệt

Internet Explorer 6 có nhiều thiết lập để quyết định các trang được hiển thị như thế nào, mức độ bảo mật nào được áp dụng và các kết quả được trình bày như thế nào. Hầu hết những thiết lập này nằm trong hộp thoại **Internet Options** (Hình 2.10).

Chương 2

Microsoft Internet Explorer 6



Hình 2.10. Hộp thoại Internet Options của Internet Explorer 6.

Tab Advanced

Các thiết lập mặc định thường đầy đủ cho việc duyệt thông thường, nhưng những thiết lập này có thể tùy biến để phù hợp với sở thích của mỗi người dùng. Lựa chọn một số trong nhiều tùy chọn quan trọng là tùy thuộc vào kinh nghiệm duyệt Web của bạn. Nếu bạn duyệt Web với một kết nối tốc độ chậm, thời gian tải trang Web về có thể giảm bằng cách bỏ chọn thiết lập **Show Pictures**, nằm trong loại **Multimedia** trên tab **Advanced**. Tắt thiết lập này (tức là bỏ chọn nó) ngăn cản trình duyệt tải về các hình ảnh do các hình ảnh có thể đòi hỏi thời gian đáng kể để tải về, vì vậy việc thay đổi này có thể tiết kiệm thời gian cho việc duyệt. *Chú ý:* Nhiều trang Web dựa vào hình ảnh để hiển thị đúng. Bỏ chọn thiết lập **Show Pictures** có thể ảnh hưởng đến chức năng của trang Web.

Tab Programs

Các chương trình mặc định dùng cho các thủ tục thông thường trên Internet như gửi thư điện tử (e-mail) được đặt trong tab **Programs**. Các thiết lập này cho phép người dùng định rõ chương trình mà Internet Explorer 6 sẽ dùng khi cần trong quá trình duyệt Web. Ví dụ nếu Microsoft Outlook Express được chỉ định là chương trình gửi e-mail mặc định, mỗi

Chương 2

Microsoft Internet Explorer 6

khi kích vào một siêu liên kết đến một địa chỉ e-mail, Outlook Express sẽ mở một hộp thoại viết thư với địa chỉ của người nhận đã được xác định.

Tab Privacy

Mức độ bảo mật của Internet Explorer 6 có thể được thiết lập trong tab **Privacy**. Có bốn mức độ bảo mật. Mức độ thấp nhất cho phép tải xuống **cookies** (các file văn bản lưu giữ hoặc tập hợp thông tin về người dùng, được tạo ra trên máy tính của bạn bởi các trang Web), mức cao nhất đưa ra một luồng các cảnh báo hoặc báo động về bảo mật khi duyệt và có thể ngăn cản các trang Web nhất định làm việc đúng.

Tab General

Một trang chủ cá nhân có thể được xác định trong tab **General**. Trang chủ là trang Web được tải về khi trình duyệt được khởi động và xuất hiện khi kích vào nút **Home** tại phần trên của cửa sổ trình duyệt. Các lựa chọn lược sử cũng có thể điều chỉnh trong phần này.

Bằng cách kích vào nút **Settings...**, số lượng bộ nhớ dùng cho **cache** có thể được thiết đặt. Cache là một vùng trên đĩa cứng mà một trình duyệt xác định để lưu các trang Web nhằm tăng tốc độ truy cập. Khi một trang Web đã xem gần đây được truy cập, Internet Explorer 6 kiểm tra xem nó có thành phần nào (như hình ảnh) đã được lưu trong cache, để giảm thời gian tải xuống. Một vùng cache lớn có thể tăng đáng kể tốc độ duyệt Web. Việc dùng cache thỉnh thoảng có lỗi vì Internet Explorer không luôn luôn kiểm tra để chắc chắn rằng trang đã lưu là phiên bản cuối cùng trên máy chủ. Kích vào nút **Refresh** trên thanh công cụ của cửa sổ trình duyệt để sửa lỗi này bằng cách bắt Internet Explorer truy lục phiên bản cuối cùng của trang Web.

Khi **Internet Options** được thiết lập, kích vào nút **Apply** và sau đó kích **OK**.

2.9. Thư điện tử

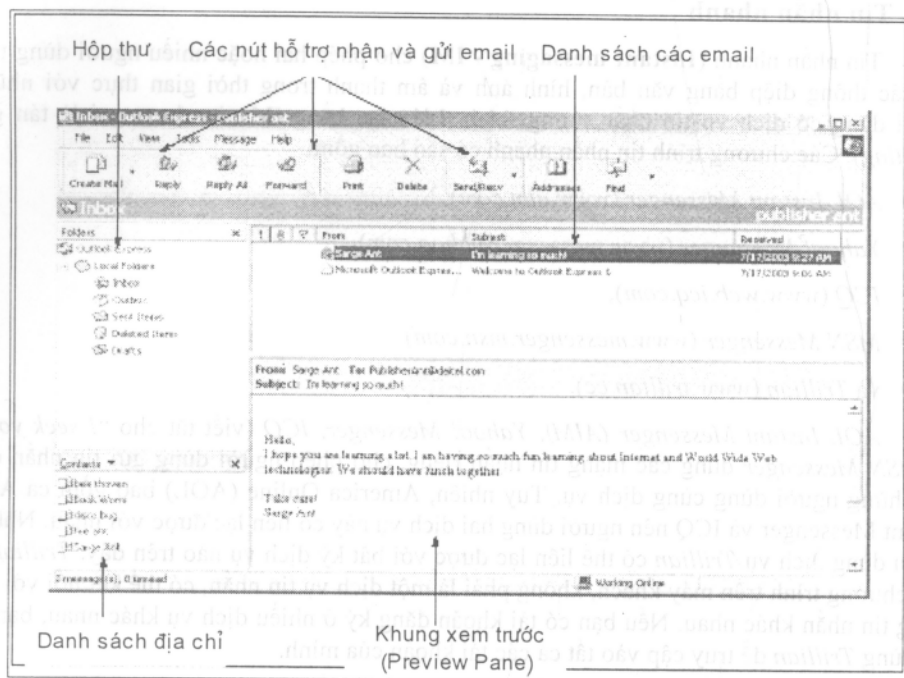
Thư điện tử (**electronic mail**, viết tắt là **e-mail** hay **email**) là một phương pháp để gửi và nhận các thông điệp và file được định dạng từ người khác thông qua Internet. Các file khác, được gọi là file đính kèm (**attachments**), có thể gửi kèm theo e-mail. Phụ thuộc vào đường truyền Internet, một bức thư điện tử có thể đến bất kỳ đâu trên thế giới chỉ sau vài giây. Các nhà cung cấp dịch vụ Internet cung cấp địa chỉ hộp thư điện tử theo dạng *<tên người dùng>@<tên miền>* (ví dụ *deitel@deitel.com*). Có nhiều chương trình thư tín như Outlook Express của Microsoft (Hình 2.11), Pegasus Mail và Eudora.

Các chương trình thư tín được thiết lập để truy cập vào một hộp thư. Các chương trình thư tín trên máy khách yêu cầu người dùng phải nhập tên của máy chủ gửi và nhận thư. Các tên này là địa chỉ của các máy chủ của nhà cung cấp dịch vụ Internet nhằm quản lý các bức thư gửi đến và gửi đi. Các địa chỉ máy chủ có thể nhận được từ người quản trị mạng hoặc nhà cung cấp dịch vụ Internet. Các chương trình thư điện tử cũng cho phép người dùng

Chương 2

Microsoft Internet Explorer 6

thiết lập các nhóm thông tin để nhận và gửi thư trong nhóm thảo luận về nhiều chủ đề khác nhau.



Hình 2.11. Giao diện chính của chương trình thư tin Outlook Express.

Post Office Protocol (POP) và **Internet Message Access Protocol (IMAP)** là hai giao thức chính để truy cập thư từ máy chủ. Dịch vụ POP lưu trữ thư điện tử trên một máy chủ từ xa cho đến khi người dùng lựa chọn để xem thư và tải chúng về máy cục bộ. Dịch vụ IMAP cũng lưu trữ thư trên một máy chủ từ xa nhưng cho phép người dùng thao tác với thư ngay trên máy chủ mà không cần phải tải thư về. Vì dịch vụ POP đòi hỏi người dùng tải thư về trước khi đọc, việc truy cập thư điện tử từ nhiều máy khác nhau sẽ rắc rối các thư trên nhiều máy truy cập. IMAP cho phép người dùng lưu trữ thư trên máy chủ, vì vậy người dùng có thể truy cập thư điện tử từ các máy khác nhau mà không làm phân tán chúng.

Để gửi thư giữa các máy chủ, hầu hết hệ thống thư điện tử dùng giao thức **Simple Mail Transfer Protocol (SMTP)**.

Có sẵn nhiều dịch vụ thư điện tử miễn phí trên mạng. Hotmail (www.hotmail.com) và Yahoo! Mail (www.mail.yahoo.com) là hai ví dụ. Các dịch vụ này sẽ thiết lập tài khoản

Chương 2

Microsoft Internet Explorer 6

cho người dùng. Các dịch vụ thư điện tử trên mạng cho phép người dùng truy cập thư của mình bằng bất kỳ trình duyệt Web nào.

2.10. Tin nhắn nhanh

Tin nhắn nhanh (**Instant messaging - IM**) cho phép hai hoặc nhiều người dùng trao đổi các thông điệp bằng văn bản, hình ảnh và âm thanh trong thời gian thực với những người dùng có dịch vụ tin nhắn tương thích. Liên lạc bằng IM còn được gọi là tán gẫu (*chatting*). Các chương trình tin nhắn nhanh có sẵn bao gồm:

- *AOL Instant Messenger* (www.aim.com),
- *Yahoo! Messenger* (www.messenger.yahoo.com),
- *ICQ* (www.web.icq.com),
- *MSN Messenger* (www.messenger.msn.com)
- và *Trillian* (www.trillian.cc).

AOL Instant Messenger (AIM), *Yahoo! Messenger*, *ICQ* (viết tắt cho “*I seek you*”) và *MSN Messenger* dùng các mạng tin nhắn riêng, cho phép người dùng gửi tin nhắn cho chỉ những người dùng cùng dịch vụ. Tuy nhiên, America Online (AOL) bao gồm cả AOL Instant Messenger và ICQ nên người dùng hai dịch vụ này có liên lạc được với nhau. Những người dùng dịch vụ *Trillian* có thể liên lạc được với bất kỳ dịch vụ nào trên đây. *Trillian* là một chương trình trên máy khách, không phải là một dịch vụ tin nhắn, có thể kết nối với các mạng tin nhắn khác nhau. Nếu bạn có tài khoản đăng ký ở nhiều dịch vụ khác nhau, bạn có thể dùng *Trillian* để truy cập vào tất cả các tài khoản của mình.

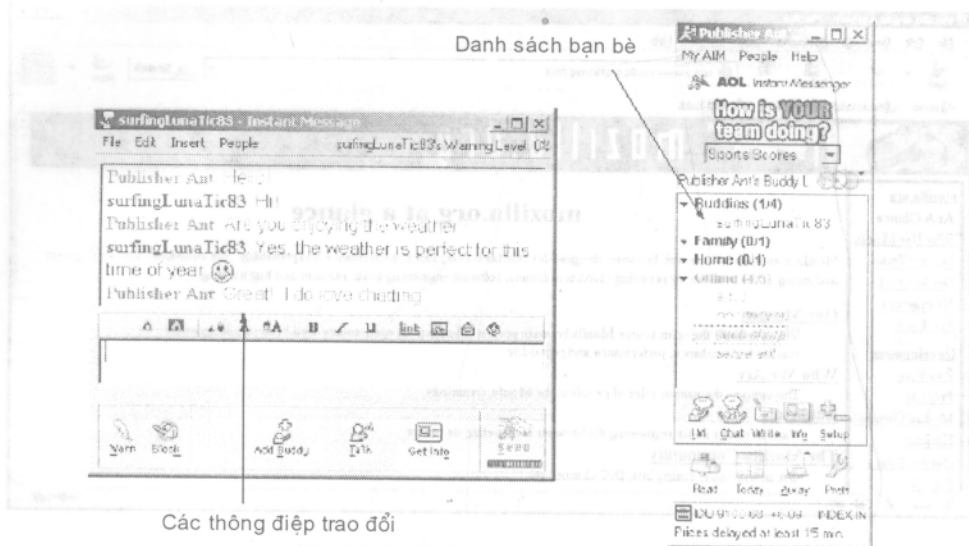
AOL Instant Messenger (Hình 2.12), *Yahoo! Messenger*, *ICQ* và *MSN Messenger* cũng cho phép người dùng thực hiện các cuộc thảo luận truyền hình và gửi tin nhắn đến các thiết bị di động (ví dụ điện thoại di động và PDA). Các chương trình tin nhắn nhanh khác như *Windows Messenger* và *NetMeeting* có nhiều công cụ cộng tác tiên tiến. Chúng cho phép một nhóm người dùng chia sẻ điều khiển của một chương trình đang chạy trên máy tính của một thành viên trong nhóm. Chúng cũng có chức năng **whiteboard**, cho phép chia sẻ các hiệu ứng trực quan (ví dụ như vẽ hình) với những người khác.

Để có thể gửi tin nhắn nhanh, người dùng phải tải về và cài đặt chương trình tin nhắn được chọn và mỗi người dùng phải tạo ra một **screen name** (hoặc dùng một tên có sẵn). Một *screen name* là một tên duy nhất mà một người dùng giữ khi liên lạc trong một mạng tin nhắn. Sau đó người dùng có thể đăng nhập vào một mạng tin nhắn.

Tin nhắn nhanh cho phép người dùng liên lạc nhanh và tiện lợi, thường được sử dụng trong các công ty nhằm tăng năng suất.

Chương 2

Microsoft Internet Explorer 6



Hình 2.12. Danh sách bạn bè và cửa sổ tin nhắn nhanh của AOL.

Mỗi chương trình tin nhắn cho phép người dùng lưu trữ một danh sách cá nhân các *screen name* gọi là danh sách bạn bè (**buddy list** hoặc **friend list**). Danh sách bạn bè trong AOL Instant Messenger cho phép người dùng tổ chức các tên vào các thư mục và hiển thị các thành viên đang ở trên mạng và có thể nói chuyện (**chat**) (Hình 2.12). Các dịch vụ IM cũng cho phép người dùng truyền file và tạo các phòng nói chuyện (**chat room**), nơi một nhóm các người dùng IM có thể gõ các tin nhắn cho mọi người trong nhóm xem và thảo luận.

2.11. Các trình duyệt Web khác

Ngoài Internet Explorer và Netscape còn có nhiều trình duyệt Web khác như Mozilla (www.mozilla.org), Opera (www.opera.com) và Amaya (www.w3.org/Amaya). Tất cả các trình duyệt này khác nhau về chức năng, tốc độ và đặc điểm. Chúng cũng sử dụng các công cụ xử lý HTML khác nhau. Các công cụ xử lý quyết định cách một trang Web hiển thị trên trình duyệt. Netscape và Mozilla (Hình 2.13) sử dụng công cụ xử lý Gecko. Opera và Internet Explorer có công cụ riêng. Opera là một trình duyệt đơn giản và nhanh, trong khi Internet Explorer và Mozilla cung cấp nhiều chức năng cho các lập trình viên. Amaya là một trình duyệt Web kết hợp với trình soạn thảo được thiết kế để tạo và cập nhật các văn bản Web và biểu diễn theo các chuẩn của W3C. Vì các trình duyệt sử dụng các công cụ xử lý HTML khác nhau, chúng có thể hiển thị khác nhau đối với cùng một trang Web. Thêm vào đó, một số khả năng được hỗ trợ trong một trình duyệt này có thể không được hỗ trợ trong các trình duyệt khác.

Chương 2

Microsoft Internet Explorer 6



Hình 2.13. Trình duyệt Web Mozilla.

2.12. Kết chương

Trong chương này các bạn đã được làm quen với các chức năng của trình duyệt Web Internet Explorer 6 (IE6), biết cách tìm kiếm thông tin cũng như sử dụng nhiều dịch vụ khác trên World Wide Web. Các bạn cũng đã được giới thiệu một số trình duyệt khác. Kết thúc phần mở đầu giới thiệu về máy tính và Internet, trong phần sau các bạn sẽ được làm quen với ngôn ngữ HTML - một ngôn ngữ cơ bản cho Web và phát triển các ứng dụng trên Web.

PHẦN 2

NỘI DUNG

Chương 3 – Ngôn ngữ cơ bản cho Web - HTML

Chương 4 - Macromedia Dreamweaver MX 2004

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Nội dung của chương:

- Tìm hiểu về HTML và cách làm việc của HTML trên các trình duyệt
- Viết các trang Web bằng HTML

3.1. Giới thiệu về Internet, Web, và HTML

Như đã giới thiệu trong chương mở đầu, “Internet” hiện đang là hệ thống mạng lớn nhất thế giới và được xem là mạng của các mạng. *World Wide Web* (WWW) là một phần của Internet bao gồm nhiều máy chủ cung cấp các dịch vụ Web nằm rải rác trên khắp thế giới. Máy chủ dịch vụ Web chứa các thông tin được tổ chức dưới hình thức các trang Web mà bất kỳ người dùng nào trên thế giới cũng có thể truy cập. Ngôn ngữ cơ bản để thiết kế nên các trang Web này chính là HTML.

World Wide Web hoạt động trên ba cơ chế để đưa các nguồn thông tin đến với người dùng, bao gồm:

- **Giao thức:** Là các qui tắc cần tuân theo để truy cập các tài nguyên thông qua Web. Giao thức truyền siêu văn bản (HTTP) là một trong các giao thức được WWW sử dụng.
- **Địa chỉ:** WWW tuân theo một định dạng đặt tên thống nhất để truy cập các tài nguyên trên Web. Các địa chỉ URL (*Uniform Resource Locator*) được sử dụng để tìm các trang và các tài nguyên trên mạng.
- **HTML:** Là ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo các tài liệu có thể truy nhập trên mạng. Tài liệu HTML được tạo ra bằng các thẻ và các phần tử của HTML. File được lưu trên máy chủ dịch vụ Web với phần mở rộng *.htm* hoặc *.html*.

Khi bạn sử dụng trình duyệt để yêu cầu một thông tin nào đó, máy chủ dịch vụ Web sẽ đáp ứng yêu cầu này. Nó gửi thông tin bạn yêu cầu đến trình duyệt của bạn dưới dạng các trang Web. Sau đó trình duyệt sẽ định dạng thông tin này và hiển thị nó.

3.2. Ngôn ngữ HTML

HTML - HyperText Markup Language có hai đặc tính cơ bản là siêu văn bản và tính phổ quát. *Siêu văn bản (hypertext)* có nghĩa là bạn tạo ra một liên kết trong trang Web định hướng người dùng đến một nơi nào đó trên Internet, còn *tính phổ quát (universality)* được thể hiện ở chỗ do hầu hết các tài liệu HTML được lưu dưới dạng tập tin ASCII hoặc Text Only nên hầu như máy tính nào cũng có thể đọc được trang Web.

Thông qua các thẻ (*tag*) và các phần tử của HTML, chúng ta có thể thực hiện được rất nhiều công việc quan trọng như:

- Điều khiển hình thức và nội dung của trang Web.
- Xuất bản các tài liệu trực tuyến và truy xuất thông tin trực tuyến bằng cách dùng các tài liệu liên kết được chèn vào các tài liệu HTML.
- Tạo các biểu mẫu trực tuyến. Những biểu mẫu này có thể được sử dụng để thu thập thông tin về người dùng, quản lý các giao dịch ...
- Chèn các đối tượng như các đoạn âm thanh, các đoạn video, các thành phần ActiveX và các applet của Java vào tài liệu HTML.

Bản thân tài liệu HTML là một file văn bản có thành phần đuôi mở rộng là **.htm** hoặc **.html**. Tài liệu này được dịch thành mã nguồn của trang Web. Khi được hiển thị trong trình soạn thảo, trình duyệt đọc file **.htm** hoặc **.html** và hiển thị trang theo các thẻ quy định cách thức hiển thị trang đó.

Chúng ta sẽ tiếp tục tìm hiểu chi tiết vấn đề này trong các phần sau.

3.3. Lập kế hoạch, tổ chức và quản lý Website của bạn

Lập kế hoạch

Trước khi bắt tay vào viết các trang HTML cho website của bạn, bạn nên phác thảo và thiết kế site trước, để sau này sẽ không mất công tổ chức lại. Công việc này bao gồm:

- Xác định mục đích tạo trang này. Chính xác thì bạn muốn diễn đạt điều gì?
- Hình dung người xem. Bạn chỉnh sửa nội dung ra sao để trang hấp dẫn được người duyệt? Ví dụ, nên thêm nhiều hình ảnh, hay tốc độ tải trang là yêu cầu quan trọng hơn?
- Số trang cần thiết là bao nhiêu? Cấu trúc mong muốn là gì?
- Bạn dự định cho người xem duyệt qua site theo định hướng cụ thể, hay để họ tùy ý khảo sát?
- Phác thảo site trên giấy.
- Vạch ra hệ thống đặt tên đơn giản, nhất quán cho các trang, hình ảnh, và các tập tin bên ngoài.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Tổ chức tập tin

- Tạo thư mục trung tâm để lưu giữ toàn bộ nội dung của Website sắp tạo.
- Phân chia thư mục trung tâm nhằm phản ánh tổ chức của Website. Có thể tạo thư mục riêng biệt cho tài liệu HTML, một thư mục chứa hình ảnh, một thư mục chứa tập tin ngoài. Nếu có site lớn với nhiều trang, hãy thử phân chia thành nhiều mức thư mục hoặc chương, đặt hình ảnh vào từng thư mục con trong đó.

3.4. Trình duyệt và trình soạn thảo

Trình duyệt

Tài liệu HTML được hiển thị thông qua một trình duyệt. Vậy trình duyệt là gì? Trình duyệt là một ứng dụng được cài đặt trên máy khách, làm nhiệm vụ đọc mã HTML và hiển thị trang theo các thẻ chỉ dẫn. Trình duyệt được sử dụng để xem các trang Web, chuyển trang.... Trình duyệt làm cho người dùng sử dụng dễ dàng do có giao diện đồ họa và hỗ trợ sử dụng chuột, bàn phím.

Trình duyệt được biết đến sớm nhất là Mosaic, do Trung tâm ứng dụng siêu máy tính quốc gia (NCSA) phát triển. Ngày nay, có nhiều trình duyệt để duyệt Internet. *Netscape's Navigator* và *Microsoft's Internet Explorer* là hai trình duyệt chính đang được sử dụng.

Trình soạn thảo

Để tạo một tài liệu trên mạng, bạn cần một trình soạn thảo HTML. Ngày nay, có nhiều trình soạn thảo đang được sử dụng.

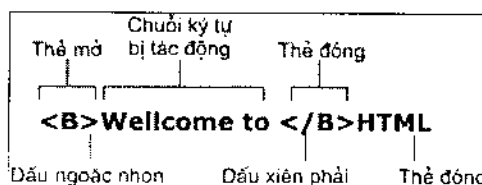
Một số trình soạn thảo tiêu biểu như:

- *Microsoft Frontpage*: là một công cụ tổng hợp có thể được dùng để tạo, thiết kế, và soạn thảo các trang Web. Bạn có thể thêm vào các văn bản, hình ảnh, bảng, biểu mẫu và các phần tử HTML khác vào trang Web của mình. Bạn chỉ cần thiết kế và tạo giao diện, trình soạn thảo Frontpage sẽ tạo mã HTML cần thiết giúp bạn.
- *Notepad*: bạn có thể sử dụng Notepad để tạo tài liệu HTML. Để xem tài liệu trong trình duyệt, bạn phải lưu tài liệu với phần tên mở rộng *.htm* hoặc *.html*.
- *Macromedia Dreamweaver MX 2004*: là một trình soạn thảo chuyên nghiệp cho phép xây dựng và quản lý websites một cách thuận tiện. Thông qua Dreamweaver người thiết kế có thể viết mã HTML thuận tiện hay sử dụng thao tác trực quan theo kiểu 'WYSIWYG' (*What You See Is What You Get*).

Trong chương tiếp chúng ta sẽ tiếp cận trình soạn thảo Dreamweaver. Và bạn sẽ không còn phải bận tâm nhiều về vấn đề thiết kế mã nữa bởi vì Dreamweaver sẽ hỗ trợ bạn tất cả.

3.5. Các thẻ của HTML

Thẻ HTML là lệnh nằm giữa hai dấu "<" và ">", cho biết cách thức trình duyệt sẽ hiển thị văn bản. Cấu trúc một thẻ HTML được mô tả như hình vẽ 3.1



Hình 3.1. Cấu trúc thẻ HTML.

Các thẻ có dạng đóng và mở, còn chuỗi ký tự bị tác động là chuỗi *nằm giữa* hai thẻ mở/đóng. Cả thẻ đóng lẫn thẻ mở đều dùng chung từ lệnh, nhưng thẻ đóng có thêm dấu xiên phải (/). Thẻ HTML bao gồm:

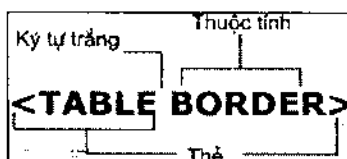
<PHẦN TỬ THUỘC TÍNH = giá trị>

Trong đó:

- Phần tử: để nhận dạng thẻ.
- Thuộc tính: mô tả thẻ.
- Giá trị: là nội dung được gán cho thuộc tính.

Các thuộc tính

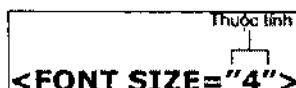
Nhiều thẻ có nhiều thuộc tính đặc biệt. Thường thì bạn có thể sử dụng một loạt thuộc tính trong chỉ một thẻ, cứ viết thuộc tính này tiếp sau thuộc tính khác – theo trật tự bất kỳ - phân cách nhau bằng ký tự trắng.



Hình 3.2. Một số thẻ có thuộc tính tùy chọn.

Giá trị

Thẻ có thuộc tính, còn thuộc tính lại có *giá trị*. Ở vài trường hợp, bạn phải chọn lựa từ nhiều giá trị cho trước. Ví dụ, thuộc tính SIZE của thẻ có thể nhận các giá trị số nguyên, bạn nên đặt các giá trị này trong ngoặc kép để chúng không bị máy hiểu nhầm



Hình 3.3. Thuộc tính SIZE.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Chú ý :

HTML không phân biệt chữ hoa hay chữ thường. Nhưng cần phải lưu ý với XHTML hay XML thì có phân biệt chữ hoa và chữ thường. Trong chương này chúng ta quy ước sử dụng tên các thẻ và thuộc tính theo dạng chữ hoa, còn giá trị ở dạng chữ thường (cho dễ đọc). Còn chúng tôi vẫn khuyên các bạn nên sử dụng tất cả ở dạng chữ thường vì đó chính là cách viết chuẩn mà chúng ta sẽ học trong chương XHTML. Còn bây giờ chúng ta hãy tiếp tục tìm hiểu về HTML.

3.6. Cấu trúc của một trang HTML

Ví dụ 3.1.

Chúng ta hãy làm một ví dụ nhỏ sau.

- Sử dụng một chương trình soạn thảo bất kỳ và gõ vào đoạn mã sau:

```
<HTML>
<HEAD>
<TITLE> Tiêu đề trang</TITLE>
</HEAD>
<BODY BGCOLOR = lavender>
<H3> Chào mừng bạn đến với HTML</H3>
</BODY>
</HTML>
```

- Lưu file này dưới dạng đuôi **.htm** hoặc **.html**. Sau đó kích hoạt chạy file bằng trình duyệt Internet Explorer, bạn sẽ nhìn thấy kết quả hiển thị như hình 3.4 .



Hình 3.4. Kết quả hiển thị trang HTML trên trình duyệt IE6.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Qua ví dụ trên các bạn có thể thấy cấu trúc một tài liệu HTML gồm 3 phần chính:

- **Phần HTML:** Mỗi tài liệu phải bắt đầu bằng một thẻ HTML kí hiệu là: `<HTML>` và kết thúc bằng một thẻ kết thúc HTML kí hiệu là: `</HTML>`. Cặp thẻ này thông báo với trình duyệt là nội dung giữa hai thẻ sẽ là một tài liệu HTML.
- **Phần tiêu đề:** Phần tiêu đề bắt đầu bằng thẻ `<HEAD>` và kết thúc bằng thẻ `</HEAD>`. Phần này chứa tiêu đề được hiển thị trong thanh tiêu đề của trang Web. Nội dung tiêu đề nằm trong cặp thẻ `TITLE`, bắt đầu bằng thẻ `<TITLE>` và kết thúc bằng thẻ `</TITLE>`. Tiêu đề là phần khá quan trọng. Trình duyệt sử dụng tiêu đề để lưu trữ các dấu mốc (*bookmarks*) được dùng để đánh dấu một Website. Khi người dùng đang tìm kiếm thông tin, tiêu đề của trang Web sẽ cung cấp các từ khoá chính mà người dùng đang tìm kiếm.
- **Phần thân văn bản:** nằm sau phần tiêu đề, bao gồm văn bản, hình ảnh và các liên kết mà bạn muốn hiển thị trên trang Web của mình. Phần thân văn bản bắt đầu bằng thẻ `<BODY>` và kết thúc bằng thẻ `</BODY>`. Nội dung trang HTML sẽ nằm giữa hai thẻ này do đó khi ta sử dụng thuộc tính của thẻ `<BODY>` thì toàn bộ trang Web sẽ chịu ảnh hưởng của thuộc tính đó.

3.7. Định dạng trang

Trong quá trình thiết kế các trang Web, HTML hỗ trợ bạn các thẻ định dạng giúp việc trình bày trang Web đó trên trình duyệt trong sẽ đẹp mắt và rõ ràng hơn.

Sau đây là một số thẻ thông dụng của HTML trong định dạng trang:

Sử dụng màu nền và màu chữ

- Thuộc tính `BGCOLOR` của thẻ `BODY` cho phép ấn định màu nền cho trang Web, còn thuộc tính `TEXT` ấn định màu font chữ trên trang.
- Có 3 màu chính là đỏ, xanh lá cây và xanh da trời. Mỗi màu chính được định nghĩa như một số có 2 chữ số trong hệ đếm thập lục phân: `#RRGGBB` (dấu # để chỉ đây là mã dịch ra). Số thập lục phân 00 chỉ 0% của màu trong khi đó số thập lục phân FF chỉ 100% của màu. Giá trị cuối cùng là một mã sáu số chỉ màu.

Bảng 3.1 Ví dụ biểu diễn màu trong hệ đếm thập lục phân

Mã thập lục phân	Màu
#FF0000	Đỏ
#00FF00	Xanh lá cây
#0000FF	Xanh da trời
#000000	Đen
#FFFFFF	Trắng

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Hiển thị dòng tiêu đề

Phần tử này được sử dụng để cung cấp tiêu đề cho phần nội dung hiển thị trên trang Web. Phần tiêu đề thường được hiển thị to và đậm hơn để phân biệt chúng với phần còn lại của văn bản. Bạn có thể hiển thị tiêu đề một trong sáu kích thước từ H1 đến H6. Nó bắt đầu bằng `<H>` và kết thúc bằng `</H>` (Trong đó *X* là một số nguyên từ 1 đến 6).

Căn giữa các phần tử trên trang với thẻ CENTER

Cú pháp: `<CENTER>` Nội dung sẽ được canh giữa `</CENTER>`

Định lề

Mặc định, các trình duyệt *Netscape* và *Internet Explorer* chứa một khoảng trống nhất định giữa nội dung trang và rìa cửa sổ hiển thị. Các thẻ thuộc tính định lề trong hai trình duyệt này lại khác nhau:

- Trình duyệt IE hỗ trợ thuộc tính *LEFTMARGIN* và *TOPMARGIN* để định lề bên trái và trên tính theo đơn vị pixel.
- Trình duyệt Netscape thì lại hỗ trợ thuộc tính *MARGINWIDTH* và *MARGINHEIGHT* với cùng ý nghĩa như trên.

Do hai trình duyệt này không hiểu rõ phương thức hiển thị của nhau, nên khi tạo các trang Web, chúng ta phải sử dụng cả hai thẻ:

```
<HTML>
<HEAD><TITLE>Canh lề</TITLE></HEAD>
<BODY LEFTMARGIN = 0 TOPMARGIN = 0
  MARGINWIDTH = 0 MARGINHEIGHT = 0>
....
</BODY>
</HTML>
```

Bắt đầu đoạn mới

Để bắt đầu một đoạn mới trong trang Web chúng ta dùng cặp thẻ `<P>` `</P>`. Thẻ này có thêm thuộc tính *ALIGN* cho phép canh chỉnh nội dung đoạn theo trái (*left*), phải (*right*), giữa (*center*). Cú pháp:

`<P ALIGN = left>` Nội dung đoạn mới `</P>`

Tạo dấu ngắt dòng

Để bắt đầu một dòng mới với khoảng cách không bị quá xa như thẻ `<P>`, chúng ta sử dụng thẻ `<BR>` và không cần phải có thẻ đóng.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Một ví dụ sử dụng các thẻ HTML để định dạng trang Web

Chúng ta đã tìm hiểu qua một số thẻ HTML giúp định dạng trang Web mà chúng ta muốn xây dựng. Quan sát một ví dụ sử dụng một số thẻ HTML này.

Ví dụ 3.2.

- Sử dụng một chương trình soạn thảo bất kỳ và gõ vào đoạn mã sau:

```
<HTML>

<HEAD><TITLE>Định dạng trang</TITLE></HEAD>

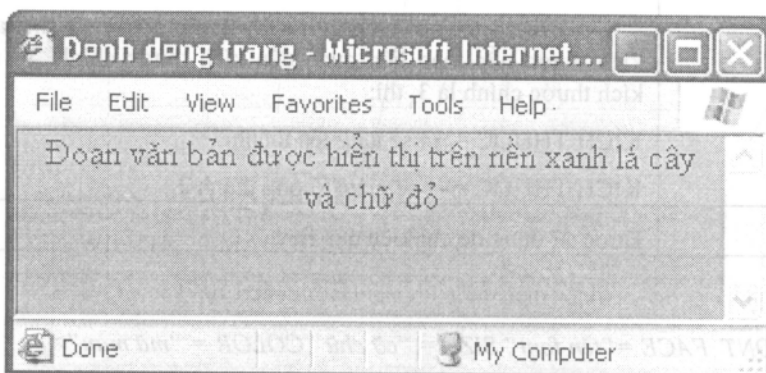
<BODY BGCOLOR="#0000FF" TEXT="#FF0000"
LEFTMARGIN = 0 TOPMARGIN = 0
MARGINWIDTH = 0 MARGINHEIGHT = 0>

<CENTER> Đoạn văn bản được hiển thị trên nền xanh lá cây và chữ đỏ
</CENTER>

</BODY>

</HTML>
```

- Lưu file này dưới dạng đuôi **.htm** hoặc **.html**. Sau đó kích hoạt chạy file bằng trình duyệt Internet Explorer, bạn sẽ nhìn thấy kết quả hiển thị như sau:



Hình 3.5. Kết quả hiển thị trang HTML với định dạng trang.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

3.8. Định dạng văn bản

Có hai phương pháp để định dạng văn bản.

- Phương pháp thứ nhất cơ bản hơn là áp dụng các thẻ HTML và thuộc tính bình thường để định dạng từng phần văn bản riêng biệt. Chúng ta sẽ làm quen với phương pháp này trong phần dưới đây.
- Phương pháp thứ hai được sử dụng trong trường hợp tài liệu quá dài, sử dụng Style Sheet cho phép tạo định dạng văn bản chỉ trong một đoạn mã. Chúng ta sẽ tìm hiểu phương pháp này trong chương 5.

Đổi phông chữ

Sử dụng thẻ `<FONT>` kết hợp với thuộc tính *FACE*, *SIZE* và *COLOR* để thiết lập loại font chữ theo ý muốn.

Bảng 3.2. Sử dụng thẻ `<FONT>` kết hợp với thuộc tính *FACE*, *SIZE* và *COLOR*

Thuộc tính	Mô tả
COLOR	Được sử dụng để chỉ màu của FONT. Bạn có thể sử dụng tên màu hoặc qua giá trị thập lục phân tương ứng với màu đó.
SIZE	Được sử dụng để chỉ kích thước của font. Bạn có thể chỉ ra một loạt các kích thước của FONT từ 1 đến 7. Kích thước lớn nhất là 7 và nhỏ nhất là 1. Ngoài ra có thể sử dụng một kích thước chính và chỉ ra những kích thước tiếp theo liên quan đến kích thước chính. Ví dụ, nếu kích thước chính là 3, thì: KÍCH THƯỚC = +4 sẽ tăng lên thành 7 KÍCH THƯỚC = -1 sẽ giảm xuống thành 2
FACE	Được sử dụng để chỉ kiểu của font

Cú pháp :

`<FONT FACE = "tên font" SIZE = "cỡ chữ" COLOR = "mã màu">`

Đoạn văn bản cần áp dụng font

`</FONT>`

Chương 3

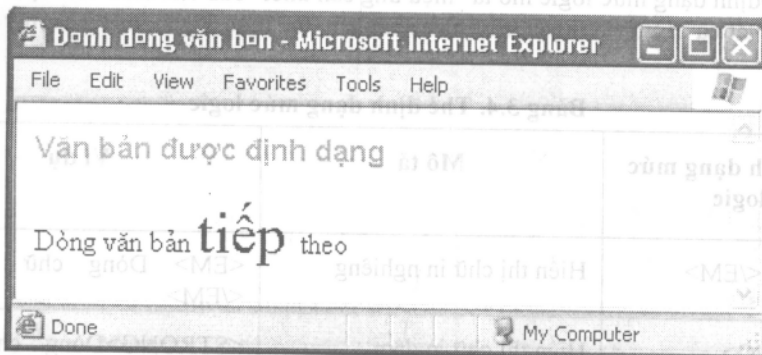
Ngôn ngữ cơ bản cho Web - HTML

Ví dụ 3.3.

- Gõ đoạn mã HTML sau:

```
<HTML>
<HEAD>
<TITLE>Định dạng văn bản</TITLE>
<BODY>
<H3><FONT SIZE = 3 COLOR = HOTPINK FACE = Arial>Văn bản được định
dạng</FONT></H3>
<P> Dòng văn bản <FONT COLOR = BLUE SIZE = 6>tiếp </FONT> theo
</BODY>
</HTML>
```

- Kết quả hiển thị trên trình duyệt:



Hình 3.6. Kết quả hiển thị trang HTML.

Thẻ định dạng mức vật lý:

Bảng 3.3. Thẻ định dạng mức vật lý

Thẻ	Mô tả	Ví dụ
 ... 	Đậm	 Dòng chữ đậm
<I>...</I>	Nghiêng	<I>Dòng chữ nghiêng </I>
<U>...</U>	Gạch chân	<U>Dòng chữ gạch chân</U>
<BIG>...</BIG>	Gỡ kích thước lớn hơn bình thường. Nó được sử dụng để hiển thị một từ, một ký tự, hoặc một cụm từ với font lớn hơn so với phần còn lại của văn bản.	<BIG>D</BIG>dòng chữ có kích thước lớn hơn so với phần còn lại

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

<SMALL>... </SMALL>	Gỡ kích thước nhỏ hơn bình thường. Nó được sử dụng để hiển thị một từ, một ký tự, hoặc một cụm từ với font nhỏ hơn so với phần còn lại của văn bản.	<SMALL>D</SMALL> dòng chữ có kích thước nhỏ hơn so với phần còn lại.
<STRIKE>	Hiển thị dấu gạch ngang cả văn bản	<STRIKE>Dòng chữ bị gạch ngang</STRIKE>
^{...}	Chỉ số trên	PI*r ²
_{...}	Chỉ số dưới	O₂

Thẻ định dạng mức logic

Thẻ định dạng mức logic mô tả "hiệu ứng cần thiết" của văn bản. Thẻ định dạng mức vật lý được tất cả các trình duyệt hiểu như nhau còn thẻ mức logic được hiểu tùy theo trình duyệt.

Bảng 3.4. Thẻ định dạng mức logic

Thẻ định dạng mức logic	Mô tả	Ví dụ
...	Hiển thị chữ in nghiêng	 Dòng chữ nghiêng
... 	Hiển thị chữ in đậm	Dòng chữ đậm
<DFN>...</DFN>	Được sử dụng cho định nghĩa	<DFN>Văn in nghiêng</DFN>
<CODE>... </CODE>	Được sử dụng trong các phần trích của mã chương trình	<CODE>x = X + 1 Y = Y + 1</CODE>
<KBD>...</KBD>	Được sử dụng trong định dạng thuần túy	<KBD>Bình thường</KBD>
<VAR>...</VAR>	Được sử dụng cho các biến số trong các chức năng chương trình	<VAR>X</VAR>

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

<CITE>...</CITE>	Được sử dụng cho các trích dẫn và tham chiếu, biến thành chữ nghiêng	<CITE> Một dạng chữ nghiêng khác</CITE>
<ABBR>...</ABBR>	Chỉ dạng viết tắt, ví dụ: URL	<ABBR>UN</ABBR>
<BLINK>...</BLINK>	Được sử dụng để nhấp nháy văn bản (xem trong trình duyệt Netscape)	Đây là dòng chữ <BLINK> nhấp nháy </BLINK>
...	Một dạng khác của chữ gạch ngang	Chữ gạch ngang
<INS>...</INS>	Một dạng khác của chữ gạch dưới	<INS>Chữ gạch dưới</INS>

Chú ý:

- Mỗi trình duyệt có cách thực hiện các phần tử INS và DEL riêng. Netscape không thực hiện các phần tử này.
- Phần tử BLINK không tương thích trong Internet Explorer.

Thêm chú thích

Nội dung chú thích sẽ không xuất hiện trong trình duyệt mà nó giúp bạn nhớ rõ mục đích của bạn cố gắng đạt được với thẻ HTML trong quá trình viết mã. Cú pháp sử dụng cặp thẻ chú thích như sau :

<!-- Lời chú giải -->

Sử dụng các ký tự đặc biệt

Bạn có thể chèn các ký tự đặc biệt vào văn bản của tài liệu HTML. Để bảo đảm trình duyệt không nhầm chúng với thẻ HTML, bạn phải thêm *mã thoát* cho các ký tự đặc biệt này. Định dạng là: &code

Bảng 3.5. Một số ký tự thông dụng

Các ký tự đặc biệt	Mã thoát
Lớn hơn (>)	>
Nhỏ hơn (<)	<
Trích dẫn	"
Ký tự "và" &	&

3.9. Thẻ đường ngang

Thẻ kẻ đường ngang <HR> được sử dụng để kẻ một đường ngang trên một trang. Nó có một thẻ bắt đầu, không có thẻ kết thúc và không chứa nội dung. Các thuộc tính sau đây giúp điều khiển các đường ngang.

Bảng 3.6. Các thuộc tính giúp điều khiển đường ngang

Thuộc tính	Mô tả
ALIGN	Chỉ vị trí của đường ngang. Bạn có thể canh lề nằm giữa/phải/trái.
WIDTH	Chỉ độ dài của dòng. Nó có thể được xác định bằng các điểm hoặc tính theo phần trăm. Mặc định là 100%, nghĩa là toàn bộ bề ngang của tài liệu.
SIZE	Chỉ độ dày của đường ngang và được xác định bằng các điểm.
NOSHADE	Chỉ đường được hiển thị bằng màu đặc thay vì có bóng.

Ví dụ 3.4.

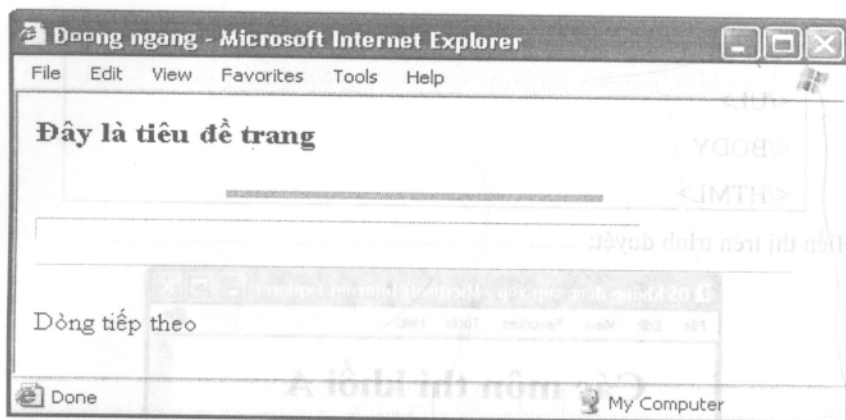
- Tạo trang HTML có nội dung sau:

```
<HTML>
<HEAD>
<TITLE>Đường ngang </TITLE>
<BODY>
<H3> Đây là tiêu đề trang</H3>
<HR NOSHADE SIZE= 5 ALIGN = center WIDTH= 50%>
<HR SIZE= 15 ALIGN = left WIDTH= 80%>
<HR>
<P> Dòng tiếp theo
</BODY>
</HTML>
```

- Hiển thị trên trình duyệt, kết quả như sau:

Chương 3

Ngôn ngữ cơ bản cho Web - HTML



Hình 3.7. Kết quả hiển thị trang HTML với định dạng văn bản.

3.10. Danh sách

Các danh sách được sử dụng để nhóm dữ liệu một cách logic. Bạn có thể thêm các danh sách vào tài liệu HTML để nhóm các thông tin có liên quan với nhau, như: nhóm các loại hoa, nhóm các loại trái cây... Các loại danh sách có thể chèn vào tài liệu HTML là:

- Danh sách không được sắp xếp.
- Danh sách được sắp xếp.
- Danh sách định nghĩa.

Danh sách không được sắp xếp

Đây là loại danh sách đơn giản nhất, các mục bắt đầu bằng các dấu bullet. Mỗi mục được đánh dấu bằng thẻ `<LI>`, viết tắt của *List Item*. Thẻ kết thúc `</LI>` không bắt buộc.

Ví dụ 3.5.

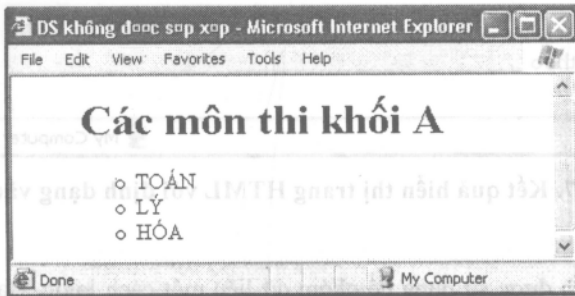
```
<HTML>
<HEAD><TITLE>DS không được sắp xếp</TITLE></HEAD>
<BODY>
<H1>Các môn thi khối A</H1>
<UL>
<LI>TOÁN
<LI>LÝ
```

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

```
<LI>HÓA  
</UL>  
</BODY>  
</HTML>
```

- Hiển thị trên trình duyệt:



Hình 3.8. Danh sách không được sắp xếp.

Sử dụng thuộc tính *TYPE* để định dạng bullet hiển thị cùng với các mục trong danh sách.

Bảng 3.7. Sử dụng thuộc tính *TYPE*

Thuộc tính	Thẻ
Bulleted	 và
Các bullet được tùy chỉnh	<LI TYPE=SQUARE> Các bullet vuông <LI TYPE = DISC> Các bullet hình cầu <LI TYPE=CIRCLE> Các bullet hình tròn

Chú ý:

- Thuộc tính *TYPE* không được hiển thị trong Internet Explorer.

Danh sách được sắp xếp

Danh sách được sắp xếp nằm trong thẻ , . Sự khác nhau là các mục danh sách hiển thị theo thứ tự được tạo ra một cách tự động.

Ví dụ 3.6.

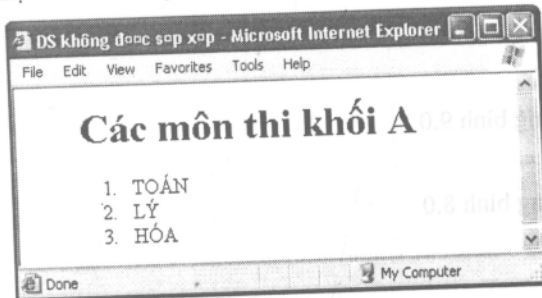
- Tạo trang HTML như sau:

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

```
<HTML>
<HEAD><TITLE>DS được sắp xếp</TITLE></HEAD>
<BODY>
<H1>Các môn thi khối A</H1>
<OL>
<LI>TOÁN
<LI>LÝ
<LI>HÓA
</OL>
</BODY>
</HTML>
```

- Kết quả hiển thị trên trình duyệt:



Hình 3.9. Danh sách được sắp xếp.

Chúng ta có thể đặt thuộc tính để xác định các thông số được tạo ra cho các mục danh sách.

Bảng 3.8. Đặt thuộc tính cho danh sách

Thuộc tính	Thẻ
Upper Roman	<LI TYPE = I>
Lower Roman	<LI TYPE = i>
Upper case	<LI TYPE = A>
Lower case	<LI TYPE = a>
Bắt đầu với số khác "1"	<OL START = n>

Bạn có thể lồng danh sách được sắp xếp trong danh sách không được sắp xếp và ngược lại.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Danh sách định nghĩa

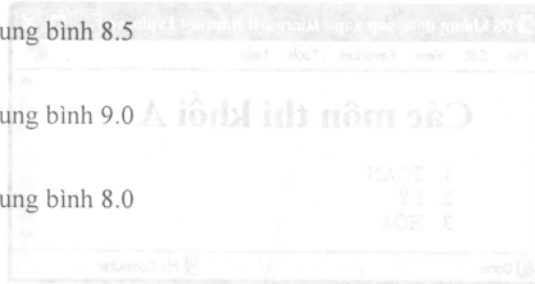
Danh sách định nghĩa được sử dụng để tạo một danh sách các thuật ngữ và định nghĩa của các thuật ngữ đó. Danh sách định nghĩa nằm trong thẻ `<DL>`, `</DL>`.

Thẻ `<DT>` được sử dụng để chỉ thuật ngữ, còn thẻ `<DD>` chỉ định nghĩa.

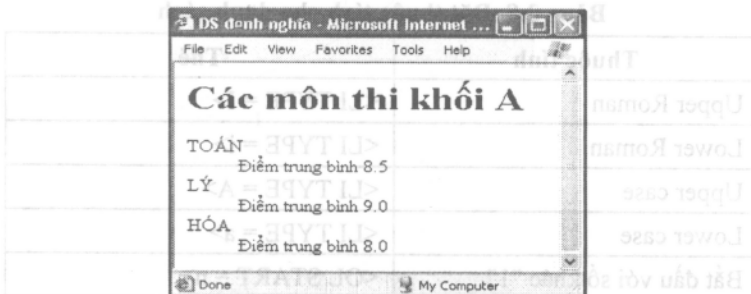
Ví dụ 3.7.

- Tạo trang HTML sau:

```
<HTML>
<HEAD> <TITLE>DS định nghĩa</TITLE>
</HEAD>
<BODY>
<H1>Các môn thi khối A</H1>
<DL>
<DT>TOÁN
<DD> Điểm trung bình 8.5
<DT>LÝ
<DD> Điểm trung bình 9.0
<DT>HÓA
<DD> Điểm trung bình 8.0
</DL>
</BODY>
</HTML>
```



- Kết quả hiển thị trên trình duyệt:



Hình 3.10. Danh sách định nghĩa.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

3.11. Chèn ảnh vào trang Web

Khi thiết kế trang Web, nếu chỉ đơn giản là những dòng chữ thôi thì trang Web của bạn sẽ không có sức hấp dẫn. HTML cung cấp cho bạn một loại thẻ có thể chèn được hình ảnh vào trong trang. Đó là cặp thẻ `<IMG></IMG>`. Để chèn hình ảnh vào tài liệu, sử dụng thẻ `IMG` rồi tham chiếu đến địa chỉ lưu giữ ảnh. Ví dụ:

`<IMG SRC = "image.gif" BORDER = 0 ALIGN = "bottom">` Dòng chữ kết hợp

Sau đây là một số thuộc tính của thẻ `IMG`:

Bảng 3.9. Một số thuộc tính của thẻ `IMG`

Thuộc tính	Ý nghĩa
<code>SRC = URL</code>	Chỉ ra nguồn chứa ảnh URL
<code>BORDER = "độ dày"</code>	Viền của ảnh
<code>ALIGN = "hướng căn"</code>	Căn lề của ảnh so với văn bản xung quanh, bao gồm TOP, BOTTOM, MIDDLE
<code>ALT = "từ thay thế"</code>	Hiển thị cụm từ thay thế
<code>WIDTH</code>	Xác định chiều rộng của ảnh
<code>HEIGHT</code>	Xác định chiều cao của ảnh

Ví dụ 3.8.

- Để có kết quả như hình 3.11, tạo trang mã HTML sau:

```
<HTML>
<HEAD> <TITLE>Chèn ảnh</TITLE>
<BODY>
<H1><CENTER><FONT SIZE=3 COLOR=FORESTGREEN>
<B>Chèn ảnh</B>
</FONT></CENTER></H1>
<HR>
<IMG ALIGN=BOTTOM SRC= "h.gif">Căn dưới</P>
<P>
<IMG ALIGN=TOP SRC= "3.gif">Căn đỉnh trên</P>
<P>
<IMG ALIGN=MIDDLE SRC= "y.gif"> Căn giữa</P>
```


Chương 3

Ngôn ngữ cơ bản cho Web - HTML

</BODY>

</HEAD>

</HTML>

- Kết quả hiển thị trên trình duyệt như hình 3.11.



