

MỤC LỤC

CHƯƠNG 1. MỞ ĐẦU.....	1
1.1. Lịch sử phát triển của Internet.....	1
1.2. Tổ chức của Internet.....	2
1.3. Quản lý Internet.....	4
1.4. Một số dịch vụ cơ bản trên Internet.....	4
1.4.1. Giao thức SMTP, POP3.....	4
1.4.2. Giao thức truyền file - FTP (File Transfer Protocol).....	8
1.4.3. Dịch vụ mạng thông tin toàn cầu WWW (World Wide Web).....	10
1.4.4. Giao thức HTTP (HyperText Transfer Protocol).....	10
1.4.5. Giao thức Chat.....	11
1.4.6. URL (Uniform Resource Locator).....	11
1.4.7. Web Browser.....	12
1.4.8. Web Server.....	16
1.5. Khai thác dịch vụ Internet.....	16
Câu hỏi và Bài tập.....	19
Chương 2: NGÔN NGỮ HTML.....	20
2.1. Giới thiệu chung về HTML.....	20
2.1.1. Lịch sử của HTML.....	20
2.1.2. Khái niệm về HTML.....	22
2.1.3. Các đặc điểm chính của HTML.....	23
2.2. Cấu trúc trang HTML.....	24
2.3. Thẻ và cấu trúc thẻ.....	26
2.4. Thẻ <META>.....	27
2.5. Thẻ <Head>.....	27
2.6. Sử dụng ký tự đặc biệt trong tài liệu HTML.....	28
2.7. Sử dụng các siêu liên kết.....	28
2.7.1. Giới thiệu siêu liên kết và URL.....	28
2.7.2. Sử dụng siêu liên kết.....	30
2.7.3. Điều hướng quanh Web site.....	35
2.8. Các thẻ mức đoạn.....	36
2.8.1. Thẻ P.....	36
2.8.2. Các thẻ định dạng đề mục.....	37
2.8.3. Thẻ xuống dòng BR.....	38
2.8.4. Thẻ ADDRESS.....	38
2.8.5. Thẻ BLOCKQUOTE.....	39
2.8.6. Thẻ PRE.....	39
2.9. Thẻ khối SPAN, DIV.....	40
2.10. Các thẻ định dạng ký tự thường dùng.....	40

2.11. Danh sách.....	41
2.11.1. Danh sách không thứ tự	41
2.11.2. Danh sách có thứ tự	42
2.11.3. Danh sách định nghĩa.....	44
2.12. Sử dụng font.....	45
2.13. Sử dụng màu sắc	45
2.14. Sử dụng hình ảnh trong tài liệu HTML	46
2.14.1. Chèn hình ảnh vào tài liệu HTML.....	46
2.14.2. Chèn ảnh động vào tài liệu HTML	47
2.14.3. Chèn âm thanh vào tài liệu HTML	47
2.14.4. Chèn Video vào tài liệu HTML	48
2.14.5. Chèn các Java Applets	49
2.15. Cách tạo bảng	49
2.15.1. Thẻ dùng để tạo bảng.....	50
2.15.2. Chèn hàng và cột	51
2.15.3. Xóa hàng và cột	52
2.15.4. Trộn ô.....	52
2.15.5. Định dạng cho ô.....	54
2.16. Lớp (Layer).....	54
2.16.1. Khái niệm lớp	54
2.16.2. Sử dụng các lớp	55
2.17. Giới thiệu biểu mẫu	56
2.17.1. Sử dụng biểu mẫu	57
2.17.2. Phần tử FORM.....	57
2.17.3. Các phần tử nhập của HTML	57
2.17.4. Tạo biểu mẫu	59
2.18. Khung (Frame).....	61
2.19. Phần tử IFRAME	63
2.20. Định dạng bằng CSS.....	64
2.20.1. Khái niệm CSS	64
2.20.2. Các loại style, mức độ ưu tiên style và các sử dụng	65
2.20.3. Các khai báo sử dụng CSS.....	67
2.20.4. Định dạng bằng CSS.....	73
2.21. Công cụ trong thiết kế Web	74
2.21.1. Tổng quan về các loại công cụ	74
2.21.2. Công cụ soạn thảo.....	75
2.21.3. Công cụ tạo và xử lý ảnh tĩnh	77
2.21.4. Công cụ Flash, tạo và xử lý ảnh động	77
2.21.5. Công cụ Download, Upload trang Web.....	78