Hình 3.11. Chèn ảnh và căn chữ.

Một số định dạng ảnh phổ biến

- **Định dạng ảnh GIF (Graphics Interchange Format)**

GIF là định dạng thông thường nhất được sử dụng trong các tài liệu HTML. Các file GIF được định dạng không phụ thuộc nền và hỗ trợ đến 256 màu. Ưu điểm của các file GIF là khá dễ chuyển tải, ngay cả đối với các kết nối sử dụng MODEM tốc độ chậm.

- **Định dạng ảnh JPEG (Joint Photographic Expert Group)**

Nén JPEG là một lược đồ nén mất thông tin. Điều này có nghĩa là ảnh sau khi bị nén không giống như ảnh gốc. Tuy nhiên, ảnh sau nén hiển thị cũng tốt gần như ảnh gốc. Khi bạn lưu một file với định dạng JPEG, bạn có thể định tỉ lệ nén. Tỉ lệ càng cao thì ảnh

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

càng ít giống ảnh gốc. JPEG hỗ trợ hơn 16 triệu màu và thường được sử dụng cho các ảnh có màu thực.

Cả các file dạng JPEG (phần mở rộng là .jpg) và các dạng GIF đều nén ảnh để bảo đảm tốc độ chuyển tải qua Internet được nhanh hơn. Ảnh JPEG có thể được nén nhiều hơn nhưng chậm hơn trong quá trình hiển thị so với ảnh .GIF. Có lẽ đó là lý do tại sao ảnh .GIF phổ biến hơn trong tài liệu HTML.

- **Định dạng ảnh PNG (Portable Network Graphics)**

Định dạng cho một file PNG là không bị mất (*lossless*). Trong nén không bị mất "*lossless*", dữ liệu ảnh được nén mà không bỏ chi tiết. Các file PNG hỗ trợ ảnh màu thực và dải màu xám. Các file PNG cũng có thể được nén để chuyển qua mạng. Ảnh định dạng này hỗ trợ cả những ảnh phức tạp cũng như nền trong suốt và rất tốt để thay thế cho hai loại định dạng trên.

Như vậy, khi dùng ảnh trong trang Web của mình, bạn phải chú ý cân nhắc xem nên dùng loại ảnh nào cho phù hợp vì điều này sẽ làm ảnh hưởng tới tốc độ tải trang của bạn.

Bản đồ ảnh

Bạn có thể sử dụng ảnh trong trang Web để tạo bản đồ ảnh. Một bản đồ ảnh có các vùng hoạt động mà bạn có thể gán các liên kết với chúng gọi là các "điểm nóng". Khi người dùng nhấp vào các "điểm nóng", thì các liên kết được kích hoạt.

Toạ độ của điểm nóng được xác định bằng cách sử dụng vùng (có thể có hình dáng khác nhau). Bạn phải xác định hình dạng và toạ độ của vùng nóng.

Bảng 3.10. Xác định hình dạng vùng

Hình dạng	Mô tả
DEFAULT	Chỉ toàn bộ vùng. Đây là mặc định.
RECT	Chỉ vùng hình chữ nhật
CIRCLE	Chỉ vùng hình tròn
POLY	Chỉ vùng hình đa giác

Toạ độ được xác định đối với góc trên và trái của đối tượng.

Bảng 3.11. Xác định toạ độ vùng

Toạ độ	Phối hợp
RECT	Trái-x, trên-y, phải-x, dưới-y
CIRCLE	giữa-x, giữa-y, bán kính
POLY	x1, y1, x2, y2, ...xN, yN

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Để tạo một bản đồ ảnh, bạn phải sử dụng các phần tử *MAP*, *AREA* và *usemap*.

- Phần tử *AREA* được sử dụng để tạo vùng "điểm nóng".
- Phần tử *MAP* được dùng để chuyển các vùng ảnh với các URL tương ứng.
- Thuộc tính *usemap* liên kết một phần tử với bản đồ ảnh của nó.

Ví dụ 3.9.

Để tạo một bản đồ ảnh:

- Tạo một file ảnh và lưu nó với phần tên mở rộng là .GIF hoặc .JPEG
- Tạo một tài liệu .htm.
- Gõ đoạn mã sau sau:

```
<HTML>
<BODY>
<H1>Bản đồ ảnh</H1>
<A HREF = " ">
<CENTER>
<IMG SRC=3.gif WIDTH=176 HEIGHT=163 BORDER=1 usemap=
"#map1">
</CENTER>
</A>
<MAP NAME= "map1">
<AREA href= "file1.html"
    alt= "Link1"
    shape= "rect"
    coords= "0, 0, 80, 80">
<AREA href= "file2.html"
    alt= "Link2"
    shape= "rect"
    coords= "0, 80, 80, 163">
```

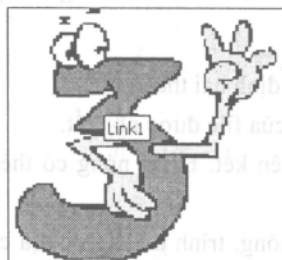
Chương 3

Ngôn ngữ cơ bản cho Web - HTML

```
<AREA href= "file3.html"
      alt= "Link3"
      shape= "circle"
      coords= "163, 175, 60">
<AREA href= "file4.html"
      alt= "Link4"
      shape= "poly"
      coords= "163, 0, 175, 28, 50, 50, 163, 0">
</MAP>
</BODY>
</HTML>
```

- Kết quả hiển thị trên trình duyệt:

Bản đồ ảnh



Hình 3.12. Bản đồ ảnh với bốn liên kết.

Trong ví dụ trên chúng ta có sử dụng thẻ liên kết `<A>`, bạn không cần phải lo lắng vì ngay sau đây chúng ta sẽ tìm hiểu cách tạo liên kết khi sử dụng thẻ HTML.

3.12. Siêu liên kết

Liên kết (link) là một đặc trưng của WWW. Chúng cho phép bạn chuyển từ trang này sang trang khác, và tải tập tin về bằng giao thức truyền file FTP (*File Transfer Protocol*).

Bạn có thể đặt liên kết đến:

- Một phần khác của cùng một tài liệu.

Chương 3

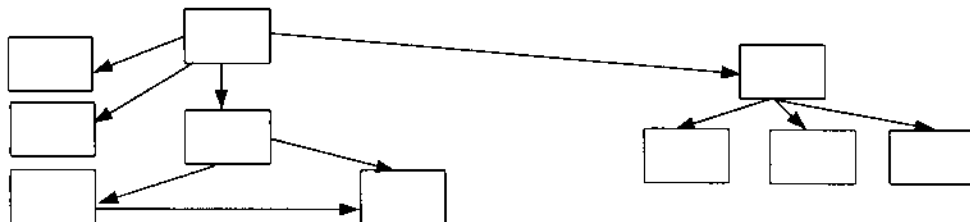
Ngôn ngữ cơ bản cho Web - HTML

- Một tài liệu khác
- Một phần khác của một tài liệu khác
- Các file khác - ảnh, âm thanh, trích đoạn video
- Một vị trí hoặc máy chủ khác

Các loại liên kết

Các liên kết được tạo ra có thể là liên kết trong hoặc liên kết ngoài:

- Liên kết trong là các liên kết với các phần trong cùng một tài liệu hoặc cùng một web site.
- Liên kết ngoài là các liên kết với các trang trên web site khác hoặc máy chủ khác.



Hình 3.13. Liên kết trong và ngoài.

Tạo siêu liên kết

Để tạo siêu liên kết, bạn phải xác định hai thành phần:

- Địa chỉ đầy đủ hoặc URL của file được nối kết.
- Điểm nóng sẽ cung cấp liên kết. Điểm nóng có thể là một dòng của văn bản hoặc thậm chí một ảnh.

Khi người dùng nhấp vào điểm nóng, trình duyệt đọc địa chỉ được xác định trong URL và chuyển đến vị trí mới.

Địa chỉ

Mỗi tài nguyên trên web có một địa chỉ duy nhất. Ví dụ, 207.46.130.149 là địa chỉ web site của Microsoft. Như chúng ta đều thấy việc nhớ các con số này rất khó và dễ nhầm lẫn. Vì vậy, người ta sử dụng các *Uniform Resource Locator* (URL) là một chuỗi kí tự cung cấp địa chỉ Internet của một web site hoặc nguồn tài nguyên trên World Wide Web. Định dạng điển hình là: **www.nameofsite.typeofsite.countrycode**. Ví dụ, 207.46.130.149 có thể được biểu diễn trong URL là **www.microsoft.com**.

- URL giúp nhận biết giao thức mà site hoặc nguồn được truy cập. Dạng URL thông thường nhất là "**http**", nó cung cấp cho Internet địa chỉ của một trang web. Một vài

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

dạng URL khác là "gopher", cung cấp địa chỉ Internet của một thư mục Gopher, và "ftp", cung cấp vị trí mạng của một nguồn FTP.

- URL có thể chỉ đến một trang cụ thể trong một web site. Ví dụ, *www.microsoft.com/ms.htm* sẽ dẫn bạn đến trang chủ của Microsoft.
- URL cũng có thể đề cập một vị trí trong một nguồn tài liệu. Ví dụ, bạn có thể tạo một liên kết với một phần nào đó trong cùng một tài liệu. Trong trường hợp đó, định danh phân mảnh được sử dụng ở cuối URL. Định dạng là: **Protocol:nameofsite / main document#fragment identifier**

Có hai dạng URL:

- **URL tuyệt đối** – là địa chỉ Internet đầy đủ của một trang hoặc file, bao gồm giao thức, vị trí mạng, đường dẫn tùy chọn và tên file. Ví dụ, *http://www.microsoft.com/ms.htm* là một URL tuyệt đối.
- **URL tương đối** - là một URL với một hoặc nhiều phần bị thiếu. Trình duyệt lấy thông tin bị thiếu từ trang chứa URL. Ví dụ, nếu giao thức bị thiếu, thì trình duyệt sẽ sử dụng giao thức của trang hiện thời.

Tạo các điểm neo (Anchors)

Thẻ neo <A> được sử dụng để xác định siêu liên kết trong tài liệu HTML. Tham số *HREF* (tham chiếu siêu văn bản) được sử dụng để chỉ địa chỉ hoặc URL của tài liệu hoặc file được liên kết. Cú pháp là:

`<A HREF = "URL">Hypertext</A>`

Mô tả cụ thể hơn:

`<A HREF = "protocol://host.domain:port/path/filename">Hypertext</A>`

- **Protocol - Giao thức:** Một số giao thức thường dùng:
 - *http* – giao thức truyền siêu văn bản
 - *telnet* - mở một phần telnet
 - *gopher* – tìm một file
 - *ftp* – giao thức truyền file
 - *mailto* - gửi e-mail
- **Host.domain:** địa chỉ Internet của máy chủ.
- **Port - Cổng:** số cổng của máy chủ đích.
- **Hypertext - Siêu văn bản:** văn bản hoặc ảnh mà người dùng phải nhấp vào để kích hoạt liên kết.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Liên kết ngoài

Chúng ta hãy cùng quan sát ví dụ sau.

Ví dụ 3.10.

Giả sử có hai tài liệu HTML trên đĩa cứng cục bộ, *Trang1.htm* và *Trang2.htm*. Để tạo một liên kết từ *Trang1.htm* đến *Trang2.htm*.

- Tạo trang *Trang doc. htm* như sau:

```
<HTML>
<HEAD>
<TITLE> Trang 1</TITLE>
</HEAD>
<BODY>
<H1><FONT SIZE = 3 COLOR = HOTPINK>TRANG 1 </FONT></H1>
<P>
<A HREF = "Trang2.htm">Liên kết đến trang 2</A>
</BODY>
</HTML>
```

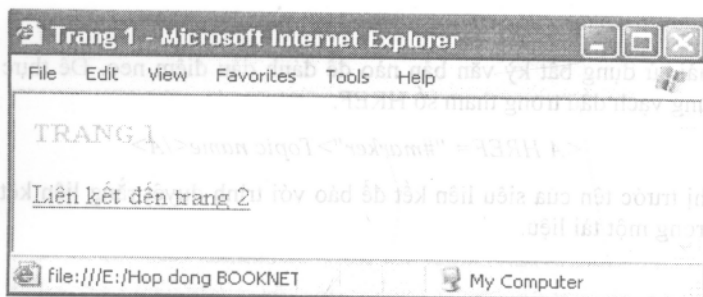
- Khi xem trang web, người dùng có thể chuyển theo một liên kết đến tài liệu khác, bạn có thể cung cấp liên kết để trở về trang chủ hoặc đến một trang khác.

```
<HTML>
<HEAD>
<TITLE> Trang 2</TITLE>
</HEAD>
<BODY>
<H1><FONT SIZE = 3 COLOR = HOTPINK>TRANG 2 </FONT></H1>
<P>
<A HREF= "Trang1.htm">Liên kết đến trang 1</A>
</BODY>
</HTML>
```

- Nếu để ý, chúng ta sẽ thấy các liên kết được gạch chân. Trình duyệt làm việc đó cho bạn. Nó cũng sẽ đổi hình dáng con trỏ khi di chuyển chuột qua liên kết.

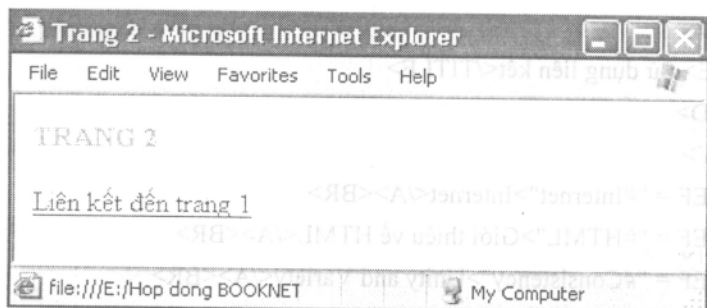
Chương 3

Ngôn ngữ cơ bản cho Web - HTML



Hình 3.14. Kết quả trang thứ nhất trên trình duyệt.

- Trình duyệt hiển thị trang 2 sau khi click vào liên kết



Hình 3.15. Kết quả trang thứ hai trên trình duyệt.

Ở ví dụ trên, các file nằm trong cùng một thư mục và vì vậy chỉ cần chỉ ra tên file trong thông số *HREF* là đủ. Tuy nhiên, khi liên kết đến các file trong các thư mục khác ta cần xác định tên đường dẫn tuyệt đối và tương đối.

- Tên đường dẫn tuyệt đối: chỉ đường dẫn đầy đủ từ file đến thư mục gốc. Ví dụ, `C:\mydirectory\html examples\Trang1.htm`.
- Tên đường dẫn tương đối: chỉ vị trí của file liên quan đến vị trí của file hiện thời. Ví dụ, nếu thư mục hiện thời là `mydirectory`, thì tên đường dẫn sẽ là:

```
<A HREF= "...\\Trang1.htm">Trang tiếp</A>
```

Liên kết trong

Chúng ta có thể sử dụng thẻ neo cho phép người dùng chuyển đến các phần khác nhau của một tài liệu. Ví dụ, bạn có thể hiển thị nội dung của trang web như một loạt các liên kết mô tả chủ đề. Khi người dùng nhấp vào một chủ đề nào đó, các chi tiết nằm trong một phần khác của tài liệu sẽ được hiển thị. Loại neo này được gọi là một neo được đặt tên vì thuộc tính "*NAME*" được sử dụng để tạo các liên kết này.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

`<A NAME= "marker">Topic name</A>`

Bạn không phải sử dụng bất kỳ văn bản nào để đánh dấu điểm neo. Để thực hiện chuyển đến, hãy sử dụng vạch dấu trong tham số HREF.

`<A HREF= "#marker">Topic name</A>`

được hiển thị trước tên của siêu liên kết để báo với trình duyệt rằng liên kết là một điểm được đặt tên trong một tài liệu.

Ví dụ 3.11.

Quan sát ví dụ về liên kết qua trang HTML sau:

```
<HTML>
<HEAD>
<TITLE> Sử dụng liên kết</TITLE>
</HEAD>
<BODY>
<A HREF = "#Internet">Internet</A><BR>
<A HREF = "#HTML">Giới thiệu về HTML</A><BR>
<A HREF = "#Consistency">Unity and Variety</A><BR>
<A name = "Internet">Internet</A><BR>
<P>Đoạn văn bản chứa nội dung về Internet</P>
<A name = "HTML">Giới thiệu về HTML</A>
<P>Đoạn văn bản chứa nội dung về HTML</P>
<A name = "Consistency">Unity and Variety</A><BR>
<P>Đoạn văn bản chứa nội dung về tính thống nhất và tính đa hình</P>
</BODY>
</HTML>
```

Liên kết đến vị trí nằm trong một tài liệu khác

Để chuyển đến một vị trí nằm trong một tài liệu khác, phải xác định tên của tài liệu đó khi tạo dấu hiệu. Trước tiên trình duyệt đọc tên tài liệu và mở tài liệu sau đó sẽ đọc dấu hiệu và chuyển đến vị trí đánh dấu này.

Các bước thực hiện:

- Ở trang 1 bạn sử dụng thẻ liên kết dạng:

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

`<A HREF = "trang1.htm#Internet">Internet</A>`

- Ở trang 2 chỉ cần đặt tên điểm neo:

`<A name = "Internet">Internet</A>`

Sử dụng E-mail

Nếu muốn người dùng gửi e-mail cho bạn, bạn có thể đưa một đặc tính vào trong trang web cho phép họ gửi e-mail từ trình duyệt. Tất cả những gì bạn cần làm là chèn giá trị *mailto* vào thẻ liên kết.

`<A HREF= "mailto:thisperson@mymail.com">`

Nếu bạn muốn đặt liên kết e-mail ở cuối trang, hãy sử dụng thẻ `<ADDRESS>` để siêu liên kết được hiển thị cuối trang Web.

Thay đổi màu liên kết

Bằng cách cung cấp các liên kết trong tài liệu HTML, bạn giúp người dùng điều hướng quanh web site hoặc ngay cả quanh cùng một tài liệu nếu đó là một tài liệu dài. Sau một lúc, người dùng có thể mất dấu vết của trang hoặc đề tài mà họ đã "xem". Bạn có thể đề nguyên như vậy, hoặc bạn có thể cung cấp một đặc điểm để chỉ liên kết đã được nhấp vào. Để làm như vậy, bạn phải thêm màu vào khu vực định dạng liên kết.

- *LINK*: là các liên kết trong trang khi hiển thị lần đầu, người dùng chưa chuyển đến lần nào. Màu mặc định là xanh da trời.
- *ALINK*: là các liên kết hoạt động. Khi người dùng nhấp vào liên kết, nó có thể đổi màu.
- *VLINK*: là liên kết đã được đến. Màu mặc định là tím, nhưng có thể chọn màu khác.

Ví dụ 3.12.

Ví dụ về đổi màu liên kết.

```
<HTML>
<HEAD><TITLE> Tô màu liên kết</TITLE></HEAD>
<BODY LINK = blue ALINK = HOOTPINK>
<H1><FONT SIZE = 3 COLOR = HOTPINK>TRANG 1
</FONT></H1>
<P>
<A HREF = "Trang2.htm">Liên kết đến trang 2</A>
</BODY>
</HTML>
```

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

3.13. Thiết kế bảng

Bạn có thể sử dụng bảng với các hàng và cột để hiển thị dữ liệu. Bảng giúp chúng ta điều khiển vị trí của văn bản và ảnh trên trang Web, thay vì giao tất cả các việc đó cho trình duyệt. Thẻ `<TABLE>` được sử dụng để tạo bảng trong tài liệu HTML.

- Các thuộc tính của phần tử `<TABLE>` áp dụng cho bảng, chứ không phải cho dữ liệu được hiển thị trong bảng.
- Đơn vị cơ bản của bảng là ô, và được định nghĩa bằng thẻ `<TD>`. Còn hàng được xác định bằng cách sử dụng thẻ `<TR>`.
- Các ô tạo thành hàng. Các hàng tạo thành bảng. Thẻ `<TD>` được lồng trong thẻ `<TR>`. Thẻ `<TR>` nằm trong thẻ `<TABLE></TABLE>`.
- Phần tử `CAPTION` được sử dụng để thêm phần chú giải cho bảng. `CAPTION` nằm ngay sau thẻ `<TABLE>` bắt đầu.
- Sử dụng thẻ `<TH>` để xác định tiêu đề cho cột.
- Thuộc tính `BORDER` có thể được sử dụng để xác định ô và cấu trúc của bảng. Thuộc tính này chỉ độ rộng của đường viền. Nếu giá trị được đặt là 0 thì đường viền sẽ không hiển thị.

Ví dụ 3.13.

Ví dụ về sử dụng bảng trong HTML.

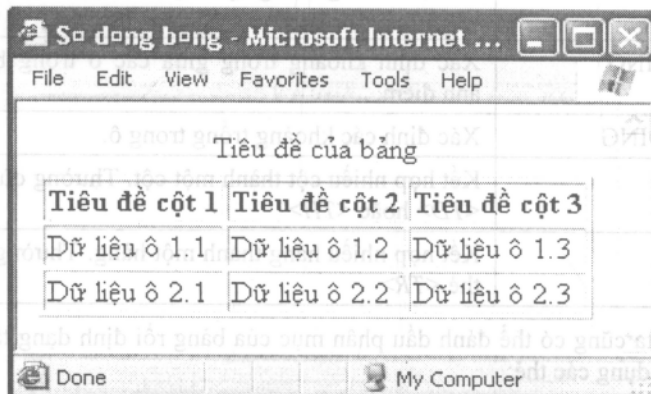
```
<HTML>
<HEAD>
<TITLE>Sử dụng bảng</TITLE>
</HEAD>
<BODY>
<TABLE BORDER = 1>
<CAPTION>Tiêu đề của bảng</CAPTION>
<TH>Tiêu đề cột 1</TH>
<TH>Tiêu đề cột 2</TH>
<TH>Tiêu đề cột 3</TH>
  <TR>
    <TD>Dữ liệu ô 1.1</TD>
    <TD>Dữ liệu ô 1.2</TD>
```

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

```
<TD>Dữ liệu ô 1.3</TD>
<TR>
<TD>Dữ liệu ô 2.1</TD>
<TD>Dữ liệu ô 2.2</TD>
<TD>Dữ liệu ô 2.3</TD>
</TABLE>
</BODY>
</HTML>
```

- Kết quả hiển thị trên trình duyệt:



Hình 3.16. Kết quả tạo bảng hiển thị trên trình duyệt.

Định dạng nội dung bảng

Sử dụng các thuộc tính trong bảng sau để định dạng bảng.

Bảng 3.12. Định dạng bảng

Thuộc tính	Giải thích
ALIGN	Căn lề ngang trong các ô của bảng. Nhận các giá trị <i>left</i> , <i>center</i> , <i>right</i> , <i>justify</i>
VALIGN	Căn lề đứng trong các ô của bảng. Nhận các giá trị <i>top</i> , <i>middle</i> , <i>bottom</i>
WIDTH	Kích thước rộng của bảng. Có thể nhận giá trị theo đơn vị % hoặc pixel.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

HEIGHT	Kích thước cao của bảng. Có thể nhận giá trị theo đơn vị % hoặc pixel.
BGCOLOR	Tô màu nền cho bảng. Giá trị theo hệ đếm thập lục phân.
BORDERCOLOR	Tô viền cho bảng
BORDERCOLORDARK	Tô viền có bóng đổ
BACKGROUND	Gán ảnh nền cho bảng hoặc ô. Giá trị là URL của ảnh
HSPACE	Thêm khoảng trắng (pixel) vào hai bên trái phải của bảng
VSPACE	Thêm khoảng trắng (pixel) vào bên trên, dưới của bảng
CELLSPACING	Xác định khoảng trống giữa các ô trong bảng bằng ảnh điểm.
CELLPADDING	Xác định các khoảng trống trong ô.
COLSPAN	Kết hợp nhiều cột thành một cột. Thường dùng với thẻ <code><TD></code> hoặc <code><TH></code>
ROWSPAN	Kết hợp nhiều hàng thành một hàng. Thường dùng với thẻ <code><TR></code>

Ngoài ra, chúng ta cũng có thể đánh dấu phân mục của bảng rồi định dạng tất cả cùng một lúc bằng cách sử dụng các thẻ:

`<THEAD>` Tiêu đề của tất cả các cột `</THEAD>`

`<TBODY>` Nội dung bảng `</TBODY>`

`<TFOOT>` Nội dung cuối cùng của bảng `</TFOOT>`

Việc sử dụng bảng linh hoạt sẽ giúp chúng ta quản lý vị trí và bố cục trang Web một cách hiệu quả. Chúng ta có thể kết hợp tất cả các thẻ định dạng cho văn bản hay liên kết trong từng ô của bảng cũng như chèn thêm các ảnh vào đó để có được một giao diện tốt nhất.

Ví dụ 3.14.

- Tạo trang HTML sau

```
<HTML>
<HEAD>
<TITLE>Trình bày nội dung bảng</TITLE>
</HEAD>
```

Chương 3
Ngôn ngữ cơ bản cho Web - HTML

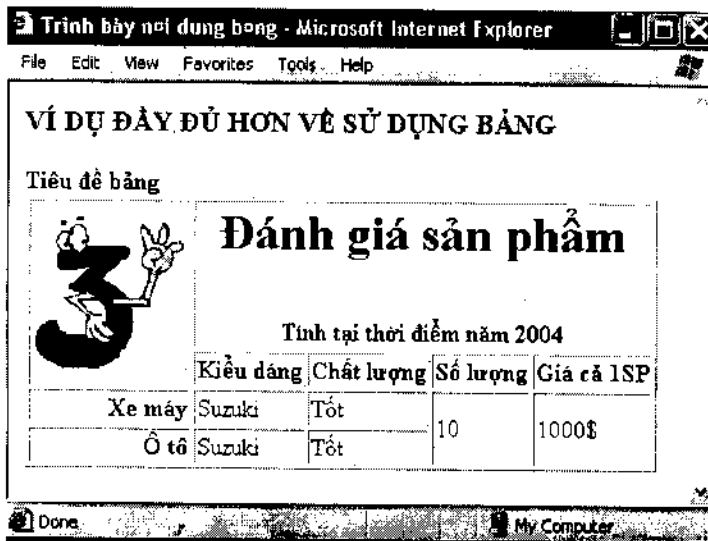
```
<BODY>
<H3>VÍ DỤ ĐẦY ĐỦ HƠN VỀ SỬ DỤNG BẢNG </H3>
<TABLE BORDER= "1">
<CAPTION><STRONG>Tiêu đề bảng</STRONG></CAPTION>
<COLGROUP>
<COL ALIGN= "right" SPAN= "1" />
</COLGROUP>
<THREAD>
<TR>
<TH ROWSPAN= "2">
<IMG SRC= "H3Y.gif" WIDTH= "100"
HEIGHT= "100" ALT= "Number Three!" />
</TH>
<TH COLSPAN= "4" VALIGN= "top">
<H1>Đánh giá sản phẩm</H1><br />
<P>Tính tại thời điểm năm 2004</P>
</TH>
</TR>
<TR VALIGN= "bottom">
<TH>Kiểu dáng</TH>
<TH>Chất lượng</TH>
<TH>Số lượng</TH>
<TH>Giá cả 1SP</TH>
</TR>
</THEAD>
<TBODY>
<TR>
<TH>Xe máy</TH>
<TD>Suzuki</TD>
```

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

```
<TD>Tốt</TD>
<TD ROWSPAN = "2">10</TD>
<TD ROWSPAN = "2">1000$</TD>
</TR>
<TR>
<TH>Ô tô</TH>
<TD>Suzuki</TD>
<TD>Tốt</TD>
</TR>
</TBODY>
</TABLE>
</BODY>
</HTML>
```

- Hiện thị trên trình duyệt:



Hình 3.17. Tạo bảng phức tạp.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Tăng tốc độ hiển thị bảng

Dù bảng cực kỳ mạnh nhưng chúng lại xuất hiện quá chậm chạp trong trình duyệt của người xem. Lý do chính là vì trình duyệt phải tính toán chiều rộng và chiều cao của bảng trước khi bắt đầu hiển thị ô. Vì thế có thể giảm thiểu phép tính toán của trình duyệt để bảng hiển thị nhanh hơn theo phương pháp sau:

- Tạo bảng càng nhỏ càng tốt.
- Định rõ kích thước bảng theo đơn vị Pixel
- Chia bảng thành nhiều nhóm cột.

3.14. Tạo khung

Khung chia cửa sổ của trình duyệt web thành các vùng riêng biệt, mỗi vùng hiển thị một trang riêng biệt và có thể sử dụng thanh cuộn. Khung được tạo ra bằng cách sử dụng thẻ `<FRAMESET>`.

Các thuộc tính như sau.

Bảng 3.13. Thuộc tính tạo khung

Thuộc tính	Mô tả
Rows	Xác định cách trình bày khung ngang theo đơn vị: điểm (pixel), phần trăm hoặc biến (có dấu *). Giá trị mặc định là 100%, nghĩa là một dòng.
Cols	Xác định cách trình bày khung dọc theo đơn vị điểm (pixel), phần trăm và hoặc biến (có dấu *). Giá trị mặc định là 100%. nghĩa là chỉ có một cột.

- *Phần tử FRAME* xác định hình thức và nội dung của một khung đơn với một `<FRAMESET>`.

Ví dụ 3.15.

- Tài liệu sau tạo ra 2 khung hình chữ nhật bằng nhau và lấp kín cửa sổ chính.

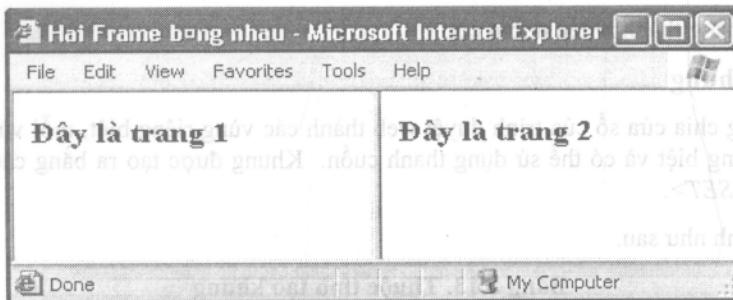
```
<HTML>
<HEAD>
<TITLE>Hai Frame bằng nhau</TITLE>
</HEAD>
<!--Tạo ra hai cột -->
```


Chương 3

Ngôn ngữ cơ bản cho Web - HTML

```
<FRAMESET COLS= "50%, *">
<FRAME SRC="f1.htm">
<FRAME SRC="f2.htm">
</FRAMSET>
</HTML>
```

- Hiện thị trên trình duyệt:



Hình 3.18. Ví dụ tạo 2 khung trong một trang.

Các thuộc tính của phần tử FRAME bao gồm:

Bảng 3.14. Thuộc tính của phần tử FRAME

Thuộc tính	Mô tả
NAME	Thuộc tính này gán tên cho khung hiện thời.
SRC	Xác định vị trí của tài liệu ban đầu nằm trong khung.
NORESIZE	Quy định cửa sổ khung không/có được đổi kích thước
SCROLLING	Thuộc tính này xác định các kiểu cuộn cho cửa sổ khung. Các giá trị có thể: ▪ Auto: cung cấp thanh cuộn cho cửa sổ khung khi cần thiết. Đây là giá trị mặc định. ▪ Yes: luôn cung cấp các thanh cuộn cửa sổ khung. ▪ No: không cho phép thanh cuộn cửa sổ khung.
FRAMEBORDER	Xác định viền khung. Các giá trị có thể là: ▪ 1 - giá trị mặc định. Dòng kẻ phân cách được vẽ giữa các khung hiện thời và mỗi khung nối liền. ▪ 0 - Giữa khung hiện thời và mỗi khung nối liền không có

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

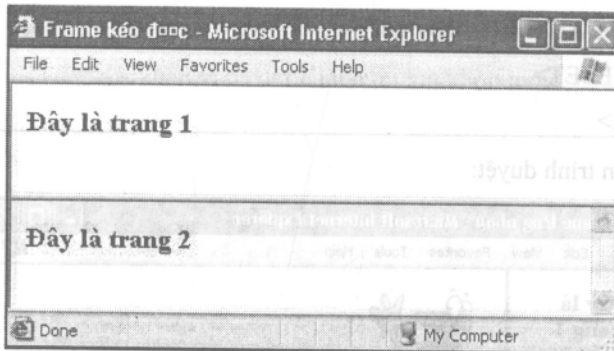
	dòng kẻ phân cách nào. Tuy nhiên, vẫn có thể hiển thị nếu được thiết lập.
MARGINWIDTH	Xác định khoảng trống được hiển thị giữa nội dung của khung và lề phải hoặc trái của khung.
MARGINHEIGHT	Xác định vị trí được hiển thị giữa nội dung của khung và lề trên, dưới của khung.

Ví dụ 3.16.

- Tạo trang HTML sau:

```
<HTML>
<HEAD><TITLE>Frame kéo được</TITLE></HEAD>
<FRAMESET ROWS= "*,*">
    <FRAME SRC= "f1.htm" SCROLLING= "no">
    <FRAME SRC= "f2.htm" SCROLLING= "yes">
</FRAMESET>
</HTML>
```

- Hiển thị trên trình duyệt:



Hình 3.19. Ví dụ tạo 2 khung có thể co giãn kích thước.

Chú ý:

Bạn không thể sử dụng phần tử đặt BODY và FRAMESET cùng với nhau. Trình duyệt nhận phần tử đầu tiên xuất hiện trong tài liệu và bỏ qua các phần tử sau. Nghĩa là, nếu bạn sử dụng phần tử BODY, thì phần tử FRAMESET sau đó sẽ bị bỏ qua và ngược lại.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Các khung lồng nhau (Nested framesets)

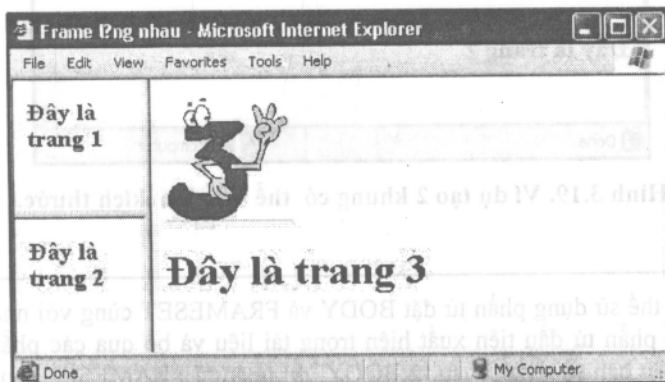
Bạn cũng có thể tạo ra các khung lồng nhau nhưng tuyệt đối phải chú ý là phải có thẻ đóng `</FRAMESET>`, nếu không trình duyệt Netscape sẽ không hiểu được.

Ví dụ 3.17.

Ví dụ sau trình bày cách kết hợp hai khung lồng nhau:

```
HTML>
<HEAD>
<TITLE>Frame lồng nhau</TITLE>
</HEAD>
<FRAMESET cols= "20%, 80%"
<!-- Chỉ rõ kích thước bằng Pixel -->
  <FRAMESET rows= "100, 200">
    <FRAME src= "f1.htm" noresize>
    <FRAME src= "f2.htm">
  </FRAMESET>
  <FRAME src= "h3y.gif">
</FRAMESET>
</HTML>
```

- Hiện thị trên trình duyệt:



Hình 3.20. Sử dụng khung lồng nhau.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

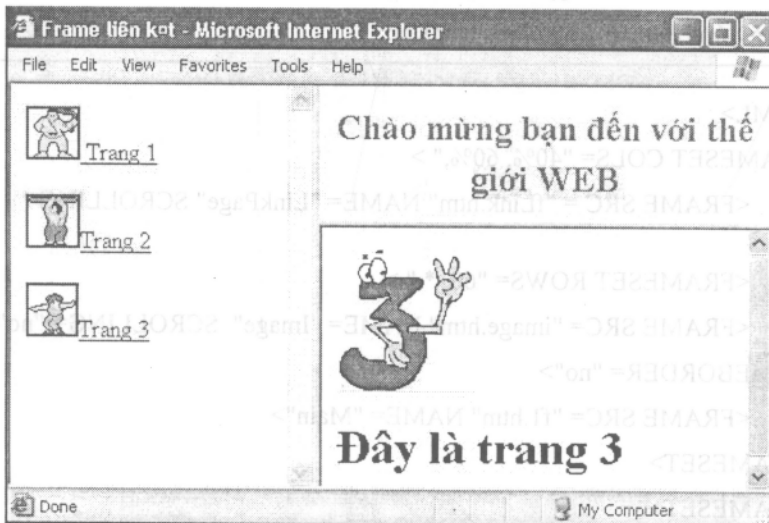
Ví dụ 3.18.

Trang HTML sau tạo ra 3 cột: cột 2 có độ rộng cố định là 250 điểm, cột 1 chiếm 25% khoảng trống còn lại và cột 3 chiếm 75% khoảng trống còn lại.

```
<HTML>
<HEAD>
<TITLE>Using frames</TITLE>
</HEAD>
<FRAMESET cols="1*, 250, 3*">
  <FRAME src="x.htm" frameborder=0>
  <FRAME src="scroll2.htm" frameborder=0 marginWIDTH=25>
  <FRAME src="y.htm">
</FRAMESET>
</HTML>
```

Liên kết với khung

Trong trang web bạn có thể tạo một khung như là mục tiêu của liên kết, như hình 3.21 sau.



Hình 3.21. Liên kết các khung.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Trước khi thực hành tạo liên kết khung, hãy chú ý một số quy định mà trình duyệt tuân theo:

- Nếu xác định một khung trong thuộc tính đích – *TARGET* - của một phần tử, thì tài liệu được phần tử xác định sẽ được tải vào trong khung khi phần tử hoạt động.
- Nếu thuộc tính *TARGET* không được đặt, thì thuộc tính *TARGET* của phần tử *BASE* sẽ được sử dụng để xác định khung.
- Nếu cả phần tử và phần tử *BASE* đều không đề cập đến *target*, thì tài liệu được tải vào khung có chứa phần tử.
- Nếu không tìm thấy khung, trình duyệt sẽ tạo một cửa sổ và khung mới và tải tài liệu vào khung mới đó.
- Thuộc tính *TARGET* sử dụng để chỉ tên của khung trong đó tài liệu nên được mở.
- Khi tạo khung, phải đặt thuộc tính tên. Tên này được sử dụng khi tạo liên kết.
- Sau khi thay đổi nội dung của một khung thì định nghĩa *FRAMESET* ban đầu sẽ mất.
- Nếu có nhiều liên kết chỉ cùng một *TARGET*, có thể đặt một *TARGET* mặc định trong phần tử *BASE*. Sau đó, việc xác định thuộc tính *TARGET* trong mỗi phần tử sẽ không cần thiết nữa.

Ví dụ 3.19.

Ví dụ sau sẽ minh họa kết quả tạo ra trong hình 3.21. Chúng ta có các trang: *Main.htm*, *fLink.htm*, các trang *f1.htm*, *f2.htm*, *f3.htm*

- Trang *Main.htm* có mã HTML như sau:

```
<HTML>
<FRAMESET COLS= "40%, 60%," >
    <FRAME SRC = "fLink.htm" NAME= "LinkPage" SCROLLING=
"yes" >
    <FRAMESET ROWS= "60, * " >
        <FRAME SRC= "image.htm" NAME= "Image" SCROLLING= "no"
FRAMEBORDER= "no">
        <FRAME SRC= "f1.htm" NAME= "Main">
    </FRAMESET>
</FRAMESET>
</HTML>
```

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

- Trang *fLink.htm* có mã HTML như sau:

```
<HTML>
<BODY>
<!-- Chỉ định tên khung hiển thị là "Main" -->
<BASE TARGET= "Main">
<A HREF= "f1.htm"><IMG SRC= "image1.gif"> Trang 1</A><P>
<A HREF= "f2.htm"><IMG SRC= "image2.gif">Trang 2</A><P>
<A HREF= "f3.htm"><IMG SRC= "image3.bmp">Trang 3</A><P>
</BODY>
</HTML>
```

- Các trang *f1.htm*, *f2.htm*, *f3.htm*:

f1.htm	f2.htm	f3.htm
<HTML>	<HTML>	<HTML>
<BODY>	<BODY>	<BODY>
<H3>Đây là trang 1	<H3>Đây là trang 2	
</H3>	</H3>	<H1> Đây là trang 3 </H1>
</BODY>	</BODY>	</BODY>
</HTML>	</HTML>	</HTML>

- Trang *image.htm*:

```
<HTML>
<BODY>
<H1>
<CENTER><FONT SIZE = 5 COLOR = RED> Chào mừng bạn đến với thế
giới WEB </FONT></CENTER>
</H1>
</BODY>
</HTML>
```

- Kết quả hiển thị trên trình duyệt xem hình 3.21.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Tạo nội dung thay thế khung

Mặc dù hai trình duyệt Netscape và Internet Explorer đã có khả năng hiển thị khung từ phiên bản 2, đồng thời khung chính thức được đưa vào chuẩn định dạng HTML ở phiên bản 4, nhưng một số ít trình duyệt vẫn từ chối hỗ trợ chúng. Trong trường hợp này, chúng ta nên cung cấp nội dung thay thế để hiển thị bằng cách: gõ `<NOFRAME>` ngay trước thẻ `</FRAMESET>` rồi tạo nội dung sẽ xuất hiện nếu khung không hiển thị và gõ tiếp `</NOFRAMES>`.

Ví dụ 3.20

Ví dụ tạo nội dung thay thế khung.

```
<NOFRAMES>
```

Khung không được hiển thị. Nhấp vào `<A href= "main.htm">` để có một phiên bản không có khung.`</A>`

```
</NOFRAMES>
```

Khung trong dòng: phần tử IFRAME

Phần tử IFRAME được sử dụng để ký hiệu khung nổi hoặc khung trong dòng. Bạn có thể tạo, chèn một khung vào một khối văn bản. Khi trình duyệt không hỗ trợ khung thì các nội dung nằm trong thẻ IFRAME sẽ bị trả lại. Khung trong dòng không thể thay đổi kích thước.

Các thuộc tính của IFRAME bao gồm:

Bảng 3.15. Thuộc tính của IFRAME

Thuộc tính	Mô tả
NAME	Gán tên cho khung hiện thời
WIDTH	Chỉ độ rộng của khung trong dòng
HEIGHT	Chỉ chiều cao của khung trong dòng

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

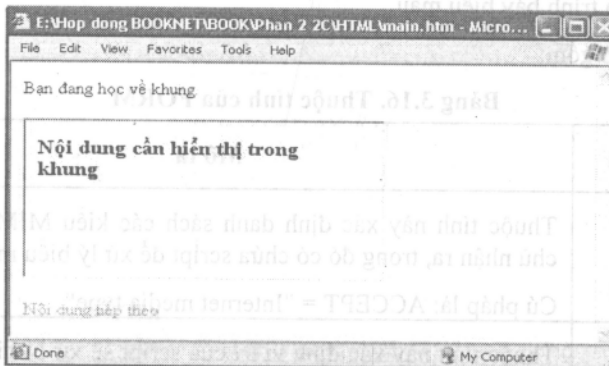
Ví dụ 3.21.

- Tạo trang HTML sau:

```
<HTML>
<P>

Bạn đang học về khung.
<BR>
<BR>
<IFRAME SRC= "f1.htm" WIDTH= "300" HEIGHT= "150"
SCROLLING= "auto" FRAMEBORDER= "1">
Nếu trình duyệt không hỗ trợ hoặc bạn không nhìn thấy khung, hãy
<A HREF = "f2.htm"> click vào đây </A>
</IFRAME>
<BR>
<P>
<FONT COLOR = hotpink>Nội dung tiếp theo</FONT>
</HTML>
```

- Kết quả hiển thị trên trình duyệt:



Hình 3.22. Một ví dụ về IFRAME.

Chú ý: Không thể thay đổi kích thước khung trong dòng.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

3.15. Sử dụng biểu mẫu (FORM)

Một biểu mẫu HTML là một phần của tài liệu chứa các phần tử đặc biệt gọi là các điều khiển (*control*). Các điều khiển được sử dụng để nhận thông tin vào từ người dùng và cung cấp một tương tác nào đó. Dữ liệu do người dùng nhập vào có thể có hiệu lực nhờ ngôn ngữ kịch bản (*script*) phía máy khách và được chuyển đến máy chủ để xử lý thêm.

Sử dụng các biểu mẫu

Các biểu mẫu trên World Wide Web khá nhiều và phát triển liên tục. Sau đây là một số cách sử dụng biểu mẫu thông thường:

- Thu thập tên, địa chỉ, số điện thoại, e-mail và các thông tin khác để đăng ký cho người dùng vào một dịch vụ,...
- Tập hợp thông tin để mua một mặt hàng nào đó. Ví dụ, nếu bạn muốn mua một quyển sách thông qua Internet, bạn cần phải điền tên, địa chỉ gửi thư bưu điện, phương thức thanh toán và các thông tin liên quan khác.
- Thu thập thông tin phản hồi về một Web site. Hầu hết các site cung cấp dịch vụ đều khuyến khích khách hàng gửi thông tin phản hồi. Ngoài việc xây dựng mối quan hệ với khách hàng, đây còn là một nguồn thông tin để trao đổi hoặc cải tiến.
- Cung cấp công cụ tìm kiếm cho web site. Các site cung cấp thông tin khác nhau thường cung cấp cho người dùng một hộp tìm kiếm để cho phép họ định vị thông tin nhanh hơn.

Phần tử FORM (Biểu mẫu)

Phần tử `<FORM>` được sử dụng để tạo một biểu mẫu trên trang. Nó có các thuộc tính quy định cách trình bày biểu mẫu.

Các thuộc tính bao gồm:

Bảng 3.16. Thuộc tính của FORM

Thuộc tính	Mô tả
ACCEPT	Thuộc tính này xác định danh sách các kiểu MIME được máy chủ nhận ra, trong đó có chứa script để xử lý biểu mẫu. Cú pháp là: ACCEPT = "Internet media type"
ACTION	Thuộc tính này xác định vị trí của script sẽ xử lý biểu mẫu hoàn chỉnh và đã được gửi đi. Cú pháp là ACTION = "URL".

Chương.3

Ngôn ngữ cơ bản cho Web - HTML

METHOD	Thuộc tính này xác định phương thức dữ liệu được gửi đến máy chủ. Cú pháp là : METHOD = (GET POST): ▪ Nếu giá trị là GET thì trình duyệt sẽ tạo một câu hỏi chứa trang URL, một dấu hỏi và các giá trị do biểu mẫu tạo ra. Trình duyệt sẽ đổi script của câu hỏi thành kiểu được xác định trong URL để xử lý. ▪ Nếu giá trị là POST, dữ liệu trên biểu mẫu sẽ được gửi đến script như một khối dữ liệu. Người ta không sử dụng chuỗi câu hỏi.
--------	---

Ví dụ 3.22.

Để đưa một biểu mẫu đến chương trình "xử lý biểu mẫu" bạn có thể sử dụng phương pháp POST như sau:

```
<FORM ACTION="http://myweb/xuly.asp" METHOD="post">  
Nội dung bên trong và các thành phần sử dụng cùng với FORM  
</FORM>
```

Các thành phần sử dụng cùng với FORM

▪ Thẻ INPUT

Thẻ `<INPUT>` xác định loại và sự xuất hiện của điều khiển được hiển thị trên biểu mẫu. Các thuộc tính của phần tử là:

Bảng 3.17. Thuộc tính của thẻ INPUT

Thuộc tính	Mô tả
TYPE	Thuộc tính này chỉ loại phần tử. Bạn có thể chọn TEXT, PASSWORD, CHECKBOX, RADIO, SUBMIT, RESET, FILE, HIDDEN và BUTTON. Mặc định là TEXT.
NAME	Thuộc tính này chỉ tên của điều khiển. Ví dụ, nếu bạn có nhiều hộp văn bản trên một biểu mẫu, bạn có thể xác định chúng bằng cách đặt tên: TEXT1, TEXT2, hoặc bất kỳ tên nào mà bạn chọn. Phạm vi hoạt động của thuộc tính tên nằm trong phần tử FORM.
VALUE	Đây là một thuộc tính tùy chọn xác định giá trị ban đầu của điều

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

	kiển. Tuy nhiên, nếu TYPE là RADIO thì bạn bắt buộc phải xác định một giá trị.
SIZE	Thuộc tính này chỉ độ rộng ban đầu của điều khiển. Nếu TYPE là "text" hoặc "password" thì kích thước được xác định bằng các ký tự. Đối với các loại nhập khác, độ rộng được xác định bằng điểm.
MAXLENGTH	Thuộc tính này được sử dụng để chỉ số ký tự lớn nhất có thể nhập vào trong phần tử "text" hoặc "password". Mặc định là không giới hạn.
CHECKED	Đây là thuộc tính logic chỉ nút đang bật hoặc tắt. Thuộc tính này được sử dụng khi dữ liệu nhập là "radio" hoặc "checkbox".
SRC	SRC= "URL". Thuộc tính này được sử dụng khi bạn nhập vào một ảnh. Thuộc tính xác định vị trí của ảnh.
READONLY	Không cho phép nhập dữ liệu vào, chỉ cho phép đọc và lưu dữ liệu.
TABINDEX	TABINDEX = n , với n là con số cho biết thứ tự TAB (0->32767). Thông qua phím TAB, người dùng có thể di chuyển qua lại giữa các thành phần trong mẫu biểu từ đầu đến cuối.
ACCESSKEY	ACCESSKEY = c , trong đó c là ký tự phím tắt (có thể là số). Thuộc tính này cho phép chọn thành phần của biểu mẫu bằng phím tắt ($ALT + c$).

▪ Tạo hộp nhập - textbox

Hộp nhập (text box) có thể chứa một dòng văn bản tùy ý, nghĩa là, bất cứ những gì người xem muốn gõ vào và thường dành cho việc nhập tên, địa chỉ ...

Ví dụ 3.23.

```
<INPUT TYPE = "text" NAME = "name" VALUE = "value" SIZE = 20 MAXLENGTH = 10>
```

Trong đó:

- *name* là chuỗi ký tự nhận diện dữ liệu nhập vào đối với máy chủ (server).
- *value* là dữ liệu đầu tiên hiển thị trong hộp nhập.
- *SIZE* = n định kích thước hộp nhập trên biểu mẫu (tính bằng ký tự).
- *MAXLENGTH* = n : kích thước tối đa cho phép gõ vào (tính bằng ký tự)

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

▪ Tạo hộp mật mã

Các thuộc tính hộp mật mã (*password box*) giống hộp nhập. Khác biệt duy nhất là toàn bộ nội dung gõ vào hộp mật mã đều bị thay bằng dấu chấm hoặc dấu sao (*). Tuy nhiên thông tin này không bị mã hóa khi gửi đến máy chủ.

Ví dụ 3.24.

```
<INPUT TYPE = "password" NAME = "name" VALUE = "value" SIZE = 9 MAXLENGTH = 10>
```

▪ Tạo vùng văn bản lớn hơn

Trong vài trường hợp, bạn muốn cung cấp thêm phạm vi nhập thông tin rộng hơn cho người xem. Trong trường hợp này chúng ta sẽ sử dụng thẻ TEXTAREA. Khác với hộp nhập, vùng nhập văn bản (*text area*) có thể chiếm nhiều dòng và có thể mở rộng nếu nội dung nhập vượt quá kích thước của nó.

Ví dụ 3.25.

```
<TEXTAREA NAME = "name" ROWS = 3 COLS = 50 WRAP>Thông tin thêm </TEXTAREA>
```

Trong đó:

- *name* là chuỗi ký tự nhận diện dữ liệu nhập vào đối với máy chủ (server).
- *ROWS* = *n*, với *n* là chiều cao của vùng văn bản tính theo hàng. Mặc định *n*=4.
- *COLS* = *n*, với *n* là chiều rộng của vùng văn bản tính bằng ký tự. Mặc định *n*=40.
- *WRAP* để khi người dùng gõ thông tin, những dòng này tự động xuống dòng trong phạm vi lề.

▪ Tạo nút radio

Nút *radio* thường được sử dụng trong trường hợp chỉ được lựa chọn duy nhất một đáp án (trường hợp) trong nhiều đáp án nêu ra. Nghĩa là không thể chọn cùng một lúc nhiều đáp án.

Ví dụ 3.26

```
<INPUT TYPE ="radio" NAME= "sex" VALUE= "male" CHECKED>Male
```

```
<INPUT TYPE ="radio" NAME= "sex" VALUE= "female">Female
```

Trong đó:

- *name* là chuỗi ký tự nhận diện dữ liệu nhập vào đối với máy chủ (server), đồng thời liên kết các nút radio với nhau, bảo đảm mỗi lần chỉ một nút được nhấp chọn.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

- *VALUE* = "data" xác định dữ liệu sẽ được server xử lý nếu nút radio được chọn.
- *CHECKED* sẽ kích hoạt nút radio theo trạng thái mặc định khi trang được mở.

▪ Tạo ô chọn

Trong khi nút **radio** chỉ cho phép chọn một đáp án trong một tập hợp thì **checkbox** lại cho phép bạn chọn được nhiều đáp án. Tương tự như nút radio, ô chọn liên kết với nhau thông qua giá trị của thuộc tính *NAME*.

Ví dụ 3.27.

```
<INPUT TYPE ="checkbox" NAME= "prefer" VALUE= "Thể thao">Thể thao  
<INPUT TYPE ="checkbox" NAME= "prefer" VALUE= "Ca nhạc">Ca nhạc  
<INPUT TYPE ="checkbox" NAME= "prefer" VALUE= "Học tập">Học tập
```

Gõ thêm từ *CHECKED* nếu bạn muốn mặc định một ô nào đó được tích chọn.

▪ Tạo menu

Menu là phương tiện cho phép người dùng lựa chọn từ một tập hợp các tùy chọn cho trước.

Ví dụ 3.28.

```
<SELECT NAME = "Khoa">  
<OPTION VALUE = "Công nghệ thông tin" SELECTED>Công nghệ  
thông tin  
<OPTION VALUE = "Tự động hóa">Tự động hóa  
<OPTION VALUE = "Điện tử viễn thông">Điện tử viễn thông  
<OPTION VALUE = "Cơ khí">Cơ khí  
</SELECT>
```

Trong đó:

- *VALUE* là thuộc tính xác định dữ liệu sẽ gửi đến máy chủ (server) nếu tùy chọn này được chọn.
- Gõ *SELECTED* để tùy chọn được theo mặc định.
- Có thể sử dụng thuộc tính *SIZE* = *n* để ấn định kích thước của menu (theo hàng).

▪ Cho phép người xem tải tập tin lên

Nếu thông tin do người xem điền vào mẫu biểu của bạn quá phức tạp, bạn nên cho phép họ tải toàn bộ tập tin lên server.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Ví dụ 3.29.

Ví dụ một form cho phép tải tập tin.

```
<FORM METHOD = "POST"
ACTION = http://www.yoursite.com/ASP/uploadfile.asp
ENCTYPE = "multipart/form-data">
Họ tên :<INPUT TYPE = "text" NAME = "hoten" SIZE = 20>
<P><B>Bạn muốn gửi những file gì?</B><BR>
<INPUT TYPE= "file" NAME= "files" SIZE = 40>
<INPUT TYPE = "submit" NAME ="submit">
</FORM>
```

Chú ý:

- Phải đảm bảo sử dụng phương thức *POST* trong thẻ *FORM*. Không được dùng *GET* cho trường hợp này.
- Đừng quên gán thuộc tính *ENCTYPE*, điều này bảo đảm cho tập tin được tải lên theo dạng thức thích hợp.

▪ Trường ẩn

Trong một số trường hợp, bạn muốn ẩn dữ liệu để gửi đi không hiển thị trên giao diện, bạn có thể sử dụng trường ẩn. Trình duyệt sẽ không hiển thị trường ẩn nhưng chúng góp phần vào mã HTML tạo nên trang Web.

Ví dụ 3.30.

```
<INPUT TYPE = "hidden" NAME ="name" VALUE ="value">
```

Trong đó:

- *name* sẽ được server sử dụng để truy cập dữ liệu.
- *value* chính là thông tin được lưu trữ.

▪ Tạo nút gửi dữ liệu

Toàn bộ thông tin do người sử dụng nhập vào chỉ có tác dụng khi nó được gửi đến máy chủ (server). Vì thế trên giao diện phải chú ý tạo nút gửi dữ liệu cho biểu mẫu để người sử dụng có thể chuyển thông tin đến cho bạn.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Ví dụ 3.31.

```
<INPUT TYPE = "submit" VALUE = "Gửi dữ liệu">
```

Trong đó:

- VALUE xác định tiêu đề của nút. Nếu bỏ qua thuộc tính này thì nút sẽ có tiêu đề mặc định là *Submit*.

▪ Tạo nút gửi dữ liệu có hình ảnh

Có thể sử dụng hình ảnh kết hợp với phần dữ liệu nhập và nút gửi dữ liệu.

Ví dụ 3.32.

```
<INPUT TYPE = "image" SRC = "button.gif" NAME = "name">
```

Trong đó:

- TYPE : Kiểu nút là image.
- SRC xác định vị trí của ảnh trên máy chủ (server).
- NAME: Tên nút. Khi người dùng click chuột vào hình ảnh, tọa độ x và y của chuột sẽ được ghi nối vào cuối tên và được gửi đến máy chủ.

▪ Tạo nút phục hồi

Đôi khi, người dùng nhập sai dữ liệu và muốn nhập lại từ đầu. Lúc đó ta phải hỗ trợ người sử dụng việc xóa hết dữ liệu đã nhập trước đó. Để làm được như vậy hãy sử dụng kiểu *reset* trong thuộc tính TYPE .

Ví dụ 3.33.

```
<INPUT TYPE = "reset" VALUE = "Nhập lại">
```

Trong đó:

- VALUE xác định tiêu đề nút. Nếu bỏ qua thuộc tính này thì nút sẽ có tiêu đề mặc định là *Reset*.

▪ Tạo nhãn cho các thành phần mẫu biểu

Bạn có thể chèn thêm thông tin giải thích bên cạnh các thành phần của mẫu biểu bằng cách sử dụng thẻ **<LABEL>**.

Ví dụ 3.34.

```
<LABEL FOR = "hoten">Họ tên:</LABEL>
```

```
<LABEL FOR = "email">E-mail address::</LABEL>
```

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Trong đó:

- FOR = "idname", với *idname* là giá trị của thuộc tính ID trong thành phần mẫu biểu tương ứng.

▪ Vô hiệu hóa các thành phần biểu mẫu

Đôi khi bạn không muốn người xem sử dụng vài thành phần nào đó của mẫu biểu. Ví dụ, bạn muốn vô hiệu hóa nút gửi dữ liệu cho đến khi tất cả các trường cần thiết trên biểu mẫu đều được điền xong. Thực hiện điều này bằng cách thêm thuộc tính *DISABLED* vào thành phần của nút.

Ví dụ 3.35.

```
<INPUT TYPE = "submit" VALUE = "Send" DISABLED>
```

Sau đó muốn thay đổi nội dung thành phần mẫu biểu, bạn phải sử dụng kịch bản trên máy chủ (server).

▪ Sắp xếp các thành phần của biểu mẫu

Để sắp xếp các thành phần của biểu mẫu, bạn hãy sử dụng bảng. Một kỹ năng không thể thiếu đó là sự kết hợp nhuần nhuyễn giữa các yếu tố của một trang Web bao gồm: "*bảng - chữ - hình ảnh - form*".

Ví dụ 3.36.

- Ví dụ tạo một trang đăng kí thông tin (hình 3.23) sau thể hiện rõ điều này:

```
<FORM ACTION="xuly.asp" METHOD="post" TARGET="_blank">
  <TABLE WIDTH="478" BORDER="0">
    <TR>
      <TD WIDTH="116">Họ đệm</TD>
      <TD COLSPAN="2"><INPUT NAME="hodem" TYPE="text"
        SIZE="40" TABINDEX="0"></TD>
      <TD WIDTH="98"><EM>Yêu cầu</EM></TD>
    <TR>
      <TD></TD>
      <TD COLSPAN="2"><INPUT NAME="ten" TYPE="text" SIZE="40"
        TABINDEX="1"></TD>
      <TD><EM>Yêu cầu</EM></TD>
```


Chương 3
Ngôn ngữ cơ bản cho Web - HTML

```
</TR>
<TR>
  <TD>E-Mail:</TD>
  <TD COLSPAN="2"><INPUT NAME="email" TYPE="text"
SIZE="40" TABINDEX="2"></TD>
  <TD><EM>Yêu cầu</EM></TD>
</TR>
<TR>
  <TD>Mật khẩu:</TD>
  <TD COLSPAN="2"><INPUT NAME="mk" TYPE="password"
SIZE="40" TABINDEX="3"></TD>
  <TD><EM>Yêu cầu</EM></TD>
</TR>
<TR>
  <TD>Xác nhận mật khẩu:</TD>
  <TD COLSPAN="2"><INPUT NAME="reMk" TYPE="password"
SIZE="40" TABINDEX="4"></TD>
  <TD><EM>Yêu cầu</EM></TD>
</TR>
<TR>
  <TD>Địa chỉ:</TD>
  <TD COLSPAN="2"><INPUT NAME="diachi" TYPE="text"
SIZE="40" TABINDEX="5"></TD>
  <TD>&nbsp;</TD>
</TR>
<TR>
  <TD>Thành phố:</TD>
  <TD COLSPAN="2"><INPUT NAME="thanhpho" TYPE="text"
SIZE="40" TABINDEX="6"></TD>
  <TD>&nbsp;</TD>
```

Chương 3

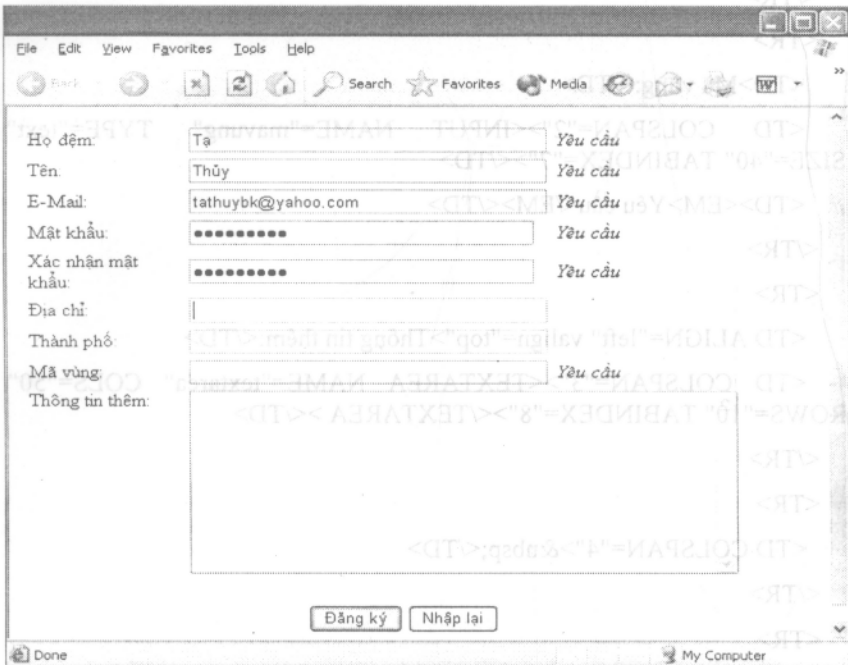
Ngôn ngữ cơ bản cho Web - HTML

```
</TR>
<TR>
  <TD>Mã vùng:</TD>
  <TD COLSPAN="2"><INPUT NAME="mavung" TYPE="text"
SIZE="40" TABINDEX="7"></TD>
  <TD><EM>Yêu cầu</EM></TD>
</TR>
<TR>
  <TD ALIGN="left" valign="top">Thông tin thêm:</TD>
  <TD COLSPAN="3"><TEXTAREA NAME="textarea" COLS="50"
ROWS="10" TABINDEX="8"></TEXTAREA ></TD>
</TR>
<TR>
  <TD COLSPAN="4">&nbsp;</TD>
</TR>
<TR>
  <TD>&nbsp;</TD>
  <TD WIDTH="135">
    <div align="right">
      <INPUT TYPE="submit" NAME="dangky" value="Đăng ký"
TABINDEX="9">
    </div></TD><TD WIDTH="101">
      <div align="left">
        <INPUT TYPE="reset" NAME="nhaplai" value="Nhập lại"
TABINDEX="10">
      </div></TD>
  <TD>&nbsp;</TD>
</TR>
</table>
</FORM>
```

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

- Kết quả hiển thị trên trình duyệt như sau:



The screenshot shows a web browser window with a menu bar (File, Edit, View, Favorites, Tools, Help) and a toolbar. The main content area displays a registration form with the following fields and labels:

Họ đệm:	Tạ	Yêu cầu
Tên:	Thủy	Yêu cầu
E-Mail:	tathuybk@yahoo.com	Yêu cầu
Mật khẩu:		Yêu cầu
Xác nhận mật khẩu:		Yêu cầu
Địa chỉ:		
Thành phố:		
Mã vùng:		Yêu cầu
Thông tin thêm:		

At the bottom of the form are two buttons: "Đăng ký" and "Nhập lại". The browser's status bar at the bottom shows "Done" and "My Computer".

Hình 3.23. Kết hợp bảng và Form.

3.16. Kết hợp khả năng đa phương tiện: âm thanh, hình ảnh

Một trong những điều làm Web trở nên phổ biến là khả năng có thể thêm hình ảnh., âm thanh, hoạt hình,... vào trang Web. Chúng ta sẽ tiếp tục tìm hiểu cách sử dụng đa phương tiện trong trang Web trong phần này.

Hình ảnh không được trình duyệt hỗ trợ

Các trình duyệt đều hỗ trợ hiển thị ảnh GIF, JPEG, PNG, nhưng với những loại hình ảnh khác bạn phải tạo một liên kết để xem bằng ứng dụng trợ giúp bên ngoài. Cách làm là tạo liên kết đến địa chỉ các hình ảnh không được trình duyệt hỗ trợ.

Ví dụ 3.37.

```
<A HREF = "image.tiff"><IMG SRC = "link.gif">225K Ảnh Tiff</A>
```

Âm thanh và video

Bạn có thể thêm âm thanh vào trang Web của mình bằng hai cách:

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

- Ở dạng tập tin được tải về toàn bộ cùng một lúc vào máy tính của người xem sau đó phát lại.
- Ở dạng tập tin liên tục tải mỗi lần một ít và bắt đầu phát lại trong khi vẫn tiếp tục tải.

Cả hai trình duyệt Netscape và IE đều kèm theo các *plug-in* cho phép phát âm thanh ngay trong trang Web mà không cần người xem phải hỗ trợ thêm. Sử dụng thẻ `<EMBED>` như sau:

```
<EMBED SRC="sound.url" CONTROLS="form" AUTOSTART=false LOOP=n  
ALIGN=direction>
```

Trong đó:

- *SRC="sound.url"* xác định đường dẫn đến tập tin âm thanh.
- *CONTROLS="form"*, ở đây *form* xác định hình thức của điều khiển âm thanh. Các tùy chọn khả dụng bao gồm *console*, *smallconsole*, *playbutton*, *pausebutton*, *stopbutton*, *columelevel*. Nếu chọn loại điều khiển, gõ thêm *WIDTH=w* *HEIGHT=h* để xác định kích thước của điều khiển.
- *AUTOSTART* cho phép để âm thanh chạy tự động hay không. Nhận hai giá trị *true* (có) và *false* (không).
- *LOOP = n*, để âm thanh tự động lặp lại *n* lần. Hoặc gõ *LOOP=true* để thực hiện lặp lại âm thanh cho đến khi người xem ấn nút Stop hoặc chuyển sang trạng thái khác.
- *ALIGN=direction*, canh chỉnh các điều khiển trên mạng. Cách dùng giống như hình ảnh.

Với định dạng film chúng ta cũng thực hiện tương tự.

Tạo liên kết đến các tập tin âm thanh và video

Cách dễ dàng để thêm âm thanh cũng như đoạn film vào trang Web là tạo liên kết đến tập tin âm thanh và đoạn film đó. Khi xem, người sử dụng phải nhấp chuột vào liên kết mới nghe được âm thanh hoặc xem được hình ảnh đó.

Ví dụ 3.38.

```
<A HREF = "sound.url"><IMG SRC="soundicon.gif">Nghe nhạc</A>
```

```
<A HREF = "movie.dat"><IMG SRC="movieicon.gif">Xem phim</A>
```

Thêm âm thanh nhạc nền cho Explorer

Internet Explorer có một thẻ đặc biệt cho phép liên kết tập tin âm thanh với trang Web và để cho tập tin tự động phát mỗi khi người xem chuyển đến trang này. Đó là thẻ `<BGSOUND>`.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Ví dụ 3.39.

Đoạn mã sau thực hiện phát bản nhạc lặp lại 2 lần:

```
<BGSOUND SRC="nhacnen.mdi" LOOP=2>
```

Nhúng các file hoạt hình Flash

Sử dụng thẻ **<EMBED>** kết hợp với đường dẫn đến định dạng file Flash (*.SWF) là bạn đã có thể có những trang Web với nhiều hình ảnh hoạt hình sinh động lấy từ Flash. Cách làm cũng tương tự như với tệp âm thanh.

Ví dụ 3.40.

```
<object classid="clsid:d27cbb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash
.cab#version=7,0,0,0" width="200" height="50" id="textbox" align="middle">
<param name="allowScriptAccess" value="sameDomain" />
<param name="movie" value="flashFile.swf" />
<param name="quality" value="high" />
<param name="bgcolor" value="#6699ff" />
<embed src="flashFile.swf" quality="high" bgcolor="#6699ff" width="200"
height="50" name="textbox" align="middle"
allowScriptAccess="sameDomain" type="application/x-shockwave-flash"
pluginspage="http://www.macromedia.com/go/getflashplayer" />
</object>
```

Đoạn mã trên nhúng một file FLASH tên là *flashFile.swf* với kích thước (w,h) là (200,50). Bạn chỉ cần chèn đoạn mã này vào trang HTML là tệp flash đó sẽ được đính kèm trang nhưng chú ý là *flashFile.swf* phải cùng vị trí của trang HTML.

3.17. Xuất bản Website của bạn

Sau khi hoàn tất việc xây dựng Website, bạn phải chuyển các trang HTML này đến máy phục vụ Website để mọi người sử dụng khác có thể truy cập được chúng.

Có rất nhiều nhà cung cấp dịch vụ Internet (ISP) và bạn có thể liên hệ để có được một nơi đặt Website của bạn. Tuy nhiên, bạn phải trả tiền cho họ. Bạn cũng có thể tìm thấy một số địa chỉ cho phép bạn thử nghiệm miễn phí hệ thống Website của bạn như *www.brinkster.com* hay *www.freedomain.com*,... Thông thường mỗi nhà cung cấp sẽ có một cách thức riêng để giúp bạn tải các trang web lên máy lưu trữ của họ và bạn chỉ cần sử dụng nó để tải vào đúng tên vùng mà họ cấp cho bạn. Khi đó trang Web của bạn sẽ có thể bắt đầu hoạt động trong thế giới Internet vô cùng rộng lớn và phong phú.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

Giúp người xem tìm thấy trang Web của bạn

Hiện nay, khi muốn tìm kiếm thông tin gì trên mạng, người sử dụng sẽ sử dụng dịch vụ tìm kiếm để tìm đến những trang thông tin cần truy cập. Như vậy, để nhiều người sử dụng sẽ biết đến Website của bạn hơn, chúng ta hãy cung cấp cho dịch vụ tìm kiếm biết thông tin về trang của bạn. Để làm được điều này làm như sau:

- Ở mục HEAD, gõ xâu kí tự “<META NAME =”keywords” CONTENT =”
- Gõ các từ ngắn gọn mô tả chủ đề thông tin trên trang Web của bạn. Phân cách mỗi từ bằng dấu phẩy và một ký tự trắng. Đây chính là các từ khóa cho các dịch vụ tìm kiếm nhận biết và sử dụng.
- Gõ kí tự “>” để hoàn tất thẻ META.

Ví dụ 3.41.

```
<HEAD><TITLE> Trang chủ của bạn! </TITLE>
<META NAME=”keywords” CONTENT=”tin3k46, H3Ygroup, ITCLUP”>
<META NAME=”description” CONTENT=”Trang Web của nhóm H3Y-
ITCLUP-DHBKHN”>
</HEAD>
```

Ngoài ra, bạn có thể sử dụng thêm một số thuộc tính của thẻ định dạng META như:

```
<META NAME=”author” CONTENT=”Tên tác giả đánh ở đây”>
```

```
<META NAME=”generator” CONTENT=”Tên phần mềm tạo trang HTML”>
```

để thông báo thêm cho người dùng biết thông tin tác giả và phần mềm thiết kế.

Phần công việc còn lại là bạn phải đăng kí site của mình cho dịch vụ tìm kiếm. Muốn vậy bạn phải tìm đến địa chỉ trang cho phép thêm URL của các dịch vụ này và đăng ký. Ví dụ: www.yahoo.com/docs/info/include.html

Chú ý:

Mặc dù dịch vụ tìm kiếm đã lập chỉ mục trang Web của bạn, nhưng chưa chắc người dùng đã quan tâm đến nó ngoại trừ nó nằm ở phần đầu của danh sách. Đây là một số cách để cải thiện vấn đề này:

- Càng bám sát với chủ đề mà người xem sẽ tìm kiếm càng tốt - đừng gõ thừa.
- Tạo tiêu đề trang. Tiêu đề càng khớp với nội dung mà người xem gõ vào thì xác suất hiển thị trang của bạn trong danh sách càng cao.

Chương 3

Ngôn ngữ cơ bản cho Web - HTML

- Thận trọng với từ khóa. Từ khóa càng khớp với nội dung mà người xem gõ vào thì xác suất tìm thấy càng cao.
- Trang phải ngắn gọn và từ ngữ khớp với tiêu chuẩn tìm kiếm.

3.18. Kết chương

Trong chương này chúng ta đã tìm hiểu về ngôn ngữ siêu văn bản HTML và làm quen với việc viết các trang Web bằng HTML. Trong chương sau, chúng ta sẽ làm quen với một công cụ cho phép soạn thảo các trang mã HTML một cách nhanh chóng, hiệu quả, hỗ trợ đầy đủ các phương thức, thuộc tính của mọi đối tượng trên một trang web HTML.

Chương 4

Macromedia Dreamweaver MX 2004

Nội dung của chương:

- Tìm hiểu cách sử dụng Dreamweaver MX 2004.
- Tạo ra các trang Web bằng Dreamweaver MX 2004.
- Tạo và áp dụng Style CSS bằng Dreamweaver MX 2004.
- Xử lý các sự kiện.

4.1. Mở đầu với Dreamweaver MX 2004

Dreamweaver là một công cụ soạn thảo trực quan chuyên nghiệp cho phép xây dựng và quản lý websites một cách thuận tiện. Dreamweaver được thiết kế cho phép người sử dụng soạn thảo, tạo các trang web với rất nhiều đặc tính mà không cần trực tiếp viết mã HTML. Nó được thiết kế theo kiểu 'WYSIWYG' (*What You See Is What You Get*).

Các đặc điểm của Dreamweaver:

- Sử dụng cùng thẻ HTML và JavaScript, trình gỡ rối JavaScript, cho phép người dùng soạn thảo trực tiếp mã JavaScript,...
- Trình soạn thảo HTML mạnh, với đặc điểm *Roundtrip HTML* - giữ nguyên mã HTML gốc.
- Có thể cài đặt để xoá và định dạng lại mã HTML nếu cần.
- Các đặc trưng soạn thảo trực quan cho phép thêm các thiết kế trang mà không cần viết mã.
- Khả năng kéo thả các đối tượng vào trang web nhanh chóng, thuận tiện.

Dreamweaver MX 2004 đã có nhiều thay đổi và cải tiến về kiểu dáng thiết kế cũng như môi trường làm việc. Hầu hết các tính năng chưa được hỗ trợ trong phiên bản trước nay đã có thể thực hiện được trong Dreamweaver MX 2004. Trong phần này của cuốn sách chúng

Chương 4

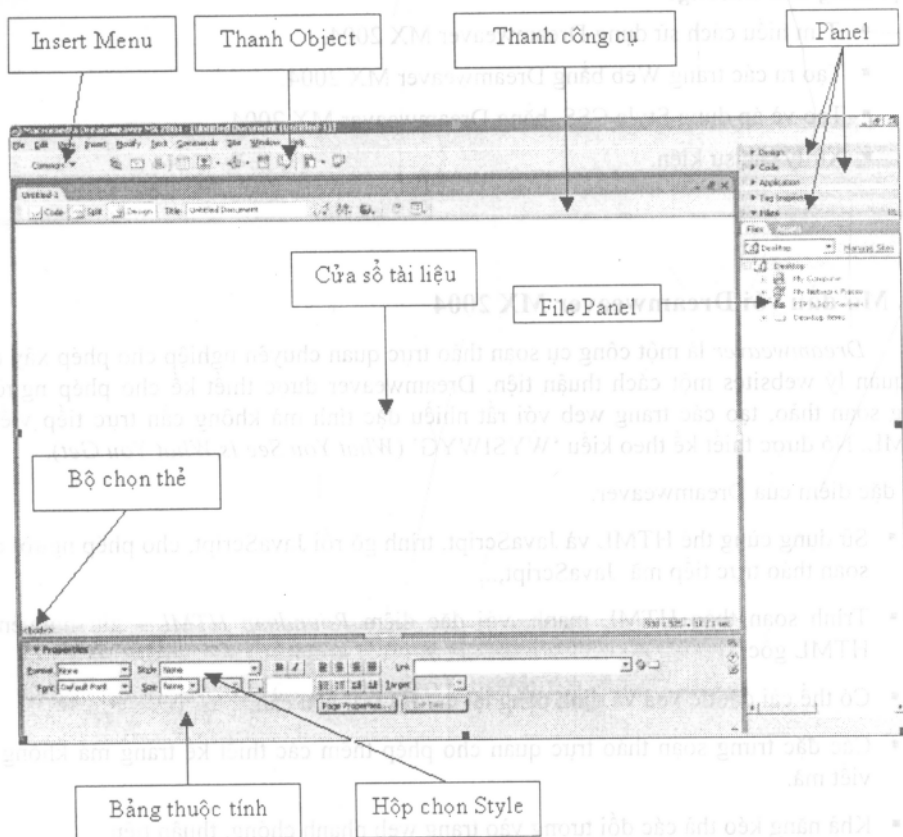
Macromedia Dreamweaver MX 2004

ta không đi vào chi tiết mà tập trung tìm hiểu những kỹ năng cơ bản nhất trong việc thiết kế trang Web thông qua các công cụ mà Dreamweaver cung cấp để không cần phải tốn một dòng mã nào.

Trước tiên chúng ta hãy cùng tìm hiểu giao diện làm việc của Dreamweaver MX 2004.

4.1.1. Tìm hiểu giao diện làm việc của Dreamweaver MX 2004

Khi khởi động Dreamweaver MX 2004, ta sẽ nhìn thấy giao diện làm việc của Dreamweaver như hình vẽ.



Hình 4.1. Vùng làm việc của Dreamweaver MX 2004.

- Cửa sổ **Document** hiển thị trang hiện thời giống như khi trang được mở trên trình duyệt web.

Chương 4

Macromedia Dreamweaver MX 2004

- **Thanh công cụ** chứa các nút và các menu thả xuống cho phép ta xem cửa sổ tài liệu theo các cách khác nhau, đặt các lựa chọn *view*, sử dụng một số tác vụ thường dùng.
- **Menu ngữ cảnh (Insert Menu)** cho phép ta nhanh chóng chọn các lệnh hữu ích liên quan đến lựa chọn hiện thời. Để hiển thị menu ngữ cảnh, click nút phải chuột trong cửa sổ *Document*.
- **Thanh Objects** chứa các nút cho phép chèn nhiều loại đối tượng như hình ảnh, bảng, lớp và *frame*. Có thể chuyển đổi giữa 2 chế độ xem *Standard* và *Layout* và truy cập các chức năng về ở chế độ *Layout*.
- **Bảng thuộc tính** hiển thị các đặc tính của đối tượng hoặc văn bản đang được chọn, và cho phép sửa đổi các đặc tính này.

Dreamweaver MX 2004 cung cấp nhiều hộp kiểm soát (*inspector*), *panel* và các cửa sổ khác như cửa sổ *History* và *Code Inspector*. Có thể mở các cửa sổ *inspector* và các *panel* bằng cách chọn từ menu **Window**.

4.1.2. Tạo và xóa một website

Quá trình thiết lập **Site** rất quan trọng bởi sau khi hoàn thành trên ổ cứng của bạn, bạn tải các trang lên máy chủ Web thì từng trang chỉ giữ lại định vị tương đối như có trên ổ cứng. Các tính năng quản lý site trong Dreamweaver bảo đảm mọi tác vụ được thực hiện trên máy chủ hoàn toàn giống như trên ổ cứng ban đầu của bạn. Theo cách này Dreamweaver không phá vỡ liên kết giữa các trang khi tải site lên máy chủ Web.

❖ Các bước để tạo một site mới:

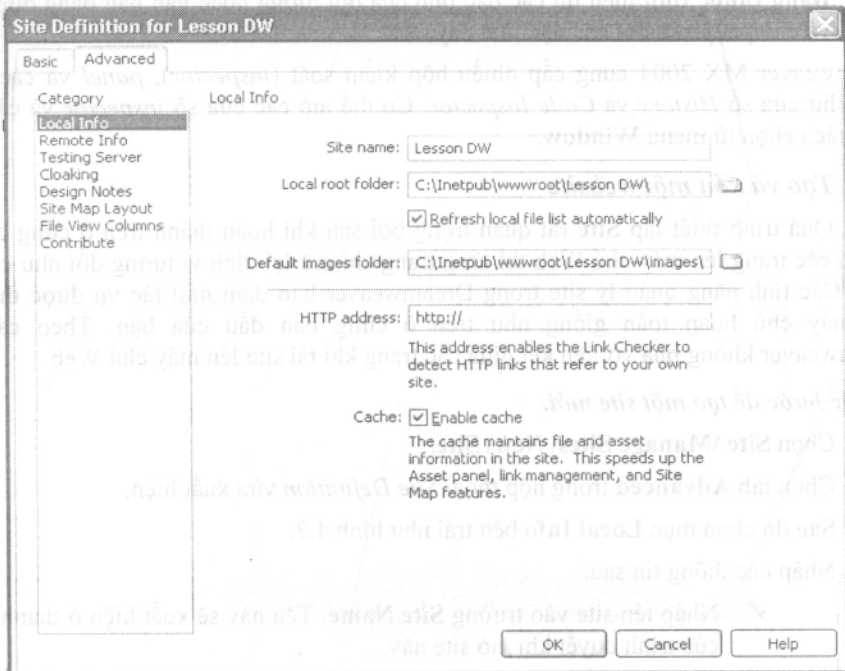
- Chọn **Site \Manage Sites\ New\ Site**.
- Chọn tab **Advanced** trong hộp thoại *Site Definition* vừa xuất hiện,
- Sau đó chọn mục **Local Info** bên trái như hình 4.2.
- Nhập các thông tin sau:
 - ✓ Nhập tên site vào trường **Site Name**. Tên này sẽ xuất hiện ở thanh tiêu đề của trình duyệt khi mở site này.
 - ✓ Trong trường **Local Root Folder**, xác định thư mục để lưu các file và các thư mục con của site. Có thể click biểu tượng thư mục để tìm thư mục cho site hoặc gõ trực tiếp đường dẫn đến thư mục này vào text box.
 - ✓ Tuỳ chọn **Refresh Local File List Automatically** cho phép tự động cập nhật danh sách tất cả các trang web bạn thêm vào trong site của mình. Bạn có thể *refresh* lại trang một cách thủ công bằng cách chọn nút *Refresh* ở cửa sổ hoặc chọn **View \ Refresh Design View (F5)**.
 - ✓ Trong trường **HTTP Address**, nhập địa chỉ URL của trang đã đăng ký với Web Server. *HTTP Address* là một URL hay còn gọi là một địa chỉ web mà site của bạn sẽ có khi nó được xuất lên máy chủ web. Nếu bạn

Chương 4

Macromedia Dreamweaver MX 2004

chưa biết địa chỉ web cho site của bạn hoặc bạn không có ý định xuất nó lên một máy chủ web, thì trường này để trống.

- ✓ Tuy chọn **Cache** để xác định có tạo bộ đệm cục bộ để cải thiện tốc độ liên kết và các tác vụ quản lý site hay không. Nên chọn tùy chọn này.
- ✓ Nhấn **OK** để đóng hộp thoại *Site Definition*. Cửa sổ *Site* xuất hiện. Sau này, khi đã sẵn sàng đưa Web site lên server, ta sẽ thêm các thông tin khác cho site.

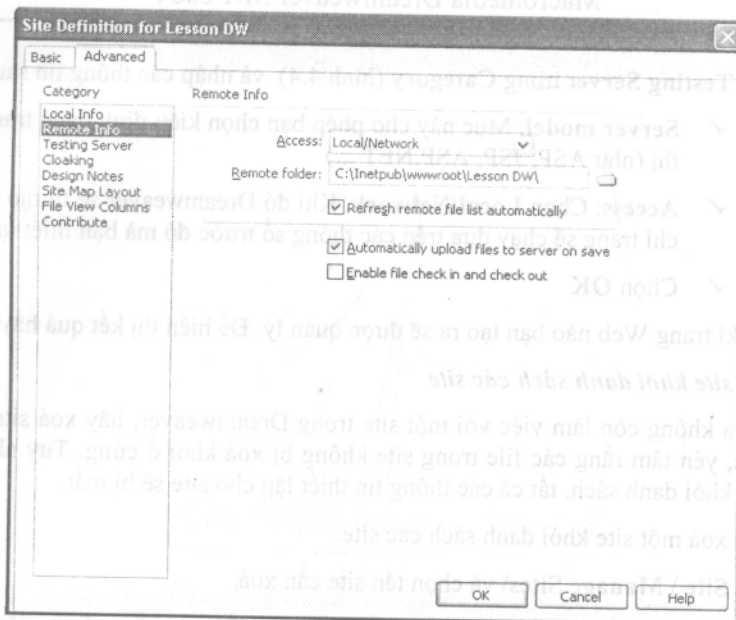


Hình 4.2. Lựa chọn mục Local Info.

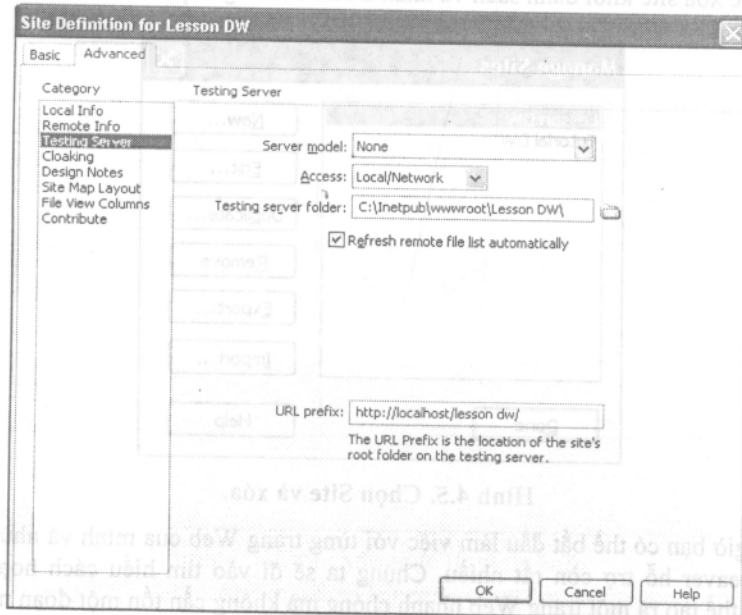
- Chọn **Remote Info** trong **Category** (hình 4.3) và nhập các thông tin sau trong trường hợp bạn muốn các file của mình tự động được tải lên server.
 - ✓ **Access:** Chọn Local/Network để tự động chạy site trên localhost trên mạng kết nối.
 - ✓ **Remote folder:** Nhập đường dẫn chứa các trang của site.
 - ✓ Tiếp đó chuyển sang bước 3 để hoàn thành công việc hiển thị trang Web.

Chương 4

Macromedia Dreamweaver MX 2004



Hình 4.3. Lựa chọn mục Remote Info.



Hình 4.4. Lựa chọn mục Testing Server.

Chương 4

Macromedia Dreamweaver MX 2004

- Chọn **Testing Server** trong **Category** (hình 4.4) và nhập các thông tin sau:

- ✓ **Server model:** Mục này cho phép bạn chọn kiểu định dạng trang cần hiển thị (như ASP, JSP, ASP.NET ...)
- ✓ **Access:** Chọn **Local/Network**. Khi đó Dreamweaver sẽ tự tạo ra URL địa chỉ trang sẽ chạy dựa trên các thông số trước đó mà bạn thiết lập.
- ✓ Chọn **OK**

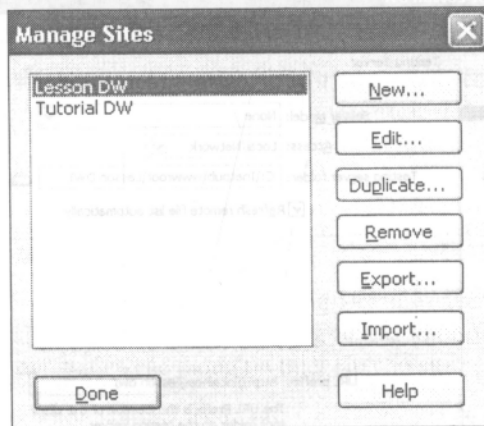
Giờ đây bất kì trang Web nào bạn tạo ra sẽ được quản lý. Để hiển thị kết quả hãy ấn F12.

❖ *Xoá một site khỏi danh sách các site*

Nếu ta không còn làm việc với một site trong Dreamweaver, hãy xoá site khỏi danh sách các site, yên tâm rằng các file trong site không bị xoá khỏi ổ cứng. Tuy nhiên, khi đã xoá một site khỏi danh sách, tất cả các thông tin thiết lập cho site sẽ bị mất.

Các bước để xoá một site khỏi danh sách các site:

- Chọn **Site \ Manage Sites** và chọn tên site cần xoá.
- Nhấn nút **Remove**. Một hộp thoại xuất hiện yêu cầu ta khẳng định việc xoá. Nhấn **Yes** để xoá site khỏi danh sách và nhấn **Done** để đóng hộp thoại.



Hình 4.5. Chọn Site và xoá.

Bây giờ bạn có thể bắt đầu làm việc với từng trang Web của mình và những công cụ mà Dreamweaver hỗ trợ còn rất nhiều. Chúng ta sẽ đi vào tìm hiểu cách hoạt động của chúng để có thể tạo ra một trang Web nhanh chóng mà không cần tốn một đoạn mã nào.

Chương 4

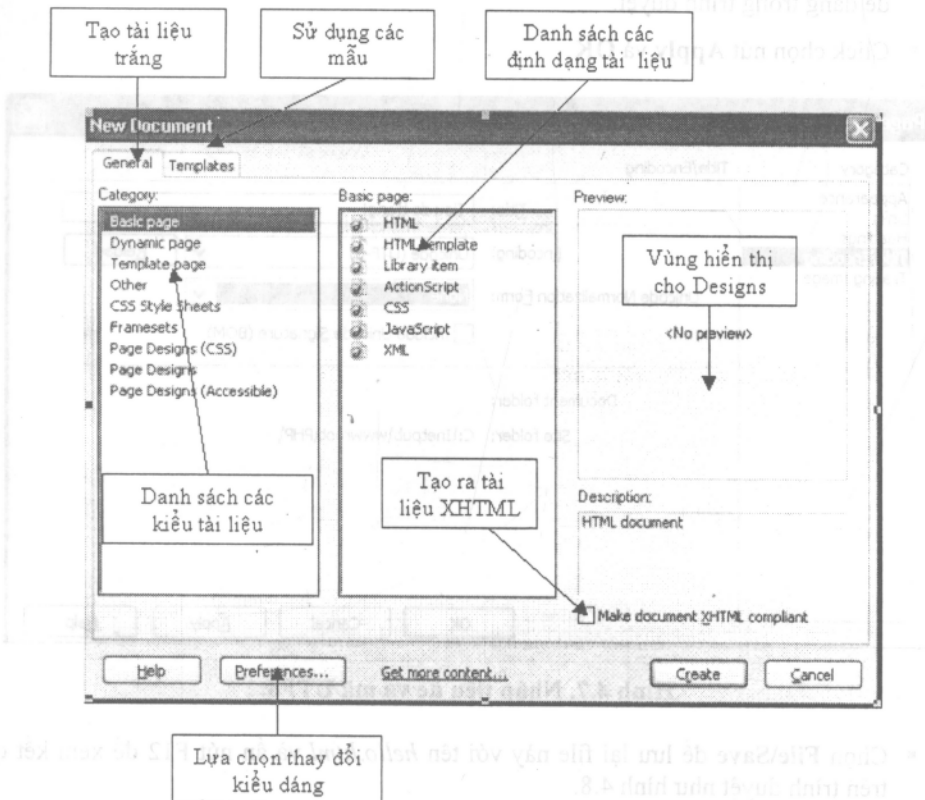
Macromedia Dreamweaver MX 2004

4.1.3. Tạo một trang tài liệu mới

Thông qua hỗ trợ của *Dreamweaver*, bạn có thể tạo ra rất nhiều kiểu định dạng và sau đó có thể tự viết mã hoặc sử dụng hỗ trợ của *Dreamweaver* để hoàn thành các tài liệu này một cách nhanh chóng.

Để tạo ra một trang tài liệu mới thực hiện các bước sau (Xem hình 4.6):

- Chọn **File \ New** và lựa chọn kiểu định dạng cho tài liệu mà bạn định tạo ra (HTML, XHTML, XML, CSS ...). Trong trường hợp bạn muốn tạo ra các trang Web động bạn phải chọn mục **Dynamic Page** rồi chọn kiểu định dạng tương ứng (ASP, PHP, JSP ...)
- Chọn ô **Make Document XHTML compliant** trong trường hợp bạn muốn tạo ra kiểu tài liệu XHTML.
- Chọn nút **Create** để hoàn tất.



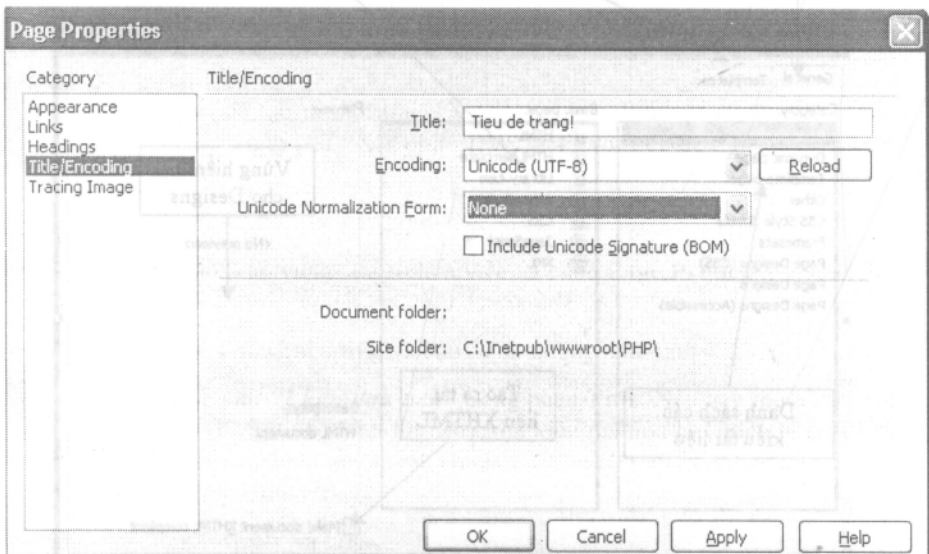
Hình 4.6. Chọn định dạng tài liệu cần tạo.

Chương 4

Macromedia Dreamweaver MX 2004

Ở ví dụ này chúng tôi đã tạo ra một trang XHTML trắng. Bây giờ chúng ta sẽ tạo ra một trang XHTML hiện lên dòng chữ “**Xin chào DW2004**” với tiêu đề “**Vì du dau tien**”. Các bước tiến hành như sau:

- Trên vùng cửa sổ làm việc hãy gõ dòng chữ : “**Xin chào DW2004!**”.
- Click phải chuột vào vùng cửa sổ làm việc và chọn **Page Properties** hoặc ấn **Ctrl+J** để tiến hành thay đổi và thiết lập định dạng trang.
- Chọn tiếp mục **Title/Encoding** và bạn sẽ nhìn thấy như hình 4.7.
- Nhập vào dòng chữ “**Vì du dau tien**” trong phần **Title**. Nếu bạn muốn hiển thị trang Web của mình bằng tiếng Việt thì nên chọn kiểu mã hóa là **Unicode (UTF-8)** trong phần Encoding. Kiểu mã này là mã chuẩn quốc tế cho phép gõ và hiển thị tiếng việt dễ dàng trong trình duyệt.
- Click chọn nút **Apply** và **OK**

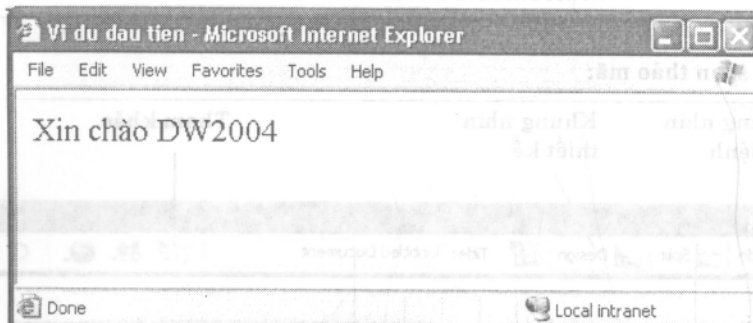


Hình 4.7. Nhập tiêu đề và mã UTF8.

- Chọn **File\Save** để lưu lại file này với tên *hello.html* và ấn nút **F12** để xem kết quả trên trình duyệt như hình 4.8.

Chương 4

Macromedia Dreamweaver MX 2004



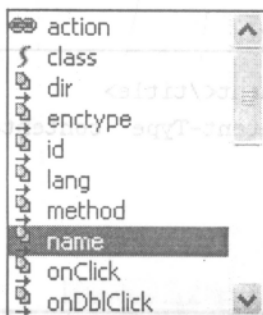
Hình 4.8. Kết quả hiển thị trang hello.html.

4.1.4. Xem và soạn thảo mã nguồn

Xét một cách toàn diện thì Dreamweaver là một chương trình thiết kế Web. Nói chung bạn không cần thiết phải thành thạo ngôn ngữ HTML mới có thể dùng chương trình này một cách hiệu quả. Tuy nhiên trong một số trường hợp chúng ta vẫn cần thay đổi các mã mà Dreamweaver không cung cấp đủ. Trong trường hợp này Dreamweaver cung cấp cho chúng ta một công cụ soạn thảo cho các mã HTML cũng như các mã kịch bản trên Server khác.

Tính năng *Roundtrip HTML* vừa cho phép bạn viết mã và cũng vừa nhìn thấy nội dung được hiển thị. Thêm vào đó mỗi khi bạn gõ một chữ nằm trong định dạng thẻ của HTML tự động xuất hiện danh sách các thẻ và bạn chỉ việc lựa chọn chúng mà không cần nhớ chi tiết từng thẻ.

Để sử dụng các trợ giúp viết mã của Dreamweaver, trong dòng soạn thảo bạn ấn **Ctrl + Space**, khi đó danh sách các thẻ hỗ trợ sẽ hiện ra cho phép bạn lựa chọn (hình 4.9)

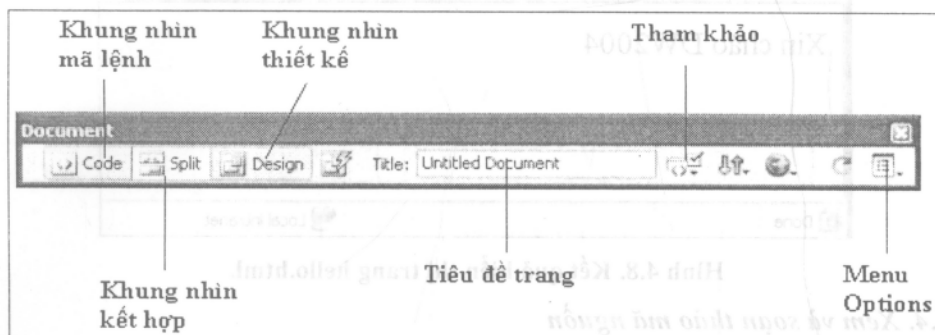


Hình 4.9. Chức năng hỗ trợ mã của Dreamweaver.

Chương 4

Macromedia Dreamweaver MX 2004

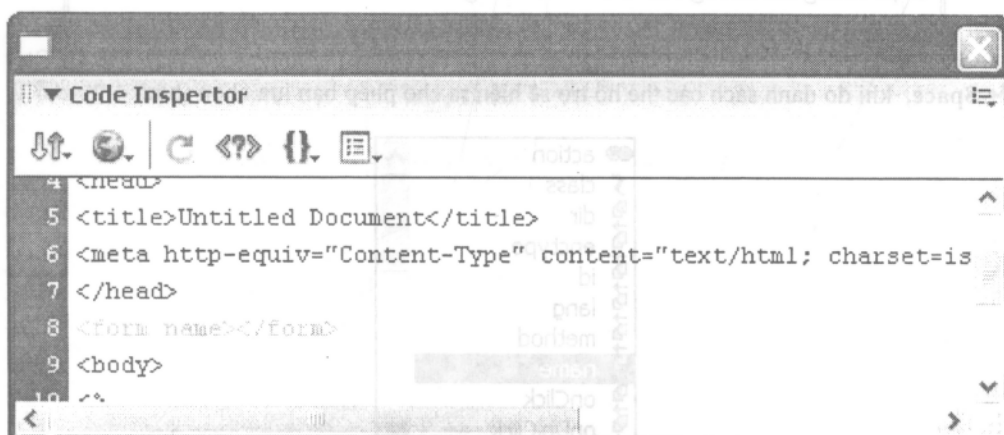
Các chế độ soạn thảo mã:



Hình 4.10. Thanh công cụ quản lý khung nhìn và soạn thảo.

- **Khung nhìn mã lệnh:** cho phép soạn thảo mã HTML hay các ngôn ngữ kịch bản khác. Trạng thái này không nhìn được kết quả.
- **Khung nhìn kết hợp:** cho phép vừa viết mã vừa xem được kết quả.
- **Khung nhìn thiết kế:** chỉ cho phép thao tác bằng tay thông qua các công cụ của Dreamweaver.

Ngoài ra trong quá trình thiết kế bạn có thể sử dụng ngay cửa sổ soạn thảo mã riêng là **Code Inspector** (Chọn Window\Code Inspector hoặc F10) để có thể cải thiện tốc độ thiết kế.



Hình 4.11. Cửa sổ Code Inspector cho phép làm việc nhanh hơn.

4.2. Định dạng văn bản và Style

4.2.1. Chèn và định dạng văn bản

Việc sử dụng văn bản trong DW2004 cũng tương tự các trình xử lý văn bản khác: cho phép bạn áp dụng các kiểu định dạng chữ nhanh chóng. Nếu bạn đang viết một trang HTML hay XHTML ... thì bạn chỉ đơn giản là thực hiện gõ vào vùng tài liệu và làm bất kỳ thiết kế nào- DW2004 sẽ tự động thêm các mã thẻ HTML hay XHTML tương ứng. Để định dạng văn bản trong vùng tài liệu, trước hết bạn phải bôi đen vùng văn bản cần định dạng sau đó sử dụng hộp thoại thuộc tính *Property* để áp dụng.

Sau đây chúng ta sẽ tìm hiểu chi tiết những chức năng tương ứng mà các thẻ HTML có thể tạo ra nhưng hoàn toàn với sự hỗ trợ của DW2004.

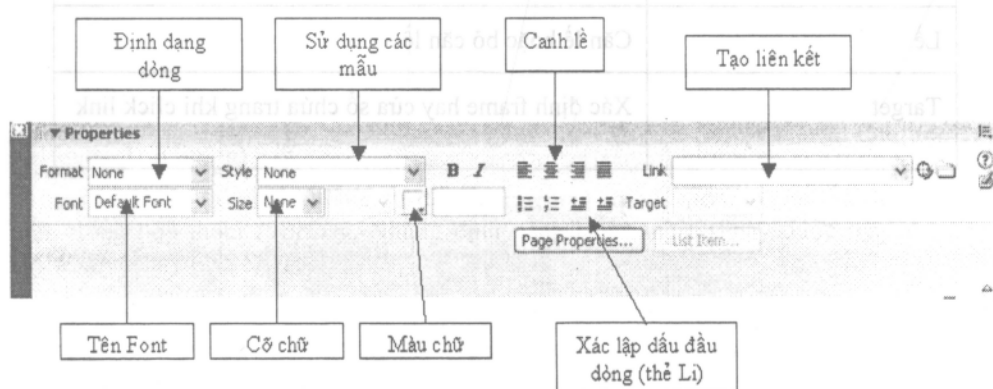
Đặt font, style, màu và cỡ

Có thể xây dựng một site sử dụng các lệnh *Text*, *Paragraph*, *Format* hay các tùy chọn trong hộp thoại *Property*. Những định dạng làm bằng tay này được thay thế bằng *HTML Style* hay *CSS Style*.

Text Properties

Dreamweaver cung cấp tùy chọn *Text Properties* giúp định dạng văn bản. Có thể dùng hộp thoại *Property Inspector* để thay đổi các thuộc tính của đoạn văn bản được lựa chọn trong cửa sổ Document, xác định các liên kết và đích đến cho đoạn văn bản được lựa chọn,...

Đặt con trỏ vào đoạn văn bản để xem các thuộc tính của đoạn văn bản này trong *Property Inspector*. Bảng này liệt kê các thuộc tính thường dùng nhất của văn bản, như hình 4.12:



Hình 4.12. Hộp thoại thuộc tính cho văn bản.