Câu hỏi và Bài tập.....	80
Chương 3: LẬP TRÌNH SCRIPT.....	83
3.1. Javascript	83
3.1.1. Giới thiệu Javascript	83
3.1.2. Nhúng Javascript trong trang web	83
3.1.3. Cách đặt biểu thức cho các thuộc tính của thẻ HTML	85
3.1.4. Dùng Javascript cho trình xử lý sự kiện	86
3.1.5. Các kiểu dữ liệu	86
3.1.6. Các toán tử	88
3.1.7. Các biểu thức	90
3.1.8. Khai báo biến, mảng	91
3.1.9. Cách lệnh điều kiện	92
3.1.10. Các lệnh lặp	93
3.1.11. Hàm (function).....	95
3.1.12. Các đối tượng trong Javascript	97
3.2. Vbscript.....	109
3.2.1. Giới thiệu Vbscript	109
3.2.2. Nhúng VBscript trong trang web.....	109
3.2.3. Cách đặt biểu thức cho các thuộc tính của thẻ HTML	110
3.2.4. Dùng VBscript cho trình xử lý sự kiện.....	110
3.2.5. Các kiểu dữ liệu	111
3.2.6. Các toán tử	112
3.2.7. Các biểu thức	113
3.2.8. Khai báo biến, mảng	113
3.2.9. Cách lệnh điều kiện	116
3.2.10. Các lệnh lặp	118
3.2.11. Hàm (function) và thủ tục Procedure	120
Câu hỏi và Bài tập.....	126
Chương 4: LẬP TRÌNH ASP	128
4.1. IIS (Internet Information Server).....	128
4.1.1. Giới thiệu IIS	128
4.1.2. Cài đặt IIS.	129
4.1.3. Thực hiện file ASP trên IIS	131
4.2. ASP (Active Server Page)	132
4.2.1. Giới thiệu về ASP.	132
4.2.2. Hoạt động của ASP.....	133
4.2.3. Cú pháp của ASP	134
4.2.4. Khai báo các ngôn ngữ Script trong ASP.....	134
4.2.5. Khai báo biến trong ASP	135

4.2.6. Phạm vi hoạt động của biến.....	136
4.2.7. Các biến phiên và biến ứng dụng	137
4.2.8. Khai báo thủ tục, hàm và cách gọi.....	137
4.2.9. Liên kết nhiều tệp trong một tệp.....	139
4.3. Các đối tượng cơ bản của ASP	141
4.3.1. Đối tượng Request	141
4.3.2. Đối tượng Response.....	147
4.3.3. Đối tượng Cookies	155
4.3.4. Đối tượng Session.....	156
4.3.5. Đối tượng Application	164
4.3.6. Đối tượng Server.....	165
4.3.7. Đối tượng ASPError	170
4.4. Tập tin Global.asa	173
4.5. Đối tượng Dictionary.....	175
4.5.1. Tạo các đối tượng Dictionary	175
4.5.2. Các thuộc tính của đối tượng Dictionary	175
4.5.3. Các phương thức của đối tượng Dictionary.....	177
4.6. Đối tượng FileSystemObject	181
4.6.1. Tạo các đối tượng FileSystemObject.....	181
4.6.2. Các thuộc tính của đối tượng FileSystemObject	181
4.6.3. Các phương thức của đối tượng FileSystemObject.....	181
4.7. Đối tượng AdRotator	188
4.7.1. Tạo các đối tượng AdRotator	188
4.7.2. Các thuộc tính của đối tượng AdRotator	189
4.7.3. Các phương thức của đối tượng AdRotator.....	190
4.8. Kết nối cơ sở dữ liệu.....	190
4.8.1. Kết nối với cơ sở dữ liệu	191
4.8.2. Các đối tượng của ADO	192
4.8.3. Tập hợp các Errors.....	198
4.8.4. Stored Procedure và truyền tham số	199
Câu hỏi và Bài tập.....	203