Chương 4
Macromedia Dreamweaver MX 2004

Các thuộc tính được mô tả trong bảng sau:

Bảng 4.1. Thuộc tính Text Properties

Thuộc tính	Mô tả
Format	Định dạng văn bản theo các style heading
Font	Font áp dụng cho văn bản
Style	Thiết lập các Style cho văn bản bằng CSS
Size	Kích thước của văn bản
Màu chữ	Màu áp dụng cho văn bản
Chữ đậm và nghiêng	Chữ đậm và chữ nghiêng cho văn bản
Căn lề trái, giữa và phải	Các chế độ căn lề cho văn bản
Link	Tạo hyper link cho đoạn văn bản được chọn
Danh sách (dấu đầu dòng)	Tạo danh sách có hay không có thứ tự thông qua thẻ LI trong HTML
Lề	Căn lề hoặc bỏ căn lề
Target	Xác định frame hay cửa sổ chứa trang khi click link

Ví dụ 4.1.

- Đoạn mã XHTML như sau:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head> <title>Dinh dang van ban</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>

<body>

<h1>Đây là dòng tiêu đề</h1>

<p><em><strong>Danh sách các chức năng:</strong></em></p>

<ul> <li>Chọn font chữ</li> <li>Chọn màu chữ</li>
      <li>Canh chỉnh</li> <li>Danh sách</li>
      <li>.....</li>
</ul>

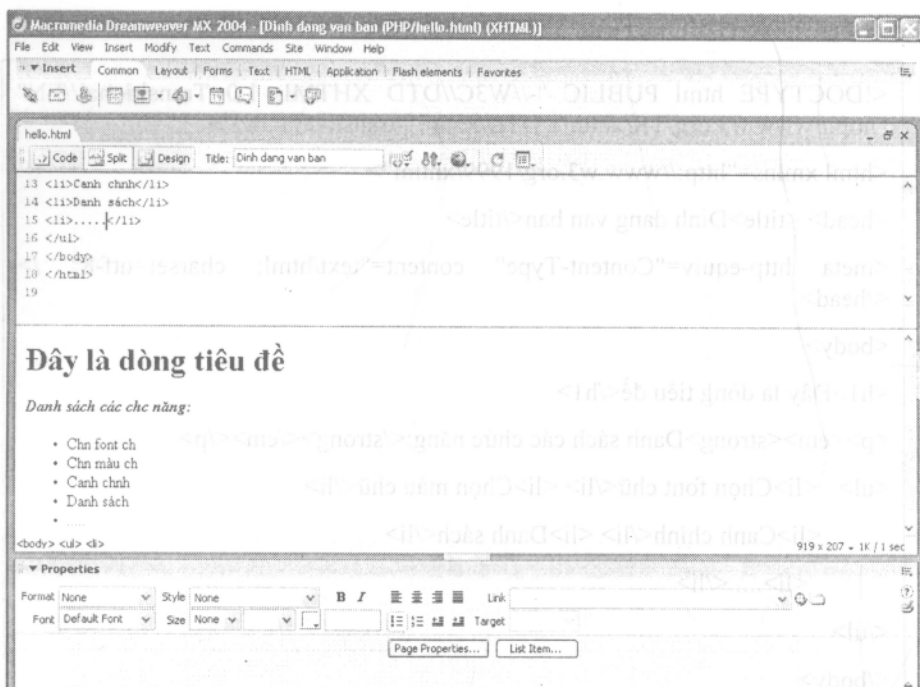
</body>

</html>
```

- Trong DW bạn chỉ cần gõ vào vùng Tài liệu và lựa chọn các chức năng trong bảng trên:

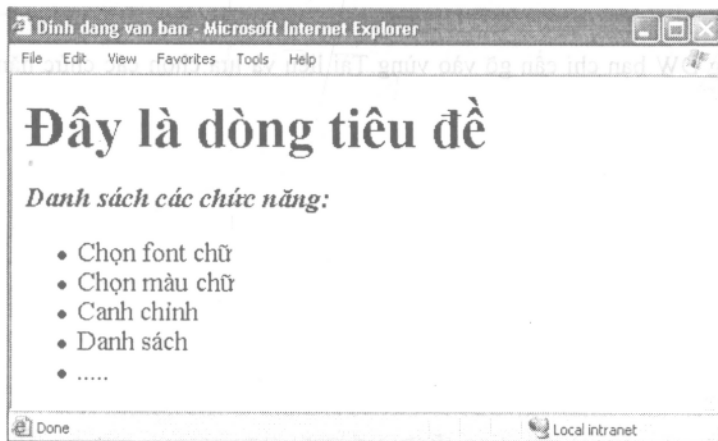
Chương 4

Macromedia Dreamweaver MX 2004



Hình 4.13. Dùng DW tạo văn bản.

- Ấn nút F12 để hiển thị trên trình duyệt cho kết quả như sau



Hình 4.14. Kết quả văn bản hiển thị trên trình duyệt.

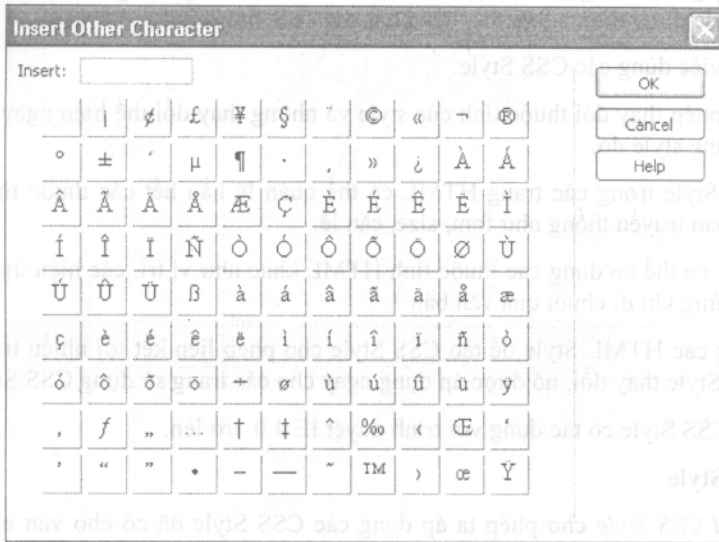
Chương 4

Macromedia Dreamweaver MX 2004

Sử dụng các biểu tượng đặc biệt (Special Symbol)

DW 2004 hỗ trợ cho bạn một phương pháp áp dụng các biểu tượng ký tự đặc biệt một cách nhanh chóng mà không phải thao tác như HTML. Bạn chỉ cần:

- Chọn: Inset\HTML\Special Characters\Other ...
- Chọn biểu tượng cần thêm và chọn OK



Hình 4.15. Bảng biểu tượng đặc biệt.

Tạo đường ngang

Trong HTML bạn sử dụng thẻ `<hr>` để tạo đường ngang, trong DW2004 bạn cũng có thể tạo loại đường thẳng này bằng cách chọn : **Inset\HTML\Horizontal Rule**.

Bạn cũng có thể sử dụng hộp thoại Object để thêm vào đường thẳng như sau:

- Chọn bảng HTML tương ứng
- Chọn biểu tượng cần áp dụng



Hình 4.16. Tạo đường ngang.

Sau đó hãy thiết lập thuộc tính độ dài (*W*) và dày (*H*) của đường thẳng này.

4.2.2. CSS Style Sheets

Một *style* là một nhóm các định dạng nhằm điều khiển hình dáng của vùng văn bản xác định trong một trang web. Một *CSS Style* có thể sử dụng trong nhiều trang khác nhau. Các *CSS Style* được định nghĩa theo tên hoặc theo các thẻ HTML.

Lợi ích của việc dùng các *CSS Style*:

- Cho phép thay đổi thuộc tính của style và những thay đổi thể hiện ngay trên văn bản áp dụng style đó.
- *CSS Style* trong các trang HTML có thể quản lý hầu hết các thuộc tính định dạng văn bản truyền thống như font, size, căn lề.
- Cũng có thể áp dụng các thuộc tính HTML khác như vị trí, các hiệu ứng đặc biệt và hiệu ứng khi di chuột qua văn bản.
- Dùng các HTML Style để tạo *CSS Style* cho phép liên kết tới nhiều trang. Khi một *CSS Style* thay đổi, nó được áp dụng ngay cho các trang sử dụng *CSS Style* đó.

Chú ý: các *CSS Style* có tác dụng với trình duyệt IE 4.0 trở lên.

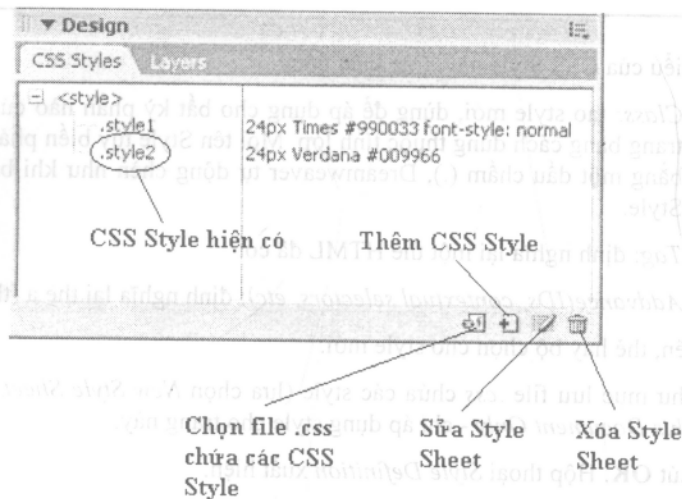
Panel CSS Style

Panel CSS Style cho phép ta áp dụng các *CSS Style* đã có cho văn bản hiện hành đang lựa chọn. Cũng có thể thêm mới, xoá, thay đổi một *CSS Style* từ panel này.

- Để hiển thị panel *CSS Style*, chọn **Window \ CSS Styles**:

Chương 4

Macromedia Dreamweaver MX 2004

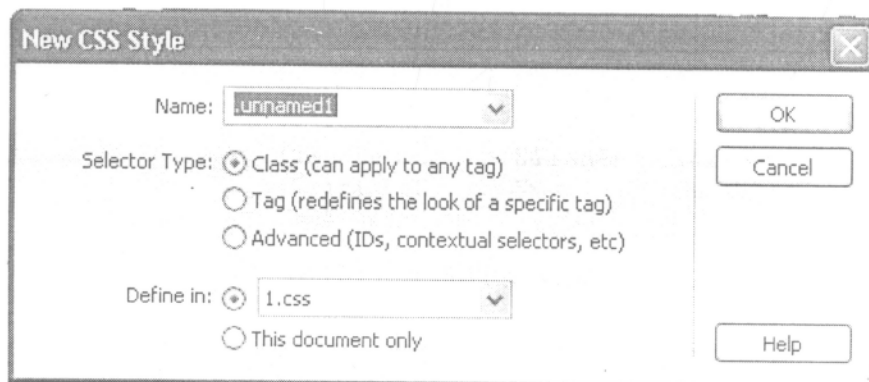


Hình 4.17. Panel CSS Style.

Tạo một CSS Style

Các bước để tạo một CSS Style:

- Mở panel CSS Style. Chọn một trong 2 cách sau:
 - ✓ Chọn nút **New** từ menu ngữ cảnh sau khi click nút phải chuột.
 - ✓ Chọn nút **Add CSS Style** ở cuối panel.
- Hộp thoại New Style xuất hiện như hình dưới.



Hình 4.18. Hộp thoại New Style.

Chương 4

Macromedia Dreamweaver MX 2004

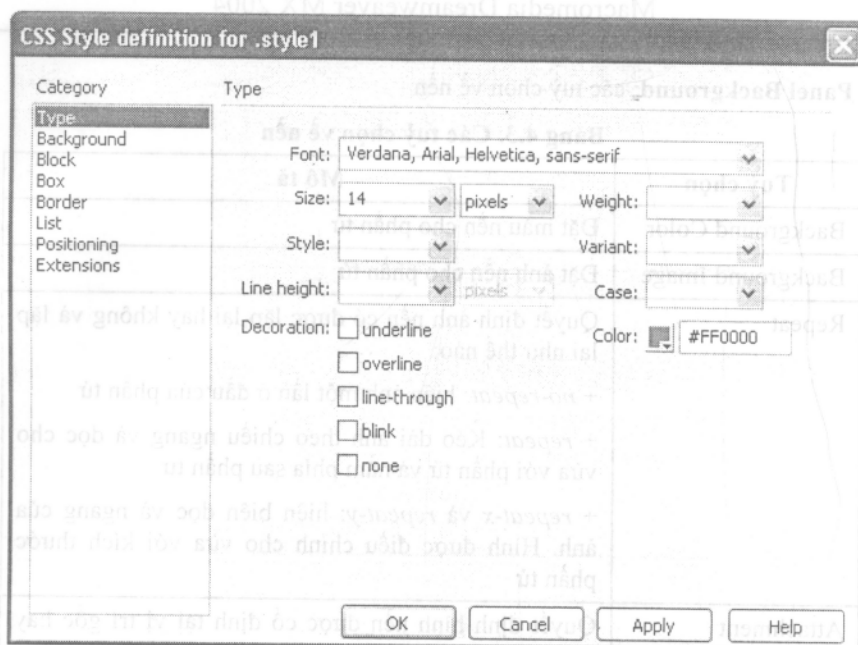
- Chọn kiểu của CSS Style này. Các kiểu gồm:
 - ✓ *Class*: tạo style mới, dùng để áp dụng cho bất kỳ phần nào của văn bản trên trang bằng cách dùng thuộc tính lớp. Mọi tên Style tùy biến phải được bắt đầu bằng một dấu chấm (.), Dreamweaver tự động chèn như khi bạn đặt tên cho Style.
 - ✓ *Tag*: định nghĩa lại một thẻ HTML đã có.
 - ✓ *Advance(IDs, contextual selectors, etc)*: định nghĩa lại thẻ a (thẻ tạo liên kết)
- Nhập tên, thẻ hay bộ chọn cho style mới.
- Chọn thư mục lưu file .css chứa các style (lựa chọn *New Style Sheet File*) hoặc lựa chọn *This Document Only* - chỉ áp dụng style cho trang này.
- Chọn nút **OK**. Hộp thoại *Style Definition* xuất hiện.
- Đặt các thiết lập cho các thuộc tính của style. Để trống thuộc tính nào không cần thiết với style đang tạo.
- Chọn nút **OK** để đóng hộp thoại *Style Definition*.

Hộp thoại *Style Definition*

Hộp thoại *Style Definition* bao gồm nhiều *panel*, sau đây chúng ta sẽ giới thiệu ý nghĩa của các lựa chọn trong mỗi panel. Bạn có thể thử nghiệm chi tiết để hiểu rõ hơn từng chức năng của chúng:

Chương 4

Macromedia Dreamweaver MX 2004



Hình 4.19. Hộp thoại định dạng Style.

- **Panel Type:** Các tùy chọn liên quan đến các ký tự

Bảng 4.2. Các tùy chọn liên quan đến các ký tự

Tùy chọn	Mô tả
Font	Đặt họ font cho style
Size	Đặt kích thước font
Style	Đặt font style: Normal (bình thường), Italic (nghe nghiêng) hay Oblique ()
Line height	Đặt chiều cao dòng chứa chữ
Decoration	Thêm gạch chân, đường ngang ở trên, gạch giữa cho chữ hay làm chữ nhấp nháy
Variant	Chọn thay đổi cho font, ví dụ : small caps
Case	Viết hoa chữ cái đầu của mỗi từ hoặc để chữ dạng hoa hay thường
Color	Đặt màu chữ

Chương 4
Macromedia Dreamweaver MX 2004

- **Panel Background:** các tùy chọn về nền

Bảng 4.3. Các tùy chọn về nền

Tùy chọn	Mô tả
Background Color	Đặt màu nền cho phần tử
Background Image	Đặt ảnh nền cho phần tử
Repeat	Quyết định ảnh nền có được lặp lại hay không và lặp lại như thế nào: • <i>no-repeat</i> : hiện ảnh một lần ở đầu của phần tử + <i>repeat</i> : Kéo dài ảnh theo chiều ngang và dọc cho vừa với phần tử và nằm phía sau phần tử + <i>repeat-x</i> và <i>repeat-y</i> : hiện biên dọc và ngang của ảnh. Hình được điều chỉnh cho vừa với kích thước phần tử
Attachment	Quyết định hình nền được cố định tại vị trí góc hay cuộn theo nội dung
Horizontal Position và Vertical Position	Xác định vị trí ban đầu của hình nền so với phần tử

- **Panel Block**

Bảng 4.4. Các tùy chọn về khoảng trống cho phần tử

Tùy chọn	Mô tả
Word Spacing	Thêm khoảng trống giữa các từ
Letter Spacing	Thêm khoảng trống giữa các ký tự
Vertical Alignment	Xác định lề dọc của phần tử
Text Align	Chế độ căn lề cho phần tử: left (trái), right (phải), center (giữa), justify (đều 2 bên)
Text Indent	Xác định lề cho dòng đầu
Whitespace	Cách sử dụng khoảng trắng trong phần tử

Chương 4
Macromedia Dreamweaver MX 2004

▪ **Panel Box**

Bảng 4.5. Các tùy chọn Panel Box

Tùy chọn	Mô tả
Width và Height	Kích thước của phần tử
Float	Dịch chuyển phần tử đến lề trái hoặc lề phải của trang
Clear	Xác định các cạnh không cho phép dùng các lớp (Layer). Nếu một lớp xuất hiện ở cạnh đặt thuộc tính clear, cạnh này bị chuyển xuống dưới layer
Padding	Số khoảng trống giữa phần nội dung của phần tử và viền của nó (hoặc lề nếu không có viền)
Margin	Số khoảng trống giữa viền (hoặc padding nếu không có viền) của phần tử và các phần tử khác

▪ **Panel Border**

Bảng 4.6. Các tùy chọn viền khung

Tùy chọn	Mô tả
Width	Đặt chiều dày cho viền của phần tử
Color	Đặt màu cho các viền
Style	Đặt kiểu cho viền, một số trình duyệt có thể không hiển thị đúng kiểu viền

▪ **Panel List:** các thiết lập cho danh sách

Bảng 4.7. Các tùy chọn cho danh sách

Tùy chọn	Mô tả
Type	Danh sách kiểu bullet (biểu tượng) hay số
Bullet Image	Xác định hình làm biểu tượng cho các phần tử danh sách
Position	Phần tử danh sách kéo dài tới lề ngoài hay lề trong

▪ **Panel Positioning:** các thiết lập về vị trí phần tử

Chương 4

Macromedia Dreamweaver MX 2004

Bảng 4.8. Các tùy chọn cho thiết lập về vị trí phần tử

Tùy chọn	Mô tả
Visibility	Điều kiện hiển thị ban đầu của lớp
Z-Index	Thứ tự theo chiều Z của lớp. Lớp có thứ tự cao hơn sẽ nằm trên lớp có thứ tự thấp hơn
Overflow	Chế độ tràn được sử dụng khi nội dung của lớp vượt quá kích thước của nó
Placement	Xác định vị trí và kích thước của lớp
Clip	Định nghĩa phần nhìn thấy của lớp

▪ Panel Extensions

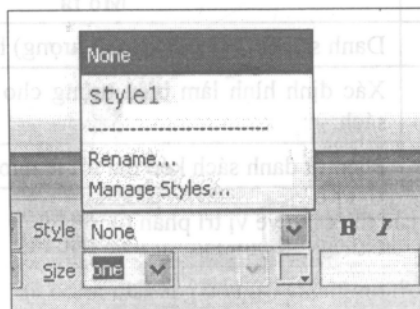
Bảng 4.9. Các tùy chọn mở rộng

Tùy chọn	Mô tả
Page Break	Thêm dấu ngắt trang phía trước hay phía sau đối tượng
Cursor	Thay đổi hình dáng con trỏ chuột khi trỏ chuột qua đối tượng.
Filter	Áp dụng các hiệu ứng đặc biệt cho đối tượng bao gồm <i>blur</i> (mờ) và <i>inversion</i> (đảo ngược).

Áp dụng các Style

Để áp dụng một style vào văn bản, hãy làm theo các bước sau:

- Chọn văn bản cần áp dụng một Style.
- Chọn tên Style trong danh sách menu từ thuộc tính Style của hộp thoại Property.



Hình 4.20. Áp dụng Style trong danh sách.

Chương 4

Macromedia Dreamweaver MX 2004

Khi đó đoạn văn bản sẽ hiển thị nội dung theo các thuộc tính mà Style đó quy định. Từ giờ trở đi bất cứ lúc nào bạn hiệu chỉnh lại Style thì văn bản đó cũng sẽ tự động cập nhật tương ứng, giúp bạn tiết kiệm thời gian hơn rất nhiều lần trong việc chỉnh sửa đơn lẻ từng văn bản.

4.3. Chèn ảnh và tạo liên kết

4.3.1. Chèn ảnh

Sử dụng Dreamweaver để thêm ảnh cho trang Web hết sức đơn giản, thông thường người ta hay sử dụng ba định dạng ảnh là GIF, JPEG, PNG.

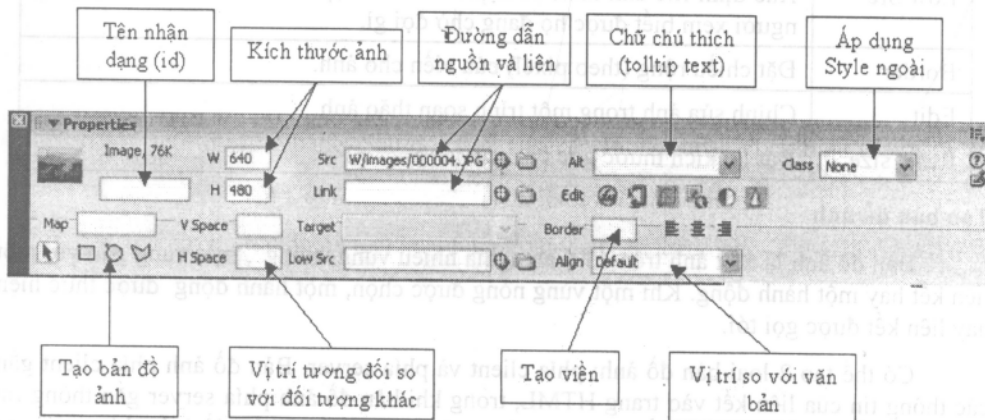
Các bước để chèn một ảnh, thực hiện một trong các cách sau:

- Đặt con trỏ tại nơi muốn ảnh xuất hiện trong cửa sổ Document, ví dụ: một vị trí trên trang, một ô trong bảng. Sau đó chọn **Insert \ Image** hoặc click biểu tượng **Insert Image** ở mục **Common** của thanh Objects rồi chọn **Image**.
- Đặt con trỏ tại nơi muốn ảnh xuất hiện trong cửa sổ Document và nhấn **Ctrl-Alt-I**.
- Kéo biểu tượng **Insert Image** từ mục **Common** của thanh Objects vào cửa sổ Document. Thao tác này sẽ làm xuất hiện hộp thoại chọn ảnh. Click nút **Browse** để chọn file một ảnh hoặc gõ đường dẫn đến file ảnh vào trường File name.

Các thuộc tính của ảnh

Chọn một ảnh đã chèn trong cửa sổ Document để xem và thay đổi các thuộc tính của nó trong *Property Inspector*.

Nếu *Property Inspector* chưa xuất hiện, chọn **Window \ Properties** hoặc nhấn tổ hợp phím **Ctrl-F3** để hiện nó.



Hình 4.21. Các thuộc tính định dạng ảnh.

Chương 4
Macromedia Dreamweaver MX 2004

Bảng sau mô tả các thuộc tính của ảnh.

Bảng 4.10. Các thuộc tính của ảnh

Thuộc tính	Mô tả
Ô text box	Chứa ID cho file ảnh.
W và H	Kích thước gốc của file, tính theo pixel.
Src	Đường dẫn đến file ảnh. Click biểu tượng thư mục để chọn lại file ảnh.
Link	Liên kết sẽ chuyển đến khi click vào file ảnh.
Align	Chế độ căn lề cho ảnh so với các phần tử khác trong đoạn.
Alt	Đoạn chữ xuất hiện tại nơi chứa ảnh cho các trình duyệt chỉ hỗ trợ văn bản hoặc khi không tìm thấy file ảnh. Với một số trình duyệt, đoạn chữ này cũng xuất hiện khi di chuột qua ảnh.
Map	Tên bản đồ cho phép tạo bản đồ ảnh. Bạn có thể vẽ bản đồ bằng hình đa giác hay vuông, tròn rồi tạo liên kết (Xem phần HTML)
V Space và H Space	Thêm khoảng trắng, theo chiều đứng và chiều ngang dọc theo các phía của ảnh. V Space thêm khoảng trắng dọc theo đỉnh và đáy của ảnh. H Space thêm khoảng trắng dọc theo phía trái và phải ảnh.
Target	Xác định frame hoặc cửa sổ sẽ chứa trang khi nhấn vào liên kết ảnh.
Low Src	Xác định file ảnh khác sẽ nạp trước khi nạp ảnh chính. Điều này giúp người xem biết được họ đang chờ đợi gì.
Border	Đặt chiều rộng (theo pixel) của viền cho ảnh.
Edit	Chỉnh sửa ảnh trong một trình soạn thảo ảnh.
Reset size	Đặt lại kích thước gốc của ảnh.

Tạo bản đồ ảnh

Bản đồ ảnh là một ảnh trên đó định nghĩa nhiều vùng “nóng”, mỗi vùng gắn với một liên kết hay một hành động. Khi một vùng nóng được chọn, một hành động được thực hiện hay liên kết được gọi tới.

Có thể tạo 2 loại bản đồ ảnh: phía client và phía server. Bản đồ ảnh phía client gắn các thông tin của liên kết vào trang HTML, trong khi bản đồ ảnh phía server gắn thông tin của liên kết trên server. Bản đồ ảnh phía client xử lý nhanh hơn bản đồ ảnh phía server vì server không cần biên dịch khi người dùng click vào liên kết. *Chú ý:* Có thể tạo cả bản đồ

Chương 4

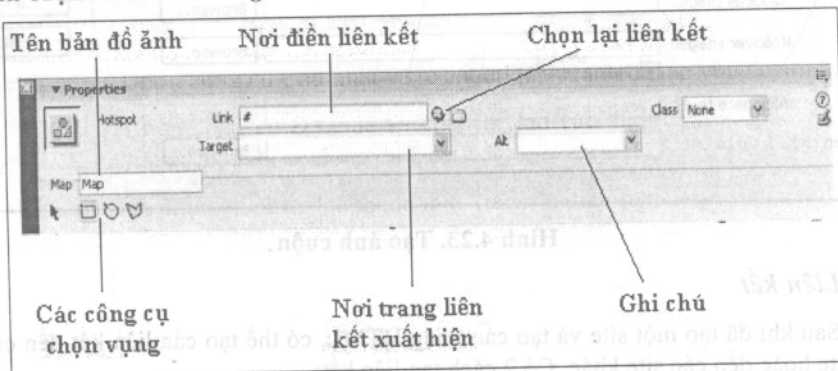
Macromedia Dreamweaver MX 2004

ảnh phía client và server trong cùng một trang. Tuy nhiên với các trình duyệt hỗ trợ cả 2 loại bản đồ này, quyền ưu tiên dành cho bản đồ ảnh phía client.

Các bước để tạo bản đồ ảnh:

- Chọn một ảnh.
- Click mũi tên mở rộng ở **Property Inspector** để thấy hết các thuộc tính.
- Trong trường **Map**, đặt một tên cho bản đồ ảnh.
- Chọn một trong các cách sau để định nghĩa các vùng cho bản đồ ảnh:
 - Chọn công cụ Tròn và rê chuột quanh ảnh để tạo vùng chọn hình tròn.
 - Chọn công cụ Chữ nhật và rê chuột quanh ảnh để tạo vùng chọn chữ nhật.
 - Chọn công cụ Đa giác và click để tạo các đỉnh cho đa giác. Click biểu tượng mũi tên để đóng đa giác.
- Khi các vùng chọn đã được tạo, chọn một vùng để định nghĩa các liên kết cho vùng. *Property Inspector* cho một vùng xuất hiện như hình 4.21.
- Click biểu tượng thư mục để chọn liên kết hoặc nhập trực tiếp liên kết vào trường *Link*.
- Chọn frame chứa trang sẽ liên kết đến từ *menu Target* hoặc nhập tên cửa sổ chứa trang.
- Trong trường *Alt*, nhập dòng văn bản sẽ xuất hiện khi hiển thị trên các trình duyệt chỉ hỗ trợ text. Một số trình duyệt hiển thị dòng này như là *Tool tip*.
- Lặp lại từ bước 4 đến 7 cho mỗi vùng ảnh trong bản đồ.

Tạo ảnh cuộn - Rollover image



Hình 4.22. Tạo bản đồ ảnh .

Chương 4

Macromedia Dreamweaver MX 2004

Rollover là ảnh có thể thay đổi khi người dùng di chuột qua nó. Một rollover gồm 2 ảnh: một ảnh thể hiện khi trang được nạp, một ảnh xuất hiện khi di chuột đi qua nó.

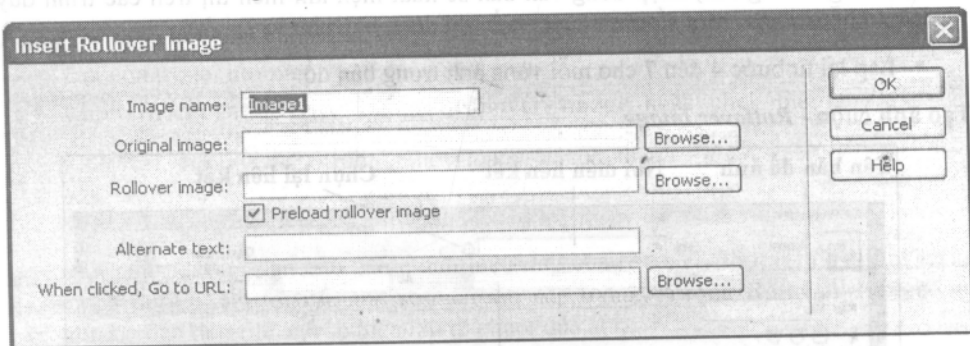
Chú ý: 2 ảnh trong *rollover* phải cùng kích thước. Nếu không, Dreamweaver sẽ tự động định lại kích thước cho ảnh thứ 2 bằng kích thước ảnh đầu.

Các bước để tạo một ảnh cuộn:

- Đặt con trỏ tại nơi muốn ảnh cuộn xuất hiện.
- Chọn **Insert/Interactive Images/Rollover Image** hoặc click biểu tượng **Insert**

Rollover Image  trong mục *Common* của thanh *Objects*. Hộp thoại *Insert Rollover Image* xuất hiện như hình 4.22.

- Đặt tên cho ảnh chính trong trường **Image Name**.
- Xác định đường dẫn cho 2 file ảnh hiệu ứng của rollover. Trường **Original Image** chứa đường dẫn tới file ảnh sẽ hiện khi nạp trang. Trường **Rollover Image** chứa đường dẫn đến file ảnh sẽ hiện khi di chuột qua ảnh.
- Chọn tùy chọn **Preload Rollover Image** để ảnh được nạp trước khi trang được gọi tới.
- Trong trường **When Clicked Go to URL**, click nút **Browse** để chọn file hay gõ đường dẫn tới file muốn mở khi người dùng click vào ảnh. Nếu không xác định một liên kết nào, Dreamweaver tự động chèn liên kết rỗng (#) trong mã HTML để liên kết với ảnh.
- Click **OK** để đóng hộp thoại.



Hình 4.23. Tạo ảnh cuộn.

4.3.2. Liên kết

Sau khi đã tạo một site và tạo các trang HTML, có thể tạo các liên kết đến các vị trí trong site hoặc đến các site khác. Có 2 cách tạo liên kết:

- Tạo các file và các trang trước, sau đó thêm các liên kết

Chương 4

Macromedia Dreamweaver MX 2004

- Tạo các liên kết tới các trang hay các file chưa tồn tại

Vị trí và đường dẫn của tài liệu

- Đường dẫn tuyệt đối: là địa chỉ URL đầy đủ của tài liệu được liên kết bao gồm cả giao thức sử dụng (thường là `http://`). Ví dụ: `http://home.vnn.vn/support/index.asp`
- Đường dẫn tương đối: là liên kết cục bộ trong website, thường dùng khi tài liệu được liên kết nằm trong thư mục hiện hành.
 - Để liên kết tới một file khác trong thư mục hiện hành, đơn giản chỉ cần cung cấp tên file. Ví dụ: `content.htm`
 - Để liên kết tới một file trong thư mục con của thư mục hiện hành, cần cung cấp tên thư mục con, sau đó là dấu sổ (/) và cuối cùng là tên file. Ví dụ: `resources/books.htm`
 - Để liên kết tới một file trong thư mục cha của thư mục hiện hành, cần đặt 2 dấu chấm và dấu sổ trước tên file. Ví dụ: `../books.htm`
- Đường dẫn tới thư mục gốc: dùng khi làm việc trên một website lớn, sử dụng nhiều server hoặc một server quản lý nhiều site. Thư mục gốc được biểu thị bằng dấu sổ (/). Ví dụ: `/support/tips.htm`

Tạo liên kết các trang trong Website

- Chọn đoạn văn bản hay hình ảnh cần tạo liên kết trong cửa sổ Document ở chế độ thiết kế.
- Mở *Property Inspector*, và thực hiện một trong hai cách sau:
 - Click biểu tượng thư mục để chọn một file cần liên kết tới. Trong hộp thoại Select File, chọn loại đường dẫn sử dụng: tương đối hay đường dẫn tới thư mục gốc.
 - Ở trường **Link**, xác định đường dẫn đến file cần liên kết.
- Chọn vị trí mở tài liệu
- Để tài liệu được liên kết xuất hiện ở vị trí khác cửa sổ hay frame hiện hành (tức là nơi chứa liên kết), chọn một lựa chọn từ menu **Target** trong hộp thoại *Property Inspector*. Có các tùy chọn sau:

Bảng 4.11. Các Target

Tùy chọn	Mô tả
_blank	Nạp tài liệu được liên kết vào một cửa sổ mới trên trình duyệt.
_parent	Nạp tài liệu được liên kết vào cửa sổ hay frame cha của frame chứa liên kết. Nếu frame chứa liên kết không bị lồng thì tài liệu được hiển thị trên toàn bộ cửa sổ trình duyệt.

Chương 4

Macromedia Dreamweaver MX 2004

_self	Nạp tài liệu được liên kết vào cùng cửa sổ hay frame của liên kết. Đây là lựa chọn mặc định.
_top	Nạp tài liệu được liên kết vào toàn bộ cửa sổ trình duyệt. Do đó bỏ hết các frame.

Tạo liên kết tới một vị trí trong cùng một trang web

Thường đánh dấu vị trí các chủ đề hay các điểm đặc biệt trong trang web (ví dụ: đầu trang, cuối trang) bằng một *anchor* - điểm neo. Có thể tạo liên kết tới các anchor qua 2 bước: tạo *anchor* và liên kết tới *anchor* vừa tạo.

Để tạo một *anchor*, thực hiện 2 bước sau:

- Trong chế độ thiết kế, đặt con trỏ tại vị trí muốn đặt anchor.
- Chọn một trong 2 cách:
 - Chọn **Insert \ Named Anchor**
 - Chọn biểu tượng **Named Anchor**  trong panel Objects
- Đặt tên cho *anchor* trong hộp thoại vừa xuất hiện.

Chú ý:

- + Tên *anchor* không được chứa khoảng trắng và không được đặt anchor trong một layer.
- + Tên *anchor* phân biệt chữ hoa, chữ thường.

Để liên kết tới một *anchor*, thực hiện 2 bước sau:

- Trong chế độ thiết kế, chọn đoạn văn bản hoặc hình ảnh muốn tạo liên kết
- Trong trường Link của hộp thoại *Property Inspector*, gõ ký hiệu # và tên *anchor*. Ví dụ: để liên kết tới một vị trí trong trang hiện hành có anchor là **topic**, ta nhập **#topic** vào trường Link.

Tạo liên kết E-mail

Liên kết email dùng để mở cửa sổ soạn thư mới với trường *To* được điền là địa chỉ mà liên kết trở tới. Các bước để tạo một liên kết email:

- Chọn đoạn văn bản hay hình ảnh muốn tạo liên kết email. Sau đó chọn một trong 2 cách:
 - Chọn **Insert \ Email Link**

Chương 4

Macromedia Dreamweaver MX 2004

- Chọn biểu tượng **Insert Email Link**  trong mục Common của panel Objects
 - Hộp thoại **Insert Email Link** xuất hiện, nhập vào trường **Text** đoạn văn bản sẽ trở thành liên kết. Nhập vào trường **Email** địa chỉ email cần gửi đến.
 - Chọn nút **OK** để đóng hộp thoại.

Tạo Navigation Bars

Navigation Bars là menu, bố trí theo chiều ngang hay dọc, cho phép di chuyển giữa các trang một cách thuận tiện. Menu này bao gồm một ảnh (hay một tập ảnh) thể hiện các thay đổi khi người dùng tác động lên menu.

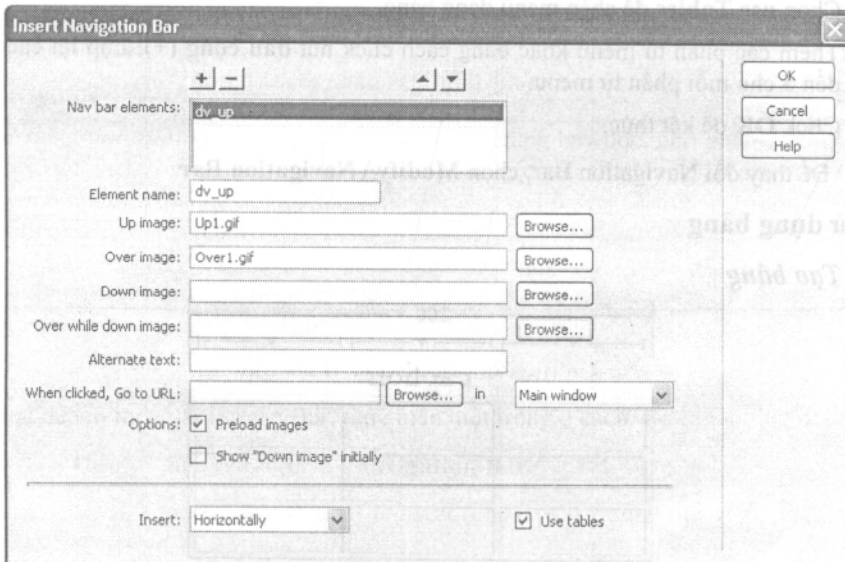
Ví dụ một navigation bar:



Hình 4.24. Ví dụ một navigation bar.

Các bước để tạo một Navigation Bar, thực hiện một trong 2 cách:

- Chọn **Insert \ Image Objects \ Navigation Bar**
- Trong mục *Common của panel Objects*, chọn biểu tượng **Insert Navigation Bar** . Hộp thoại *Insert Navigation Bar* xuất hiện như hình 4.24.



Hình 4.25. Navigation Bar.

Chương 4

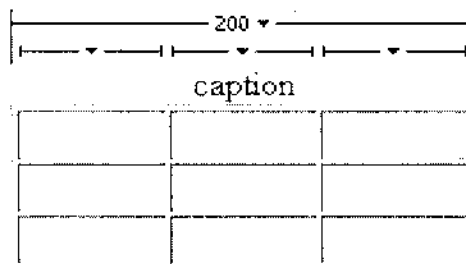
Macromedia Dreamweaver MX 2004

- Trong trường **Element Name**, nhập tên cho phần tử menu. Mỗi phần tử menu ứng với một nút có 4 ảnh cho 4 trạng thái của nút. Xác định đường dẫn đến các file ảnh ở 4 trường tiếp theo.
- Trong trường **When Clicked, Go to URL**, chọn đích đến khi nút tương ứng được chọn.
- Trường **in** có các lựa chọn:
 - ✓ **Main Window**: mở file được liên kết trong cùng cửa sổ với menu.
 - ✓ Chọn một frame để mở file trên frame đó.
- Lựa chọn nạp hình ảnh:
 - ✓ **Preload Images**: nạp hình ảnh khi trang được nạp. Nếu không chọn, có thể có khoảng thời gian trễ khi người dùng di chuột qua hình ảnh.
 - ✓ Chọn **Show “Down Image” Initially** để hiện phần tử menu ở trạng thái nhấn xuống khi nạp trang thay vì hiển thị ở trạng thái chưa nhấn như bình thường.
- Chọn lựa chọn **Insert**:
- Chọn **vertically** để menu xuất hiện theo chiều đứng, **horizontally** để menu xuất hiện theo chiều ngang.
- Chọn **use Tables** để chèn menu dạng bảng.
- Thêm các phần tử menu khác bằng cách click nút **dấu cộng (+)**. Lặp lại các bước 2 đến 5 cho mỗi phần tử menu.
- Click **OK** để kết thúc.

Chú ý: Để thay đổi Navigation Bar, chọn **Modify \ Navigation Bar**

4.4. Sử dụng bảng

4.4.1. Tạo bảng



Hình 4.26. Bảng trong Dreamweaver.

Chương 4

Macromedia Dreamweaver MX 2004

Bảng là công cụ chủ yếu để tạo bố cục cho trang HTML. Bảng gồm 3 thành phần chính:

- Các hàng (theo chiều ngang)
- Các cột (theo chiều dọc)
- Các ô tạo bởi các hàng và các cột

Bảng là phần quan trọng trong các thiết kế web. Bảng mang lại nhiều công dụng quan trọng như:

- Có thể sử dụng bảng để kết xuất dữ liệu, thiết kế các cột trong trang, hiển thị văn bản hay hình ảnh.
- Có thể chèn văn bản, hình ảnh, các đối tượng multimedia vào các ô trong bảng.
- Có thể thể xóa, chia, trộn các ô, chỉnh sửa thuộc tính của bảng, hàng hay ô; thêm màu sắc, căn lề; cắt, dán các ô.

Chèn bảng

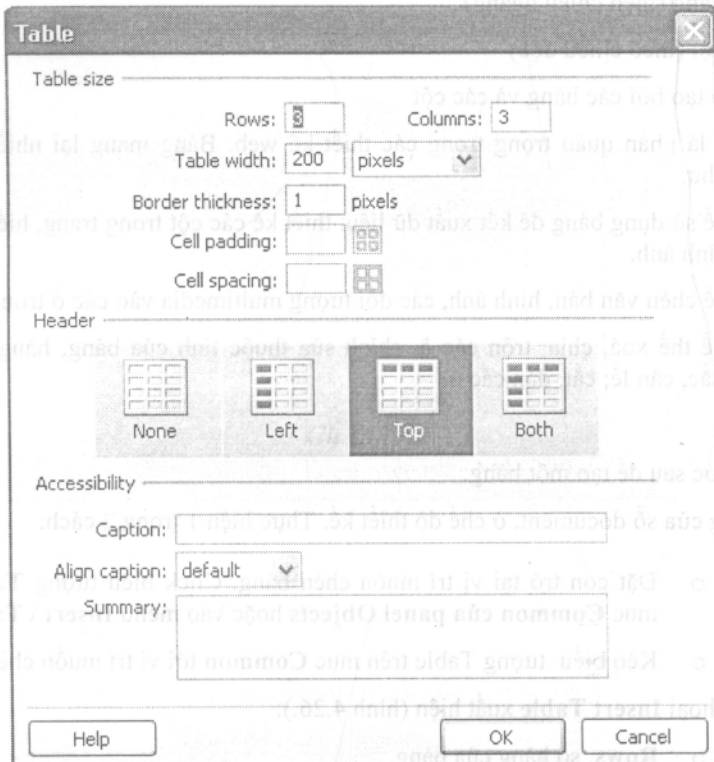
Theo các bước sau để tạo một bảng:

- Trong cửa sổ document, ở chế độ thiết kế. Thực hiện 1 trong 2 cách:
 - Đặt con trỏ tại vị trí muốn chèn bảng. Click biểu tượng **Table**  trên mục **Common** của panel **Objects** hoặc vào menu **Insert \ Table**.
 - Kéo biểu tượng **Table** trên mục **Common** tới vị trí muốn chèn bảng.
- Hộp thoại **Insert Table** xuất hiện (hình 4.26.):
 - **Rows**: số hàng của bảng
 - **Columns**: số cột của bảng
 - **Cell Padding**: số pixel thêm vào giữa phần nội dung trong ô và viền. Mặc định cell padding là 1 pixel. Nhập 0 nếu không dùng cell padding.
 - **Cell Spacing**: số pixel giữa mỗi ô trong bảng, hay khoảng cách giữa các ô. Mặc định cell spacing là 2 pixel. Nhập 0 nếu không cần cell spacing.
 - **Width**: chiều rộng của bảng, tính theo pixel hoặc phần trăm so với cửa sổ.
 - **Border**: chiều dày 4 đường viền của bảng, tính theo pixel. Nhập 0 nếu không cần đường viền.
 - **Header** có 4 lựa chọn để ghi tiêu đề bảng.

Chương 4

Macromedia Dreamweaver MX 2004

- Click **OK** để tạo bảng.



Hình 4.27. Xác lập các thông số của bảng.

Các bảng lồng nhau

Một bảng lồng là bảng nằm trong một bảng khác. Có thể chỉnh sửa một bảng lồng như mọi bảng bình thường. Một hạn chế duy nhất là chiều rộng của nó bị hạn chế bởi ô chứa bảng lồng.

- Để tạo một bảng nằm trong một ô của bảng khác, thực hiện 1 trong 2 cách sau:
 - Chọn ô muốn chèn bảng vào, rồi chọn **Insert \ Table**.
 - Kéo nút **Table** trên panel **Object**, ở mục **Common** tới ô muốn chèn bảng vào.
- Nhập các thông số cho bảng ở hộp thoại **Insert Table** vừa xuất hiện.
- Click **OK** để tạo bảng.

4.4.2. Các thao tác với bảng

Chọn bảng, các hàng, các cột

DreamweaverMX 2004 cung cấp cho bạn khả năng tương tác lựa chọn các thành phần của bảng rất linh hoạt thông qua hệ thống “lưới” phía trên bảng. Bạn có thể chèn thêm cột, dòng cũng như thực hiện một vài thao tác khác trên đó.

❖ Để chọn toàn bộ bảng, có các cách sau:

- Click góc trên bên trái của bảng hoặc click bất kỳ vị trí nào phía dưới hoặc phía phải của bảng.
- Click một lần vào bảng và chọn **Modify \ Table \ Select Table**.
- Chọn một ô trong bảng và chọn **Edit \ Select All**.
- Đặt con trỏ tại bất kỳ ô nào trong bảng và chọn tag **<table>** ở góc dưới bên trái của sổ Document.

Dấu hiệu một bảng đã được chọn là có một đường viền đậm và các hộp điều khiển kích thước xuất hiện xung quanh bảng.

❖ Để chọn các hàng hay các cột của bảng, có các cách sau:

- Đặt con trỏ tại lề trái của một hàng hoặc phía đỉnh của một cột. Click khi xuất hiện mũi tên.
- Đặt con trỏ vào một ô và kéo xuống hoặc kéo ngang để chọn các hàng hay cột mong muốn.

❖ Để chọn các ô không liên kề nhau, có các cách sau:

- Giữ phím **Ctrl** và click vào các ô cần chọn
- Chọn nhiều ô liên tiếp trong bảng. Tiếp đó giữ **Ctrl** và click vào các ô cần bỏ chọn.

Thêm, xóa các hàng, các cột

❖ Thêm hàng hoặc cột:

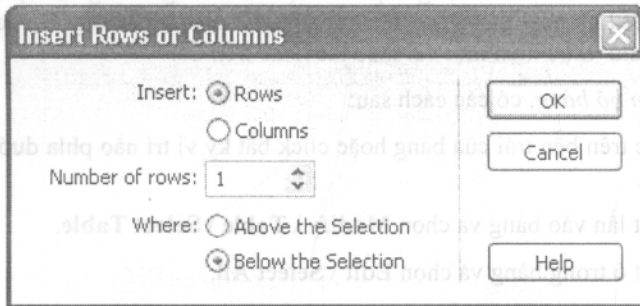
Click vào ô nơi muốn thêm hàng hoặc cột mới. Thực hiện theo một trong các cách sau:

- Để thêm một hàng, chọn **Modify \ Table \ Insert Row** hoặc chọn **Table \ Insert Row** từ menu ngữ cảnh.
- Để thêm một cột, chọn **Modify \ Table \ Insert Column** hoặc chọn **Table \ Insert Column** từ menu ngữ cảnh.
- Để thêm các hàng hoặc cột, chọn **Modify \ Table \ Insert Rows or Columns** hoặc chọn **Table \ Insert Rows or Columns** từ menu ngữ cảnh. Trong hộp thoại vừa xuất

Chương 4

Macromedia Dreamweaver MX 2004

hiện, nhập số hàng hoặc cột mong muốn, xác định hàng hoặc cột mới sẽ xuất hiện phía trên hay dưới hàng hay cột đang được chọn, như hình 4.27.



Hình 4.28. Thêm hàng và cột.

- Click **OK** để thêm.

❖ Xoá một hàng hay cột:

Click vào ô thuộc hàng hay cột cần xoá. Thực hiện một trong các cách sau:

- Để xoá một hàng, chọn **Modify \ Table \ Delete Row** hoặc chọn **Table \ Delete Row** từ menu ngữ cảnh.
- Để xoá một cột, chọn **Modify \ Table \ Delete Column** hoặc chọn **Table \ Delete Column** từ menu ngữ cảnh.

Chú ý: Có thể sử dụng **Property Inspector** để thêm hay xoá các hàng hoặc cột ở phía dưới hoặc góc phải của bảng.

Tách, ghép các ô

❖ Ghép 2 hay nhiều ô trong một bảng

- Chọn các ô cần ghép. Các ô được chọn phải liên tiếp và tạo thành một hình chữ nhật.
- Chọn **Modify \ Table \ Merge Cells** hoặc click nút **Merge Cell**  trong **Property Inspector**. Nội dung của các ô được ghép lại trong một ô mới tạo ra. Các đặc tính của ô đầu sẽ áp dụng cho ô mới tạo ra.

❖ Tách một ô trong bảng

- Click vào ô cần tách.
- Chọn **Modify \ Table \ Split Cell** hoặc click nút **Split Cell**  trong **Property Inspector**.

Chương 4

Macromedia Dreamweaver MX 2004

- Trong hộp thoại *Split Cell* vừa xuất hiện, chọn sẽ chia ô thành các hàng hay các cột và nhập vào số hàng/số cột cần thiết.

Cắt, dán các ô

Có thể cắt hay dán nhiều ô trong bảng, bảo toàn định dạng của ô hoặc chỉ cắt / dán nội dung của ô.

❖ Cắt / copy các ô trong bảng

- Chọn một hay nhiều ô cần cắt hay copy. Các ô được chọn phải liên tiếp nhau và tạo thành một hình chữ nhật như trường hợp ghép các ô.
- Chọn **Edit \ Cut** hay **Edit \ Copy** tương ứng để cắt / copy các ô.

❖ Dán các ô vào bảng

- Chọn nơi muốn dán các ô vào bảng và làm tiếp như sau:
 - Để dán các ô vào phía trước hay phía trên một ô, chọn ô đó.
 - Để tạo bảng mới chứa các ô sẽ dán, click nơi muốn bảng xuất hiện.

- Chọn **Edit \ Paste**.

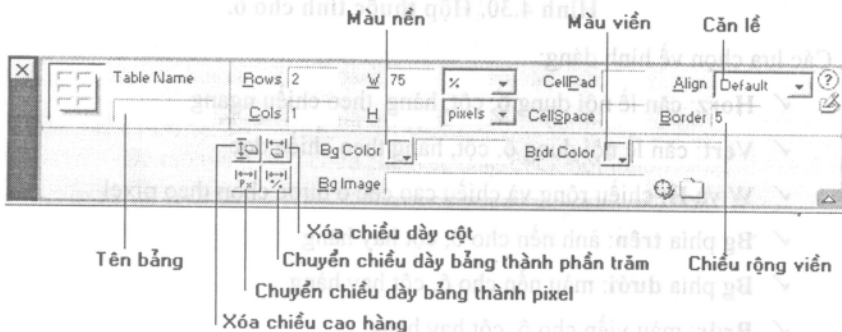
❖ Xóa nội dung các ô nhưng vẫn giữ nguyên các ô

- Chọn một hay nhiều ô cần xóa nội dung.
- Chọn **Edit \ Clear** hay nhấn nút **Delete**.

4.4.3. Định dạng bảng và các ô

Định dạng bảng:

- Chọn bảng.
- Mở *Property Inspector* và click nút mũi tên mở rộng để thấy hết các thuộc tính của bảng. Xem hình 4.29 dưới.



Hình 4.29. Hộp thuộc tính cho bảng.

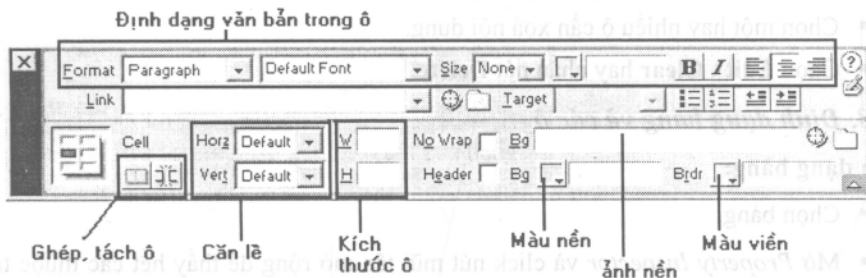
Chương 4

Macromedia Dreamweaver MX 2004

- Đặt tên cho bảng trong trường **Tên bảng**
- Chọn các lựa chọn bố trí bảng:
 - ✓ **Rows** và **Cols**: số hàng, số cột của bảng.
 - ✓ **W** và **H**: chiều rộng và chiều cao của bảng theo % so với cửa sổ trình duyệt hoặc theo pixel.
 - ✓ **Căn lề**: Căn lề bảng (Trái, Giữa, Phải) so với các phần tử khác trong cùng đoạn như văn bản, hình ảnh.
 - ✓ **Xóa chiều cao hàng** và **Xóa chiều dày cột**: thu hẹp chiều cao hàng và chiều rộng cột đến mức tối thiểu.
- Chọn các lựa chọn cho nền bảng: **Màu nền**, **màu viền**, **hình nền**.

Định dạng các ô:

Chọn các ô cần định dạng và sử dụng *Property Inspector* để thay đổi thuộc tính cho các ô, các hàng, các cột.



Hình 4.30. Hộp thuộc tính cho ô.

- Các lựa chọn về hình dáng:
 - ✓ **Horz**: căn lề nội dung ô, cột, hàng theo chiều ngang
 - ✓ **Vert**: căn lề nội dung ô, cột, hàng theo chiều dọc
 - ✓ **W** và **H**: chiều rộng và chiều cao các ô được chọn theo pixel
 - ✓ **Bg phía trên**: ảnh nền cho ô, cột hay hàng
 - ✓ **Bg phía dưới**: màu nền cho ô, cột hay hàng
 - ✓ **Brdr**: màu viền cho ô, cột hay hàng
- Các lựa chọn về bố cục

Chương 4

Macromedia Dreamweaver MX 2004

- ✓ **Merge Cells:** ghép các ô, hàng, cột thành một ô
- ✓ **Split Cell:** chia một ô thành nhiều ô
- ✓ **No Wrap:** không xuống dòng
- ✓ **Header:** định dạng các ô hay các hàng thành dạng tiêu đề bảng

Các cách thay đổi chiều rộng và chiều cao:

- Kéo các hộp kích thước phía dưới hoặc cạnh bên phải của bảng.
- Chọn **Modify \ Table \ Clear Cell Heights** hoặc **Clear Cell Widths**.
- Mở chế độ *Show Code View* và thay đổi chiều rộng và chiều cao thông qua mã HTML.
- Nhập giá trị cho **W** và **H** trong *Property Inspector*.

4.5. Sử dụng Forms

4.5.1. Form là gì?

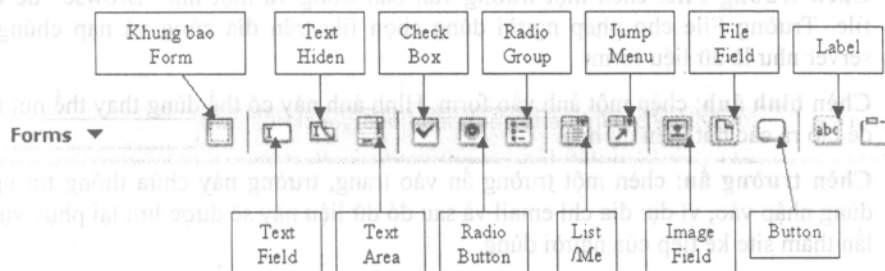
Form đóng vai trò quan trọng cho hoạt động của nhiều đặc tính tương tác trên Web. Sử dụng form, người thiết kế web có thể tương tác với người sử dụng và thu lại các thông tin để sử dụng theo nhiều cách. Một trong những cách mang tính tương tác nhất để thu thập thông tin là yêu cầu tên truy nhập và địa chỉ email hoặc cung cấp thông tin phản hồi cho website.

Form bao gồm 2 phần:

- Mã HTML mô tả form (ví dụ: các trường và các nút nhìn thấy được trên web).
- Kịch bản xử lý trên Server như CGI, ASP, ASP.NET, JSP, PHP ...

Các đối tượng sử dụng trong form

Trong Dreamweaver, form chứa nhiều phần tử, được gọi là các đối tượng form. Chúng được nhóm lại trong panel *Forms* của bảng *Objects* như hình dưới. Có thể sử dụng các đối tượng form này thông qua panel Form hay bằng cách chọn từ menu **Insert \ Form** và **Insert \ Form Objects**.



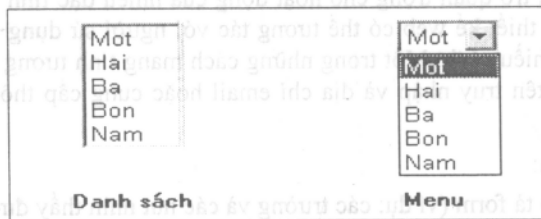
Hình 4.31. Các chức năng của Form.

Chương 4

Macromedia Dreamweaver MX 2004

Nhóm Form bao gồm các biểu tượng sau:

- **Form:** chèn một form vào cửa sổ Document. Dreamweaver chèn các thẻ mở và đóng form trong mã nguồn HTML.
- **Text Field:** chèn trường văn bản vào form. Trường văn bản chấp nhận các kiểu ký tự, số.
- **Hidden Field:** Chèn trường văn bản vào form, nhưng không được hiển thị.
- **Button:** chèn nút nhấn vào form. Khi nút được nhấn, một hành động được thực hiện như gửi thông tin đi hoặc khởi đầu lại các trường trong form.
- **List/Menu:** cho phép tạo nhiều lựa chọn trong một danh sách. Tùy chọn **Danh sách** hiển thị các giá trị khác nhau trong danh sách và cho phép chọn nhiều giá trị. Tùy chọn **Menu** hiển thị các giá trị trong một menu thả xuống và chỉ cho phép chọn một giá trị.



Hình 4.32. Ví dụ về menu và danh sách.

- **Checkbox:** chèn một checkbox vào form. *Checkbox* cho phép người dùng chọn nhiều lựa chọn trong một nhóm các lựa chọn.
- **Radio:** chèn một nút radio vào form. Các nút radio đại diện cho các lựa chọn kiểu loại trừ (ví dụ: yes hoặc no). Chỉ chọn được một lựa chọn trong nhóm các nút radio.
- **Chèn trường File:** chèn một trường văn bản trống và một nút "Browse" để chọn file. Trường File cho phép người dùng chọn file trên đĩa cứng và nạp chúng lên server như là dữ liệu form.
- **Chèn hình ảnh:** chèn một ảnh vào form. Hình ảnh này có thể dùng thay thế nút nhấn để tạo ra các nút kiểu đồ họa.
- **Chèn trường ẩn:** chèn một trường ẩn vào trang, trường này chứa thông tin người dùng nhập vào, ví dụ: địa chỉ email và sau đó dữ liệu này sẽ được lưu lại phục vụ cho lần thăm site kế tiếp của người dùng.
- **Chèn menu Jump:** Tương tự như chèn danh sách/menu, nhưng mỗi phần tử của menu sẽ có một liên kết.

Chương 4

Macromedia Dreamweaver MX 2004

4.5.2. Tạo form

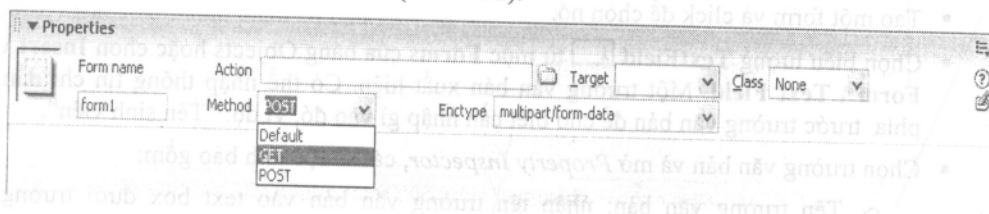
Một form có 3 thành phần cơ bản:

- Các thẻ form: bao gồm địa chỉ URL của kịch bản trên Server sẽ xử lý form và phương thức gửi dữ liệu đến server.
- Các trường dùng trong form: trường văn bản, checkbox, nút radio, danh sách/menu,...
- Nút nhấn: gửi dữ liệu để cho các kịch bản trên server xử lý.

Các bước để tạo một form cho trang web:

Trong cửa sổ Document, đặt con trỏ tại vị trí muốn form xuất hiện và thực hiện theo một trong các cách sau:

- Chọn **Forms** từ menu xổ xuống của *Panel Object*
- Chọn biểu tượng **Form**. Khi một form được tạo, một hình chữ nhật nét đứt viền màu đỏ xuất hiện trong cửa sổ document, biểu thị cho form. Nếu không thấy hình chữ nhật này, chọn **View \ Visual Aids \ Invisible Elements**.
- Click chọn đường viền biểu diễn form để chọn form và mở *Property Inspector* để xem các thuộc tính của form (hình 4.32).



Hình 4.33. Hộp thuộc tính cho của Form.

- Đặt tên form trong trường **Form Name**. Trong trường **Action**, xác định địa chỉ URL nơi sẽ xử lý dữ liệu gửi từ form bằng cách:
- Click biểu tượng thư mục để chọn file. Nhập đường dẫn đến file.
- Trong trường menu thả xuống **Method**, chọn phương thức gửi dữ liệu lên server:
 - ✓ **POST**: gửi dữ liệu form vào phần *head* của trang.
 - ✓ **GET**: gửi dữ liệu form vào sau chuỗi URL.
 - ✓ **Default**: sử dụng giá trị mặc định (GET).

4.5.3. Cách tạo các trường sử dụng trong form

Chú ý: Để dữ liệu từ các đối tượng giới thiệu dưới đây được gửi lên server thì bắt buộc chúng phải nằm trong form.

Trường văn bản

Trường văn bản cho phép người dùng nhập các thông tin vào, ví dụ như: username, password, địa chỉ email, ý kiến,...

Có 3 loại trường văn bản trong Dreamweaver:

- *Trường văn bản trên một dòng:* để nhập một đoạn văn bản ngắn, ví dụ: tên, địa chỉ.
- *Trường văn bản trên nhiều dòng:* cho phép nhập văn bản dài hơn, ví dụ: ý kiến khách hàng.
- *Trường password:* dùng để nhập password. Các ký tự người dùng gõ vào sẽ không được hiện ra, thay vào đó là một ký tự đặc biệt (để đảm bảo tính bảo mật).

Các bước để thêm một trường văn bản:

- Tạo một form và click để chọn nó.
- Chọn biểu tượng **TextField**  từ mục **Forms** của bảng **Objects** hoặc chọn **Insert \ Form \ Text Field**. Một trường văn bản xuất hiện. Có thể nhập thông tin chỉ dẫn phía trước trường văn bản để cho biết cần nhập gì vào đó, ví dụ: "Tên sinh viên".
- Chọn trường văn bản và mở *Property Inspector*, các thuộc tính bao gồm:
 - Tên trường văn bản: nhập tên trường văn bản vào text box dưới trường **TextField**.
 - **Char Width:** nhập số ký tự tối đa sẽ hiển thị trong trường văn bản. Mặc định giá trị này là 20.
 - **Max Chars:** số ký tự tối đa có thể nhập vào trường văn bản đối với văn bản trên một dòng.
 - **Type:** loại trường văn bản, bao gồm 3 loại như trên.
 - **Num Lines:** số dòng đối với trường văn bản trên nhiều dòng. Mặc định là 2.
 - **Init Value:** chuỗi ký tự khởi đầu của trường văn bản khi trang được nạp.
 - **Wrap:** có xuống dòng đối với trường văn bản nhiều dòng không. Có các lựa chọn:

Default: mặc định

Off  không xuống dòng

Chương 4

Macromedia Dreamweaver MX 2004

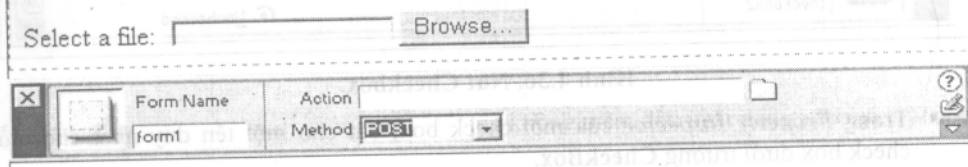
Virtual: xuống dòng nhưng không thêm dấu ngắt dòng vào mã HTML.

Physical: xuống dòng và thêm dấu ngắt dòng vào mã HTML.

Trường File

Trường file gồm một *text box* và một nút nhấn để người dùng có thể chọn file. Các bước để tạo một trường file trong form:

- Chọn **Insert \ Form \ File Field** hoặc click biểu tượng **File Field** trên mục **Forms** của bảng **Objects**.
- Chọn trường file vừa thêm để định nghĩa các thuộc tính cho nó trong **Property Inspector**. Có thể thay đổi tên trường ở mục **FormFieldName**.
- Chọn form chứa trường file vừa tạo, ở mục **Method**, chọn phương thức **POST**. File người dùng chọn sẽ được gửi đi theo địa chỉ điền trong trường **Action**.



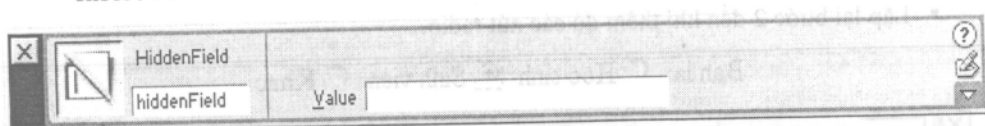
Hình 4.34. Trường file.

Trường ẩn

Trường ẩn thực ra là một trường văn bản không nhìn thấy được trên trang web, dùng để thu thập hay gửi thông tin đi. Dreamweaver thêm biểu tượng đánh dấu trường ẩn vào cửa sổ Document khi nó được thêm. Nếu biểu tượng đánh dấu không xuất hiện khi thêm trường ẩn, chọn **View \ Visual Aids \ Invisible Elements** để nó hiện ra.

Các bước để tạo một trường ẩn:

- Chọn biểu tượng **Hidden Field** trên mục **Forms** của bảng **Objects** hoặc chọn **Insert \ Form \ Hidden Field**.



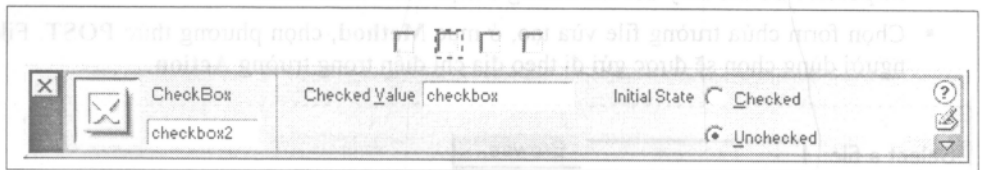
Hình 4.35. Trường ẩn Hidden Field.

- Trong **Property Inspector** của trường ẩn vừa tạo, nhập tên cho trường này.
- Trong trường **Value**, nhập giá trị muốn gán cho trường này (tùy chọn).

Nút Checkbox

Check box cho phép người dùng chọn nhiều lựa chọn trong một nhóm các lựa chọn. Mỗi check box có một tên duy nhất. Để tạo check box:

- Click chọn form cần chèn check box.
- Chọn biểu tượng **Checkbox**  trên mục Forms của bảng Objects hoặc chọn **Insert \ Form \ Check Box**.
- Lặp lại bước 2 đến khi chèn đủ các check box mong muốn.



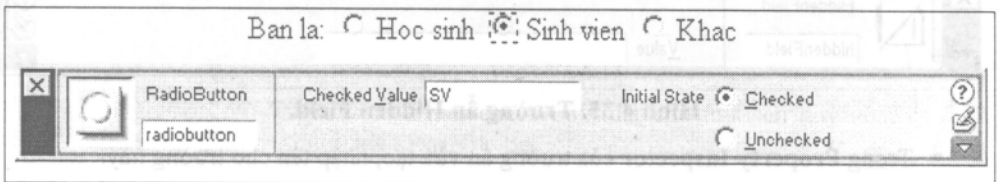
Hình 4.36. Nút Checkbox.

- Trong *Property Inspector* của mỗi check box, đặt cho một tên duy nhất cho mỗi check box dưới trường **CheckBox**.
- Nhập giá trị cho mỗi check box trong trường **CheckedValue**.
- Chọn một trong 2 tùy chọn **Checked** hay **Unchecked** trong trường **Initial State** để cho phép có đánh dấu check box hay không khi trang được nạp.

Nút Radio

Nút Radio chỉ cho phép chọn một trong một nhóm các lựa chọn. Tất cả các nút radio trong một nhóm phải có chung một tên và chứa các giá trị khác nhau trong trường **Value**. Các bước để thêm các nút radio vào form:

- Chọn form cần thêm các nút radio.
- Chọn biểu tượng **Radio**  trên mục **Forms** của bảng Objects hoặc chọn **Insert \ Form \ Radio Button**.
- Lặp lại bước 2 đến khi thêm đủ các nút radio.



Hình 4.37. Nút Radio.

Chương 4

Macromedia Dreamweaver MX 2004

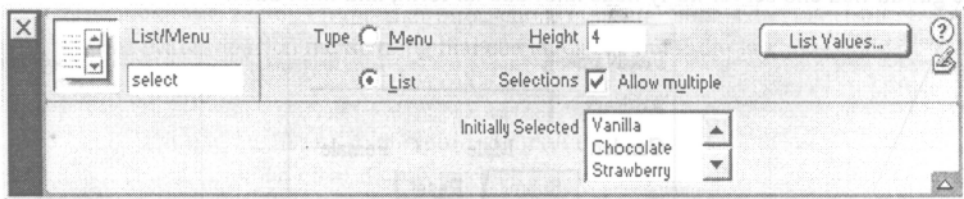
- Trong trường **RadioButton**, đặt một tên cho tất cả các nút radio muốn gộp chung vào một nhóm các lựa chọn.
- Trong trường **CheckedValue**, nhập giá trị cho mỗi nút radio.
- Nếu muốn nút radio ở trạng thái được chọn, click tùy chọn **Checked** trong trường **Initial State**; hoặc có thể để trạng thái của tất cả các nút radio là không chọn.

Danh sách và menu

Danh sách hay menu cho phép người dùng nhiều lựa chọn. Danh sách và menu gần giống nhau, chúng chỉ khác nhau ở cách thể hiện các lựa chọn. Menu là giải pháp lý tưởng cung cấp cho người dùng nhiều lựa chọn trong khoảng không chật hẹp, hiện các lựa chọn chỉ khi người dùng click nút mũi tên. Còn danh sách được thiết kế để hiển thị hết các lựa chọn.

Các bước để tạo một danh sách cuộn / menu thả xuống:

- Tạo một form và click để chọn nó.
- Chọn biểu tượng **List/Menu**  từ mục Forms của bảng Objects hoặc chọn **Insert \ Form \ List/Menu**.
- Nhập thêm các lời chỉ dẫn vào trước menu / danh sách nếu cần thiết.
- Chọn danh sách / menu vừa chèn vào form và chỉnh sửa các thuộc tính trong *Property Inspector* như hình dưới:

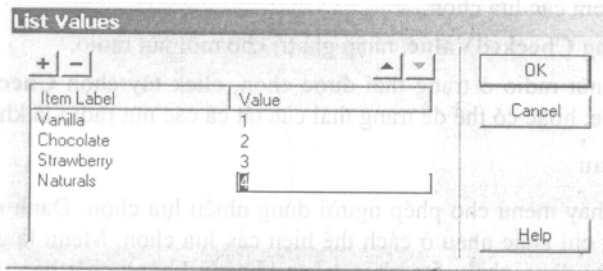


Hình 4.38. Menu.

- Đặt tên cho danh sách/menu trong hộp văn bản List/Menu. Mỗi danh sách/menu phải có một tên duy nhất để phân biệt nhau khi sắp xếp dữ liệu.
- Chọn kiểu danh sách hay menu trong trường **Type**: List hay Menu .
- Nếu là danh sách, đặt chiều cao trong trường **Height**. Đây là số phần tử nhìn thấy trong khung danh sách. Nếu số phần tử nhìn thấy lớn hơn chiều cao, thanh cuộn dọc sẽ xuất hiện.
- Chọn tùy chọn *Allow multiple* để cho phép chọn nhiều lựa chọn trong danh sách.
- Chọn nút **List Values** để nhập các phần tử cho danh sách/menu trong hộp thoại xuất hiện:

Chương 4

Macromedia Dreamweaver MX 2004



Hình 4.39. Danh sách.

- o Thêm mỗi phần tử bằng cách nhập tên và giá trị trong trường Item Label và Value rồi click nút **dấu cộng (+)**.
- o Xóa một phần tử đã thêm bằng cách chọn phần tử và click nút **dấu trừ (-)**.
- o Click nút **OK** khi đã thêm xong các phần tử cho danh sách / menu.
- o Trong trường **Initially Selected** của Property Inspector, có thể chọn các phần tử ở trạng thái được chọn khi trang được nạp.

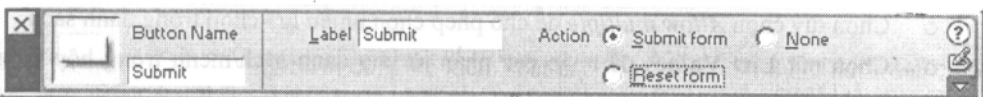
Nút nhấn

Các nút nhấn trong form dùng để kiểm soát các hành động. Sử dụng nút trong form để gửi dữ liệu cho server xử lý hoặc khởi đầu lại form, như hình dưới:

Hình 4.40. Nút trong form để gửi dữ liệu.

Các bước để tạo nút nhấn:

- Click chọn form.
- Đặt con trỏ tại vị trí muốn chèn nút và click biểu tượng **Button**  trên mục Forms của bảng Objects hoặc chọn **Insert \ Form \ Button**. Nút Submit xuất hiện, chọn nút để mở *Property Inspector* của nó.



Hình 4.41. Button.

Chương 4

Macromedia Dreamweaver MX 2004

- Trong trường **ButtonName**, nhập tên duy nhất cho nút.
- Trong trường **Label**, nhập dòng văn bản sẽ xuất hiện trên nút. Ví dụ: Search, Clear, Reset, Submit, OK, ...
- Chọn hành động sẽ thực hiện khi nhấn nút trong trường **Action**: Submit form, Reset hay None.

Các bước để tạo nút nhấn dạng đồ họa

Nút nhấn dạng đồ họa cũng có thể đóng vai trò như một nút nhấn bình thường, thực hiện các thao tác như gửi dữ liệu, khởi đầu form, chạy một file âm thanh.... Ngoài ra, nó làm hấp dẫn người dùng hơn.

- Chọn form cần thêm nút.
- Chọn biểu tượng **Image Field**  từ mục Forms của bảng Objects hoặc chọn **Insert \ Form Objects \ Image Field**. Hộp thoại chọn ảnh xuất hiện. Chọn một ảnh cho nút.
- Trong *Property Inspector* của ảnh vừa chọn, đặt các thuộc tính:
 - **ImageField**: nhập tên cho nút.
 - **SRC**: đường dẫn đến file ảnh.
 - **Alt**: dòng văn bản xuất hiện dạng Tool tip.
 - **Align**: lựa chọn căn lề cho ảnh.

Chú ý: Để cho các phần tử như hộp văn bản, checkbox, ... trong form được căn lề thẳng hàng, có thể tạo bảng để chứa các phần tử này.

4.6. Xử lý sự kiện (Behavior)

4.6.1. Behavior là gì?

Behavior là các đoạn mã kịch bản được xây dựng trên ngôn ngữ JavaScript, cho phép thêm tính năng cho trang web một cách dễ dàng. Một **behavior** là sự kết hợp của một **sự kiện** và một **hành động** gây ra bởi sự kiện đó.

Sự kiện là các thông báo do các trình duyệt sinh ra để báo hiệu rằng người dùng đã thực hiện một hành động gì đó. Khi một sự kiện xảy ra, người dùng có cơ hội trả lời nó. Sự trả lời này gọi là một hành động - bao gồm mã JavaScript định nghĩa trước để thực hiện một nhiệm vụ cụ thể (ví dụ: mở cửa sổ trình duyệt, ẩn / hiện một lớp, chơi một bản nhạc).

Sau khi gắn một *behavior* với một phần tử của trang HTML, thì khi một sự kiện xảy ra với phần tử đó, trình duyệt sẽ gọi hành động (một hàm JavaScript) được kết hợp với sự

Chương 4

Macromedia Dreamweaver MX 2004

kiện. Các sự kiện để gọi một hành động có thể thay đổi theo các trình duyệt. Một hành động có thể gọi vài sự kiện khác nhau, và ta có thể đặt thứ tự thực hiện các hành động đó.

Các sự kiện

Hầu hết các sự kiện chỉ có thể sử dụng với một số phần tử xác định trong trang HTML. Do đó cần biết các sự kiện nào được các trình duyệt hỗ trợ cho phần tử nào trong trang ta chèn phần tử vào trang HTML và gắn cho nó một sự kiện, rồi mở menu Events để biết các sự kiện nào có thể gắn với phần tử đó.

Một số sự kiện quan trọng được liệt kê trong bảng sau:

Bảng 4.12. Các sự kiện quan trọng

Sự kiện	Mô tả
onClick (NS3, NS4, IE3, IE4, IE5)	Được gọi khi người dùng nhấn chuột vào phần tử: như liên kết, nút nhấn hay bản đồ ảnh. (Việc nhấn chuột chưa hoàn thành đến khi người dùng nhả nút chuột).
onDbClick (NS4, IE4, IE5)	Được gọi khi người dùng nhấn đúp vào phần tử.
onKeyDown (NS4, IE4, IE5)	Được gọi khi người dùng nhấn một phím bất kỳ (sự kiện này sinh ra mà không cần chờ đến khi người dùng nhả phím).
onKeyPress (NS4, IE4, IE5)	Được gọi khi người dùng nhấn và nhả một phím bất kỳ.
onLoad (NS3, NS4, IE3, IE4, IE5)	Được gọi khi một hình ảnh hay một trang HTML được nạp xong.
onMouseDown (NS4, IE4, IE5)	Được gọi khi người dùng nhấn nút chuột (sự kiện này sinh ra mà không cần chờ đến khi người dùng nhả nút chuột).
onMouseMove (IE3, IE4, IE5)	Được gọi khi người dùng di chuyển chuột qua một phần tử nào đó trong trang HTML (con trỏ di chuyển trong phạm vi biên của phần tử).
onMouseOver (NS3, NS4, IE3, IE4, IE5)	Được gọi khi chuột <i>lần đầu</i> di chuyển để chỉ đến một phần tử (khi con trỏ di chuyển từ vị trí không chỉ đến vị trí chỉ vào phần tử).

Chương 4

Macromedia Dreamweaver MX 2004

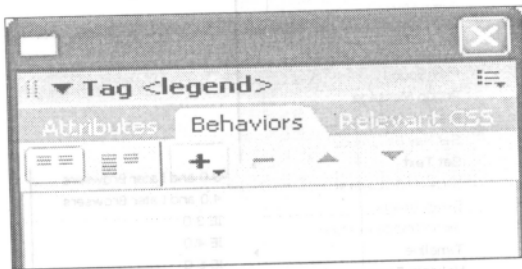
onMouseOut (NS3, NS4, IE4, IE5)	Được gọi khi con trỏ di chuyển ra khỏi phần tử HTML (thường là một hình ảnh, một liên kết được gắn với một hình ảnh).
onSubmit (NS3, NS4, IE3, IE4, IE5)	Được gọi khi người dùng gửi một form.

4.6.2. Gắn behavior cho một phần tử HTML

Panel Behaviors

Có thể dùng *panel Behavior* để gắn các hành động cho các phần tử HTML (đặc biệt là các thẻ) và thay đổi các tham số của các hành động đã gắn cho một phần tử.

- Chọn **Window \ Behaviors** hoặc **Windows \ Launcher \ Behavior** hoặc nhấn tổ hợp phím **Shift-F3** để mở panel Behaviors như hình vẽ:



Hình 4.42. Hộp thoại Panel Behaviors.

Thẻ HTML đang được chọn trong cửa sổ *Document* xuất hiện trên thanh tiêu đề của cửa sổ *Behavior*. Các *behavior* đã gắn với phần tử đang được chọn, xuất hiện trong danh sách các behavior được liệt kê theo thứ tự ABC theo các sự kiện. Có rất nhiều hành động cho cùng một sự kiện, các hành động xuất hiện theo thứ tự chúng sẽ được thực hiện. Nếu không có *behavior* nào xuất hiện trong danh sách các behavior, tức là chưa có *behavior* nào được gắn cho phần tử HTML đang được chọn.

Panel Behaviors có các tùy chọn sau:

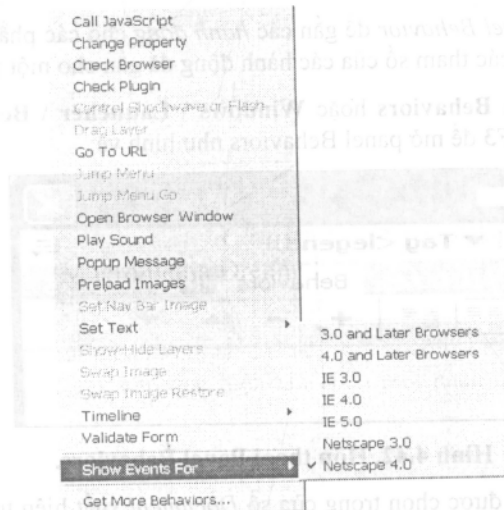
- **Các hành động (+):** khi nhấn vào nút dấu cộng, một menu liệt kê các hành động có thể gắn với phần tử HTML đang được chọn xuất hiện, Chọn một hành động từ menu. Các hành động chứa cả mã JavaScript và mã HTML.
- **Xóa (-):** nhấn nút dấu trừ sẽ khiến hành động và sự kiện được chọn bị xóa khỏi danh sách các *behavior*.
- **Mũi tên lên và xuống:** nhấn các mũi tên này để thay đổi thứ tự thực hiện hành động cho phần tử HTML đang được chọn.

Chương 4

Macromedia Dreamweaver MX 2004

- **Các sự kiện:** là một menu gồm tất cả các sự kiện có thể đặt để thực hiện hành động được chọn. Menu này chỉ xuất hiện khi chọn một hành động trong danh sách các *behavior*.
- **Show Event For** (menu con của menu các sự kiện) giới hạn các *behavior* cho các trình duyệt. Lựa chọn trong menu này sẽ quyết định các sự kiện nào sẽ xuất hiện trong menu Các sự kiện.

Gắn behavior cho một phần tử HTML



Hình 4.43. Gắn behavior cho một phần tử HTML.

Có thể gắn các *behavior* cho toàn bộ trang HTML (thẻ *body*) hoặc các liên kết, hình ảnh, các phần tử trong form và các phần tử HTML khác.

Các bước để gắn một *behavior* cho một phần tử HTML:

- Chọn phần tử, ví dụ một hình ảnh hay một liên kết. Để gắn một *behavior* cho toàn bộ trang, chọn thẻ *<body>* ở phía cuối, bên trái cửa sổ Document, vùng chọn thẻ.
- Mở panel Behaviors. Thẻ HTML đang được chọn sẽ hiện trên thanh tiêu đề của panel này.
- Nhấn nút dấu cộng (+) trên panel và chọn một hành động cho phần tử từ menu thả xuống. Nếu một hành động trên danh sách có màu xám thì nghĩa là nó không làm việc với phần tử đang chọn. Ví dụ, hành động *Control Shockwave or Flash* chỉ làm việc nếu một file shockwave hoặc flash được chọn. Sau khi chọn hành động, một hộp thoại gắn với hành động sẽ xuất hiện yêu cầu định nghĩa các tham số cho behavior tương ứng (xem hình trên).

Chương 4

Macromedia Dreamweaver MX 2004

- Xác định các tham số cho hành động vừa chọn để behavior có thể làm việc và nhấn nút **OK** để đóng hộp thoại.
- Nhấn nút tam giác ngược (phía phải của mục *Events*) trong *panel Behavior* và chọn một sự kiện sẽ gọi hành động. Dreamweaver liệt kê các sự kiện tùy thuộc vào trình duyệt được chọn cho behavior này.

Chú ý: không thể gán một *behavior* cho văn bản bình thường, nhưng có thể gán một behavior cho một liên kết.

4.6.3. Các hành động cho behavior trong Dreamweaver

Các hành động cho *behavior* trong Dreamweaver được tạo ra để làm việc với nhiều trình duyệt nhất. Nếu ta bỏ đoạn mã trong một hành động hoặc thay nó bằng đoạn mã riêng của mình, trang web của ta có thể mất tính tương thích giữa các trình duyệt.

Mặc dù các hành động trong Dreamweaver được thiết kế để tương thích với nhiều trình duyệt, một số hành động cũng không thể thực hiện được trong các trình duyệt cũ. Hơn nữa có một số trình duyệt không hỗ trợ JavaScript hoặc tắt tùy chọn hỗ trợ JavaScript. Có thể đạt được tính tương thích chéo tốt nhất giữa các trình duyệt bằng cách sử dụng cặp thẻ `noscript` để những trình duyệt không hỗ trợ JavaScript vẫn xem được site.

Bảng 4.13. Các hành động trong Dreamweaver

Hành động	Mô tả
Call JavaScript	Thực hiện một đoạn mã JavaScript khi xảy ra sự kiện.
Change Property	Thay đổi thuộc tính của một đối tượng (ví dụ: màu nền của một lớp hoặc hành động của một form).
Check Browser	Đưa người dùng đến các trang khác nhau dựa trên trình duyệt và phiên bản của họ.
Check Plugin	Hướng người dùng đến các trang khác nhau nếu họ đã cài plugin.
Shockwave or Flash	Thêm các trình diễn multimedia cho trang web.
Drag Layer	Cho phép người dùng định vị lại lớp lên trên một lớp khác.
Go To URL	Mở một trang mới trong cửa sổ hiện hành hoặc trong một frame, thường dùng để thay đổi nội dung của một hay hai frame chỉ với một lần click chuột.
Jump Menu	Danh sách gồm nhiều lựa chọn được liên kết với các URL. Menu Jump thường đi với nút Go.
Open Browser Window	Mở một URL trong cửa sổ mới. Có thể định các thuộc tính cho cửa sổ mới.

Chương 4
Macromedia Dreamweaver MX 2004

Play Sound	Chơi một file âm thanh.
Poup Message	Dùng JavaScript để hiển thị một hộp thông báo.
Preload Images	Nạp những ảnh không xuất hiện ngay trên trang (ví dụ những hình ảnh trong timelines, behaviors, JavaScript). Tùy chọn này chống hiện tượng trễ do thời gian tải hình ảnh xuống.
Set Navigation Bar	Chuyển đổi một ảnh thành một ảnh " <i>navigation bar</i> ", hay thay đổi cách hiển thị và các hành động cho hình ảnh trong thanh <i>navigation</i> .
Set Text	Thay đổi văn bản trong frame, lớp, trường văn bản hay thanh trạng thái, thường dùng để hiển thị thông tin một cách động.
Show-Hide layers	Thay đổi thuộc tính thấy được của lớp theo sự phản ứng của người dùng với các sự kiện, do đó hiện ra hình ảnh hay thông tin một cách từ từ. Đây là một công cụ mạnh để tạo các hiệu ứng thú vị.
Swap Image	Tráo một ảnh này thành một ảnh khác bằng cách thay đổi thuộc tính <i>src</i> của thẻ <i>img</i> . Nó dùng để tạo ra các nút "cuộn", tạo ra các phần tử tương tác trên trang web.
Timelines	<i>Behavior</i> kết hợp với tiến trình tạo ra các hoạt hình, cung cấp sự tương tác mạnh cho trang web. Có thể dùng các behavior để chạy, dừng hay chuyển đến một frame trong tiến trình bằng cách liên kết nó với một <i>behavior</i> , hoặc thực hiện các <i>behavior</i> trong tiến trình.
Validate Form	Kiểm tra nội dung các hộp văn bản để yêu cầu người dùng phải nhập đúng định dạng dữ liệu. Nó kiểm tra các trường người dùng nhập vào <i>form</i> khi người dùng nhấn nút <i>submit</i> để gửi thông tin từ <i>form</i> lên <i>server</i> .

4.7. Kết chương

Trong chương này chúng ta đã làm quen với công cụ *MX Dreamweaver*, một công cụ soạn thảo trực quan chuyên nghiệp cho phép người sử dụng soạn thảo, tạo các trang web với rất nhiều đặc tính mà không cần trực tiếp viết mã HTML. Trong phần sau chúng ta sẽ tìm hiểu một số ngôn ngữ tiên tiến trên Web với các tính năng nâng cao hơn HTML như XML, XHTML.

PHẦN 3

CSS, XML và XHTML

NỘI DUNG

Chương 5 - Giới thiệu về Cascading Style Sheets (CSS)

Chương 6 - Giới thiệu về XML

Chương 7 - Giới thiệu về XHTML

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

Nội dung của chương:

- Tìm hiểu về CSS hay Style.
- Cách tạo và sử dụng Style trong Website của bạn.

5.1. Giới thiệu CSS

Ban đầu, HTML được thiết kế chủ yếu như một tập quy ước mã hóa chung cho phép mọi người sử dụng đều có thể xem trang Web dù đang sử dụng trên bất cứ nền hệ máy tính nào. HTML đưa ra kiểu định dạng có cấu trúc, nhưng không hỗ trợ các nhà thiết kế có nhiều quyền chi phối diện mạo trang Web. Để đáp ứng nhu cầu này các hãng sản xuất trình duyệt cạnh tranh nhau bằng cách bổ sung thêm các thẻ HTML mới độc quyền, không theo chuẩn, dẫn đến sự không hiểu nhau giữa các trình duyệt. Trước tình hình này tổ chức World Wide Web Consortium (W3C) nỗ lực hình thành một hệ thống duy trì tính phổ quát chung của HTML, đồng thời cho phép nhà thiết kế có quyền chi phối hình thức trang Web ở mức cao hơn. Kết quả của nỗ lực này là sự ra đời của *Cascading Style Sheets*.

Cascading Style Sheets, còn gọi là CSS, hay đơn giản chỉ là Style, cho phép bạn gán nhiều thuộc tính cùng lúc cho tất cả các phần tử được đánh dấu bằng cùng một thẻ cụ thể trên trang Web. Ví dụ, có thể hiển thị tất cả đề mục <H1> với phông chữ, cỡ chữ, màu chữ cụ thể. Mặc dù có thể định dạng bằng thẻ HTML, nhưng sử dụng Style vẫn mang lại nhiều ưu điểm vượt trội.

5.2. Tại sao phải sử dụng Style?

Style tiết kiệm thời gian

Bạn thử hình dung Style tiết kiệm thời gian đến thế nào cho bạn trong trường hợp phải ấn định phông chữ cho mỗi tiêu đề của mỗi đoạn trong trang Web rất dài bằng thao tác thủ công với HTML, trong khi với Style bạn chỉ cần gõ một dòng cho mỗi phần tử ở đầu trang là xong.

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

CSS rất dễ thay đổi

Giả sử trang Web của bạn sau khi đã hoàn thành mà bạn lại muốn thay đổi phông chữ, cỡ chữ và màu chữ cho mỗi đoạn bằng tay, quả là sẽ rất vất vả nếu như bạn không dùng Style. Với Style bạn sẽ hiệu chỉnh nhanh hơn: chỉ cần sửa các quy định về định dạng này tại một chỗ trong tệp Style mà thôi.

Sự nhất quán

Sử dụng Style sẽ tạo ra sự nhất quán và thống nhất trong bố cục trang Web, bạn không cần phải nhớ nhiều về từng định dạng nếu bạn sử dụng Style.

Khả năng chi phối đa dạng

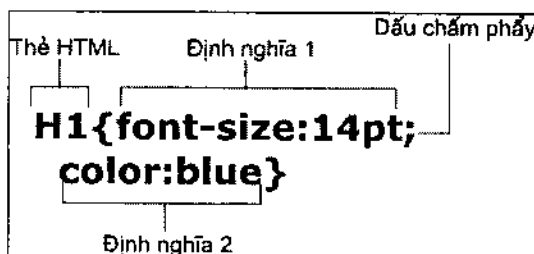
Thông qua Style, bạn có thể định dạng văn bản theo nhiều cách khác nhau ngay cả khi các thẻ HTML không có khả năng đó. Ví dụ bạn có thể ấn định khoảng cách dòng, màu nền, xóa định dạng in đậm và in nghiêng...

Tạo ra định dạng chung cho toàn bộ Website

Style giúp dễ dàng tạo kiểu định dạng chung cho mọi trang Web của bạn. Bạn chỉ phải định nghĩa Style một lần duy nhất. Khi thay đổi chỉ cần thay đổi ở một nơi và tất cả các trang đều được cập nhật ngay lập tức.

5.3. Cấu trúc của một Style

Cấu trúc của một Style được tạo thành từ một thẻ gốc HTML và một hay nhiều định nghĩa xác định style đó. Mỗi định nghĩa chứa một thuộc tính, dấu hai chấm, và một hay nhiều giá trị. Theo sau dấu hai chấm không có khoảng trắng. Các định nghĩa phải được phân cách bằng dấu chấm phẩy. Ví dụ: *color:blue* với *color* là thuộc tính còn *blue* là giá trị.



Hình 5.1. Style tạo thành từ một thẻ H1.

Một số thuộc tính cho phép nhóm lại với nhau dựa vào một thuộc tính bao trùm đặc biệt (như *font*, *background*, *border* ...).

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

Ví dụ 5.1.

font:bold 14pt Tahoma

tương đương với

font-size:14pt; font-weight:bold; font-family:Tahoma

Một số thuộc tính tự động được thừa hưởng từ thẻ này qua thẻ kia. Ví dụ nếu bạn định nghĩa chuỗi văn bản đi kèm thẻ `<H1>` màu xanh, thì mọi ký tự đi kèm thẻ `<EM>` trong `<H1>` cũng sẽ có màu xanh. Tất nhiên, bên ngoài thẻ `<H1>` sẽ không có màu xanh kể cả thẻ `<EM>`.

Có thể phân loại thành ba loại Style sau:

- Style trong (*Internal Style Sheet*)
- Style ngoài (*External Style Sheet*)
- Style cục bộ (*Inline Style Sheet*)

Bạn có thể sử dụng Style bất kỳ hoặc cả ba loại nhưng có một lưu ý là: Style cục bộ sẽ giành quyền của style trong, còn style trong sẽ giành quyền của style ngoài. Chúng ta sẽ đi vào tìm hiểu chi tiết cách thiết lập và áp dụng từng loại style trong phần dưới đây.

5.4. Style trong (Internal Style Sheet)

Style trong rất lý tưởng để sử dụng cho những trang tài liệu riêng lẻ chứa đầy chữ. Chúng cho phép bạn ấn định style ở đầu trang và sẽ được sử dụng thông qua tài liệu HTML. Nếu dự định áp dụng style cho nhiều trang tốt hơn hết bạn nên sử dụng style ngoài.

Cách thiết lập Style trong

- Ở đầu tài liệu HTML, gõ thẻ `<STYLE>` giữa hai thẻ `<HEAD>` và `</HEAD>`.
- Gõ tên của thẻ nào bạn muốn định nghĩa thuộc tính (ví dụ: `H1`, `P`, `A` ...).
- Gõ ký tự “{” để đánh dấu điểm bắt đầu thuộc tính của thẻ này.
- Định nghĩa bao nhiêu thuộc tính tùy thích cho thẻ, chú ý phân cách mỗi thuộc tính bằng dấu chấm phẩy (;).
- Gõ ký tự “}” nhằm kết thúc phần định nghĩa thuộc tính cho một thẻ.
- Lặp lại nếu muốn định nghĩa thêm thẻ khác.
- Gõ thẻ `</STYLE>` để hoàn tất style.

Chú ý:

- Mỗi tập hợp thuộc tính phải bắt đầu dấu móc mở “{” và kết thúc bằng dấu móc đóng “}”. Mỗi thuộc tính và giá trị của nó phải được phân cách bằng dấu chấm phẩy.
- Có thể viết các thuộc tính và giá trị của một thẻ trên mỗi dòng khác nhau.

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

- Có thể định nghĩa thuộc tính cho nhiều thẻ cùng lúc bằng cách dùng dấu phẩy làm dấu phân cách thẻ: *H1, H2, H3 {color:blue}* sẽ hiển thị 3 cấp đề mục có màu xanh.
- Bổ sung cặp thẻ *comment* (*<!-- -->*) vào giữa cặp thẻ *<STYLE> </STYLE>* nhằm che giấu style khỏi những trình duyệt không hỗ trợ chúng. Nếu không, các trình duyệt này có thể hiển thị cả phần định nghĩa style cho người xem.

Ví dụ 5.2.

- Tạo trang HTML sau:

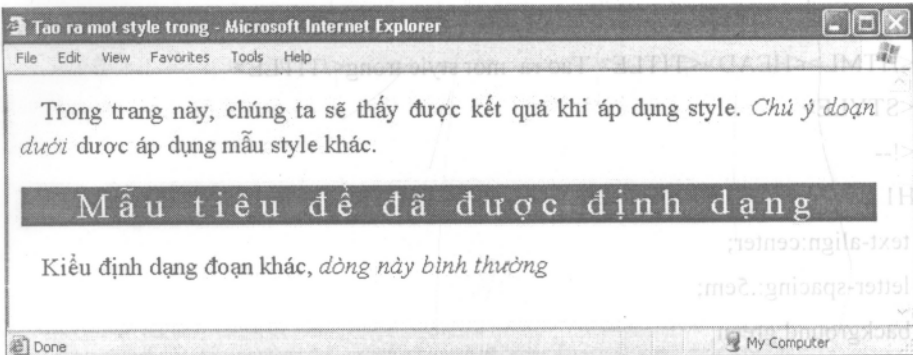
```
<HTML><HEAD><TITLE> Tao ra  mot style trong</TITLE>
<STYLE>
<!--
H1 {
  text-align:center;
  letter-spacing:.5em;
  background:green;
  color:yellow;
  font:normal 20pt "Times New Roman", "Tahoma"
}
P{
  text-align:justify;
  text-indent:12pt;
  font:15pt/20pt "Arial", "Verdana"
}
P:intro {text-indent:0;font:italic bold}
P:intro EM {font-style:normal}
-->
</STYLE>
</HEAD>
<BODY>
<P class=intro>Trong trang này,chúng ta sẽ thấy được kết quả khi áp dụng style.
<EM> Chú ý đoạn dưới</EM> được áp dụng mẫu style khác.
```

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

```
<H1> Mẫu tiêu đề đã được định dạng </H1>  
<P> Kiểu định dạng đoạn khác, <EM>dòng này bình thường </EM>  
<BODY>  
</HTML>
```

- Kết quả hiển thị trên trình duyệt :



Hình 5.2. Kết quả khi áp dụng style trong.

5.5. Style ngoài (External Style Sheet)

Style ngoài thích hợp đối với mục đích muốn đồng nhất cho mọi trang Web trong một site theo cùng một định dạng. Thay vì lấy style từ từng bảng style trong tại đầu mỗi trang Web, bạn có thể ấn định cho mỗi trang Web tham chiếu đến bảng style ngoài. Và như vậy, mỗi khi có một sự thay đổi của mẫu style ngoài, các trang Web sẽ tự cập nhật và hiển thị theo mẫu mới mà không cần bạn phải mở từng trang ra sửa.

Cách thiết lập Style ngoài

- Tạo một file mới.
- Gõ tên thẻ có thuộc tính cần định nghĩa (ví dụ: H1, P ..).
- Gõ kí tự “{” để đánh dấu điểm bắt đầu thuộc tính của thẻ này.
- Định nghĩa bao nhiêu thuộc tính tùy thích cho thẻ, chú ý phân cách mỗi thuộc tính bằng dấu chấm phẩy (;).
- Gõ kí tự “}” nhằm kết thúc phần định nghĩa thuộc tính cho một thẻ.
- Lặp lại nếu muốn định nghĩa thêm thẻ khác.
- Lưu file này dưới dạng Text Only vào thư mục mong muốn với phần mở rộng là .CSS để chỉ đây là tài liệu Cascading Style Sheet.

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

Cách sử dụng Style ngoài

Để sử dụng file style đã tạo, bạn phải liên kết nó với mỗi trang cần định dạng. Ví dụ trong phần *HEAD* của từng trang HTML sẽ áp dụng mẫu style *url.css*, bạn hãy gõ đoạn mã sau:

```
<HEAD>
<LINK REL=stylesheet TYPE="text/css HREF ="url.css">
</HEAD>
```

Chú ý:

- URL phải tương quan so với tập tin được áp dụng. Có nghĩa là bạn phải định đúng đường dẫn đến style theo phương pháp định địa chỉ URL đã học ở chương 2.
- Bạn có thể viết chú thích cho style bên trong cặp thẻ */* */*.

Kết hợp Style ngoài và Style trong

Bạn có thể vừa áp dụng Style ngoài vừa áp dụng Style trong hay cục bộ trong cùng một tài liệu HTML.

Ví dụ 5.3.

Ví dụ bạn tạo một style ngoài lưu lại với tên *styles.css* như sau:

```
/* styles.css _ Đây là dòng chú thích */
<!--
H1 {
text-align:center;
letter-spacing:.5em;
background:green;
color:yellow;
font:normal 20pt "Times New Roman", "Tahoma"
}
P{
text-align:justify;
text-indent:12pt;
```


Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

```
font:15pt/20pt "Arial", "Verdana"  
}  
P:intro {text-indent:0;font:italic bold}  
P:intro EM {font-style:normal}  
-->
```

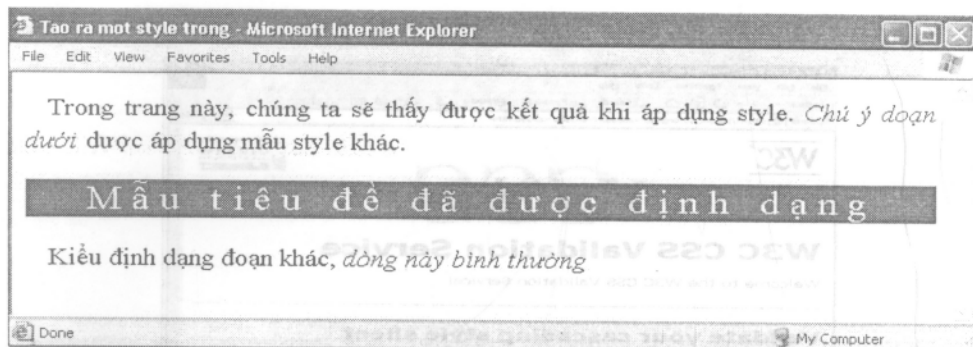
Sau đó tạo một trang HTML và chèn thêm đoạn mã sử dụng style ngoài *styles.css*. Chú ý là ở đây file *styles.css* nằm cùng vị trí của trang HTML này.

```
<HTML>  
<HEAD> <TITLE>Tạo một Style ngoài</TITLE>  
<LINK REL=stylesheet TYPE="text/css" HREF="styles.css">  
</HEAD>  
<BODY>  
<P class=intro>Trong trang này, chúng ta sẽ thấy được kết quả khi áp dụng style.  
<EM> Chú ý đoạn dưới </EM> được áp dụng mẫu style khác.  
<H1> Mẫu tiêu đề đã được định dạng </H1>  
<P> Kiểu định dạng đoạn khác, <EM> dòng này bình thường </EM>  
</BODY>  
</HTML>
```

Kết quả hiển thị trên trình duyệt tương tự như ví dụ trước (hình 5.3).

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)



Hình 5.3. Kết quả khi áp dụng style ngoài.

5.6. Style cục bộ (Inline Style Sheet)

Đây là phương pháp dễ dàng nhất cho phép bạn áp dụng các style bằng cách sử dụng định nghĩa style ngay bên trong mỗi thẻ của HTML. Với phương pháp này bạn có thể tạo ra những kiểu định dạng mới mà bản thân các thẻ HTML không thể có được.

Ví dụ 5.4.