CHƯƠNG 1. MỞ ĐẦU

1.1. Lịch sử phát triển của Internet

Tiền thân của mạng Internet ngày nay là mạng ARPANET. Vào năm 1960 khi một cơ quan của Bộ Quốc phòng Mỹ, cơ quan quản lý dự án nghiên cứu phát triển (ARPA) đề nghị liên kết 4 địa điểm đầu tiên vào tháng 7 năm 1968. Bốn địa điểm đầu tiên đó là Viện Nghiên cứu Stanford, Trường Đại học tổng hợp California ở Los Angeles, đại học Santa Barbara và trường Đại học tổng hợp Utah. Đó là mạng liên khu vực (Wide area Network) hay mạng Wan đầu tiên được xây dựng (mặc dù nó nhỏ hơn nhiều so với các mạng WAN ngày nay). Bốn địa điểm trên được nối thành mạng vào năm 1969 đã đánh dấu sự ra đời của Internet ngày nay, mạng được biết đến dưới cái tên ARPANET đã hình thành.

Nếu xét về thời gian thì thuật ngữ Internet xuất hiện lần đầu vào khoảng năm 1974. Lúc đó mạng vẫn được gọi là ARPANET. Năm 1983, giao thức TCP/IP chính thức được coi như một chuẩn đối với ngành quân sự Mỹ, và tất cả các máy tính nối với ARPANET phải sử dụng chuẩn mới này. Năm 1984, ARPANET đã được chia ra thành hai phần : Phần thứ nhất vẫn được gọi là ARPANET- dành cho việc nghiên cứu và phát triển; phần thứ hai được gọi là MILNET- là mạng dùng cho các mục đích quân sự.

Giao thức TCP/IP ngày càng thể hiện rõ các điểm mạnh của mình, quan trọng nhất là khả năng liên kết các mạng khác với lại nhau một cách dễ dàng. Chính điều này cùng với các chính sách mở cửa đã cho phép các mạng dùng cho nghiên cứu và thương mại kết nối được với ARPANET, thúc đẩy việc tạo ra một siêu mạng (SuperNetwork). Nhưng năm 1980 ARPANET được đánh giá là mạng trụ cột của mạng Internet. Mốc lịch sử quan trọng của Internet được chọn vào giữa thập kỷ 1980, khi tổ chức khoa học quốc gia Mỹ NSF thành lập mạng liên kết các trung tâm máy tính lớn với nhau gọi là NSFNET. Nhiều doanh nghiệp đã chuyển từ ARPANET sang NSFNET và do đó sau gần 20 năm hoạt động, ARPANET không còn hiệu quả nữa và đã ngừng hoạt động vào khoảng năm 1990.

Sự hình thành mạng backbone của NSFNET và những mạng vùng khác đã tạo ra một môi trường thuận lợi cho sự phát triển của Internet. Tới năm 1995, NSFNET thu lại thành một mạng nghiên cứu còn Internet thì vẫn tiếp tục phát triển.

Với khả năng kết nối mở như thế, Internet đã trở thành một mạng lớn nhất trên thế giới, mạng của các mạng. Ngày nay chúng ta thấy Internet xuất hiện trong mọi lĩnh vực : thương mại, chính trị, quân sự, nghiên cứu, giáo dục, văn hoá, xã hội ..., và cũng từ đó các dịch vụ trên Internet không ngừng phát triển tạo ra cho nhân loại một kỷ nguyên mới : Kỷ nguyên thương mại điện tử trên Internet.