```
<p style="color: sienna; margin-left: 20px">
```

Đây là một đoạn

```
</p>
```

Cách thiết lập Style cục bộ

- Trong phạm vi thẻ HTML nào cần định dạng gõ xâu kí tự: **‘STYLE=’**
- Gõ định nghĩa thuộc tính và giá trị theo định dạng: *thuộc tính: giá trị*
- Muốn tạo thêm định nghĩa style, gõ dấu ‘;’ và thực hiện lại bước 2
- Gõ dấu “ để kết thúc định nghĩa style.

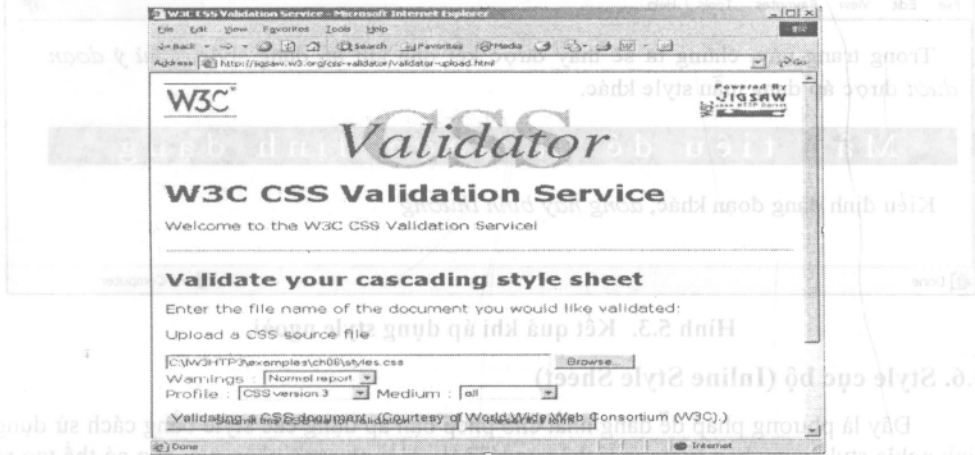
Chú ý:

- Nhớ phân cách các thuộc tính bằng dấu chấm phẩy ;
- Đừng thêm khoảng trống vào giữa giá trị và đơn vị. Ví dụ nếu bạn để thuộc tính căn lề là **“margin-left: 20 px”** thì nó chỉ làm việc đúng với trình duyệt IE6 còn không làm việc trong trình duyệt Mozilla hoặc Netscape.

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

5.7. Kiểm tra cú pháp của style bằng dịch vụ W3C CSS



Hình 5.4. Giao diện trang upload file css của W3C để kiểm tra.

Bạn có thể vào trang Web của tổ chức W3C để kiểm tra file style của mình có đúng cú pháp và phù hợp với chuẩn W3C hay không. Để làm được điều này hãy vào địa chỉ sau để tải file style của bạn lên để kiểm tra: <http://jigsaw.w3.org/css-validator/validator-upload.html>. (hình 5.4)

Làm theo các chỉ dẫn và chờ kết quả trả lời, bạn sẽ biết được những lỗi cần phải sửa, còn nếu nhận được thông báo không có lỗi có nghĩa là bạn hoàn toàn có thể yên tâm vì mình đã trở thành một người viết mã thành thạo.

5.8. Một số cách định nghĩa Style khác

Style dành cho lớp (class)

Có thể áp dụng Style một cách có chọn lọc bằng cách chỉ rõ kiểu style cho các thuộc tính của thẻ HTML cùng tên.

Ví dụ 5.5.

- Ban đầu bạn hãy định nghĩa kiểu style cho từng thẻ theo kiểu:

```
p.right {text-align: right}
p.center {text-align: center}
```

- Tiếp theo sử dụng thuộc tính class trong mỗi thẻ HTML cần sử dụng style như sau:

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

```
<p class="right">  
Đoạn này sẽ được căn lề phải.  
</p>  
<p class="center">  
Đoạn này sẽ được căn giữa.  
</p>
```

Chú ý:

- Chỉ dùng một thuộc tính *class* trong một thẻ HTML. Ví dụ sau là sai vì dùng hai *class*

```
<p class="right" class="center">  
Đây là một đoạn.  
</p>
```

- Cũng có thể bỏ qua tên thẻ trong phần định nghĩa style. Khi đó tất cả các thẻ HTML, nào có thuộc tính này sẽ có thể áp dụng được style đã định dạng. Quan sát ví dụ sau, trong ví dụ này tất cả các thẻ HTML có chứa *class="center"* sẽ được căn giữa:

```
.center {text-align: center}  
  
<h1 class="center">  
Phân tiêu đề được căn giữa.  
</h1>  
  
<p class="center">  
Đoạn này cũng được căn giữa.  
</p>
```

Style kiểu nhận dạng (id)

Thay vì tạo một lớp toàn thẻ HTML, bạn cũng có thể xác định từng thẻ riêng lẻ, sau đó chỉ rõ tên nhận diện. Một thuộc tính ID phải xác định duy nhất trong tài liệu HTML.

Ví dụ 5.6.

Ví dụ sau xác định ID cho thẻ *p* là *tenID* (phân cách bằng dấu #).

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

```
p#tenID
{
text-align: center;
color: red
}
```

Để áp dụng kiểu style này bạn phải sử dụng thuộc tính *ID=tenID*:

```
<p ID=tenID>
Đây là một đoạn được căn giữa và chữ màu đỏ
</p>
```

Ngoài ra chúng ta có thể chỉ định *tenID* cho nhiều thẻ cùng một lúc. Khi đó mọi thẻ đều có thể sử dụng style này thông qua *tenID*.

```
*#tenID {color: green}
```

5.9. Thiết lập vị trí với Style

Thuộc tính xác định vị trí **:position** cho phép ta chỉ rõ vị trí của một đối tượng. Nó cũng cho phép bạn thiết lập hình dạng của đối tượng, điểm đặt ... Sau đây là một vài cách thiết lập vị trí trong tài liệu HTML thông qua thuộc tính xác định vị trí *position*.

Thiết lập vị trí tương đối

Mỗi một thành phần trong trang đều có một vị trí tự nhiên. Di chuyển vị trí thành phần so với vị trí ban đầu được gọi là *di chuyển tương đối*. Khi di chuyển tương đối các thành phần xung quanh sẽ không chịu ảnh hưởng gì cả.

Cú pháp xác định vị trí tương đối với thuộc tính *position*:

- Gõ *position:relative*;
- Gõ *top*, *right*, *bottom*, hoặc *left*
- Gõ *tx*; với *x* là khoảng cách muốn dịch lệch tương đối so với vị trí ban đầu. Ví dụ: 20pt hoặc 2cm.

Ví dụ 5.7.

```
<html>
<head>
```

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

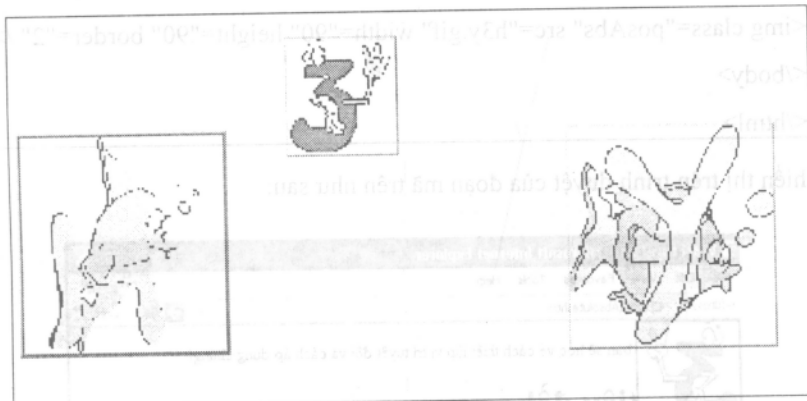
```
<title>Vị trí tương đối</title>
<style>
    img.pos1 {position:relative;top:50;left:0}
    img.pos2 {position:relative;top:10;left:25}
    img.pos3 {position:relative;top:50;left:100}
</style>
</head>
<body>



</body>
</html>
```

Chú ý là vị trí tương đối ở đây chính là so với vị trí ban đầu của thành phần đó, chứ không phải là so với các thành phần xung quanh.

Kết quả hiển thị trên trình duyệt của đoạn mã trên như sau:



Hình 5.5. Vị trí tương đối của ảnh thông qua style.

Thiết lập vị trí tuyệt đối

Thông thường, mỗi thành phần trong trang Web lưu chuyển theo thứ tự hiển thị. Chúng ta cũng có thể thiết lập vị trí tuyệt đối bằng cách định rõ vị trí chính xác của chúng so với thành phần cha. Phải chú ý hiện tượng phủ chồng lên nhau của các đối tượng.

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

Cú pháp:

- Gõ `position: absolute;`
- Gõ `top`, `right`, `bottom`, hoặc `left`
- Gõ `:x`; với `x` là khoảng cách lệch so với vị trí của thành phần cha. Ví dụ: 10pt, 2cm...

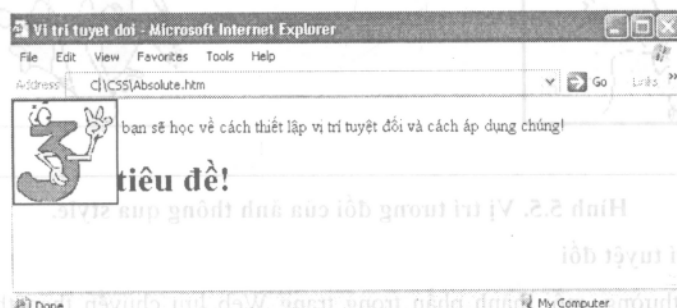
Ví dụ 5.8.

Bạn tạo trang Web sau:

```
<html>
<head>
<title>Vị trí tương đối</title>
<style> img.posAbs{position:absolute;top:0;left:0} </style>
</head>
<body>
<p>Trong bài này bạn sẽ học về cách thiết lập vị trí tuyệt đối và cách áp dụng chúng!</p>
<h1>Dòng tiêu đề!</h1>

</body>
</html>
```

Kết quả hiển thị trên trình duyệt của đoạn mã trên như sau:



Hình 5.6. Vị trí tuyệt đối của ảnh thông qua style.

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

Bố trí thứ tự ưu tiên

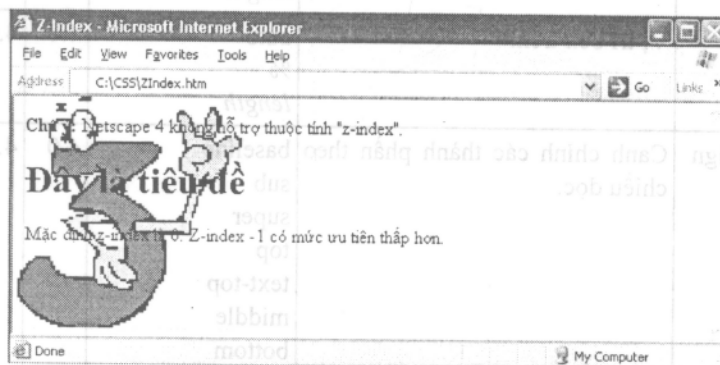
Sau khi đi chuyển tương đối và tuyệt đối, sẽ có lúc bạn gặp trường hợp các đối tượng phủ chồng lên nhau. Có thể chọn thứ tự ưu tiên cho thành phần nào được nằm bên trên hay ở dưới bằng cách sử dụng thuộc tính **z-index**. Tuy nhiên có một điểm cần chú ý là thuộc tính này không được Netscape 4 hỗ trợ.

Ví dụ 5.9.

```
<html>
<head><title>Z-Index</title>
<style>
img.x{position:absolute; left:0px; top:0px; z-index:-1}
</style>
</head>
<body>
<p><b>Chú ý:</b> Netscape 4 không hỗ trợ thuộc tính "z-index".</p>
<h1>Đây là tiêu đề</h1>

<p>Mặc định z-index là 0. Z-index -1 có mức ưu tiên thấp hơn.</p>
</body>
</html>
```

Kết quả hiển thị trên trình duyệt (hình 5.7)



Hình 5.7. Thiết lập thứ tự của thành phần bằng z-index.

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

Bảng sau mô tả một số thuộc tính khác của position. Cách sử dụng các thuộc tính này tương tự. (Kí hiệu: NN: Netscape, IE: Internet Explorer, W3C: Web Standard)

Bảng 5.1. Các thuộc tính khác của position

Thuộc tính	Mô tả chức năng	Giá trị	NN	IE	W3C
bottom	Vị trí phía dưới.	auto % <i>length</i>	6.0	5.0	CSS2
clip	Thiết lập hình dạng thành phần. Các thành phần được bao bên trong hình dạng này, và hiển thị.	<i>shape</i> auto	6.0	4.0	CSS2
left	Vị trí bên trái.	auto % <i>length</i>	4.0	4.0	CSS2
overflow	Thiết lập những hành động nếu nội dung của đối tượng quá lớn.	visible hidden scroll auto	6.0	4.0	CSS2
position	Đặt vị trí là cố định, tương đối, tuyệt đối hay di chuyển.	static relative absolute fixed	4.0	4.0	CSS2
right	Vị trí bên phải.	auto % <i>length</i>		5.0	CSS2
top	Vị trí bên trên.	auto % <i>length</i>	4.0	4.0	CSS2
vertical-align	Canh chỉnh các thành phần theo chiều dọc.	baseline sub super top text-top middle bottom text-bottom	4.0	4.0	CSS1

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

		<i>length</i> %			
z-index	Thiết lập thứ tự của các thành phần.	auto <i>number</i>	6.0	4.0	CSS2

5.10. Thiết lập nền với Style (background)

Thuộc tính **Background** cho phép bạn thiết lập màu nền của một thành phần, thiết lập một bức ảnh là ảnh nền, lặp lại ảnh nền theo chiều dọc hay ngang,... trong một trang HTML. Một số thuộc tính cơ bản của background mô tả trong bảng dưới. (Kí hiệu NN: Netscape, IE: Internet Explorer, W3C: Web Standard).

Bảng 5.2. Một số thuộc tính cơ bản của background

Thuộc tính	Mô tả chức năng	Giá trị	NN	IE	W3C
background	Nhanh chóng thiết lập thuộc tính nền trong khai báo.	<i>background-color</i> <i>background-image</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-position</i>	6.0	4.0	CSS1
background-attachment	Thiết lập cho dù ảnh nền là cố định hay có thể kéo cuộn trong phạm vi trang.	scroll fixed	6.0	4.0	CSS1
background-color	Thiết lập màu nền.	<i>color-rgb</i> <i>color-hex</i> <i>color-name</i> transparent	4.0	4.0	CSS1
background-image	Thiết lập một ảnh là ảnh nền.	<i>url</i> none	4.0	4.0	CSS1
background-position	Thiết lập vị trí bắt đầu của ảnh nền.	top left top center top right center left center center center right bottom left bottom center	6.0	4.0	CSS1

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

		bottom x-% x-pos y-pos	right y-%			
background-repeat	Thiết lập cách lặp lại của ảnh nền.	repeat repeat-x repeat-y no-repeat		4.0	4.0	CSS1

Cú pháp thiết lập màu nền:

- Gõ *background*:
- Gõ *transparent* (không màu) hoặc *color*, với *color* là tên màu hoặc giá trị màu hệ 16
- Gõ *URL(image.gif)* để sử dụng hình ảnh cho nền, nếu muốn. Gõ *repeat* để sắp xếp ảnh theo phương ngang và dọc.
- Gõ *fixed* hay *scroll* để xác định ảnh nền có cuộn cùng với khung nhìn hay không. Ấn định vị trí ảnh nền bằng cách gõ *x*, *y*. Hoặc áp dụng các giá trị *top*, *center*, *bottom* cho *x* và *left*, *center*, *right* cho *y*.

Ví dụ 5.10.

Bạn thiết lập màu nền, ảnh nền cho toàn trang như sau:

```
body{ background: #FF0000}
body{ background: url("h3y.gif") no-repeat top}
body{ background: #00FF00 url("h3y.gif") no-repeat fixed top}
```

Đoạn mã sau xác định ảnh nền với vị trí ban đầu và được di chuyển cùng khung nhìn

```
body
{
background-image:url("h3y.gif");
background-attachment: scroll;

background-repeat:no-repeat;
background-position:0%0%
}
```

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

5.11. Thiết lập kích thước (Dimension) với Style

Bạn có thể điều chỉnh kích thước của một thành phần thông qua thuộc tính **dimension**. Các thành phần đó có thể là hình ảnh, mẫu biểu hay thậm chí cả khối văn bản. Một số thuộc tính căn bản của Dimension được mô tả trong bảng sau. (Kí hiệu NN: Netscape, IE: Internet Explorer, W3C: Web Standard).

Bảng 5.3. Một số thuộc tính căn bản của Dimension

Thuộc tính	Mô tả chức năng	Giá trị	NN	IE	W3C
height	Thiết lập chiều cao của thành phần	auto length %	6.0	4.0	CSS1
line-height	Thiết lập khoảng cách giữa các dòng	normal number length %	4.0	4.0	CSS1
max-height	Thiết lập chiều cao lớn nhất của thành phần	none length %			CSS2
max-width	Thiết lập chiều rộng lớn nhất của thành phần	none length %			CSS2
min-height	Thiết lập chiều cao nhỏ nhất của thành phần	length %			CSS2
min-width	Thiết lập chiều rộng nhỏ nhất của thành phần	length %			CSS2
width	Thiết lập chiều rộng của thành phần	auto % length	4.0	4.0	CSS1

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

Cú pháp:

- Gõ *width:w*, với *w* là độ rộng của thành phần.(biểu diễn bằng giá trị tuyệt đối hoặc là phần trăm của thành phần cha).
- *height:h*, với *h* là độ cao của thành phần (chỉ biểu diễn bằng giá trị tuyệt đối).

Ví dụ 5.11.

Các cách định dạng dòng trong một đoạn văn bản:

```
p{ line-height: 1.4}
p{ line-height: 14pt}
p{line-height: 140%}
```

Định dạng kích thước ảnh:

```
img{height: 230px ; width: 230px}
img{height: 200px ; width: 100%}
```

Chú ý:

- Không thể ấn định thuộc tính *WIDTH* cho hầu hết các thẻ văn bản (như *P* và *H1*). Muốn làm như vậy thì phải đưa chúng vào thẻ *<DIV>* đã được định dạng *WIDTH* rồi.

5.12. Định dạng văn bản với Style

Style cung cấp nhiều khả năng hơn thẻ HTML. Với Style bạn cũng có thể thay đổi cỡ chữ, dạng chữ, độ nghiêng chữ, chiều cao dòng, màu nền, màu chữ, khoảng trắng, các kiểu canh chỉnh văn bản, xác định xem có nên gạch dưới, gạch trên, hay gạch giữa chữ, chuyển đổi hoàn toàn sang chữ hoa, chữ thường hay chữ nhỏ,...

Tổ chức W3C khuyến nghị người sử dụng ngừng sử dụng thẻ FONT cùng với khoảng 20 thẻ và thuộc tính khác có chức năng định dạng văn bản, danh sách, bảng biểu trên trang Web. Thay vào đó nên hợp nhất các thông tin này trong một style.

Lựa chọn họ font chữ

Sử dụng thuộc tính **font-family** (*họ font chữ*) để xác định rõ nhiều font khi font đầu tiên không khả dụng trong máy tính tại phía người xem.

Cách sử dụng:

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

- Gõ *font-family: tênFont*, với *tênFont* là font chữ được chọn đầu tiên.
- Nếu muốn chọn thêm nhiều font chữ khác gõ tiếp *tênFont2* với *tênFont2* là font chữ thứ hai được chọn. Chú ý phải phân cách bằng dấu phẩy.

Ví dụ 5.12.

```
H1 {font-family:"tahoma","verdana"}  
P {font-family:"Times New Roman","verdana"}
```

Lựa chọn kiểu dáng chữ

Sử dụng thuộc tính *font-style* để áp dụng định dạng chữ in nghiêng, và *font-weight* để áp dụng định dạng chữ in đậm. Ngoài ra bạn có thể tạo chữ dạng gạch chân (*text-decoration*), chữ nhấp nháy hoặc viết chữ hoa, thường (*text-transform*). Phần dưới đây hướng dẫn bạn sử dụng một số kiểu chữ cơ bản này.

Chữ in nghiêng:

- Gõ *font-style*:
- Gõ *oblique* để có chữ xiên, hoặc *italic* để có chữ nghiêng. (Muốn loại bỏ kiểu chữ nghiêng gõ *normal*).

Ví dụ 5.13.

```
H1 {font-family:"tahoma","verdana"; font-style:normal}  
P {font-family:"Times New Roman","verdana"; font-style:italic}
```

Chữ đậm

- Gõ *font-weight*:
- Gõ *bold* để có chữ đậm, hoặc *lighter* để sử dụng giá trị tương quan với độ đậm nhạt hiện hành của chữ. (Muốn loại bỏ kiểu chữ đậm gõ *normal*).
- Ngoài ra bạn cũng có thể gõ bội của 100, trong khoảng 100 và 900, với 400 hiển thị chữ bình thường và 700 hiển thị chữ đậm.

Ví dụ 5.14.

```
H1 {font-family:"tahoma","verdana"; font-weight:normal}  
P {font-family:"Times New Roman","verdana"; font-weight:bold}
```

Chữ gạch và nhấp nháy

- Gõ *text-decoration*:

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

- Gỡ **underline** để gạch dưới chữ.
- Gỡ **overline** để gạch trên chữ.
- Gỡ **line-through** để gạch ngang chữ.
- Gỡ **none** để xóa bỏ để đường gạch.
- Gỡ **blink** để tạo chữ nhấp nháy.

Ví dụ 5.15.

```
H1{
text-decoration:underline;
text-decoration:blink;
font-family:"tahoma","verdana";
font-weight:normal
}
```

Chữ hoa, thường:

- Gỡ **text-trasform:**
- Gỡ **capitalize** để viết hoa chữ đầu tiên của mỗi từ.
- Gỡ **uppercase** để biến đổi mọi ký tự thành chữ hoa.
- Gỡ **lowercase** để biến đổi mọi ký tự thành chữ thường.
- Gỡ **none** để hiển thị văn bản bình thường.

Ví dụ 5.16.

```
H1{
text-trasform:uppercase ;
font-family:"tahoma","verdana";
font-weight:normal
}
P{
text-trasform:capitalize;
font-weight:normal
}
```

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

Lựa chọn cỡ chữ (size)

Sử dụng thuộc tính **font-size** để xác định rõ kích cỡ chữ chính xác tính theo đơn vị *point/pixel*, hay với từ mô tả, hoặc định cỡ chữ tương đối so với thành phần cha.

Cú pháp:

- Gõ **font-size**:
- Để có cỡ chữ tuyệt đối, gõ: *xx-small*, *x-small*, *small*, *medium*, *large*, *x-large*, hay *xx-large*.
- Để có cỡ chữ tương đối, gõ: *larger* hay *smaller*.
- Để có cỡ chữ chính xác, gõ: Ví dụ 12pt, 17px.
- Hoặc gõ giá trị phần trăm tương quan với style cha bất kỳ: ví dụ 150%.

Ví dụ 5.17.

```
H1 {font-family:"tahoma","verdana"; font-size:150%}  
P {font-family:"Times New Roman","verdana"; font-size:20pt}
```

Lựa chọn màu cho văn bản

Bạn có thể thay đổi màu của bất kỳ chuỗi văn bản nào kèm theo thẻ, cho dù đó là toàn bộ văn bản hay chỉ là một vài từ. Sử dụng thuộc tính *color* để ấn định màu cho chữ. Và thuộc tính *background* để ấn định màu nền cho văn bản.

Cú pháp:

- Gõ *color*:
- Gõ *tên Màu* là một trong 16 tên màu chỉ định sẵn (*red*, *blue*, *green* ...).
- Hoặc gõ *rgb(r, g, b)* với *r*, *g*, *b* là số nguyên từ 0 đến 255, chỉ rõ tỷ lệ pha màu (*xem lại phần HTML*).

Ví dụ 5.18.

```
H1 {font-family:"tahoma","verdana"; color:yellow}  
P {font-family:"Times New Roman","verdana"; background:green}
```


Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

Chi phối giãn cách cho văn bản

Bạn có thể tăng hay giảm khoảng cách giữa các dòng (*line-height*), giữa các ký tự (*letter-spacing*), giữa các từ (*word-spacing*). Ngoài ra còn tùy ý bổ sung một khoảng trắng lớn (*text-indent*), hay khoảng thụt vào trước một đoạn văn bản. Cách sử dụng như sau:

- Giãn cách dòng: *line-height: n*, với *n* là một số được nhân với cỡ chữ nhằm có được chiều cao dòng mong muốn. (Có thể thay *n* là giá trị %, hoặc giá trị tuyệt đối đơn vị pt).
- Giãn cách từ: *word-spacing: length*, với *length* là trị số tính bằng pixel(px), point(pt), em ...
- Giãn cách ký tự: *letter-spacing: length*, với *length* là trị số tính bằng pixel(px), point(pt), em ...
- Khoảng thụt dòng: *text-indent: length*, với *length* là khoảng thụt vào tính bằng giá trị tuyệt đối (âm/dương), hay tỷ lệ phần trăm.

Ví dụ 5.19.

```
P{
text-indent:8pt
line-height:15pt;
font-size:10pt/15pt;
font-family:"Times New Roman","verdana"; background:green
}
```

Canh chỉnh văn bản

Bạn có thể canh phải, canh trái, canh giữa, hay canh thẳng hàng hai bên tùy ý cho văn bản (*text-align*).

Cú pháp:

- Gõ *text-align*:
- Gõ *left* để canh trái văn bản.
- Gõ *right* để canh phải văn bản.
- Gõ *center* để canh giữa văn bản.
- Gõ *justify* để canh văn bản thẳng hàng hai bên.

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

Ví dụ 5.20.

```
P{
text-align:justify;
text-indent:8pt
line-height:1.5pt;
font-size:10pt/15pt;
font-family:"Times New Roman","verdana"; background:green
}
```

Định nghĩa style dành cho liên kết

Nếu không thích liên kết ở dạng gạch dưới, bạn có thể thay đổi màu nền và màu mặc định hay bất cứ thuộc tính chữ nào ở trên cho liên kết nhằm làm cho chúng đẹp mắt và nổi bật lên. Style dành cho liên kết sử dụng 4 hành động:

- *A.link*: Trạng thái ban đầu (chưa tác động).
- *A.visited*: Trạng thái sau khi tác động (sau khi click chuột).
- *A.hover*: Trạng thái khi di chuột qua.
- *A.active*: Trạng thái trong lúc tác động (đang click chuột).

Tương ứng với 4 hành động trên bạn có thể chèn thêm các kiểu dáng style cho văn bản đã nêu ở trên.

Ví dụ 5.21.

```
A.link{
font-size:1.5pt;
font-family:"Times New Roman","verdana";
background:green;
color:yellow
}
A.visited{
font-size:1.5pt;
font-family:"Times New Roman","verdana";
```

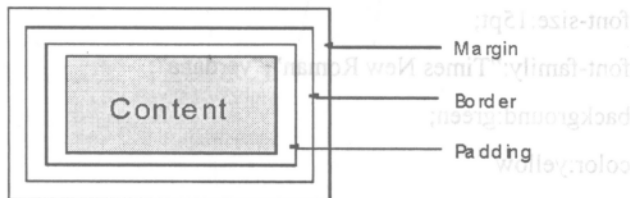
Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

```
background:green;
color:red
}
A:hover{
font-size:20pt;
font-family:"Times New Roman","verdana";
background:green;
color:yellow
}
A.active{
font-size:18pt;
font-family:"Times New Roman","verdana";
background:blue;
color:yellow
}
```

5.13. Làm việc với khung bao

Mỗi một thành phần trong trang được nằm trong một khung bao và đây là cách để đóng gói thành phần nhằm thiết lập vị trí bất kỳ cho nó. Chúng ta có thể ấn định đường viền khung bao (*border*), tạo khoảng đệm xung quanh khung bao (*padding*), hay canh chỉnh thành phần bên trong khung bao (*margin*).



Hình 5.8. Kiểu khung bao của mỗi thành phần.

Ấn định đường viền

Có thể tạo đường viền thành phần, sau đó tạo kiểu dáng đường viền, màu sắc.

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

Cú pháp:

- Gõ ***border:***
- Nếu muốn, hãy gõ *-top*, *-bottom*, *-left* hoặc *-right* không có khoảng trắng ngay sau *border* (ví dụ *border-top*) để giới hạn vị trí đường viền.
- Gõ *thin*, *medium*, *thick* hoặc giá trị tuyệt đối (như 5px) để xác định độ dày mảnh của đường viền. Mặc định là *medium*.
- Gõ *none*, *dotted*, *dashed*, *double*, *groove*, *ridge*, *inset* hoặc *outset* để xác định kiểu đường viền.
- Gõ *color*, với *color* là một trong 16 màu mặc định để thiết lập màu đường viền.

Ví dụ 5.22.

```
IMG
{
position:relative;
top:0; left:0;
border:thick double green
}
```

Tạo khoảng trống xung quanh thành phần

Khoảng đệm (*padding*) chỉ là không gian xung quanh nội dung thành phần nhưng nằm trong đường viền. Có thể thay đổi độ rộng hẹp của khoảng đệm ngoại trừ màu sắc và kết cấu.

Cú pháp:

- Gõ *padding* (có thể nối thêm *-top*, *-bottom*, *-left* hoặc *-right* vào sau *padding*) để giới hạn khoảng đệm ở một bên của đối tượng.
- Gõ *:x*, với *x* là không gian sẽ thêm vào, biểu diễn bằng đơn vị hoặc phần trăm của thành phần cha.

Ví dụ 5.23.

Đoạn mã sau để thêm khoảng đệm 10 pixel xung quanh văn bản trong đề mục

```
H1 {padding :10}
```

Và thêm khoảng đệm 15 pixel vào phần trên và dưới của ảnh.

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)

```
IMG{  
padding-top :15;  
padding-bottom :15  
}
```

Định lề xung quanh thành phần

Lề (*margin*) là khoảng không gian trong suốt giữa thành phần này và thành phần kia, thêm vào và nằm ngoài khoảng đệm hoặc đường viền xung quanh thành phần.

Cú pháp:

- Gõ *margin*: (có thể nối thêm *-top*, *-bottom*, *-left* hoặc *-right* vào sau *margin*) để giới hạn khoảng đệm ở một bên của đối tượng.
- Gõ *x*, với *x* là không gian sẽ thêm vào, biểu diễn bằng đơn vị hoặc phần trăm độ rộng của giá trị tương ứng trong thành phần cha.

Ví dụ 5.24.

Thêm khoảng đệm 10 pixel xung quanh văn bản, đồng thời định lề trái cách 30 pixel.

```
H1 {padding :10; margin-left: 30}
```

5.14. Kết chương

Trong chương này các bạn đã tìm hiểu về CSS hay Style, cách tạo và sử dụng Style trong Website của bạn. CSS cho phép người thiết kế có quyền chi phối hình thức trang Web ở mức cao hơn, tạo ra sự nhất quán và thống nhất trong bố cục trang Web. Trong chương sau các bạn sẽ được làm quen với XML - một ngôn ngữ tổng quát hiệu quả dùng để định nghĩa dữ liệu thông qua các thẻ chứ không hiển thị nội dung dữ liệu.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Nội dung của chương:

- Tìm hiểu về XML.
- Định dạng dữ liệu bằng XML.
- Một số ngôn ngữ tương đương với XML.
- Mối quan hệ giữa DTDs, Schemas với XML.
- Cơ sở của DOM và SAX.
- Tạo ra một tài liệu XSL đơn giản.
- Đôi nét về SOAP.

6.1. Giới thiệu XML

XML là gì?

XML, viết tắt của chữ *eXtensible Markup Language*, là một tập qui ước chuẩn về cách phân chia một tài liệu ra thành nhiều phần, đánh dấu từng phần theo đặc trưng của phần đó rồi ghép các phần đã đánh dấu lại để dễ dàng hiểu và nhận diện tài liệu hơn. Không giống như HTML, XML là một ngôn ngữ tổng quát dùng để định nghĩa dữ liệu thông qua các thẻ chú không hiển thị nội dung dữ liệu. Nếu như ở HTML chỉ sử dụng các thẻ có sẵn thì ở XML chúng ta có thể tự định nghĩa các thẻ.

Nguồn gốc của XML

XML được hình thành và phát triển khoảng trong mười năm gần đây. Cách đây hơn chục năm chúng ta có chuẩn SGML (*Standard Generalised Markup Language*) là một ngôn ngữ rất phức tạp, dùng để đặc tả tài liệu đa dạng và có cấu trúc tinh vi. Tuy SGML có nhiều công dụng nhưng lại có nhược điểm lớn là rắc rối, khó học, khó dùng và nhất là đòi hỏi các công cụ đi kèm theo khá đắt tiền.

Do đó, vào năm 1990 Tim Berners-Lee ở CERN đã sáng chế ra HTML, một phần nhỏ của SGML, nhưng dễ dùng, thích hợp với người sử dụng, có thể được viết bằng tay mà

Chương 6

Giới thiệu về Extensible Markup Language (XML)

không cần công cụ trợ giúp. Sau khi ra đời, sự thành công của HTML lúc đó và sau này đã vượt quá sức tưởng tượng của chính tác giả.

Đến khoảng năm 1995, với sự phát triển nhanh của công nghệ Web, người ta bắt đầu thấy những giới hạn của HTML, mặc dầu HTML đã được Netscape và Microsoft hỗ trợ tối đa. Nhiều chuyên gia có ý định quay trở lại SGML, tuy nhiên vẫn vấp phải trở ngại là SGML quá khó để sử dụng.

Mùa hè 1996, Jon Bosak ở Sun Microsystem khởi đầu nhóm cộng tác W3C SGML, bây giờ được gọi là nhóm XML. Mục đích của nhóm là đơn giản hoá SGML để nó dễ dùng như HTML, đồng thời làm cho nó mạnh và linh động, mềm dẻo hơn. Các nhà nghiên cứu mong muốn XML hội tụ các điều kiện sau:

- Đơn giản cho lập trình viên áp dụng.
- Không giới hạn ngôn ngữ (tiếng Anh hay các thứ tiếng khác).
- Dễ dàng cho các Search Engine (như Yahoo, AltaVista, Infoseek .v.v..) sắp xếp.

Bản đặc tả đầu tiên của XML được phát hành vào tháng 11 năm 1996. Tháng 7 năm 1997 Microsoft giới thiệu ứng dụng thật sự đầu tiên của XML: *Channel Definition Format* (CDF) để xuất bản các trang Web đến khách hàng. CDF là một phần của Internet Explorer 4.0. Sau đó Microsoft và Inso Corporation xuất bản XSL (*eXtensible Style Language*) là Style Sheet mô tả cách trình bày một trang XML.

Tháng 1 năm 1998, Microsoft đưa ra một chương trình miễn phí MSXSL để tạo ra một trang HTML từ một cặp trang XML và XSL. Sau này Internet Explorer 5 đã có thể hiển thị trực tiếp một trang XML (với điều kiện phải chỉ rõ địa chỉ tham chiếu trang XSL).

Tháng 2 năm 1998, W3C phê chuẩn chính thức công nhận version 1.0 của đặc tả XML.

XML trông ra sao?

Nếu như HTML là những trang Web có giao diện đẹp và sinh động thì chúng ta sẽ không có được điều đó ở XML nếu chúng ta không biết các kết hợp trang XML với các trang CSS hay XSL. Dưới đây là ví dụ về một trang XML đơn giản khi chưa kết hợp với các trang CSS hay XSL.

Ví dụ 6.1.

```
<?xml version="1.0" standalone="yes" encoding="UTF-8"?>
<!--Here comes the document element.-->
<document>
<!--The next element contains a heading.-->
<heading>
```

Chương 6

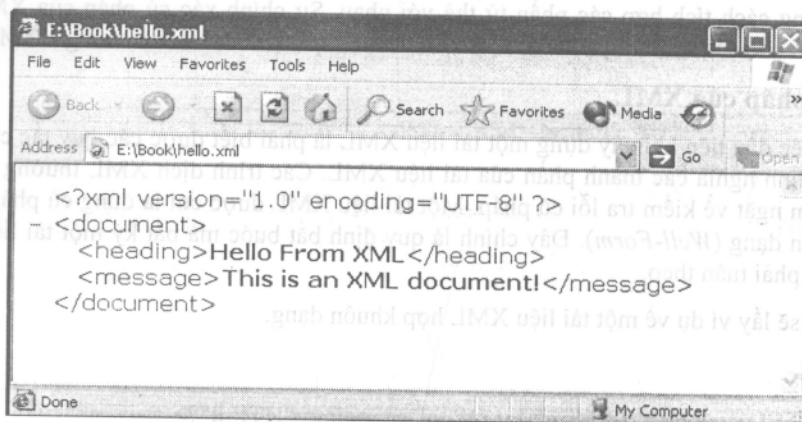
Giới thiệu về Extensible Markup Language (XML)

```

Hello From XML
</heading>
<!--The next element contains the actual message.-->
<message>
This is an XML document!
</message>
</document>

```

- Sử dụng trình soạn thảo bất kì như Notepad, Word, hay Dreamweaver MX 2004 và gõ đoạn mã trên vào.
- Sau đó lưu tài liệu này với tên file là *hello.xml*. Sử dụng trình duyệt Internet Explorer để xem kết quả như hình sau.



Hình 6.1. Trang hello.xml hiển thị trên trình duyệt.

- Kết quả hiển thị không giống như một trang HTML. Trình duyệt không hiển thị bất kỳ định dạng nào của XML ngoài việc đưa toàn bộ nội dung của file *hello.xml* ra màn hình. Đây là điều đương nhiên vì tệp XML trên mới chỉ là một khung chứa dữ liệu mà thôi. Để có thể hiển thị được như một trang HTML thì nó phải được sử dụng kết hợp với một số kỹ thuật khác (như CSS, XSL)

Tại sao phải là XML?

XML trở nên rất phổ biến là do có rất nhiều nguyên nhân, trong đó nguyên nhân quan trọng nhất là XML cho phép dễ dàng xử lý, chuyển tải và trao đổi dữ liệu giữa nhiều ứng dụng và tài liệu ở các định dạng khác nhau.

Các tính năng chính của XML bao gồm:

Chương 6

Giới thiệu về Extensible Markup Language (XML)

- *Dễ dàng trao đổi dữ liệu:* Trong XML dữ liệu và định dạng được lưu ở dạng text và có thể dễ dàng cấu hình cũng như thay đổi chúng bằng các trình soạn thảo thông thường. Dữ liệu và các thẻ trong XML cũng không mã hóa theo một thuật giải đòi hỏi bản quyền nào. Thêm vào đó, XML đã được chuẩn hóa nên mọi người đều có thể hiểu và sử dụng nó một cách dễ dàng.
- *Tùy biến ngôn ngữ định dạng:* XML có khả năng tạo ra các ngôn ngữ định dạng, đây là một trong những khả năng mạnh nhất của XML. Trên thực tế đã có rất nhiều ngôn ngữ định dạng chuyên dụng dựa trên XML phục vụ cho các lĩnh vực toán học, hóa học, tin học, đồ họa, tài chính ... Chúng ta sẽ tìm hiểu một số ngôn ngữ thông dụng này ở phần sau.
- *Dữ liệu mô tả:* Dữ liệu trong tài liệu XML tự mô tả nội dung và ý nghĩa của nó. Chỉ dựa trên các thẻ có thể hình dung ra nội dung và cách dữ liệu muốn thể hiện.
- *Dữ liệu có cấu trúc tích hợp:* XML cho phép tổ chức dữ liệu theo cấu trúc phân cấp bằng cách tích hợp các phần tử thẻ với nhau. Sự chính xác cú pháp của XML đảm bảo cho các trình duyệt hiển thị nhanh và hiệu quả hơn so với các trang HTML.

6.2. Cú pháp của XML

Việc đầu tiên khi xây dựng một tài liệu XML là phải biết được các quy tắc cũng như cú pháp định nghĩa các thành phần của tài liệu XML. Các trình dịch XML thường yêu cầu rất nghiêm ngặt về kiểm tra lỗi cú pháp. Một tài liệu XML được coi là đúng cú pháp khi nó hợp khuôn dạng (*Well-Form*). Đây chính là quy định bắt buộc mà bất kỳ một tài liệu XML nào cũng phải tuân theo.

Chúng ta sẽ lấy ví dụ về một tài liệu XML hợp khuôn dạng.

Ví dụ 6.2.

```
<?xml version="1.0" standalone="no" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="hello.css"?>
<!--Here comes the document element.-->
<document>
  <!--The next element contains a heading.-->
  <heading>
    Hello From XML
  </heading>
  <!--The next element contains the actual message.-->
  <message>
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
This is an XML document!  
</message>  
</document>
```

Để nắm chắc cách xây dựng một tài liệu XML hợp khuôn dạng, dưới đây chúng ta sẽ tìm hiểu một số quy tắc và đặc điểm của tài liệu XML.

6.2.1. Một số quy tắc chung trong XML

Trong HTML, chỉ có thể sử dụng các thẻ có sẵn (khoảng 100 thẻ), trình duyệt sẽ kiểm tra các thẻ này và hiển thị nội dung tài liệu bằng các quy định có sẵn. Trái lại, XML hoàn toàn tự do định nghĩa các thẻ sau đó xây dựng cho mình cách sử dụng chúng thông qua CSS hoặc XSL.

Thẻ XML

- Một phần tử XML bao gồm cặp thẻ mở đầu và kết thúc, bên trong là dữ liệu. Trình xử lý XML sẽ thông qua thẻ này để lấy dữ liệu.

<heading>Hello From XML</heading>

- Trong trường hợp phần dữ liệu muốn định dạng một số kí tự đặc biệt (như dấu > hay <) thì phải tham chiếu đến một số thực thể quy ước bắt đầu bằng kí tự “&” “kết thúc bởi kí tự “;” ; ví dụ như:

< - Thay thế cho “<”

> - Thay thế cho “>”

& - Thay thế cho “&”

" - Thay thế cho “ ” ”

' - Thay thế cho “ ’ ”

Chẳng hạn : *<message> a > b </message>* cho kết quả là *a>b*

- Nếu không có thẻ kết thúc thì thẻ đó phải được định nghĩa là thẻ rỗng. Tức là chỉ có một thẻ và thêm một kí tự / phía cuối: *<heading />*
- Trong mỗi thẻ đều có thể định nghĩa thêm các thuộc tính.

Tên thẻ

XML phân biệt kiểu chữ hoa và chữ thường do đó phải chú ý khi đặt tên thẻ. Tên thẻ bắt đầu có thể là chữ cái, dấu gạch dưới ‘_’, hay dấu hai chấm “:”. Các kí tự tiếp theo là chữ cái, số, dấu gạch dưới, dấu gạch nối, hoặc dấu “:”. Nhưng không có kí tự trống “ ”.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Ví dụ 6.3.

Tên đúng	Tên sai
<DOCUMENT>	<2005>
<document>	<Davita Number>
<Chapter15>	<.text>
<Section-19>	<*yay*>
<_text>	<EMPLOYEE(ID)>

Tạo một khai báo XML

Theo khuyến cáo của W3C, mỗi một tài liệu XML nên có một (và duy nhất) khai báo đặt ở dòng đầu tiên bằng cách sử dụng thẻ `<?xml ...?>`. Cụ thể như sau:

```
<?xml version = "1.0" standalone="no" encoding="UTF-8"?>
```

Khai báo này sẽ thông báo cho trình xử lý XML biết các thông tin sau:

- *version="1.0"*: Cho biết số hiệu phiên bản XML đang sử dụng. Hiện tại chỉ có hai phiên bản là 1.0 và 1.1 được dùng, tuy nhiên nên dùng 1.0 vì 1.1 vẫn chưa được nhiều trình duyệt hỗ trợ. Thuộc tính này là bắt buộc nếu sử dụng khai báo XML.
- *standalone="no"*: Thiết lập giá trị *yes* nếu tài liệu không liên quan đến bất kì tài liệu nào khác. Chọn *no* trong trường hợp ngược lại. Thuộc tính này là tùy chọn.
- *encoding="UTF-8"*: Kiểu mã hóa ngôn ngữ trong tài liệu. Mặc định chọn UTF-8 là kiểu mã hóa Unicode 8 bit cho hiển thị tiếng Việt hiện nay. Cũng có thể sử dụng kiểu mã như Unicode, UCS, hay ISO. Thuộc tính này là tùy chọn.

Bảng 6.1. Một số kiểu mã hóa

Tên	Kiểu mã hóa
US-ASCII	U.S. ASCII
UTF-8	Compressed Unicode
UTF-16	Compressed UCS
ISO-10646	UCS-2— Unicode
ISO-10646	UCS-4— UCS
ISO-2022-JP	Japanese
ISO-2022-CN	Chinese
ISO-8859-5	ASCII and Cyrillic

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Chỉ thị xử lý - thẻ `<?xml?>`

Các chỉ thị xử lý được dùng để chỉ dẫn cho bộ phân tích cách xử lý tài liệu trong quá trình phân tích sau này. Những chỉ thị này thường bắt đầu bằng `<?>` kết thúc bằng `?>` tương tự như phần khai báo. Tuy nhiên không nên sử dụng trùng lại từ khóa khai báo `<?xml?>`. Những chỉ thị xử lý này đặc trưng cho bộ phân tích xử lý tài liệu ở ngữ cảnh xác định.

Ví dụ 6.4.

Ví dụ chỉ thị yêu cầu bộ phân tích kết hợp dữ liệu của XML với bảng định kiểu CSS đã học ở chương trước là `<?xml-stylesheet?>`. Trình duyệt IE6 hay Netscape 6 sẽ hiểu được chỉ thị này:

```
<?xml-stylesheet type="text/css" href="hello.css"?>
```

Vấn đề này chúng ta sẽ quay lại nghiên cứu trong các phần sau.

Tạo dòng ghi chú

Tạo ghi chú trong XML tương tự như đối với HTML. Chỉ cần viết dòng ghi chú bao trong cặp: `<!-- -->`.

Tạo thẻ rỗng

Trong XML, thẻ rỗng chỉ có một thẻ, không phải là dạng thẻ bắt đầu và thẻ kết thúc. Tương tự như một số thẻ của HTML là `<img>`, `<li>`, `<hr>`, và `<br>` là thẻ rỗng, có nghĩa là nó không chứa nội dung gì.

Tạo thẻ rỗng bằng cách thêm kí tự `/` vào cuối tên thẻ (nên thêm một kí tự trống trước `/`).

Ví dụ 6.5.

Ví dụ nếu thẻ `<heading>` là một thẻ rỗng thì nó sẽ biểu diễn là: `<heading />`. Có thể bổ sung thuộc tính cho thẻ này như sau: `<heading text = "Hello From XML" />`

Tạo thẻ gốc (root element)

Để tài liệu XML hợp khuôn dạng thì nó phải có một thẻ gốc để chứa tất cả các thẻ và dữ liệu text bên trong. Trong ví dụ trên, thẻ gốc là *document*.

Thẻ gốc là một phần rất quan trọng của các tài liệu XML, đặc biệt là khi quan sát chúng từ quan điểm của trình xử lý XML, bởi vì tài liệu XML được phân tích bắt đầu từ thẻ gốc và từ đó tác động tiếp đến các thẻ bên trong. Có thể đặt tên bất kì cho thẻ gốc theo quy tắc đặt tên thẻ ở trên và mỗi một tài liệu chỉ có duy nhất một thẻ gốc.

Tạo thuộc tính

Các thuộc tính có thể xuất hiện bên trong các thẻ XML hoặc khai báo XML.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

- Sử dụng dạng **attributename = "value"** bên trong các thẻ.
- Không giống như HTML, giá trị chỉ định cho thuộc tính phải được bao trong nháy kép “ ” hoặc nháy đơn ‘ ’ (kể cả với số). Các trình xử lý XML sẽ nhận biết các giá trị đã được chỉ định cho thuộc tính.

Ví dụ 6.6.

```
<text language="en-US">
It was a dark and stormy night. A shot rang out!
</text>
```

- Tên thuộc tính phải được đặt theo cách đặt tên cho thẻ.
- Giá trị chỉ định cho thuộc tính có thể chứa các kí tự đặc biệt như dấu “ ” hay các kí tự đặc biệt ở trên. Trong trường hợp này cần dùng đến tham chiếu. Ví dụ chỉ định giá trị là He said: "No worries!" thì hãy sử dụng dấu nháy đơn:

```
<citation text='He said, "No worries."' />
```

Như vậy cả thuộc tính và thẻ đều lưu trữ dữ liệu. Thuộc tính không thể xác định được cấu trúc tài liệu khi có nhiều hơn một thông tin lưu trong một thẻ được định nghĩa.

Thuộc tính hữu dụng **xml:lang**

Thuộc tính *xml:lang* được sử dụng trong trường hợp muốn chỉ định kiểu ngôn ngữ của dữ liệu. Khi gán giá trị thuộc tính này là mã quốc gia theo chuẩn ISO 693 bao gồm hai kí tự tắt, *search engine* sẽ sử dụng như là dấu hiệu nhận dạng về ngôn ngữ.

Ví dụ 6.7.

Ví dụ chúng ta có thể định dạng dữ liệu bên trong thẻ `<text>` với thuộc tính *xml:lang* như sau:

```
<text xml:lang="en"> Hello </text>
```

Hoặc dùng mã quốc gia và mã vùng theo chuẩn ISO cho phân loại ngôn ngữ. Ví dụ phân biệt ngôn ngữ tiếng Anh British và American như sau:

```
<text xml:lang="en-GB"> Hello </text>
```

```
<text xml:lang="en-US"> Hi </text>
```

6.2.2. Tạo tài liệu XML hợp khuôn dạng (Well-Formed)

Tất cả các quy tắc khai báo phần tử, thẻ, thuộc tính, cách đặt tên thẻ... nhằm tạo ra chuẩn để xác định một tài liệu XML hợp khuôn dạng. Việc này rất cần thiết vì các trình xử

Chương 6

Giới thiệu về Extensible Markup Language (XML)

lý XML sẽ không nhận biết được nếu không có một quy định cho trước. Các quy tắc cơ bản cần tuân theo bao gồm:

1. Các khai báo XML cần được đặt ở dòng đầu tiên của tài liệu.
2. Mỗi một tài liệu chỉ có một phần tử gốc. Các phần tử khác, nếu có phải là con của phần tử gốc.
3. Mọi phần tử XML khác rỗng phải bao gồm đầy đủ thẻ đóng và thẻ mở.
4. Đóng các phần tử rỗng với chuỗi đóng ">"
5. Mọi phần tử trong tài liệu XML khác phần tử gốc đều phải nằm giữa cặp thẻ gốc.
6. Các phần tử lồng nhau phải có thẻ đóng, thẻ mở hợp vị trí. Trong HTML có thể đặt thẻ đóng và thẻ mở lồng nhau ở những vị trí tùy ý như ví dụ 6.8 sau:

Ví dụ 6.8.

```
<font size="7"> Hello from HTML!  
<p>  
</font>  
</p>
```

Trong XML nếu đặt vị trí các thẻ như ví dụ 6.8 trên thì tài liệu này sẽ coi là không hợp khuôn dạng, vì các thẻ `<p>` và thẻ `<font>` đặt lồng nhau giữa thẻ đóng và mở.

Ví dụ 6.9.

Ví dụ về tài liệu không hợp lệ:

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
<employee>
  <projects>
    <project>
      Information of Project 1
    <project>
    </project>
    Information of Project 2
  </project>
</projects>
</employee>
```

Cần sửa lại vị trí đặt thẻ *<project>* ở trên như sau để hợp lệ:

```
<employee>
  <projects>
    <project>
      Information of Project 1
    </project>
    <project>
      Information of Project 2
    </project>
  </projects>
</employee>
```

7. Sử dụng tên thuộc tính là duy nhất. Không được đặt trong cùng một thẻ các thuộc tính trùng tên.
8. Chỉ nên sử dụng năm tham chiếu thực thể định nghĩa trước trong XML đã nêu ở trên. Bằng cách này có thể sử dụng các kí tự đặc biệt trong giá trị thuộc tính hay dữ liệu.
9. Bao giá trị thuộc tính bằng cặp dấu nháy đơn hoặc kép.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

10. Chỉ nên sử dụng các ký tự < và & đối với thẻ mở và các thực thể tham chiếu. Bộ xử lý XML luôn coi < là ký tự bắt đầu của thẻ, còn & là ký tự để nhận dạng tham chiếu thực thể. Trong trường hợp buộc phải dùng các ký tự này có thể dùng tham chiếu thay thế tương ứng là < hay &
11. Có một điều cần lưu ý là nên lập kế hoạch xây dựng cấu trúc lưu trữ dữ liệu bằng XML để có thể sử dụng chúng trong các chương trình ứng dụng khác nhau.

Ví dụ 6.10.

Ví dụ một tài liệu hợp khuôn dạng hoàn chỉnh: tệp *employee.xml*

```
<?xml version = "1.0" encoding="UTF-8" standalone="yes"?>
<document>
  <employee>
    <name>
      <lastname>Kelly</lastname>
      <firstname>Grace</firstname>
    </name>
    <hiredate>October 15, 2005</hiredate>
    <projects>
      <project>
        <product>Printer</product>
        <id>111</id>
        <price>$111.00</price>
      </project>
      <project>
        <product>Laptop</product>
        <id>222</id>
        <price>$989.00</price>
      </project>
    </projects>
  </employee>
</employee>
```


Chương 6

Giới thiệu về Extensible Markup Language (XML)

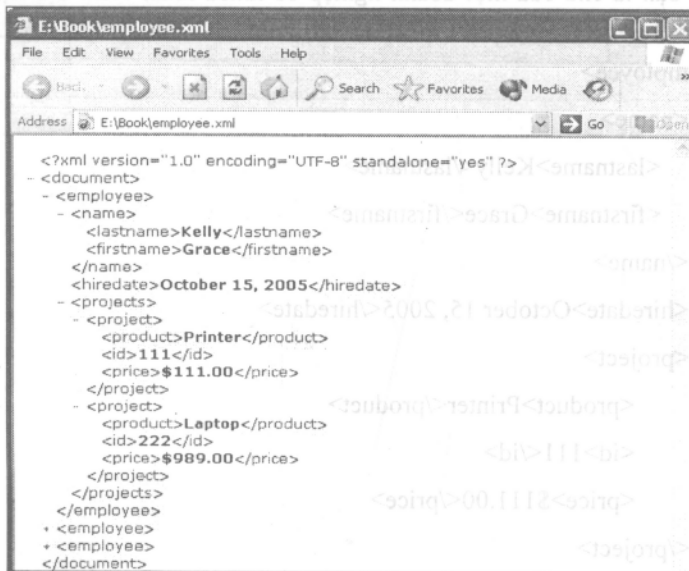
```
<name>
  <lastname>Grant</lastname>
  <firstname>Cary</firstname>
</name>
<hiredate>October 20, 2005</hiredate>
<projects>
  <project>
    <product>Desktop</product>
    <id>333</id>
    <price>$2995.00</price>
  </project>
  <project>
    <product>Scanner</product>
    <id>444</id>
    <price>$200.00</price>
  </project>
</projects>
</employee>
<employee>
  <name>
    <lastname>Gable</lastname>
    <firstname>Clark</firstname>
  </name>
  <hiredate>October 25, 2005</hiredate>
  <projects>
    <project>
      <product>Keyboard</product>
      <id>555</id>
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
<price>$129.00</price>
</project>
<project>
  <product>Mouse</product>
  <id>666</id>
  <price>$25.00</price>
</project>
</projects>
</employee>
</document>
```

- Kết quả hiển thị trên trình duyệt:



Hình 6.2. Trang employee.xml hiển thị trên trình duyệt.