Theo các chuyên gia về tin học, Internet được xem như là một mạng của các mạng (Network of network). Các mạng được nối vào Internet là mạng cục bộ (LAN—Local Area

Network, kết nối máy tính trong phạm vi hẹp, thường được sử dụng trong nội bộ một cơ quan hay tổ chức), mạng địa phương (MAN—Metropolitan Area Network, kết nối máy tính trong phạm vi một thành phố), mạng miền rộng (WAN—Wide Area Network, kết nối trong phạm vi một quốc gia hay nhiều quốc gia trong châu lục). Trên Internet thông tin được cung cấp qua dịch vụ World Wide Web (mạng toàn cầu) viết tắt là www hay Web; Web gồm có nhiều trang, gọi là các trang web (web site), mỗi trang chứa thông tin về một chủ đề riêng biệt nào đó. Internet chứa một khối lượng thông tin khổng lồ cho phép các máy nối mạng có thể truy nhập và khai thác các cơ sở dữ liệu thuộc nhiều lĩnh vực. Ngoài ra một thuật ngữ khác là Intranet cũng hay được sử dụng với nghĩa là một mạng máy tính được thiết lập trong phạm vi của một cơ quan nhằm chia sẻ thông tin qua việc sử dụng Internet. Intranet có thể bị ngăn cách với Internet bằng một bức tường lửa (firewall); Firewall là một máy chủ đứng chắn giữa Intranet và thế giới bên ngoài, theo dõi thông tin ra vào, ngăn cản sự phá rối, đánh cắp tài liệu mật hay tìm đến những website bị cấm trên Internet.

1.2. Tổ chức của Internet

Internet là một liên mạng, tức là mạng của các mạng con. Vấn đề đầu tiên là vấn đề kết nối hai mạng con. Để kết nối hai mạng con với nhau, có hai vấn đề cần giải quyết. Về mặt vật lý, hai mạng con chỉ có thể kết nối với nhau khi có một máy tính có thể kết nối với cả hai mạng này. Việc kết nối đơn thuần về vật lý chưa thể làm cho hai mạng con có thể trao đổi thông tin với nhau. Vấn đề thứ hai là máy kết nối được về mặt vật lý với hai mạng con phải hiểu được cả hai giao thức truyền tin được sử dụng trên hai mạng con này và các gói thông tin của hai mạng con sẽ được gửi qua nhau thông qua đó. Máy tính này được gọi là internet gateway hay router



Khi kết nối đã trở nên phức tạp hơn, các máy gateway cần phải biết về sơ đồ kiến trúc của các mạng kết nối. Ví dụ trong hình sau đây cho thấy nhiều mạng được kết nối bằng 2 router.



Như vậy, router R1 phải chuyển tất cả các gói thông tin đến một máy nằm ở mạng Net 2 hoặc Net 3. Với kích thước lớn như mạng Internet, việc các routers làm sao có thể quyết định về việc chuyển các gói thông tin cho các máy trong các mạng sẽ trở nên phức tạp hơn.

Để các routers có thể thực hiện được công việc chuyển một số lớn các gói thông tin thuộc các mạng khác nhau người ta đề ra quy tắc là: Các routers chuyển các gói thông tin dựa trên địa chỉ mạng của nơi đến, chứ không phải dựa trên địa chỉ của máy máy nhận .

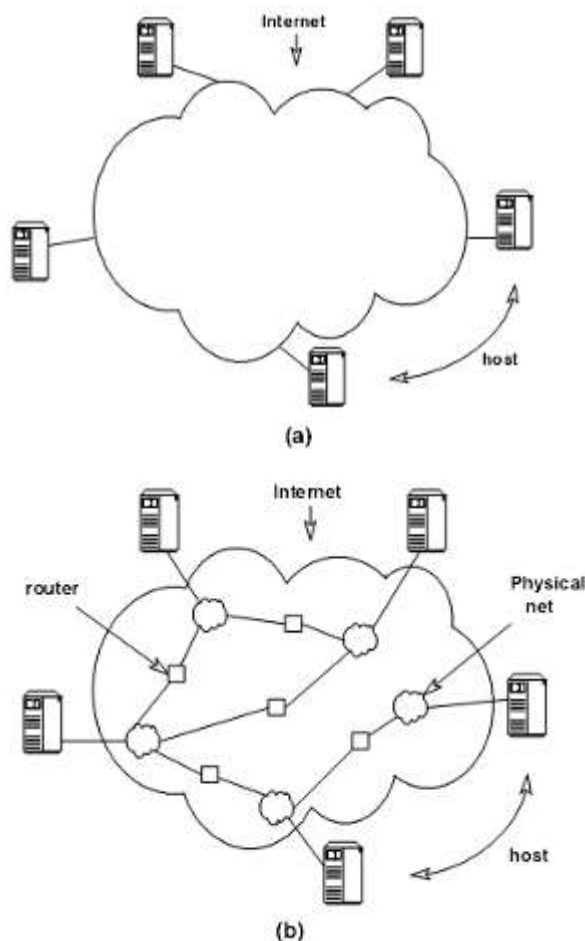
Như vậy, dựa trên địa chỉ mạng nên tổng số thông tin mà router phải lưu giữ về sơ đồ kiến trúc mạng sẽ tuân theo số mạng trên Internet chứ không phải là số máy trên Internet.

Trên Internet, tất cả các mạng đều có quyền bình đẳng cho dù chúng có tổ chức hay số lượng máy là rất chênh lệch nhau. Giao thức TCP/IP của Internet hoạt động tuân theo quan điểm sau:

Tất cả các các mạng con trong Internet như là Ethernet, một mạng diện rộng như NSFNET back bone hay một liên kết điểm - điểm giữa hai máy duy nhất đều được coi như là một mạng.

Điều này xuất phát từ quan điểm đầu tiên khi thiết kế giao thức TCP/IP là để có thể liên kết giữa các mạng có kiến trúc hoàn toàn khác nhau, khái niệm "mạng" đối với TCP/IP bị ẩn đi phần kiến trúc vật lý của mạng. Đây chính là điểm giúp cho TCP/IP tỏ ra rất mạnh. Như vậy, người dùng trong Internet hình dung Internet làm một mạng thống nhất và bất kỳ hai máy nào trên Internet đều được nối với nhau thông qua một mạng duy nhất.

Hình vẽ sau mô tả kiến trúc tổng thể của Internet.



1.3. Quản lý Internet

Thực chất Internet không thuộc quyền quản lý của bất kỳ ai. Nó không có giám đốc, không có ban quản trị. Chúng ta có thể tham gia hoặc không tham gia vào Internet, đó là quyền của mỗi thành viên. Mỗi mạng thành phần sẽ có một giám đốc hay chủ tịch, một cơ quan chính phủ hoặc một hãng điều hành, nhưng không có một tổ chức nào chịu trách nhiệm về toàn bộ Internet.

Hiệp hội Internet (Internet Society - ISOC) là một hiệp hội tự nguyện có mục đích phát triển khả năng trao đổi thông tin dựa vào công nghệ Internet. Hiệp hội bầu ra Internet Architecture Board IAB (Ủy ban kiến trúc mạng). Ban này có trách nhiệm đưa ra các hướng dẫn về kỹ thuật cũng như phương hướng để phát triển Internet. IAB họp định kỳ để bàn về các vấn đề như các chuẩn, cách phân chia tài nguyên, địa chỉ ...