6.3. Không gian tên (Namespaces) của XML

XML cho phép tự định nghĩa và đặt tên cho các thẻ. Nếu các thẻ chỉ sử dụng riêng cho một mục đích nào đó mà không cần kết hợp với các tài liệu XML khác thì điều này thực hiện rất đơn giản. Tuy nhiên khi ứng dụng phát triển đòi hỏi các định nghĩa thẻ XML kết hợp với nhau thì tên của thẻ là vấn đề cần phải chú ý tới. Chẳng hạn, nếu trong tài liệu có

Chương 6

Giới thiệu về Extensible Markup Language (XML)

định nghĩa thẻ mang tên <DATA> để chứa dữ liệu. Sau này chúng ta cần tích hợp tài liệu XML của mình với các ứng dụng khác như VML, MathML mà các ứng dụng này cũng khai báo một phần tử mang tên <DATA> thì khi đó trình phân tích nên hiểu thẻ <DATA> của ngôn ngữ nào cho đúng?

Do đó cần phải có sự phân biệt về phạm vi hay không gian tên (*namespace*) mà một thẻ có ý nghĩa. Khái niệm *Namespaces* cho phép tạo và sử dụng các thẻ cùng tên độc lập mà không gây xung đột. Chúng ta chỉ cần đặt thêm một *prefix* ở trước tên của thẻ và tên của thuộc tính là xong. Bạn đọc có thể xem thêm thông tin chi tiết tại địa chỉ website <http://www.w3.org/TR/REC-xml-names/>.

Tạo không gian tên

Dưới đây là một ví dụ tạo ra *Namespaces* và lý do tại sao chúng quan trọng đến vậy.

Ví dụ 6.11.

Giả sử bạn là chủ của một doanh nghiệp có nhiều nhân viên. Dữ liệu về các nhân viên được lưu trữ như sau trong một tệp .xml

```
<employee>
  <name>
    <lastname>Kelly</lastname>
    <firstname>Grace</firstname>
  </name>
  <hiredate>October 15, 2005</hiredate>
  <project>
    <product>Printer</product>
    <id>111</id>
    <price>$111.00</price>
  </project>
</employee>
```

Bây giờ bạn muốn có thêm những dòng chú giải vào dữ liệu của nhân viên với phần tử thẻ tên là <comment>. Vấn đề là dữ liệu của nhân viên này lưu trong phòng nhân sự, do đó không phải bất kỳ ai cũng có thể can thiệp để thay đổi được. Chúng ta có thể giải quyết vấn đề này bằng cách tự tạo ra thẻ <comment>, nhưng trước hết phải buộc dữ liệu này với *namespace* của nó để chỉ ra rằng dòng chú giải này không phải là một phần của phòng nhân sự đã thiết lập trước đó.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Để định nghĩa một *namespace* cho các thẻ, đưa thêm thuộc tính *xmlns:prefix* vào trước vị trí định sử dụng như một không gian tên, với *prefix* là không gian tên muốn chỉ định. Ở ví dụ này, định nghĩa một *namespace* mới là *hr* cho phòng nhân sự sẽ khai báo như sau:

```
<employee
xmlns:hr="http://www.superduperbigco.com/human_resources">
  <name>
    <lastname>Kelly</lastname>
    <firstname>Grace</firstname>
  </name>
  <hiredate>October 15, 2005</hiredate>
  <project>
    <product>Printer</product>
    <id>111</id>
    <price>$111.00</price>
  </project>
</employee>
```

Mỗi không gian cần mang một định danh URIs (*Uniform Resource Identifiers*) duy nhất. Ở đây thay cho URIs sử dụng một địa chỉ Web URLs (*Uniform Resource Locators*) vì hiện nay URI vẫn chưa phổ biến.

Sau khi tạo ra không gian tên *hr*, áp dụng tên này cho tất cả các thẻ nào trong tài liệu XML muốn duy nhất và độc quyền thuộc về *hr* (phòng nhân sự) bằng cách thêm cụm kí tự *hr:* vào trước tên thẻ hoặc tên thuộc tính.

Ví dụ 6.12.

```
<hr:employee
xmlns:hr="http://www.superduperbigco.com/human_resources">
  <hr:name>
    <hr:lastname>Kelly</hr:lastname>
    <hr:firstname>Grace</hr:firstname>
  </hr:name>
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
<hr:hiredate>October 15, 2005</hr:hiredate>
<hr:project>
  <hr:product>Printer</hr:product>
  <hr:id>111</hr:id>
  <hr:price>$111.00</hr:price>
</hr:project>
</hr:employee>
```

Theo cách trên thì dữ liệu bây giờ được chỉ định thuộc về *hr* và đã tránh được xung đột với các thẻ trùng tên. Việc còn lại là định nghĩa thêm một không gian tên mới là *boss* để thêm vào lời chú giải trong thẻ *<comment>*.

```
<hr:employee
  xmlns:hr="http://www.superduperbigco.com/human_resources"
  xmlns:boss="http://www.superduperbigco.com/big_boss">
  <hr:name>
    <hr:lastname>Kelly</hr:lastname>
    <hr:firstname>Grace</hr:firstname>
  </hr:name>
  <hr:hiredate>October 15, 2005</hr:hiredate>
  <boss:comment>Needs much supervision.</boss:comment>
  <hr:project>
    <hr:product>Printer</hr:product>
    <hr:id>111</hr:id>
    <hr:price>$111.00</hr:price>
  </hr:project>
</hr:employee>
```

Cũng có thể thêm vào thuộc tính trong thẻ *comment* bằng cách thêm *boss:* vào trước tên thuộc tính. Ví dụ:

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
<boss:comment boss:date="10/15/2006">
Needs much supervision.
</boss:comment>
```

Không gian tên cục bộ (local namespace)

Thuộc tính `xmlns:prefix` có thể được đặt ở bất kỳ thẻ nào chứ không chỉ giới hạn ở phần tử gốc. Ví dụ dưới đây di chuyển định nghĩa không gian tên `boss` đến nơi nó bắt đầu được sử dụng.

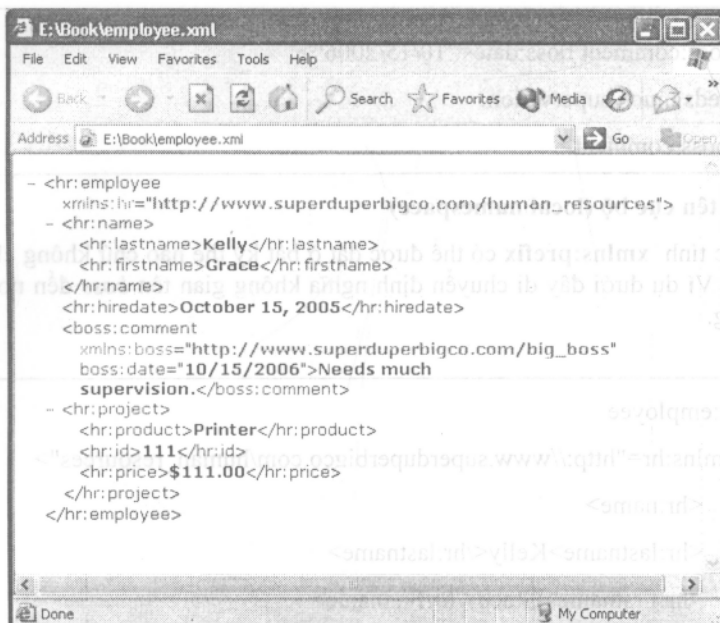
Ví dụ 6.13.

```
<hr:employee
  xmlns:hr="http://www.superduperbigco.com/human_resources">
  <hr:name>
    <hr:lastname>Kelly</hr:lastname>
    <hr:firstname>Grace</hr:firstname>
  </hr:name>
  <hr:hiredate>October 15, 2005</hr:hiredate>
  <boss:comment
    xmlns:boss="http://www.superduperbigco.com/big_boss"
    boss:date="10/15/2006">
    Needs much supervision.
  </boss:comment>
  <hr:project>
    <hr:product>Printer</hr:product>
    <hr:id>111</hr:id>
    <hr:price>$111.00</hr:price>
  </hr:project>
</hr:employee>
```

Kết quả hiển thị trên trình duyệt:

Chương 6

Giới thiệu về Extensible Markup Language (XML)



Hình 6.3. Không gian tên cục bộ.

Không gian tên mặc định

Các tài liệu XHTML (sẽ đề cập ở phần sau) và HTML thông thường không hiển thị được các biểu thức toán học. Nếu muốn sử dụng các công thức toán học cần phải dùng một ngôn ngữ bắt nguồn từ XML là MathML (sẽ đề cập ở phần sau) kết hợp với XHTML. Tuy nhiên khi kết hợp hai loại ngôn ngữ này sẽ dẫn đến trường hợp trùng tên thẻ. Do đó trước tiên cần phải định một không gian tên cho tập các thẻ XHTML. Đây là không gian tên mặc định có tác dụng cho toàn bộ thẻ trong tài liệu XML và nó chỉ sử dụng thuộc tính **xmlns** theo sau là định danh tài nguyên chứ không có tiền tố **prefix**.

Không gian tên mặc định cho XHTML:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```

Tiếp theo là tạo một không gian tên (ở đây chọn là *m*) cho MathML:

```
xmlns:m="http://www.w3.org/1998/Math/MathML"
```

Theo ví dụ trên, cũng có thể hiểu toàn bộ dữ liệu là của phòng nhân sự do đó có thể tạo ra không gian tên mặc định thay thế cho *hr*. Chúng ta chỉ còn phải định nghĩa *boss* mà thôi:

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Ví dụ 6.14.

```
<employee
  xmlns="http://www.superduperbigco.com/human_resources">
  <name>
    <lastname>Kelly</lastname>
    <firstname>Grace</firstname>
  </name>
  <hiredate>October 15, 2005</hiredate>
  <boss:comment
    xmlns:boss="http://www.superduperbigco.com/big_boss"
    boss:date="10/15/2006">
    Needs much supervision.
  </boss:comment>
  <project>
    <product>Printer</product>
    <id>111</id>
    <price>$111.00</price>
  </project>
</employee>
```

6.4. Tài liệu XML hợp lệ

Như đã trình bày ở trên, các thẻ XML không có ý nghĩa và có thể đặt tên bất kì cho các thẻ, vậy làm thế nào để biết loại thẻ nào hợp lệ trong tài liệu? Để làm được điều này sử dụng một trong hai cách sau:

- *Documents Type Definitions (DTDs)*: Định nghĩa kiểu tài liệu.
- *Schemas*: Giản đồ tương tự như DTD nhưng cho phép định nghĩa nhiều kiểu dữ liệu hơn.

Cả hai đều chỉ định các phần tử nào được phép trong tài liệu XML và cho phép một tài liệu XML hợp khuôn dạng trở thành một tài liệu XML hợp lệ.

6.4.1. Documents Type Definitions (DTDs)

Một tài liệu XML được xem là hợp lệ và có giá trị khi toàn bộ phần tử trong tài liệu được định nghĩa kiểu mà nó chứa. Việc định nghĩa kiểu dữ liệu cho các phần tử thẻ còn gọi

Chương 6

Giới thiệu về Extensible Markup Language (XML)

là định nghĩa kiểu tài liệu (DTD). DTD là một file text định nghĩa cấu trúc của một tài liệu XML, nhưng bản thân nó không phải là XML – nó có cú pháp riêng.

Tạo khai báo và định nghĩa kiểu dữ liệu DTD

Dùng thẻ khai báo **<!DOCTYPE>** để định nghĩa kiểu dữ liệu cho các phần tử XML. Có nhiều mẫu khai báo được sử dụng:

- **<!DOCTYPE rootname [DTD]>**
- **<!DOCTYPE rootname SYSTEM URL>**
- **<!DOCTYPE rootname SYSTEM URL [DTD]>**
- **<!DOCTYPE rootname PUBLIC identifier URL>**
- **<!DOCTYPE rootname PUBLIC identifier URL [DTD]>**

Phần khai báo DTD thường được đặt ở phần khởi đầu của tài liệu XML như cấu trúc sau.

```
<?xml version = "1.0" standalone="yes"?>
<!DOCTYPE document [
    ...
    <!-- DTD goes here! -->
    ...
]>
<document>
    <employee>
        ....
    </employee>
    ....
</document>
```

Ví dụ 6.15.

Ví dụ sau đây là một kiểu DTD.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
<?xml version = "1.0" standalone="yes"?>
<!DOCTYPE document [
<!ELEMENT document (employee)*>
<!ELEMENT employee (name, hiredate, projects)>
<!ELEMENT name (lastname, firstname)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT hiredate (#PCDATA)>
<!ELEMENT projects (project)*>
<!ELEMENT project (product,id,price)>
<!ELEMENT product (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
<document>
  <employee>
    .....
  </employee>
  ....
</document>
```

Khai báo <!ELEMENT>

Mỗi phần tử được bắt đầu định nghĩa bằng thẻ khai báo <!ELEMENT>. Cú pháp như sau:

<!ELEMENT name content_model>

Trong đó:

- *name*: Là tên của phần tử muốn định nghĩa.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

- *content_model*: Có thể đặt là EMPTY (không có giá trị) hay ANY (nếu cho phép chứa dữ liệu bất kì) hoặc trộn lẫn cả hai nội dung (bao gồm dữ liệu hoặc các phần tử con khác).

Ví dụ 6.16.

Ví dụ về khai báo thẻ `<!ELEMENT>`

```
<?xml version = "1.0" standalone="yes"?>
<!DOCTYPE document [
<!ELEMENT document (employee)*>
]>
<document>
  <employee>
    ...
  </employee>
</document>
```

Ở đây, thẻ `<!ELEMENT>` không chỉ có nhiệm vụ khai báo phần tử `<document>` mà còn chỉ ra rằng trong phần tử `<document>` có thể chứa các phần tử `<employee>`. Dấu * cho biết `document` có thể chứa nhiều phần tử `employee`. Tuy nhiên tài liệu trên vẫn chưa hợp lệ vì phần tử `employee` chưa được khai báo có chứa những phần tử khác nữa. Ví dụ dưới đây là tài liệu đã coi là hợp lệ.

Ví dụ 6.17.

```
<?xml version = "1.0" standalone="yes"?>
<!DOCTYPE document [
<!ELEMENT document (employee)*>
<!ELEMENT employee (name, hiredate, projects)>
<!ELEMENT name (lastname, firstname)>
<document>
  <employee>
    <name>
      <lastname>Kelly</lastname>
      <firstname>Grace</firstname>
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
</name>
<hiredate>October 15, 2005</hiredate>
<projects>
  <project>
    <product>Printer</product>
    <id>111</id>
    <price>$111.00</price>
  </project>
  <project>
    <product>Laptop</product>
    <id>222</id>
    <price>$989.00</price>
  </project>
</projects>
</employee>
</document>
```

Để định nghĩa rằng trong phần tử *employee* có chứa các phần tử như *name*, *hiredate* hay *projects* chúng ta phải sử dụng dấu phẩy để phân cách chúng:

```
<!ELEMENT employee (name, hiredate, projects)>
```

Dữ liệu văn bản : #PCDATA

Trong trường hợp một phần tử chỉ chứa dữ liệu văn bản thì phải sử dụng #PCDATA để thông báo cho trình phân tích biết dữ liệu thể hiện của các phần tử thể trong tài liệu XML. Chú ý là #PCDATA chỉ đơn thuần biểu diễn dữ liệu thô.

Ví dụ dưới đây chỉ định cho một số phần tử chứa kiểu dữ liệu văn bản.

Ví dụ 6.18.

```
<?xml version = "1.0" standalone="yes"?>
<!DOCTYPE document [
  <!ELEMENT document (employee)*>
  <!ELEMENT employee (name, hiredate, projects)>
  <!ELEMENT name (lastname, firstname)>
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT hiredate (#PCDATA)>
<!ELEMENT projects (project)*>
<!ELEMENT project (product,id,price)>
<!ELEMENT product (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
<document>
  <employee>
    ...
  </employee>
</document>
```

Làm việc với nhiều phần tử con

Khi muốn khai báo một phần tử có khả năng chứa nhiều phần tử con khác, thông thường sử dụng dấu phẩy để phân cách giữa các phần tử như trên. Hoặc có thể sử dụng một số kí tự đặc biệt như *, + để định nghĩa. Trong đó:

- x+ – Có nghĩa là x có thể xuất hiện ≥ 1 lần
- x* – Có nghĩa là x có thể xuất hiện ≥ 0 lần
- x? – Có nghĩa là x có thể xuất hiện 1 lần hoặc không.
- x,y – Có nghĩa là x rồi đến y
- x|y – x hoặc y xuất hiện nhưng không đồng thời

Trình tự đặt các phần tử sẽ quyết định đến trình tự xuất hiện của chúng trong phần tử cha. Do đó cần phải chú ý đến điều này khi tạo một DTD. Ngoài ra không cần quan tâm đến khoảng trắng trong khai báo.

Ví dụ 6.19.

Ví dụ có thể áp dụng đồng thời nhiều kiểu để chỉ ra rằng *employee* chứa ≥ 1 phần tử *name*, có thể có 1 hoặc 0 phần tử *hiredate* và chứa ≥ 0 phần tử *projects* như sau :

Chương 6

Giới thiệu về Extensible Markup Language (XML)

<!ELEMENT employee (name+, hiredate?, projects*)>

Định nghĩa cho các phần tử rỗng

Khi một phần tử không chứa kiểu dữ liệu nào thì nó phải được định nghĩa là phần tử rỗng. Sử dụng từ khóa *EMPTY* để làm điều này.

Ví dụ 6.20.

Chỉ định phần tử *intern* là phần tử rỗng :

<!ELEMENT intern EMPTY>

Tạo chú thích trong DTD

Việc tạo chú thích trong DTD cũng giống như cách làm đối với XML. Chỉ cần bao dòng chú thích trong cặp **<!--** và **-->**

Một ví dụ về DTD hoàn chỉnh

Ví dụ 6.21.

Ví dụ sau đây là mô hình dữ liệu về nhân viên với các phần tử mô tả *<employee>* (nhân viên), *<name>* (tên), *<hiredate>* (ngày thuê)...

Trang XML cho mô hình này được khai báo và định nghĩa như sau:

```
<?xml version = "1.0" encoding="UTF-8" standalone="yes"?>
<!-- Tạo khai báo DTD -->
<!DOCTYPE document [
  <!--Tạo phần tử element đầu tiên -->
  <!ELEMENT document (employee)*>
  <!--Mỗi employee cần một tên, ngày thuê, và các dự án -->
  <!ELEMENT employee (name, hiredate, projects)>
  <!-- Có thể đảo ngược thứ tự này ? -->
  <!ELEMENT name (lastname, firstname)>
  <!ELEMENT lastname (#PCDATA)>
  <!ELEMENT firstname (#PCDATA)>
  <!-- Sử dụng định dạng ngày tháng chuẩn cho phần tử hiredate -->
  <!ELEMENT hiredate (#PCDATA)>
```

Chương 6
Giới thiệu về Extensible Markup Language (XML)

```
<!ELEMENT projects (project)*>
<!ELEMENT project (product, id, price)>
<!--Tên sản phẩm nên đặt dạng 8547382 -->
<!ELEMENT product (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!--Phần tử rỗng -->
<!ELEMENT intern EMPTY>
]>
```

```
<document>
  <employee>
    <intern/>
    <name>
      <lastname>Kelly</lastname>
      <firstname>Grace</firstname>
    </name>
    <hiredate>October 15, 2005</hiredate>
    <projects>
      <project>
        <product>Printer</product>
        <id>111</id>
        <price>$111.00</price>
      </project>
      <project>
        <product>Laptop</product>
        <id>222</id>
        <price>$989.00</price>
```

Chương 6

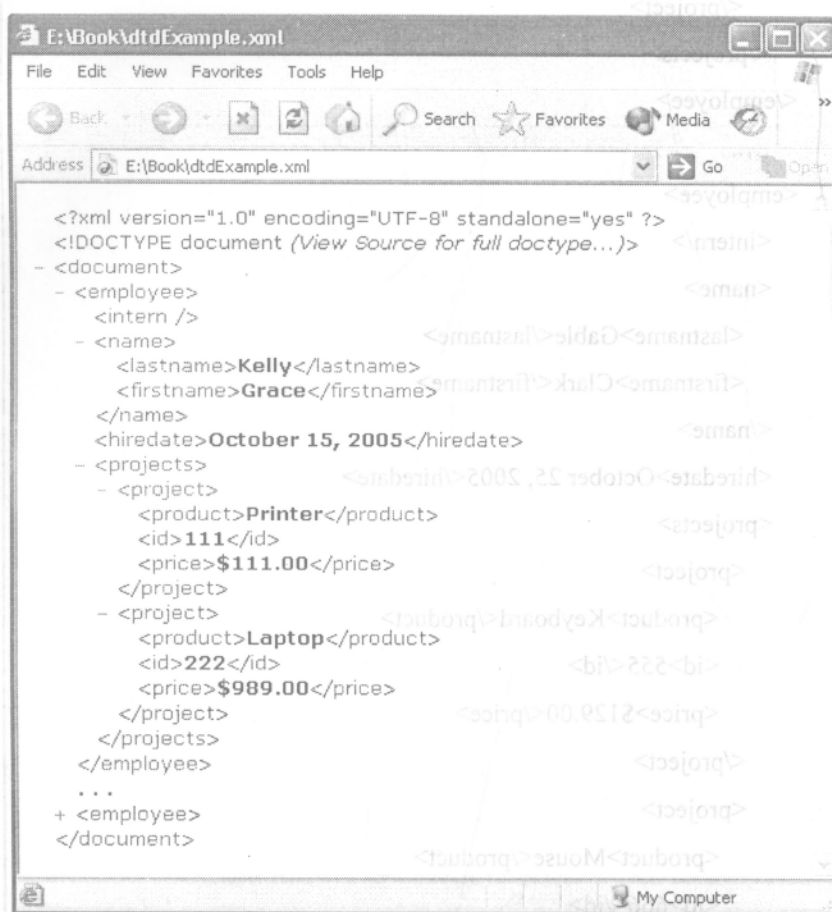
Giới thiệu về Extensible Markup Language (XML)

```
</project>
</projects>
</employee>
.....
<employee>
  <intern/>
  <name>
    <lastname>Gable</lastname>
    <firstname>Clark</firstname>
  </name>
  <hiredate>October 25, 2005</hiredate>
  <projects>
    <project>
      <product>Keyboard</product>
      <id>555</id>
      <price>$129.00</price>
    </project>
    <project>
      <product>Mouse</product>
      <id>666</id>
      <price>$25.00</price>
    </project>
  </projects>
</employee>
</document>
```


Chương 6

Giới thiệu về Extensible Markup Language (XML)

Kết quả hiển thị trên trình duyệt:



Hình 6.4. Ví dụ hoàn chỉnh về DTD.

Định nghĩa DTD ngoài (external DTD)

Những DTD đã được khai báo và tạo ra ở trên đều tham chiếu đến các phần tử định nghĩa chung và chứa trong cùng một tài liệu XML. Chúng được gọi là định nghĩa DTD trong (*Internal DTD*).

Cũng có thể định nghĩa DTD cho phần tử thẻ dựa trên các tham chiếu đã được định nghĩa sẵn từ các tài liệu XML trước đó (thường được lưu dạng file .dtd). Sử dụng định nghĩa DTD tham chiếu ngoài khiến cho các ứng dụng dễ dàng chia sẻ và dùng chung với các ứng dụng khác. Trên thực tế đây là cách mà hầu hết các ứng dụng XML đều hỗ trợ.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Có hai kiểu tham chiếu ngoài:

- Tham chiếu ngoài riêng (*private DTDs*): Được dùng cho một nhóm người và mang tính cá nhân – chúng không được dùng cho mục đích phân phối chung.
- Tham chiếu ngoài chung (*public DTDs*): Dùng cho mục đích phân phối.

Tham chiếu ngoài riêng

Sử dụng từ khóa *SYSTEM* trong chỉ thị khai báo `<!DOCTYPE>` như sau:

`<!DOCTYPE rootname SYSTEM URL>`

Ví dụ 6.22.

```
<?xml version = "1.0" standalone="no"?>
<!DOCTYPE document SYSTEM "order.dtd">
<document>
....
</document>
```

Nội dung của file *order.dtd* như sau.

```
<!ELEMENT document (employee)*>
<!ELEMENT employee (name, hiredate, projects)>
<!ELEMENT name (lastname, firstname)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT hiredate (#PCDATA)>
<!ELEMENT projects (project)*>
<!ELEMENT project (product,id,price)>
<!ELEMENT product (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT price (#PCDATA) >
<!ELEMENT intern EMPTY>
```

Chú ý phải đặt file *order.dtd* trong thư mục hiện hành cùng cấp với file tài liệu XML.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Ngoài ra có một cách khác là sử dụng URL để truy xuất theo địa chỉ tuyệt đối Internet:

```
<?xml version = "1.0" standalone="no"?>
<!DOCTYPE document SYSTEM
"http://www.xmlpowercorp.com/dtds/order.dtd">
<document>
....
</document>
```

Tham chiếu ngoại chung

Sử dụng từ khóa *PUBLIC* trong chỉ thị khai báo <!DOCTYPE> đồng thời tạo ra định danh hình thức như sau:

<!DOCTYPE rootname PUBLIC identifier URL>

Ví dụ 6.23.

```
<?xml version = "1.0" standalone="no"?>
<!DOCTYPE document PUBLIC "-//DTDS4ALL//Custom DTD Version
1.0//EN"
"http://www.xmlpowercorp.com/dtds/order.dtd">
<document>
...
</document>
```

Sử dụng chung tham chiếu nội và tham chiếu ngoại

Có thể sử dụng cả hai loại này trong cùng một tài liệu XML vào cùng một thời điểm. Lúc này khai báo <!DOCTYPE> được sử dụng ở dạng:

<!DOCTYPE rootname SYSTEM URL [DTD]>

hoặc

<!DOCTYPE rootname PUBLIC identifier URL [DTD]>

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Ở đây [DTD] được sử dụng cho các định nghĩa kiểu DTD trong.

Ví dụ 6.24.

```
<?xml version = "1.0" standalone="no"?>
<!DOCTYPE document PUBLIC "-//DTDS4ALL//Custom DTD Version
1.0//EN"
"http://www.xmlpowercorp.com/dttds/order.dtd" [
<!ELEMENT price (#PCDATA)>
]>
<document>
....
</document>
```

Theo cách trên định nghĩa DTD trong sẽ được ưu tiên hơn các định nghĩa DTD ngoài.

6.4.2. Schema

Sử dụng DTD có hai nhược điểm là:

- Chúng không phải là XML.
- Không thể chỉ định kiểu dữ liệu như *integer*, *data* ... cho mỗi phần tử.

Do đó, Microsoft đã đưa *Schema* vào W3C, là những XML tương đương với DTD. *Schema* cho phép chỉ định các kiểu dữ liệu khác nhau như *boolean*, *string*.... Dưới đây là một ví dụ dùng Schema.

Ví dụ 6.25.

```
<Schema ID="document">
<Element name="lastname" type="string" />
<Element name="firstname" type="string" />
<Element name="hiredate" type="date" />
<Element name="product" type="string" />
<Element name="id" type="integer" />
<Element name="price" type="string" />
</Schema>
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Tuy nhiên *Schema* chưa phải là một chuẩn được chấp nhận rộng rãi như DTD mà nó đang ở giai đoạn xem xét, do đó chúng ta sẽ không đi sâu vào việc tìm hiểu *Schema* nữa.

6.5. Ứng dụng của XML

XML có thể tự tạo ra các ngôn ngữ con khác (ngôn ngữ định dạng dựa trên XML – based Language). Khi nói về ứng dụng XML, ta sẽ không nhằm đến một chương trình cụ thể sử dụng XML nào. Ví dụ như Microsoft và Netscape đều dùng XML để cấu hình cho trình duyệt, nhưng ứng dụng XML này chỉ được xem là loại ứng dụng XUL (*Extensible User Interface Language*).

Các ứng dụng XML mang ý nghĩa thực tiễn là một tập các thẻ hay tập con XML hoạt động trong riêng một lĩnh vực nào đó như toán học, hóa học, tin học, đồ họa, sản xuất, kinh doanh,... Mỗi lĩnh vực hình thành tương ứng một ngôn ngữ dựa trên đặc tả XML.

Bây giờ chúng ta sẽ tìm hiểu một số ngôn ngữ thông dụng kể trên:

MathML - Ngôn ngữ định dạng hóa toán học

MathML ra đời nhằm đưa đến khả năng hiển thị và định dạng được các biểu thức toán học trong các tài liệu về Web. Hầu như cho đến nay chưa có trình duyệt nào có thể hiển thị được các ký hiệu của ngôn ngữ toán học hoàn chỉnh. Dưới đây là một ví dụ hiển thị ra biểu thức: $4X^2 - 5X + 6 = 0$

Ví dụ 6.26.

```
<?xml version="1.0"?>
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mrow>
      <mn>4</mn>
      <mo>&InvisibleTimes;</mo>
      <msup>
        <mi>x</mi>
        <mn>2</mn>
      </msup>
      <mo>-</mo>
      <mrow>
        <mn>5</mn>
        <mo>&InvisibleTimes;</mo>
        <mi>x</mi>
      </mrow>
    </mrow>
  </math>
```

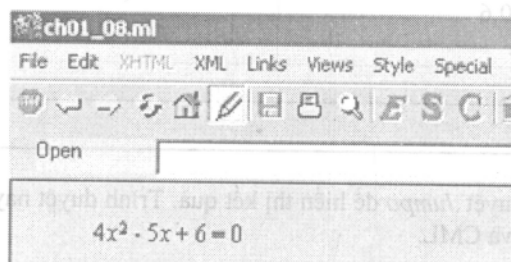
Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
</mrow>
<mo>+</mo>
<mn>6</mn>
</mrow>
<mo>=</mo>
<mn>0</mn>
</mrow>
</math>
```

Hiện nay chỉ có một số trình duyệt chuyên dụng có khả năng hiển thị ngôn ngữ định dạng MathML. Một trong những trình duyệt đó là như Amaya Browser. Bạn đọc có thể *download* trình duyệt Amaya về sử dụng tại địa chỉ: <http://www.w3.org/Amaya>

Kết quả hiển thị trên trình duyệt:



Hình 6.5. Tài liệu MathML hiển thị trong trình duyệt Amaya.

CML – Ngôn ngữ định dạng hóa học

Peter Murray – Rust đã phát triển **CML** như là một trong những ứng dụng XML sớm nhất. CML mô tả và hiển thị nội dung của cấu trúc một phân tử hóa học. CML còn cung cấp một danh sách rất nhiều công thức hóa học của phân tử được định nghĩa sẵn.

Dưới đây là một ví dụ về CML tạo ra phân tử Amoniac.

Ví dụ 6.27.

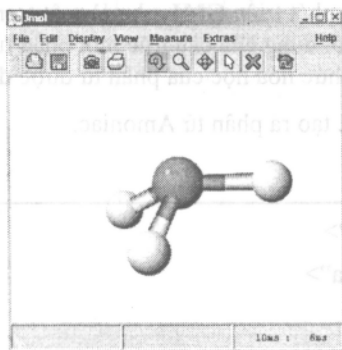
```
<?xml version="1.0"?>
<molecule id="amonia">
  <atomArray>
    <stringArray builtin="id">
      Nitrogen Hydrogen1 Hydrogen2 Hydrogen3
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
</stringArray>
<stringArray builtin="elementType">
N H H H
</stringArray>
<floatArray builtin="x3">
-0.7 -1.3 -1.2 -0.7
</floatArray>
<floatArray builtin="y3">
-0.0 0.2 0.8 -0.9
</floatArray>
<floatArray builtin="z3">
0.0 -0.9 0.6 0.6
</floatArray>
</atomArray>
</molecule>
```

- Sử dụng trình duyệt *Jumbo* để hiển thị kết quả. Trình duyệt này được thiết kế để làm việc trên XML và CML.
- Bạn đọc có thể download trình duyệt Jumbo về sử dụng tại địa chỉ: <http://www.xml-cml.org/jumbo.html>
- Kết quả hiển thị trên trình duyệt:



Hình 6.6. Tài liệu CML hiển thị trong trình duyệt Jumbo.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

SMIL - Ngôn ngữ tích hợp multimedia đồng bộ

Là giải pháp nhằm khắc phục vấn đề xử lý các file *multimedia* trong trình duyệt. **SMIL** cho phép trình diễn đồng thời nhiều file multimedia như âm thanh, hình ảnh, phim ... và biến trình duyệt thành một tivi thu nhỏ trên máy tính.

Ý tưởng ở đây là *SMIL* cho phép chỉ định các file *multimedia* muốn trình diễn trong khi bản thân tài liệu SMIL không mang theo một nội dung của file multimedia nào cả.

Hiện nay IE6 vẫn chưa cài đặt và hỗ trợ nhiều SMIL. Mặc dù vậy có thể sử dụng SMIL thông qua Applet của Java. Bạn đọc có thể tìm hiểu thêm thông tin về SMIL chạy trên Applet của Java ở địa chỉ www.empirenet.com

Dưới đây là ví dụ về trang SMIL dùng trình diễn ảnh, chơi nhạc nền, hiển thị chữ....

Ví dụ 6.28.

```
<?xml version="1.0"?>
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 1.0//EN"
"http://www.w3.org/TR/REC-smil/SMIL10.dtd">
<smil>
  <body>
    <par id="show">
      <audio src="river.wav" region="background__audio"
type="audio/x-wav" dur="20s"/>
      <seq id="slides">
        <par id="slide01">
          
          <text src="welcome.txt" type="text/plain" dur="5s"/>
        </par>
      </seq>
    </par>
  </body>
</smil>
```

VML – Ngôn ngữ định dạng vector

VML là ngôn ngữ định dạng vector cho phép vẽ trực tiếp ảnh vector trong trình duyệt chỉ bằng khai báo các thẻ. Trình duyệt IE hỗ trợ rất mạnh VML. Dưới đây là nội dung một tài liệu VML.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Ví dụ 6.29.

Trang *vml.html*

```
<html xmlns:v="urn:schemas-microsoft-com:vml">
<head>
<title>
Using VML
</title>
<style>
v\:.*{behavior:url(#default#VML);}
</style>
</head>
<body>
<center>
<h1> Using VML </h1>
</center>
<p>
<v:oval STYLE='width:100pt; height:75pt' fillcolor="yellow"/>
<p>
<v:rect STYLE='width:100pt; height:75pt'
fillcolor="blue" strokecolor="red" STROKEWEIGHT="2pt" />
<p>
<v:polyline
POINTS="20pt, 55pt, 100pt, 10pt, 180pt, 65pt, 260pt, 25pt"
strokecolor="red" STROKEWEIGHT="2pt" />
</body>
</html>
```

Các ngôn ngữ định dạng khác

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Trên đây chỉ là một vài ngôn ngữ phổ biến, còn rất nhiều ngôn ngữ khác như WML, EBRL, CDF, hay XUL... Nói chung tùy thuộc vào từng lĩnh vực cụ thể mà bạn đọc nên tìm hiểu sâu hơn, trong phạm vi cuốn sách này chúng tôi chỉ dừng lại ở mức giới thiệu như trên.

6.6. Document Object Model (DOM)

Document Object Model – DOM (mô hình đối tượng tài liệu) là một API (*Application Programming Interface* – giao tiếp lập trình ứng dụng) đối với HTML và XML, định nghĩa cấu trúc logic của tài liệu và các xử lý của chúng. DOM có thể xử lý dữ liệu được lưu trong XML.

Theo mô hình này, tài liệu XML là một cây bao gồm tập hợp các nút. Những nút này nội dung có thể chứa phân tử, dữ liệu (text), thuộc tính ... và các nút con khác. Có tất cả các loại nút sau đây trong mô hình DOM:

Bảng 6.2. Các loại nút trong mô hình DOM

Nút	Mô tả
Element	Phần tử XML
Attribute	Thuộc tính
Text	Dữ liệu dạng text
CDATA section	Phân đoạn CDATA
Entity reference	Tham chiếu thực thể
Entity	Thực thể
Processing Instruction	Chỉ thị xử lý
Comment	Chú thích
Document	Tài liệu
Document Type	Kiểu tài liệu
Document Fragment	Đoạn tài liệu
Notation	Ghi chú

Ví dụ có tài liệu XML sau.

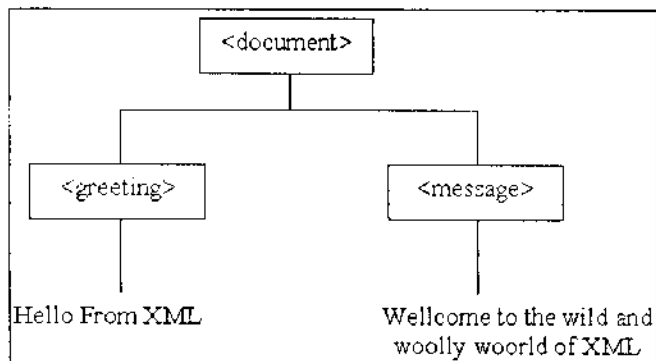
Chương 6

Giới thiệu về Extensible Markup Language (XML)

Ví dụ 6.30.

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <greeting>
    Hello From XML
  </greeting>
  <message>
    Wellcome to the wild and woolly woorld of XML.
  </message>
</document>
```

Tài liệu này có thể phân theo cấu trúc cây bao gồm các nút đối tượng:



Hình 6.7. Cấu trúc cây các nút đối tượng của tài liệu.

Nếu có nhiều con hơn chúng ta sẽ có một cây sâu hơn theo mô hình phân cấp: Nút cha, các nút con, cháu, dừ liệu

W3C định nghĩa rất nhiều cấp độ cho mô hình DOM:

- *Level 0:* Là đặc tả DOM không chính thức, khởi đầu của mô hình DOM áp dụng cho các trình duyệt thông dụng trước đây là Netscape Navigator 3.0 và IE 3.0.
- *Level 1:* Đặc tả và định nghĩa mô hình DOM ở cấp độ này được cài đặt và sử dụng nhiều nhất. DOM ở cấp độ này tập trung vào sự kết hợp giữa tài liệu HTML và XML.
- *Level 2:* Cấp độ nâng cao này chỉ đang là ứng cử viên được W3C xem xét. Mô hình DOM ở cấp độ 2 mở rộng cho phép sử dụng không gian tên và kết hợp với mô hình định kiểu như CSS hay XSL.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

- **Level 3:** Cấp độ này chỉ đang ở mức độ hoạch định. Nó cho phép nạp, xử lý, lưu lại mô hình nội dung tài liệu (như các định nghĩa DTD và Schema). Hiện chưa có tài liệu chính thức về mô hình DOM này.

Các đối tượng cơ bản của DOM

Để đại diện cho bản chất kiến trúc tầng, DOM cũng cung cấp một tập các đối tượng, phương thức và tính chất cho phép chúng ta xử lý DOM. Một vài đối tượng cơ bản được giới thiệu trong bảng sau.

Bảng 6.3. Các đối tượng cơ bản

Đối tượng	Mô tả
Node	Một nút đơn trong kiến trúc
NodeList	Một tập hợp nút
NamedNodeMap	Một tập các nút cho phép truy xuất theo tên cũng như chỉ số

Một số phương thức trong DOM

Một số phương thức sau cho phép duyệt qua các nút trong DOM (bảng 6.4)

Bảng 6.4. Một số phương thức cho phép duyệt qua các nút

Tính chất	Mô tả
childNodes	Trả về NodeList chứa các con của nút
firstChild	Trả về con đầu tiên của nút hiện hành
lastChild	Trả về con cuối cùng của nút hiện hành
parentNode	Trả về nút cha của nút hiện hành
previousSibling	Trả về nút phía trước cùng mức của nút hiện hành
nextSibling	Trả về nút kế tiếp cùng mức của nút hiện hành
nodeName	Tên của nút
nodeValue	Giá trị của nút

Bây giờ chúng ta xem lại cấu trúc của một nút cùng với các đối tượng và các thuộc tính để hiểu rõ hơn. Ở đây, chúng ta có thể sử dụng những phương thức này để duyệt qua XML DOM.

Ví dụ 6.31.

Giả sử có một đối tượng nút tên là *nodRoot* chỉ đến nút gốc *document*. Khi đó tương ứng với ví dụ ta có:

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Bảng 6.5. Cấu trúc của một nút cùng với các đối tượng và các thuộc tính

Mã	Chỉ đến
nodRoot.childNodes(0)	Greeting
nodRoot.childNodes(1)	Message
nodRoot.childNodes(0).nextSibling	Message
nodRoot.childNodes(1).parentNode	Document

Các phương thức trong DOM

Bảng 6.6. giới thiệu toàn bộ các phương thức trong DOM.

Bảng 6.6. Các phương thức trong DOM

Phương thức	Mô tả
getNodeName	Trả về kiểu định dạng nút
getNodeName	Trả về tên của nút. Nếu nút không có tên thì trả về # và kiểu nút
getNodeValue	Trả về một giá trị kiểu chuỗi hoặc giá trị null phụ thuộc vào kiểu của nút
getParentNode	Chả về nút cha của nút hiện hành
getChildNodes	Trả về một NodeList với tất cả các con của nút đó.
getFirstChild	Trả về nút đầu tiên trong NodeList.
getLastChild	Trả về nút cuối cùng trong NodeList.
getPreviousSibling	Trả về nút phía trước nút hiện hành hoặc.
getNextSibling	Trả về nút kế sau nút hiện hành hoặc
getAttributes	Trả về một NamedNodeMap bao gồm cả thuộc tính của nút
insertBefore	Chèn thêm nút (thông qua đối số) trước một nút có sẵn. Nếu nút mới đã tồn tại trong cây, nó sẽ xóa bỏ nút cũ trước khi thêm.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Bảng 6.7. Một vài phương thức của DOM NodeList

Phương thức	Mô tả
item	Thông qua một chỉ số mà trả về một nút tương ứng. Phạm vi là từ 0 đến <i>length-1</i> .
getLength	Trả về tổng số nút trong danh sách.

Bảng 6.8. Một vài phương thức của DOM NamedNodeMap

Phương thức	Mô tả
getNamedItem	Trả về hoặc một nút trong NamedNodeMap với tên xác định hoặc null.
setNamedItem	Lưu trữ một nút vào trong NamedNodeMap. Hai nút cùng tên sẽ không thể được lưu trữ trong cùng một NamedNodeMap.
removeNamedItem	Xóa một nút trong NamedNodeMap.
getLength	Trả về tổng số nút NamedNodeMap.
getValues	Trả về một NodeList mà chứa tất cả các nút trong NamedNodeMap.

Bảng 6.9. Một vài phương thức của DOM Document

Phương thức	Mô tả
getDocumentElement	Trả về nút gốc của Document.
createElement	Tạo và trả về một phần tử nút với tên thẻ được chỉ ra
createAttribute	Tạo và trả về một nút thuộc tính với tên và giá trị được chỉ ra
createTextNode	Tạo và trả về một nút text mà chứa giá trị text được chỉ ra
createComment	Tạo một chú thích

Chương 6
Giới thiệu về Extensible Markup Language (XML)

Bảng 6.10. Một vài phương thức của DOM Element

Phương thức	Mô tả
getTagName	Trả về tên của thẻ
setTagName	Thay đổi tên thẻ
getAttribute	Trả về giá trị của thuộc tính
setAttribute	Thay đổi giá trị thuộc tính. Đối số đầu là giá trị cũ, đối số thứ hai là giá trị mới
removeAttribute	Xóa bỏ thuộc tính được chỉ ra
getAttributeNode	Trả về nút thuộc tính được chỉ
setAttributeNode	Thêm vào một nút thuộc tính với tên được chỉ

Bảng 6.11. Một vài phương thức của DOM Attr

Phương thức	Mô tả
getValue	Trả về giá trị của thuộc tính
setValue	Thay đổi giá trị của thuộc tính
getName	Trả về tên của thuộc tính

Bảng 6.12. Một vài phương thức của DOM Text và Comment

Phương thức	Mô tả
getData	Trả về dữ liệu chứa trong nút (text hay comment).
setData	Thiết lập dữ liệu của nút
getLength	Trả về số kí tự chứa trong nút

Tất nhiên tùy vào trường hợp đang kết hợp sử dụng lấy dữ liệu trong XML từ ứng dụng cụ thể nào mà sử dụng các phương thức trên cho đúng (không phải là toàn bộ). Ví dụ sau chỉ ra cách lấy dữ liệu được lưu trong XML bằng Javascript:

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Ví dụ 6.32.

Dữ liệu được lưu trong file *data.xml* như sau:

```
<?xml version="1.0" encoding="UTF-8"?>
<session>
  <committee type="monetary">
    <title>Finance</title>
    <number>17</number>
    <subject>Donut Costs</subject>
    <date>7/15/2005</date>
    <attendees>
      <senator status="present">
        <firstName>Thomas</firstName>
        <lastName>Smith</lastName>
      </senator>
      <senator status="absent">
        <firstName>Frank</firstName>
        <lastName>McCoy</lastName>
      </senator>
      <senator status="present">
        <firstName>Jay</firstName>
        <lastName>Jones</lastName>
      </senator>
    </attendees>
  </committee>
</session>
```

Trang *getData.html* sử dụng Javascript để lấy tài liệu và hiển thị. Trước tiên cần tạo ra một đối tượng tài liệu tham chiếu đến file *data.xml*.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
xmlDocumentObject = new ActiveXObject("Microsoft.XMLDOM")
xmlDocumentObject.load("data.xml")
```

Sau đó sử dụng đối tượng này để lấy về phần tử gốc để tham chiếu sau này:

```
sessionNode = xmlDocumentObject.documentElement
```

Cụ thể như sau:

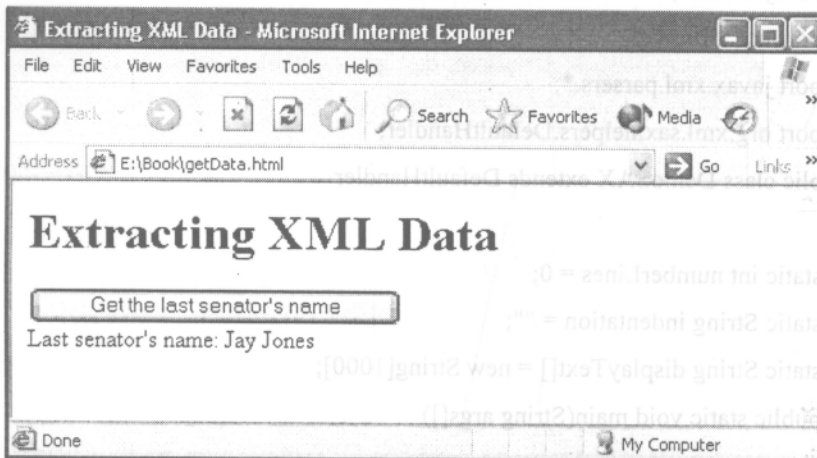
```
<HTML>
  <HEAD>
    <TITLE> Extracting XML Data </TITLE>
    <SCRIPT LANGUAGE="JavaScript">
      function readXMLData()
      {
        var xmlDocumentObject, sessionNode, committeeNode,
            attendeesNode
        var firstNameNode, lastNameNode, displayText
        xmlDocumentObject = new ActiveXObject("Microsoft.XMLDOM")
        xmlDocumentObject.load("data.xml")
        sessionNode = xmlDocumentObject.documentElement
        committeeNode = sessionNode.firstChild
        attendeesNode = committeeNode.lastChild
        senatorNode = attendeesNode.lastChild
        firstNameNode = senatorNode.firstChild
        lastNameNode = firstNameNode.nextSibling
        displayText = "Last senator's name: " +
            firstNameNode.firstChild.nodeValue + "
            + lastNameNode.firstChild.nodeValue
        displayDIV.innerHTML = displayText
      }
    </SCRIPT>
  </HEAD>
  <BODY>
    <div id="display">
    </div>
  </BODY>
</HTML>
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
</SCRIPT>
</HEAD>
<BODY>
  <H1>      Extracting XML Data      </H1>
  <INPUT TYPE="BUTTON" VALUE="Get the last senator's name"
    ONCLICK="readXMLData()">
  <BR>
  <DIV ID="displayDIV"></DIV>
</BODY>
</HTML>
```

Kết quả hiển thị trên trình duyệt. Click vào nút, thông tin sẽ hiển thị ở dưới vị trí nút:



Hình 6.8. Dùng Javascript để lấy dữ liệu từ XML.

6.7. Simple API for XML (SAX)

Mô hình DOM xem tài liệu xử lý như một cấu trúc cây và cung cấp các thuộc tính và phương thức để duyệt đến từng nút của cây nhằm trích rút dữ liệu cần lấy - tức là phải chủ động tìm đến dữ liệu.

Mô hình SAX đơn giản hơn DOM, cho phép xử lý dữ liệu XML theo mô hình hướng sự kiện rất đơn giản. Có nghĩa là dữ liệu mà ta mong muốn sẽ tự động gửi đến khi một sự

Chương 6

Giới thiệu về Extensible Markup Language (XML)

kiện phát sinh thay vì chúng ta phải tự lấy dữ liệu ra bằng cách tìm đến từng nút (đi từ nút gốc).

Đối với mô hình SAX, muốn xử lý một kiểu nút nào, chỉ cần đặt phương thức cụ thể tiếp cận kiểu nút cần xử lý. Trình phân tích SAX sẽ đọc và diễn dịch toàn bộ nội dung tài liệu. Khi phát hiện ra một kiểu nút nào đó, sự kiện tương ứng sẽ phát sinh và hàm xử lý sự kiện dành cho nút sẽ được gọi. Bằng cách này, SAX đã gửi nội dung tài liệu đến cho bạn thay vì bạn phải đi tìm kiếm nó như trong DOM.

Ví dụ 6.33.