Mọi người trên Internet thể hiện nguyện vọng của mình thông qua uỷ ban kỹ thuật Internet (Internet Engineering Task Force - IETF). IETF cũng là một tổ chức tự nguyện, có mục đích thảo luận về các vấn đề kỹ thuật và sự hoạt động của Internet. Nếu một vấn đề được coi trọng, IETF lập một nhóm kỹ thuật để nghiên cứu vấn đề này gọi là Nhóm đặc trách nghiên cứu phát triển Internet (Internet Researching Task Force) viết tắt là IRTF.

Trung tâm thông tin mạng (Network Information Center - NIC) gồm có nhiều trung tâm khu vực như APNIC - khu vực Châu Á - Thái Bình Dương. NIC chịu trách nhiệm phân tên và địa chỉ cho các mạng máy tính nối vào Internet.

1.4. Một số dịch vụ cơ bản trên Internet

1.4.1. Giao thức SMTP, POP3

Chuẩn Internet cho E-mail là Simple Mail Transport Protocol (SMTP). SMTP là nghi thức cấp ứng dụng (application-level) dùng để điều khiển các dịch vụ thông điệp thông qua các mạng TCP/IP. SMTP được nói rõ trong RFC 321 và 822. SMTP sử dụng cổng TCP 25. Bên cạnh SMTP, hai nghi thức khác dùng để phân phát mail đến client là POP3 và IMAP4.

MIME (Multipurpose Internet Mail Extensions) bổ sung thêm cho SMTP để cho phép gắn kèm các thông điệp đa phương tiện (không phải là văn bản) bên trong thông điệp SMTP chuẩn. MIME sử dụng mã hóa Base 64 để chuyển các file phức tạp sang ASCII. MIME được mô tả trong các RFC 2045-2049.

Các lệnh SMTP thường sử dụng:

Lệnh	Ý nghĩa
HELO	Sử dụng để định danh máy gửi với máy nhận. Lệnh này phải kèm với tên máy tính của máy gửi. Trong nghi thức mở rộng (ESMTP), lệnh EHLO thay cho lệnh này. MAIL - Khởi tạo một phiên gửi mail. Đối số bao gồm trường

	“FROM” và địa chỉ email của người gửi.
RCPT	Định danh người nhận thông điệp . Đi kèm với “TO” và địa chỉ email người nhận.
DATA	Thông báo bắt đầu gửi dữ liệu thực sự của mail (phần thân của message). Dữ liệu kết thúc bằng một dòng trống và một dấu chấm (.).
RSET	Hủy (reset) phiên gửi mail hiện hành
VERFY	Sử dụng để xác nhận một người nhận mail
NOOP	Viết tắt của “no operation”, lệnh này không làm gì cả
QUIT	Đóng kết nối
SEND	Báo cho host nhận biết là message phải được gửi đến một terminal khác
HELP	Yêu cầu thông tin trợ giúp từ host nhận

Ví dụ về truyền thông của SMTP:

Sau khi kết nối giữa người gửi (trình khách) và người nhận (trình chủ) đã được thiết lập, những việc làm sau đây là những việc hoàn toàn hợp lệ, đối với một phiên giao dịch dùng giao thức SMTP. Trong cuộc hội thoại dưới đây, những gì trình khách gửi được đánh dấu bằng chữ C: đứng trước, còn những gì trình chủ gửi được đánh dấu bằng S:. Các hệ thống máy tính đều có thể thiết lập một kết nối, bằng cách dùng những dòng lệnh của phần mềm telnet, trên một máy khách. Chẳng hạn:

```
telnet www.example.com 25
```

khởi động một kết nối SMTP từ máy gửi thông điệp đến máy chủ www.example.com.

```
S: 220 www.example.com ESMTP Postfix
```

```
C: HELO mydomain.com
```

```
S: 250 Hello mydomain.com
```

```
C: MAIL FROM:<sender@mydomain.com>
```

```
S: 250 Ok
```

```
C: RCPT TO:<friend@example.com>
```

```
S: 250 Ok
```

```
C: DATA
```

```
S: 354 End data with <CR><LF>.<CR><LF>
```

```
C: Subject: test message
```

```
C: From: sender@mydomain.com
```

```
C: To: friend@example.com
```

```
C: Hello,
```

```
C: This is a test.
```

C: Goodbye.

S: 250 Ok: queued as 12345

C: QUIT

S: 221 Bye

Các mã trạng thái SMTP

Khi một sending host gửi một lệnh SMTP đến receiving host, host nhận trả về một mã trạng thái cho máy gửi biết là điều gì đã xảy ra. Danh sách bên dưới là code được nhóm theo số đầu tiên (5xx là lỗi, 4xx lỗi tạm thời, 1xx-3xx thành công):

Mã	Ý nghĩa
211	Trả lời trợ giúp, trạng thái hệ thống
214	Help message
220	Dịch vụ sẵn sàng (Service ready)
221	Đóng kết nối
250	Hành động yêu cầu được chấp nhận
251	Người sử dụng không ở mạng cục bộ, chuyển đến
354	Bắt đầu nhập mail
421	Dịch vụ không sẵn sàng
450	Hành động không chấp nhận, mailbox bận
451	Hành động bị hủy, lỗi cục bộ
452	Hành động không chấp nhận, thiếu không gian lưu trữ
500	Không hiểu lệnh hoặc lỗi cú pháp
501	Lỗi cú pháp ở tham số
502	Lệnh không được hỗ trợ
503	Sai thứ tự các lệnh
504	Tham số của lệnh không được hỗ trợ
550	Hành động không chấp nhận, mailbox không có.
551	Không phải là người sử dụng cục bộ
552	Hủy lệnh do vượt quá không gian lưu trữ
553	Hành động không chấp nhận, tên mailbox không cho phép
554	Phiên dao dịch bị lỗi

Post Office Protocol (POP)

POP cho phép các mail client (UA – User Agent) ở máy cục bộ kết nối vào pop server (MTA – Message Transfer Agent) và lấy mail về máy tính cục bộ nơi mà người sử dụng có

thể đọc và trả lời các message . POP được định nghĩa đầu tiên vào năm 1984, được nâng cấp trong POP2 vào năm 1988. Chuẩn hiện hành là POP3.

POP3 UA kết nối với TCP /IP đến server (cổng chuẩn 110). UA nhập vào một tên đăng nhập (username) và mật mã (password). Sau khi đăng nhập , UA sử dụng các lệnh POP3 để lấy và xóa mail.

POP3 là nghi thức chỉ để nhận mail. POP3 UA sử dụng SMTP để gửi mail đến server. POP3 được mô tả trong RFC 1939.

Bảng sau đây mô tả các lệnh của POP3

Lệnh	Ý nghĩa
USER	Chỉ rõ username
PASS	Chỉ rõ password
STAT	Yêu cầu trạng thái của mailbox (số lượng message, kích thước của các message)
LIST	Liệt kê các message
RETR	Lấy một message cụ thể
DELE	Xóa một message cụ thể
NOOP	Không làm gì cả
RSET	Hủy hành động của các lệnh DELE (rollback)
QUIT	Chấp nhận các thay đổi và cắt kết nối

Internet Mail Access Protocol (IMAP)

POP3 là một nghi thức đơn giản. IMAP4 là một chuẩn mail khác có hỗ trợ nhiều tính năng hơn. IMAP4 được mô tả trong RFC 2060. IMAP4 sử dụng cổng TCP 143.