Dưới đây là ví dụ sử dụng phương thức SAX trong JAVA để lấy về dữ liệu trong file *data.xml* ở trên. Lớp chương trình chính kế thừa lớp *DefaultHandler*, đây là lớp cho phép xử lý các sự kiện mặc định trong quá trình phân tích. Ban đầu bạn khởi tạo bộ phân tích SAX rồi sau đó nạp dữ liệu vào bộ phân tích SAX tương tự như với mô hình DOM. Lưu file mã nguồn này với tên là *DemoSAX.java*, file này khi chạy cần một tệp đối số truyền vào là *data.xml*.

```
import java.io.*;
import org.xml.sax.*;
import javax.xml.parsers.*;
import org.xml.sax.helpers.DefaultHandler;

public class DemoSAX extends DefaultHandler
{
    static int numberLines = 0;
    static String indentation = "";
    static String displayText[] = new String[1000];
    public static void main(String args[])
    {
        //phần khởi tạo và nạp dữ liệu
        DemoSAX parser = new DemoSAX();
        parser.childLoop(args[0]);
        //Hiển thị dữ liệu bằng hàm displayText
        for(int loopIndex = 0; loopIndex < numberLines; loopIndex++){
            System.out.println(displayText[loopIndex]);
        }
    }
}
```

Chương 6
Giới thiệu về Extensible Markup Language (XML.)

```
    }  
}  
public void childLoop(String uri)  
{  
    DefaultHandler saxHandler = this;  
    SAXParserFactory saxFactory = SAXParserFactory.newInstance();  
    try {  
        SAXParser saxParser = saxFactory.newSAXParser();  
        saxParser.parse(new File(uri), saxHandler);  
    } catch (Throwable t) {}  
}  
public void startDocument()  
{  
    displayText[numberLines] = indentation;  
    displayText[numberLines] += "<?xml version=\"1.0\" encoding=\"" +  
        "UTF-8\" + "\"?>";  
    numberLines++;  
}  
public void processingInstruction(String target, String data)  
{  
    displayText[numberLines] = indentation;  
    displayText[numberLines] += "<?";  
    displayText[numberLines] += target;  
    if (data != null && data.length() > 0) {  
        displayText[numberLines] += ' ';  
        displayText[numberLines] += data;  
    }  
}
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
        displayText[numberLines] += ">";
        numberLines++;
    }

    public void startElement(String uri, String localName,
        String qualifiedName, Attributes attributes)
    {
        displayText[numberLines] = indentation;
        indentation += "  ";
        displayText[numberLines] += '<';
        displayText[numberLines] += qualifiedName;
        if (attributes != null) {
            int numberAttributes = attributes.getLength();
            for (int loopIndex = 0; loopIndex < numberAttributes; loopIndex++){
                displayText[numberLines] += ' ';
                displayText[numberLines] += attributes.getQName(loopIndex);
                displayText[numberLines] += "=\"" ;
                displayText[numberLines] += attributes.getValue(loopIndex);
                displayText[numberLines] += "\"";
            }
        }
        displayText[numberLines] += '>';
        numberLines++;
    }

    public void characters(char characters[], int start, int length)
    {
        String characterData = (new String(characters, start, length)).trim();
        if (characterData.indexOf("\n") < 0 && characterData.length() > 0) {
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

```
        displayText[numberLines] = indentation;
        displayText[numberLines] += characterData;
        numberLines++;
    }
}

public void ignorableWhitespace(char characters[], int start, int length)
{
    //characters(characters, start, length);
}

public void endElement(String uri, String localName, String qualifiedName)
{
    indentation = indentation.substring(0, indentation.length() - 4);
    displayText[numberLines] = indentation;
    displayText[numberLines] += "<";
    displayText[numberLines] += qualifiedName;
    displayText[numberLines] += ">";
    numberLines++;
}

public void warning(SAXParseException exception)
{
    System.err.println("Warning: " +
        exception.getMessage());
}

public void error(SAXParseException exception)
{
    System.err.println("Error: " +
        exception.getMessage());
}
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

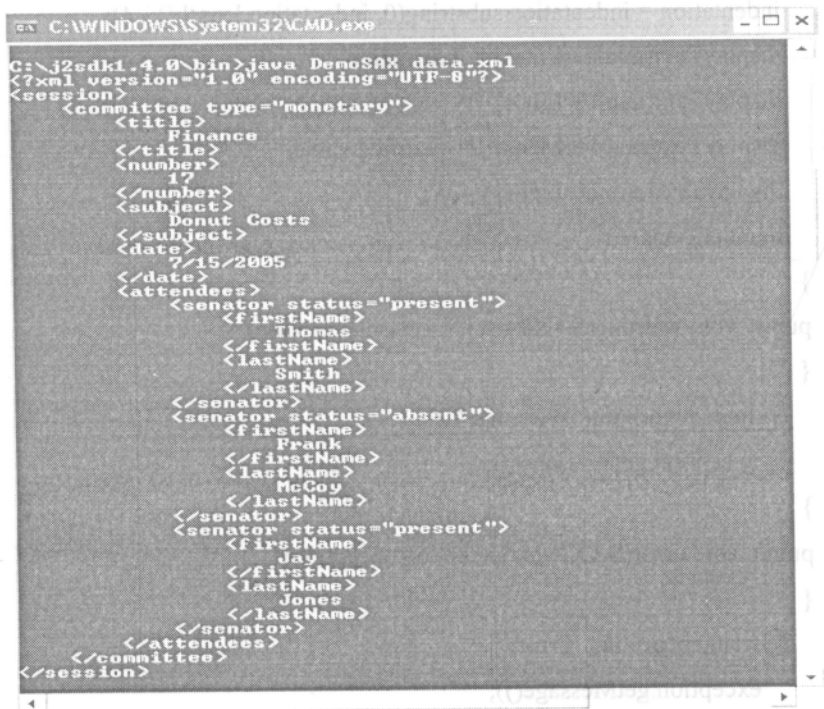
```
public void fatalError(SAXParseException exception) throws SAXException {
    System.err.println("Fatal error: " + exception.getMessage());
}
```

Lưu hai file *DemoSAX.java* và *data.xml* vào trong thư mục chạy của Java. Biên dịch và chạy chương trình:

```
C:\j2sdk1.4.0\bin\javac DemoSAX.java
```

```
C:\j2sdk1.4.0\bin\java DemoSAX data.xml
```

Kết quả hiển thị trên màn hình *console* như sau:



```
C:\WINDOWS\System32\CMD.exe
C:\j2sdk1.4.0\bin>java DemoSAX data.xml
<?xml version="1.0" encoding="UTF-8"?>
<session>
  <committee type="monetary">
    <title>
      Finance
    </title>
    <number>
      17
    </number>
    <subject>
      Donut Costs
    </subject>
    <date>
      7/15/2005
    </date>
    <attendees>
      <senator status="present">
        <firstName>
          Thomas
        </firstName>
        <lastName>
          Smith
        </lastName>
      </senator>
      <senator status="absent">
        <firstName>
          Frank
        </firstName>
        <lastName>
          McCoy
        </lastName>
      </senator>
      <senator status="present">
        <firstName>
          Jay
        </firstName>
        <lastName>
          Jones
        </lastName>
      </senator>
    </attendees>
  </committee>
</session>
```

Hình 6.9. Phân tích tài liệu XML theo mô hình SAX bằng ngôn ngữ JAVA.

6.8. XML và CSS

Trong chương trước, chúng ta đã tìm hiểu về CSS và ứng dụng của nó trong việc áp dụng các kiểu dáng định dạng cho thẻ HTML. Trong phần này chúng ta sẽ nói đến cách sử dụng CSS để định nghĩa cách hiển thị cho các thẻ XML.

Để áp dụng các định nghĩa trong CSS vào tài liệu XML, chúng ta làm như sau:

- Tạo một tài liệu XML với các thẻ để nhớ.
- Tạo ra một file kiểu CSS (tất nhiên đây là Style ngoài) có chứa định nghĩa các style của các thẻ trong tài liệu XML đó.
- Quay trở lại tài liệu XML này và chèn đoạn mã khai báo sử dụng định nghĩa CSS cho các thẻ vào dòng thứ hai sau dòng khai báo:

```
<?xml-stylesheet type="text/css" href="format.css"?>
```

Chú ý: Có thể sử dụng địa chỉ tương đối (hai file *.xml* và *.css* phải nằm cùng thư mục) hoặc địa chỉ tuyệt đối (trường hợp hai file *.xml* và *.css* không nằm cùng thư mục, khi đó phải chỉ rõ URL của file *.css*).

Ví dụ 6.34.

Ví dụ có một tài liệu XML tên là *infor.xml* chứa thông tin. Bây giờ chúng ta sẽ định dạng hiển thị nó như một tài liệu HTML.

Tệp *infor.xml*:

```
<?xml version="1.0" standalone="yes"?>
<!-- Khai báo sử dụng Style -->
<?xml-stylesheet type="text/css" href="format.css"?>
<document>
  <title>The Discourses</title>
  <philosopher>Epictetus</philosopher>
  <book>Book Four</book>
  <paragraph>
```

He is free who lives as he wishes to live; who is neither subject to compulsion nor to hindrance, nor to force; whose movements to action are not impeded, whose desires attain their purpose, and who does not

Chương 6

Giới thiệu về Extensible Markup Language (XML)

fall into that which he would avoid.

</paragraph>

<paragraph>

Who, then, chooses to live in error? No man. Who chooses to live deceived, liable to mistake, unjust, unrestrained, discontented, mean? No man.

</paragraph>

<paragraph>

Not one then of the bad lives as he wishes; nor is he, then, free. And who chooses to live in sorrow, fear, envy, pity, desiring and failing in his desires, attempting to avoid something and falling into it? Not one.

</paragraph>

<paragraph>

Do we then find any of the bad free from sorrow, free from fear, who does not fall into that which he would avoid, and does not obtain that which he wishes? Not one; nor then do we find any bad man free.

</paragraph>

</document>

Chúng ta tạo ra một file CSS là *format.css* để định nghĩa cách hiển thị các thẻ trên:

```
title {display: block; font-size: 36pt; font-weight: bold;
```

```
text-align: center; text-decoration: underline}
```

```
philosopher {display: block; font-size: 16pt;
```

```
text-align: center}
```

```
book {display: block; font-size: 28pt; text-align: center;
```

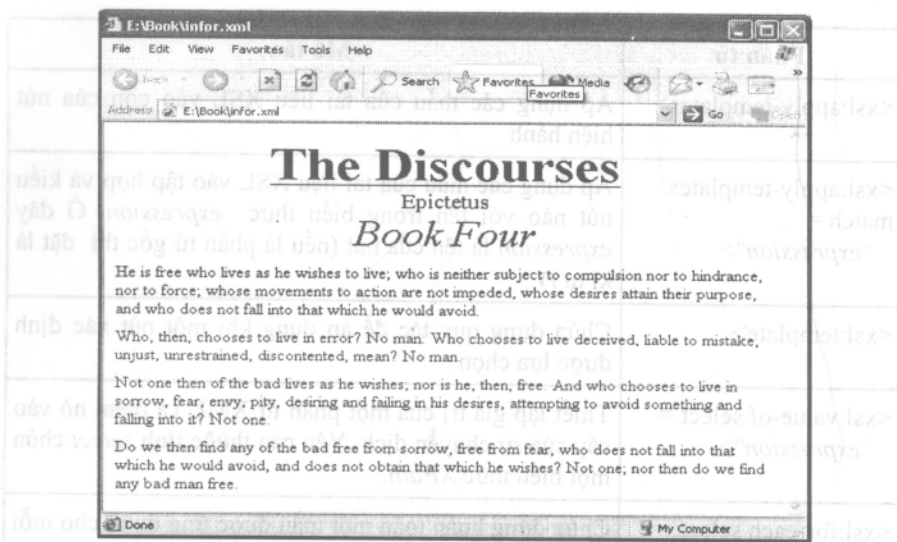
```
font-style: italic}
```

```
paragraph {display: block; margin-top: 10}
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Kết quả hiển thị trên trình duyệt:



Hình 6.10. Kết hợp với CSS để hiển thị XML.

6.9. Extensible Stylesheet Language (XSL)

Cũng giống như CSS, XSL được dùng để định dạng tài liệu XML với mục đích có được kết quả như một trang HTML. Hay nói một cách khác, kết quả mà người dùng xem được trên trình duyệt chính là trang HTML kết hợp bởi hai nội dung tài liệu XML (chứa dữ liệu) và XSL (chứa các lệnh chuyển dịch và định dạng).

Có hai loại XSL:

- *XSLT (XSL Transform)*: Là ngôn ngữ XSL chuyên về chuyển dịch (*transform*) với nhiệm vụ chủ yếu là trích rút dữ liệu XML đưa vào khuôn dạng HTML.
- *XSL – FO (XSL Formatted Objects)*: Là ngôn ngữ XSL chuyên về định dạng, cung cấp các thuộc tính thiết lập các font chữ, màu sắc, hình ảnh ...

Nói chung XSL-FO có chức năng như CSS nhưng chưa được linh động và uyển chuyển. Do đó thường dùng XSLT cho mục đích chuyển dịch tài liệu XML thành định dạng HTML hiển thị bởi trình duyệt.

Một tài liệu XSL phải được viết theo đúng cú pháp và hợp khuôn dạng với quy tắc XML. Dưới đây là một số phần tử được sử dụng trong XSL.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Bảng 6. 13. Một số phần tử được sử dụng trong XSL

Phần tử	Mô tả
<xsl:apply-templates>	Áp dụng các mẫu của tài liệu XSL vào con của nút hiện hành
<xsl:apply-templates match = "expression">	Áp dụng các mẫu của tài liệu XSL vào tập hợp và kiểu nút nào với tên trong biểu thức <i>expression</i> . Ở đây <i>expression</i> là tên của nút (nếu là phần tử gốc thì đặt là ký tự /)
<xsl:template>	Chứa định quy tắc để áp dụng khi một nút xác định được lựa chọn.
<xsl:value-of select = "expression">	Thiết lập giá trị của một phần tử XML và thêm nó vào cây của sự chuyển dịch. Yêu cầu thuộc tính <i>select</i> chứa một biểu thức <i>XPath</i> .
<xsl:for-each select = "expression">	Chứa định hoàn toàn một mẫu được ứng dụng cho mỗi nút được chọn bởi <i>XPath</i> bằng thuộc tính lựa chọn.
<xsl:sort select = "expression">	Được sử dụng như một nút con của một phần tử <xsl:apply-templates> hoặc <xsl:for-each>. Sắp xếp các nút được lựa chọn bằng các phần tử <apply-template> hoặc <for-each> do đó các nút này được xử lý theo thứ tự.
<xsl:output>	Có các thuộc tính khác nhau để định nghĩa các định dạng (ví dụ xml, html) và phiên bản (ví dụ 1.0,2.0), kiểu tài liệu và kiểu truyền thông của tài liệu. Thẻ này là phần tử top-level, có nghĩa là nó có thể được sử dụng chỉ như là phần tử con của một stylesheet.
<xsl:copy>	Thêm nút hiện hành vào cây xuất ra.

Ví dụ 6.35.

Giả sử có một tài liệu XML là file *data.xml*. Công việc chúng ta cần làm để áp dụng chuyển dữ liệu XML thành tài liệu có thể được hiển thị và đọc được bởi người dùng là tạo ra tài liệu XSLT, sau đó chèn thêm đoạn mã vào file *data.xml* này.

Chương 6

Giới thiệu về Extensible Markup Language (XML)

- Tập *data.xml* như sau:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Đoạn mã khai báo sử dụng chuyển dịch XSL -->
<?xml-stylesheet type="text/xsl" href="transform.xsl"?>
<states>
  <state>
    <name>California</name>
    <population units="people">33871648</population><!--2000 census-->
    <capital>Sacramento</capital>
    <bird>Quail</bird>
    <flower>Golden Poppy</flower>
    <area units="square miles">155959</area>
  </state>
  <state>
    <name>Massachusetts</name>
    <population units="people">6349097</population><!--2000 census-->
    <capital>Boston</capital>
    <bird>Chickadee</bird>
    <flower>Mayflower</flower>
    <area units="square miles">7840</area>
  </state>
  <state>
    <name>New York</name>
    <population units="people">18976457</population><!--2000 census-->
    <capital>Albany</capital>
    <bird>Bluebird</bird>
    <flower>Rose</flower>
    <area units="square miles">47214</area>
  </state>
</states>
```

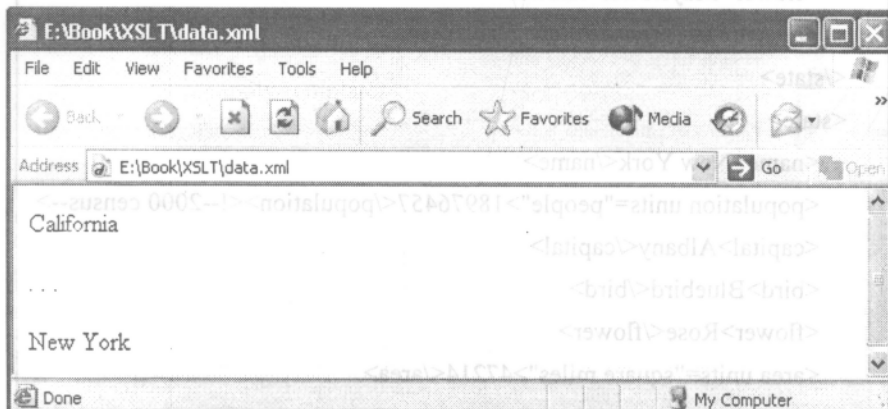
Chương 6

Giới thiệu về Extensible Markup Language (XML)

- Tập XSLT: file *transform.xml*

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0"
  <xsl:template match="states">
    <HTML>
      <BODY>
        <xsl:apply-templates/>
      </BODY>
    </HTML>
  </xsl:template>
  <xsl:template match="state">
    <P>
      <xsl:value-of select="name"/>
    </P>
  </xsl:template>
</xsl:stylesheet>
```

- Kết quả hiển thị trên trình duyệt:



Hình 6.11. Kết hợp với XSLT để hiển thị XML.

6.10. Simple Object Access Protocol (SOAP)

SOAP là giao thức dựa trên nền tảng XML. Giao thức SOAP cho phép các thành phần đối tượng và ứng dụng Internet có thể dễ dàng trao đổi với nhau thông qua chuẩn giao thức HTTP. Nói cách khác, SOAP là giao thức được xây dựng dựa trên giao thức sẵn có HTTP. Nếu như HTTP chỉ cung cấp cơ chế gửi nhận (get/pos) dữ liệu thô một cách thuần túy thì SOAP sẽ giúp bạn gửi dữ liệu theo mô hình XML có cấu trúc và dễ xử lý hơn. Hiện SOAP đang dần trở thành nền tảng của hệ điều hành Windows và một phần định nghĩa quan trọng trong bộ công cụ phát triển Visual Studio.NET. Có thể dùng giao thức SOAP để gọi một đối tượng từ xa.

Ví dụ 6.36.

Quan sát ví dụ về trao đổi giá thị trường chứng khoán giữa máy chủ và máy khách sau:

```
<soap-env:Envelope>
  <soap-env:Body>
    <xmlns:m="http://www.stock.org/stock" />
    <m:GetStockPrice>
      <StockName>IBM</StockName>
    </m:GetStockPrice>
  </soap-env:Body>
</soap-env:Envelope>
```

Để dễ hình dung, đây là một lời gọi hàm có khai báo *GetStockPrice(String StockName)*. Nếu diễn đạt theo ngôn ngữ lập trình thông thường sẽ được viết là *GetStockPrice("IBM")*.

Trình chủ Server có thể tiếp nhận yêu cầu triệu gọi hàm *GetStockPrice* và trả về kết quả cũng bằng giao thức SOAP như sau:

```
<soap-env:Envelope>
  <soap-env:Body>
    <xmlns:m="http://www.stock.org/stock" />
    <m:GetStockPriceResponse>
      <Price>25.5</Price>
    </m:GetStockPriceResponse>
  </soap-env:Body>
</soap-env:Envelope>
```

Chương 6

Giới thiệu về Extensible Markup Language (XML)

Giao thức SOAP hiện đang được nghiên cứu và triển khai rộng rãi bởi rất nhiều hãng như: Microsoft, UserLand, Ariba, Commerce One, Compaq, Developmentor, HP, IBM, IONA, Lotus, và SAP.

6.11. Kết chương

Bản thân ngôn ngữ XML rất dễ dàng để sử dụng và những ứng dụng của XML rất đa dạng và phong phú. Sự hữu ích của XML được tăng thêm khi sử dụng thêm các khái niệm DTD hay Schemas và là XSL. Vì XML chứa dữ kiện theo cấu trúc rõ ràng và phổ biến (không dựa vào kỹ thuật riêng của một công ty hay tổ chức nào) nên nó có thể được dùng để tổ chức các cơ sở dữ liệu (*database*). Với vị trí như hiện nay XML chắc chắn sẽ là một lĩnh vực quan trọng lập trình trên Web nên chúng ta sẽ rất cần thiết phải nắm chắc và hiểu rõ để nhằm đáp ứng tốt các công việc yêu cầu ở hệ thống thông tin của mình.

Chương 7
Giới thiệu về XHTML

Nội dung của chương:

- Giới thiệu về XHTML và HTML.
- Cấu trúc tài liệu XHTML.
- Các tập định nghĩa DTD cho XHTML.
- Chuyển đổi tài liệu HTML sang XHTML.
- Phê chuẩn tài liệu XHTML.

7.1. Giới thiệu về XHTML

XHTML là gì?

- XHTML là viết tắt của chuẩn cho EXtensible HyperText Markup Language
- XHTML là mục tiêu thay thế HTML
- XHTML hầu như giống hệt HTML 4.01
- XHTML là một phiên bản chặt chẽ và rõ ràng của HTML
- XHTML là HTML mà mô tả một ứng dụng XML

XHTML là chuẩn chính thức của HTML

- XHTML 1.0 trở thành một sự tiên cử chính thức của World Wide Web Consortium (W3C) vào 26/1/2000 và đã trở thành chuẩn Web hiện nay.
- W3C định nghĩa XHTML là phiên bản cuối cùng của HTML. XHTML sẽ dần dần thay thế HTML.

Tất cả trình duyệt mới đều hỗ trợ XHTML

- XHTML tương thích với HTML 4.01. Và hiện nay tất cả các trình duyệt mới đều hỗ trợ XHTML.

Chương 7

Giới thiệu về XHTML

- XHTML là sự kết hợp của HTML và XML (eXtensible Markup Language).
- XHTML chứa đựng tất cả thẻ trong HTML 4.01 kết hợp với cú pháp của XML.

7.2. Tại sao lại là XHTML?

Như đã nói ở trên, XHTML là sự kết hợp hài hòa giữa ngôn ngữ định dạng HTML và cú pháp cũng như qui tắc XML. Tập định nghĩa của XHTML bao gồm tất cả các phần tử thẻ của HTML 4.01 và do đó tài liệu XHTML của bạn hoàn toàn có thể đọc và hiển thị chính xác bởi hầu hết các trình duyệt hiểu HTML.

Đối với HTML là một ngôn ngữ định dạng không yêu cầu chặt chẽ, nghiêm ngặt trong việc thiết kế và ràng buộc tính đúng đắn của tài liệu. Nên ngay cả khi đoạn mã HTML dưới đây không đúng cú pháp thì nó vẫn được hiển thị tốt trên trình duyệt.

Ví dụ 7.1.

```
<html>
<head>
<title>This is bad HTML</title>
<body>
<h1>Bad HTML
</body>
```

Hiển thị trên trình duyệt:



Hình 7.1. Trình duyệt vẫn hiển thị tốt đoạn mã HTML tồi.

Như vậy, rất cần có một ngôn ngữ có thể định dạng chính xác kết quả trong một tài liệu đúng khuôn dạng (*well-formed*).

Sự cần thiết của việc dùng XHTML

- XML là một ngôn ngữ mô tả cấu trúc và định nghĩa dữ liệu tuyệt vời nhưng nó lại không dùng để hiển thị dữ liệu.

Chương 7

Giới thiệu về XHTML

- HTML lại được thiết kế để hiển thị dữ liệu.
- Ngày nay có nhiều loại trình duyệt và chúng có thể là chạy trên máy tính, hay trên điện thoại. Nhưng không có một công cụ mạnh để thông dịch được các ngôn ngữ định dạng tài.

Do đó bằng cách phối hợp HTML và XML với sức mạnh của chúng, chúng ta đã có một ngôn ngữ định dạng hữu ích, đó chính là XHTML. XHTML cho chúng ta viết các kiểu tài liệu đúng khuôn dạng (*well-formed*) và có thể làm việc trong tất cả trình duyệt (ngay cả với những trình duyệt cũ).

7.3. Sự khác nhau quan trọng giữa XHTML và HTML

Chính do đặc điểm ra đời của XHTML bắt nguồn từ HTML nên XHTML sử dụng lại toàn bộ các thẻ của HTML. Tuy nhiên, với qui tắc cú pháp tuân thủ theo XML và một số chuẩn nghiêm ngặt khác đã dẫn đến một số khác biệt chủ yếu sau:

- Các thẻ của XHTML phải được sắp xếp đúng thứ tự.
- Các tài liệu XHTML phải ở định dạng đúng và nghiêm ngặt.
- Tên các thẻ và thuộc tính phải ở dạng chữ thường.
- Tất cả các thẻ của XHTML phải đóng.

7.3.1. Các thẻ phải được sắp xếp đúng thứ tự

Trong HTML một vài thẻ có thể không được sắp xếp đúng mà kết quả hiển thị vẫn không thay đổi.

Ví dụ 7.2.

`<b><i>This text is bold and italic</b></i>`

Nhưng trong XHTML tất cả các thẻ phải được sắp xếp đúng vị trí.

Ví dụ 7.3.

`<b><i>This text is bold and italic</i></b>`

Chú ý: Một lỗi phổ biến trong danh sách sắp xếp là chúng ta hay quên danh sách nằm trong phạm vi một cặp thẻ `<li> </li>`,

7.3.2. Các tài liệu phải hợp khuôn dạng

Tất cả các thẻ của XHTML phải được sắp xếp bên trong thẻ gốc `<html>`. Các thẻ khác có thể có các thẻ con. Các thẻ con phải là từng cặp và sắp xếp chính xác trong thẻ cha của chúng.

Chương 7

Giới thiệu về XHTML

Cấu trúc cơ bản như sau:

```
<html>
<head> ... </head>
<body> ... </body>
</html>
```

7.3.3. Tên các thẻ và thuộc tính phải ở dạng chữ thường

Trong HTML bạn có thể viết các thẻ và thuộc tính lẫn lộn cả chữ hoa và chữ thường nhưng điều này không được phép đối với XHTML. Nguyên nhân là bởi các tài liệu XHTML là các ứng dụng XML. Mà XML lại phân biệt chữ hoa chữ thường do đó XHTML cũng phân biệt chữ hoa chữ thường. Ví dụ các thẻ như là `<br>` và `<BR>` được thông dịch thành 2 thẻ khác nhau.

Ví dụ 7.4.

Đoạn mã sai:

```
<BODY>
<P>This is a paragraph</P>
</BODY>
```

Đoạn mã đúng:

```
<body>
<p>This is a paragraph</p>
</body>
```

7.3.4. Tất cả các thẻ phải được đóng

Trong HTML cho phép hai loại thẻ: thẻ đóng (*có thẻ kết thúc*) và thẻ mở (*không có thẻ kết thúc*). Trong XHTML thì tất cả các loại thẻ phải là thẻ đóng.

Ví dụ 7.5.

Đoạn mã sai:

```
<p>This is a paragraph
<p>This is another paragraph
```

Đoạn mã đúng:

Chương 7

Giới thiệu về XHTML

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

Chú ý:

- Các thẻ mở phải hoặc là có một thẻ kết thúc hoặc là có thẻ bắt đầu nhưng phải kết thúc bằng chuỗi " />".
- Các thẻ liên kết <a> không được chứa lồng thẻ <a> khác. Ngoài ra thẻ <button> không thể chứa các phần tử như <input>, <select>, <label>, <form> ...
- Để làm cho tài liệu XHTML của bạn tương thích với các trình duyệt hiện nay, bạn nên thêm một khoảng trống trước dấu "/". Ví dụ :
, và <hr /> .

Ví dụ 7.6.

Đoạn mã sai:

```
This is a break<br>
Here comes a horizontal rule:<hr>
Here's an image 
```

Đoạn mã đúng:

```
This is a break<br />
Here comes a horizontal rule:<hr />
Here's an image 
```

7.4. Cú pháp XHTML

Một vài quy tắc cho cú pháp XHTML:

- Tên thuộc tính phải ở dạng chữ thường.
- Các giá trị thuộc tính phải được trích dẫn (*bao trong dấu nháy kép ""*).
- Thuộc tính hạn chế phải bị cấm.
- Thuộc tính *id* thay thế cho tên thuộc tính.
- Bắt buộc phải có các thẻ định nghĩa XHTML DTD.

Chúng ta sẽ tìm hiểu sâu hơn về các quy tắc này trong phần dưới đây.

Chương 7

Giới thiệu về XHTML

7.4.1. Tên thuộc tính phải ở dạng chữ thường

Vì XHTML chỉ chấp nhận tên các thẻ HTML và các thuộc tính của chúng bằng chữ thường, nên bạn phải luôn cố gắng sử dụng tên chữ thường trong trang Web.

Ví dụ 7.7.

Đoạn mã sai:

```
<table WIDTH="100%">
```

Đoạn mã đúng:

```
<table width="100%">
```

7.4.2. Giá trị thuộc tính phải được trích dẫn

Theo khuyến cáo của W3C XHTML 1.0 thì giá trị của các thuộc tính phải nằm trong ngoặc kép, mỗi trang sẽ được kiểm tra xem giá trị của thuộc tính có nằm trong dấu ngoặc kép không. Đây là một công việc tốn nhiều thời gian nên chúng ta phải chắc chắn không bao giờ quên đặt giá trị trong ngoặc kép.

Ví dụ 7.8.

Đoạn mã sai:

```
<table width=100%>
```

Đoạn mã đúng:

```
<table width="100%">
```

7.4.3. Không thể sử dụng các thuộc tính mà không có giá trị

Các thuộc tính trong HTML có thể không cần giá trị. Tuy nhiên trong XHTML thì các thuộc tính hạn chế này phải bị loại bỏ bằng cách thêm giá trị cho chúng.

Ví dụ 7.9.

Đoạn mã sai:

```
<dl compact>
<input checked>
<input readonly>
```

Chương 7

Giới thiệu về XHTML

```
<input disabled>
<option selected>
<frame noresize>
```

Đoạn mã đúng:

```
<dl compact="compact">
<input checked="checked" />
<input readonly="readonly" />
<input disabled="disabled" />
<option selected="selected" />
<frame noresize="noresize" />
```

Sau đây là danh sách các thuộc tính hạn chế trong HTML và cách chúng nên được viết trong XHTML.

Bảng 7.1. Các thuộc tính hạn chế trong HTML và cách viết trong XHTML

HTML	XHTML
compact	compact="compact"
checked	checked="checked"
declare	declare="declare"
readonly	readonly="readonly"
disabled	disabled="disabled"
selected	selected="selected"
defer	defer="defer"
ismap	ismap="ismap"
noreferrer	noreferrer="noreferrer"
noshade	noshade="noshade"
nowrap	nowrap="nowrap"
multiple	multiple="multiple"
noresize	noresize="noresize"

Chương 7

Giới thiệu về XHTML

7.4.4. Thuộc tính *id* thay cho thuộc tính *name*

HTML 4.01 định nghĩa thuộc tính *name* cho một số thẻ như *a*, *applet*, *frame*, *iframe*, *img*, và *map*. Trong XHTML, thuộc tính *name* không được chấp nhận mà sử dụng *id* để thay thế.

Ví dụ 7.10.

Đoạn mã sai:

```

```

Đoạn mã đúng:

```

```

Chú ý:

- Để phù hợp với các trình duyệt cũ, bạn nên sử dụng cả *name* và *id*, với những giá trị thuộc tính giống nhau, ví dụ như:

```

```

- Để làm cho XHTML thích hợp với các trình duyệt hiện nay, bạn nên thêm một khoảng trống trước dấu *"/*.

7.4.5. Thuộc tính *lang*

Thuộc tính *lang* áp dụng cho mỗi thành phần XHTML. Nó chỉ rõ ngôn ngữ nội dung bên trong một thành phần. Nếu bạn sử dụng thuộc tính *lang* trong một thành phần, bạn phải thêm thuộc tính *xml:lang*, như sau:

```
<div lang="no" xml:lang="no">Heia Norge!</div>
```

7.4.6. Các thành phần bắt buộc của XHTML

Với tài liệu HTML, bạn có thể định dạng tự do, thậm chí không cần dùng cặp thẻ *<html></html>*, các trình duyệt vẫn có khả năng hiểu và hiển thị đúng nội dung tài liệu. Tuy nhiên, XHTML yêu cầu tài liệu của bạn phải tuân thủ cấu trúc chuẩn gồm:

- Phải có một khai báo DOCTYPE.
- Các thành phần *html*, *head* and *body* phải được trình bày, và tiêu đề phải được viết bên trong thành phần *head*.

Chương 7

Giới thiệu về XHTML

Ví dụ dưới đây là một mẫu tài liệu XHTML thu gọn.

Ví dụ 7.11.

```
<!DOCTYPE Doctype goes here>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Title goes here</title>
</head>
<body>
Body text goes here
</body>
</html>
```

Bạn có thể thêm vào chỉ thị xử lý XML ở đầu tài liệu như:

```
<?xml version="1.0" ?>
<!DOCTYPE Doctype goes here>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Title goes here</title>
</head>
<body>
Body text goes here </body>
</html>
```

Chú ý:

- Khai báo *DOCTYPE* không phải là một phần của bản thân tài liệu XHTML. Nó luôn phải đặt ở dòng đầu tiên của tài liệu, và không nên có thẻ đóng.
- Thuộc tính *xmlns* bên trong thẻ *<html>* là bắt buộc trong tài liệu XHTML. Tuy nhiên khi thuộc tính này không có trong tài liệu XHTML thì do là giá trị tiền tố nên sẽ được tự động thêm vào.

Dưới đây là một tài liệu XHTML hoàn chỉnh

Chương 7

Giới thiệu về XHTML

Ví dụ 7.12.

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>simple xhtml document</title>
</head>
<body>
<p>Hello XHTML</p>
</body>
</html>
```

Trong đó:

- Khai báo DOCTYPE định nghĩa kiểu tài liệu:

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

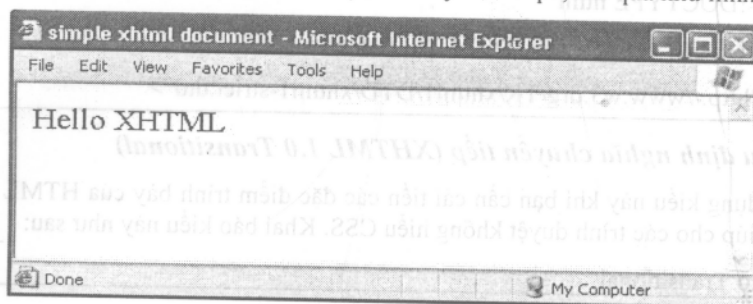
- Phần còn lại giống như HTML:

```
<html>
<head>
<title>simple document</title>
</head>
<body>
<p>a simple paragraph</p>
</body>
</html>
```

Chương 7

Giới thiệu về XHTML

Đừng bận tâm nhiều về đoạn mã trong thẻ DOCTYPE vì đó là một trong những kiểu định nghĩa tài liệu chuẩn mà sẽ được nói rõ ở phần tới. Bạn hãy lưu tài liệu này vào file *hello.html* sau đó hiển thị nó bằng trình duyệt IE hay Netscape, kết quả như hình 7.2.



Hình 7.2. Kết quả trang XHTML hiển thị trong trình duyệt.

Mặc dù đặt tên file với phần mở rộng là .html nhưng rõ ràng tài liệu của bạn được viết theo cú pháp XML. Hiện nay, chưa có nhiều trình duyệt hỗ trợ XHTML nên nhìn chung bạn có thể xem XHTML là một tài liệu HTML và không có gì rắc rối khi hiển thị với các trình duyệt.

7.5. Kiểu định nghĩa tài liệu DTD (Document Type Definitions)

Đặc điểm của DTD:

- DTD chỉ rõ cú pháp của một trang Web trong SGML.
- DTD được sử dụng bởi các ứng dụng SGML, như là HTML, để chỉ rõ quy tắc được ứng dụng vào các tài liệu định dạng cụ thể, bao gồm thiết lập các thành phần và khai báo thực thể.

XHTML được chỉ rõ trong một kiểu định nghĩa tài liệu SGML hay 'DTD'. Một XHTML DTD mô tả chính xác ngôn ngữ máy tính có thể đọc cú pháp và ngữ pháp của định dạng XHTML.

Có 3 kiểu định nghĩa DTD cho XHTML:

- STRICT
- TRANSITIONAL
- FRAMESET

7.5.1. Kiểu định nghĩa hạn chế (XHTML 1.0 Strict)

Tập định nghĩa này bao gồm những định nghĩa thẻ XHTML đơn giản, thường sử dụng nhất và diễn dịch được bởi hầu hết mọi trình duyệt. Tài liệu sử dụng kiểu định nghĩa này thường kết hợp với các Style CSS. Khai báo kiểu này như sau:

Chương 7

Giới thiệu về XHTML

XHTML 1.0 Strict

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

7.5.2. Kiểu định nghĩa chuyển tiếp (XHTML 1.0 Transitional)

Sử dụng kiểu này khi bạn cần cải tiến các đặc điểm trình bày của HTML và khi bạn muốn trợ giúp cho các trình duyệt không hiểu CSS. Khai báo kiểu này như sau:

XHTML 1.0 Transitional

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

7.5.3. Kiểu định nghĩa chuyển tiếp (XHTML 1.0 Frameset)

Sử dụng kiểu này khi bạn muốn sử dụng các khung (*frameset*) HTML để chia cửa sổ trình duyệt thành 2 hay nhiều khung. Khai báo kiểu này như sau:

XHTML 1.0 Frameset

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Chú ý: Thẻ `<!DOCTYPE>` là bắt buộc và nên được đặt ở đầu trang.

7.6. Chuyển đổi HTML sang XHTML

Ngoại trừ giới hạn nghiêm ngặt về cú pháp, các thẻ định dạng XHTML hoàn toàn giống với HTML. Xu hướng trong tương lai XHTML sẽ được sử dụng thay thế hoàn toàn HTML nhưng mặc dù vậy vẫn còn tồn tại rất nhiều trang tài liệu HTML và chúng ta phải có cách để chuyển chúng sang XHTML. Bạn có thể dùng phần mềm hỗ trợ hoặc tự làm thủ công.

Các bước thực hiện chuyển đổi HTML thành XHTML:

- Chuyển hết tên và thuộc tính của thẻ thành chữ thường.
- Thay đổi các thẻ mở và đóng sao cho đúng quy ước của XHTML.

Chương 7

Giới thiệu về XHTML

- Bổ sung giá trị cho các thuộc tính thiếu (giá trị) theo bảng trên nhưng nhớ phải cho giá trị trong ngoặc kép.
- Thêm định nghĩa *DOCTYPE* vào dòng đầu tiên của mỗi trang HTML.

Ví dụ 7.13.

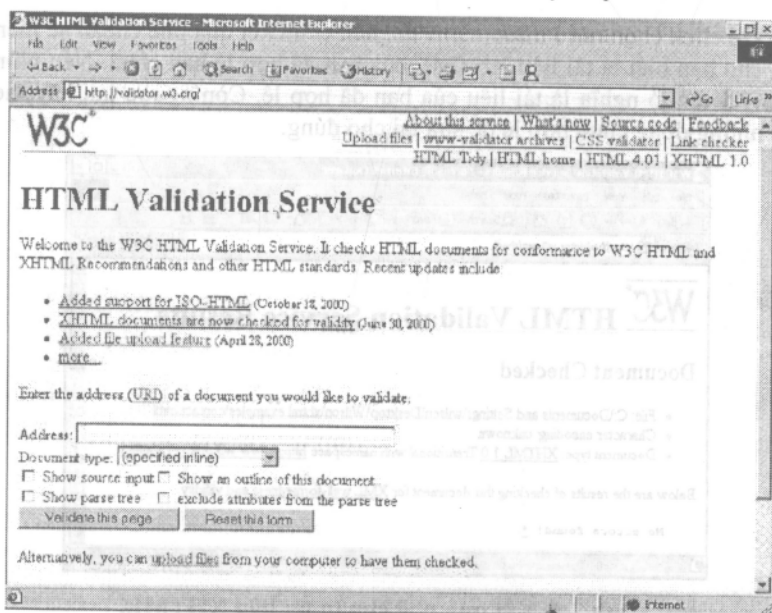
```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Chú ý: Chúng ta đã sử dụng kiểu định nghĩa *transitional DTD*. Chúng ta có thể chọn kiểu *strict DTD*, nhưng có thể nhận thấy một điều rõ ràng là càng quá “chính xác” thì càng “khó” để làm cho phù hợp.

Ngoài ra có thể sử dụng phần mềm chuyển đổi TID tài về miễn phí từ địa chỉ: <http://validator.w3.org/people/raggett/tidy>.

7.7. Phê chuẩn cú pháp tài liệu XHTML

Để kiểm tra xem tài liệu XHTML của bạn có viết đúng cú pháp theo tiêu chuẩn của W3C hay chưa hãy sử dụng trang Web hỗ trợ của tổ chức này: <http://validator.w3.org>

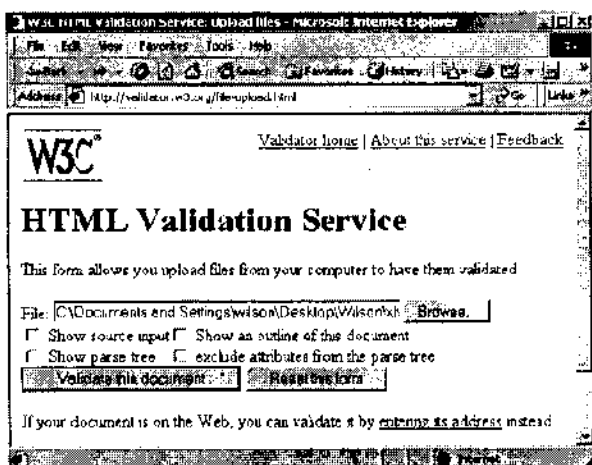


Hình 7.3. Trang web hỗ trợ phê chuẩn tài liệu XHTML.

Chương 7

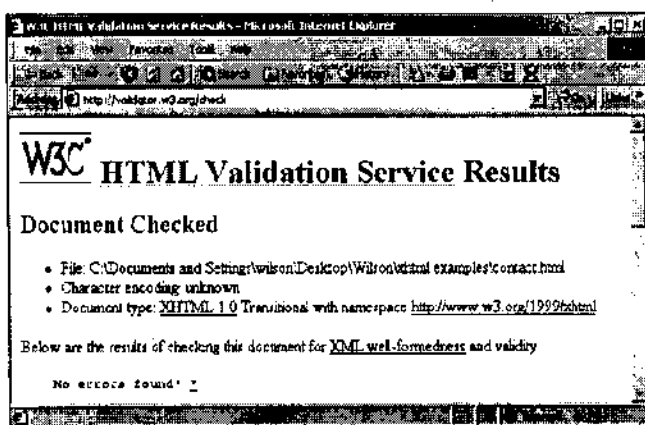
Giới thiệu về XHTML

Theo hướng dẫn của trang click chọn link *upload file*. Khi đó trình duyệt sẽ điều hướng bạn đến trang web cho phép bạn tải tài liệu của bạn lên để kiểm tra. Hãy gõ đầy đủ tên đường dẫn đến file đó hoặc click chọn nút *Browse* để chọn (hình 7.4).



Hình 7.4. Trang web cho phép tải tài liệu XHTML cần phê chuẩn.

- Sau khi click chọn nút *Validate this document* thì kết quả phê chuẩn sẽ hiển thị thông báo cho bạn biết là tài liệu XHTML của bạn đã hợp lệ hay chưa. Nếu nhìn thấy như hình 7.5 thì có nghĩa là tài liệu của bạn đã hợp lệ. Còn ngược lại, hãy theo chỉ dẫn đến những dòng chưa hợp lệ để sửa lại cho đúng.



Hình 7.5. Kết quả phê chuẩn tài liệu XHTML.

Chương 7

Giới thiệu về XHTML

7.8. Kết chương

Trong chương này chúng ta đã tìm hiểu qua ngôn ngữ định dạng mới dựa trên XML và HTML đó là XHTML. Thực ra, nó không có gì mới mẻ đối với bạn vì bạn đã được làm quen với HTML ở chương trước. Chỉ có một vài quy tắc nhỏ bạn cần phải tuân theo khi viết các tài liệu XHTML. Có thể hiện giờ cả HTML và XHTML cùng tồn tại, nhưng XHTML sẽ là xu hướng định dạng của mọi tài liệu trong tương lai.

TÀI LIỆU THAM KHẢO

1. Paul Deitel, Andrew Goldberg. *Internet and World Wide Web How to Program*. ISBN: 0131246828, Publisher Prentice Hall, 2004.
2. Thomas Powell. *HTML & XHTML: The Complete Reference (Fourth Edition)*. McGraw-Hill, 2003.
3. Robert W. Sebesta. *Programming the World Wide Web*. 3rd Edition, Addison-Wesley, 2002, ISBN: 0321149459.
4. Craig D. Knuckles. *Introduction to Interactive Programming on the Internet: Using HTML and JavaScript*. John Wiley & Sons, 2001.
5. Kate Kalata. *Internet Programming with VBScript and JavaScript*. Kalata Course Technology Media, 2001, ISBN 0619015233.
6. William Rott. *Wireless Markup Language (WML) Scripting and Programming using WML, cHTML, and xHTML*. Publisher: Althos, 2003.
7. Kris A. Jamsa. *WML & WMLScript: A Beginner's Guide*. Publisher: McGraw-Hill Osborne Media, 2001, ISBN: 0072192941.
8. John L. Cook. *WAP Servlets: Developing Dynamic Web Content With Java and WML (With CD-ROM)*. Publisher: John Wiley & Sons, 2000.
9. Steve McConnell. *Code Complete*. Second Edition, MicroSoft Press, 2004, ISBN: 1-55615-4844.
10. Harvey Deitel, Paul Deitel. *Simply Java Programming: An Application-Driven™ Tutorial Approach*. Prentice Hall, 2004, ISBN: 0131426486.

MỤC LỤC

Lời nói đầu	3
-------------------	---

Phần 1

MÁY TÍNH VÀ INTERNET

Chương 1

Giới thiệu về máy tính và Internet.....	6
--	----------

1.1. Giới thiệu.....	6
1.2. Máy tính là gì?	8
1.3. Các kiểu ngôn ngữ lập trình	9
1.4. Các ngôn ngữ bậc cao khác.....	10
1.5. Lập trình có cấu trúc.....	11
1.6. Lịch sử của Internet.....	11
1.7. Máy tính cá nhân	12
1.8. Lịch sử của World Wide Web.....	13
1.9. World Wide Web Consortium (W3C).....	14
1.10. Khuynh hướng phát triển phần cứng.....	14
1.11. Xu hướng phần mềm chủ đạo - công nghệ hướng đối tượng	15
1.12. JavaScript - kịch bản dựa vào đối tượng dành cho Web	16
1.13. Các trình duyệt Web.....	16
1.14. C và C++	17
1.15. Java.....	17
1.16. Microsoft .NET	18
1.17. HTML động (Dynamic HTML).....	18
1.18. Tài nguyên trên Web (Web Resources).....	19
1.19. Kết chương.....	19

Chương 2

Microsoft Internet Explorer	20
--	-----------

2.1. Giới thiệu về trình duyệt Web Internet Explorer 6.....	20
2.2. Kết nối Internet	20

Lập trình ứng dụng WEB INTERNET và mạng không dây

2.3. Đặc điểm của Internet Explorer 6.....	22
2.4. Tìm kiếm trên Internet.....	27
2.5. Trợ giúp và hướng dẫn trực tuyến.....	28
2.6. Lưu giữ địa chỉ những trang Web yêu thích của bạn.....	29
2.7. Giao thức truyền file (File Transfer Protocol - FTP).....	31
2.8. Tùy biến các thiết lập cho trình duyệt.....	32
2.9. Thư điện tử.....	34
2.10. Tin nhắn nhanh.....	36
2.11. Các trình duyệt Web khác.....	37
2.12. Kết chương.....	38

Phần 2

Chương 3

Ngôn ngữ cơ bản cho Web - HTML..... 40

3.1. Giới thiệu về Internet, Web và HTML.....	40
3.2. Ngôn ngữ HTML.....	41
3.3. Lập kế hoạch, tổ chức và quản lý Website của bạn.....	41
3.4. Trình duyệt và trình soạn thảo.....	42
3.5. Các thẻ của HTML.....	43
3.6. Cấu trúc của một trang HTML.....	44
3.7. Định dạng trang.....	45
3.8. Định dạng văn bản.....	48
3.9. Thẻ đường ngang.....	52
3.10. Danh sách.....	53
3.11. Chèn ảnh vào trang Web.....	57
3.12. Siêu liên kết.....	61
3.13. Thiết kế bảng.....	68
3.14. Tạo khung.....	73
3.15. Sử dụng biểu mẫu (FORM).....	82
3.16. Kết hợp khả năng đa phương tiện: âm thanh, hình ảnh.....	92
3.17. Xuất bản Website của bạn.....	94
3.18. Kết chương.....	96

Chương 4

Macromedia Dreamweaver MX 2004	97
4.1. Mở đầu với Dreamweaver MX 2004	97
4.1.1. Tìm hiểu giao diện làm việc của Dreamweaver MX 2004	98
4.1.2. Tạo và xóa một website	99
4.1.3. Tạo một trang tài liệu mới	103
4.1.4. Xem và soạn thảo mã nguồn	105
4.2. Định dạng văn bản và Style	107
4.2.1. Chèn và định dạng văn bản	107
4.2.2. CSS Style Sheets	112
4.3. Chèn ảnh và tạo liên kết	119
4.3.1. Chèn ảnh	119
4.3.2. Liên kết	122
4.4. Sử dụng bảng	126
4.4.1. Tạo bảng	126
4.4.2. Các thao tác với bảng	129
4.4.3. Định dạng bảng và các ô	131
4.5. Sử dụng Forms	133
4.5.1. Form là gì?	133
4.5.2. Tạo form	135
4.5.3. Cách tạo các trường sử dụng trong form	136
4.6. Xử lý sự kiện (Behavior)	141
4.6.1. Behavior là gì?	141
4.6.2. Gắn behavior cho một phần tử HTML	143
4.6.3. Các hành động cho behavior trong Dreamweaver	145
4.7. Kết chương	146

Phần 3

CSS, XML VÀ XHTML

Chương 5

Giới thiệu về Cascading Style Sheets (CSS)	148
5.1. Giới thiệu CSS	148
5.2. Tại sao phải sử dụng Style?	148

5.3. Cấu trúc của một Style	149
5.4. Style trong (Internal Style Sheet)	150
5.5. Style ngoài (External Style Sheet).....	152
5.6. Style cục bộ (Inline Style Sheet)	155
5.7. Kiểm tra cú pháp của style bằng dịch vụ W3C CSS	156
5.8. Một số cách định nghĩa Style khác.....	156
5.9. Thiết lập vị trí với Style	158
5.10. Thiết lập nền với Style (background).....	163
5.11. Thiết lập kích thước (Dimension) với Style	165
5.12. Định dạng văn bản với Style	166
5.13. Làm việc với khung bao.....	172
5.14. Kết chương.....	174

Chương 6

Giới thiệu về Extensible Markup Language (XML)

6.1. Giới thiệu XML.....	175
6.2. Cú pháp của XML	178
6.2.1. Một số quy tắc chung trong XML	179
6.2.2. Tạo tài liệu XML hợp khuôn dạng (Well-Formed)	182
6.3. Không gian tên (Namespaces) của XML	187
6.4. Tài liệu XML hợp lệ.....	193
6.4.1. Documents Type Definitions (DTDs).....	193
6.4.2. Schema	205
6.5. Ứng dụng của XML	206
6.6. Document Object Model (DOM)	211
6.7. Simple API for XML (SAX).....	219
6.8. XML và CSS.....	225
6.9. Extensible Stylesheet Language (XSL).....	227
6.10. Simple Object Access Protocol (SOAP)	231
6.11. Kết chương.....	232

Chương 7
Giới thiệu về XHTML..... 233

7.1. Giới thiệu về XHTML.....	233
7.2. Tại sao lại là XHTML?	234
7.3. Sự khác nhau quan trọng giữa XHTML và HTML.....	235
7.3.1. Các thẻ phải được sắp xếp đúng thứ tự	235
7.3.2. Các tài liệu phải hợp khuôn dạng.....	235
7.3.3. Tên các thẻ và thuộc tính phải ở dạng chữ thường.....	236
7.3.4. Tất cả các thẻ phải được đóng.....	236
7.4. Cú pháp XHTML	237
7.4.1. Tên thuộc tính phải ở dạng chữ thường.....	238
7.4.2. Giá trị thuộc tính phải được trích dẫn.....	238
7.4.3. Không thể sử dụng các thuộc tính mà không có giá trị	238
7.4.4. Thuộc tính id thay cho thuộc tính name.....	240
7.4.5. Thuộc tính lang.....	240
7.4.6. Các thành phần bắt buộc của XHTML	240
7.5. Kiểu định nghĩa tài liệu DTD (Document Type Definitions).....	243
7.5.1. Kiểu định nghĩa hạn chế (XHTML 1.0 Strict)	243
7.5.2. Kiểu định nghĩa chuyển tiếp (XHTML 1.0 Transitional)	244
7.5.3. Kiểu định nghĩa chuyển tiếp (XHTML 1.0 Frameset).....	244
7.6. Chuyển đổi HTML sang XHTML.....	244
7.7. Phê chuẩn cú pháp tài liệu XHTML.....	245
7.8. Kết chương.....	247



BÁCH KHOA
1956 - 2006

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

50 NĂM XÂY DỰNG
VÀ PHÁT TRIỂN

CÁC BẠN TÌM ĐỌC CÁC SÁCH ĐÃ XUẤT BẢN:

- Huỳnh Quyết Thắng, Lê Tấn Hùng. **KỸ THUẬT ĐỒ HOẠ.**
Nhà xuất bản Khoa học và Kỹ thuật. 2004
- Lương Mạnh Bá, Nguyễn Thanh Thủy. **NHẬP MÔN XỬ LÝ ẢNH SỐ.**
Nhà xuất bản Khoa học và Kỹ thuật. 2003.
- Lê Đức Trung. **CÔNG NGHỆ PHẦN MỀM.**
Nhà xuất bản Khoa học và Kỹ thuật. 2005.
- Thạc Bình Cường. **PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN.** Nhà xuất bản Khoa học và Kỹ thuật. 2004
- Thạc Bình Cường. **QUẢN LÝ DỰ ÁN PHẦN MỀM.**
Nhà xuất bản Khoa học và Kỹ thuật. 2005.
- Lương Mạnh Bá. **TƯƠNG TÁC NGƯỜI MÁY.**
Nhà xuất bản Khoa học và Kỹ thuật. 2005.

205257



Giá: 36.000đ