Các lệnh IMAP4

Lệnh	Ý nghĩa
CAPABILITY	Yêu cầu một danh sách các chức năng có hỗ trợ
AUTHENTICATE	Chỉ rõ cơ chế đăng nhập
LOGIN	Cung cấp username và password
SELECT	Chỉ rõ mailbox
EXAMINE	Chỉ định mailbox ở chế độ chỉ đọc
CREATE	Tạo một mailbox
DELETE	Xóa một mailbox
RENAME	Đổi tên một mailbox
SUBSCRIBE	Thêm một mailbox vào danh sách cho phép
UNSUBSCRIBE	Loại mailbox ra khỏi danh sách cho phép

LIST	Liệt kê các mailbox
LSUB	Liệt kê các mailbox cho phép
STATUS	Yêu cầu trạng thái của mailbox
APPEND	Thêm một message vào mailbox
CHECK	Yêu cầu một mailbox checkpoint
CLOSE	Chấp nhận các thao tác xóa và đóng mailbox
EXPUNGE	Chấp nhận các thao tác xóa
SEARCH	Tìm trong mailbox các message thỏa điều kiện cụ thể
FETCH	Lấy một phần của một message cụ thể
STORE	Thay đổi dữ liệu của message cụ thể
COPY	Chép message sang một mailbox khác
NOOP	Không làm gì cả
LOGOUT	Đóng kết nối

Các lệnh RFC thực sự không phải là các lệnh ngôn ngữ lập trình, nó là các chuỗi dùng để gửi đến Server qua Socket được tạo với cổng tương ứng. Ví dụ, nếu muốn gửi mail, tạo một socket TCP cổng 25 với Server (ví dụ, Post Office) sau đó lần lượt gửi các lệnh RFC (kết thúc lệnh là ký tự ENTER) và nhận các trả lời từ server để gửi mail.

Cách nhanh nhất để thử các lệnh RFC là dùng chương trình Telnet.

Ví dụ để gửi một mail, làm như sau:

Ở Command Prompt gõ lệnh: Telnet mail.myserver.com 25

Sau đó gõ các lệnh như trong ví dụ đã nêu phần trước.

Nếu viết một Windows Application thì có thể dùng MAPI control của Windows, nếu viết (asp) Webmail có thể dùng CDO, các control này giúp cho người lập trình không cần biết RFC nhưng vẫn có thể viết được các ứng dụng mail bằng cách sử dụng các phương thức và các thuộc tính của chúng. Nhưng các control trên đều dùng các lệnh RFC để làm việc gửi và nhận mail thực sự.

Việc viết một ứng dụng bằng cách sử dụng trực tiếp RFC đòi hỏi người lập trình phải code nhiều (hầu như là mọi thứ), phải tìm hiểu nhiều chuẩn khác (MIME), các kiểu mã hóa (Base64, unicode, ...), nếu viết Webmail phải tìm hiểu thêm upload file (để attach), cách tổ chức trên Webserver để quản lý mail đọc rồi và chưa đọc, hoặc tạo thêm các thư mục.

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

Dịch vụ FTP dùng để truyền tải các file dữ liệu giữa các host trên Internet. Công cụ để thực hiện dịch vụ truyền file là chương trình ftp, nó sử dụng một giao thức của Internet là giao thức FTP (File Transfer Protocol). Như tên của giao thức đã nói, công việc của giao

thức này là thực hiện chuyển các file từ một máy tính này sang một máy tính khác. Giao thức này cho phép truyền file không phụ thuộc vào vấn đề vị trí địa lý hay môi trường hệ điều hành của hai máy. Điều duy nhất cần thiết là cả hai máy đều có phần mềm hiểu được giao thức FTP. ftp là một phần mềm như vậy trên hệ điều hành Unix.

Muốn sử dụng dịch vụ này trước hết chúng ta phải có một đăng ký người dùng ở máy remote và phải có một password tương ứng. Việc này sẽ giám số người được phép truy cập và cập nhật các file trên hệ thống ở xa. Một số máy chủ trên Internet cho phép chúng ta login với một account là anonymous, và password là địa chỉ e-mail của chúng ta, nhưng tất nhiên, khi đó chúng ta chỉ có một số quyền hạn chế với hệ thống file ở máy remote.

Để phiên làm việc FTP thực hiện được, ta cũng cần 2 phần mềm. Một là ứng dụng FTP client chạy trên máy của người dùng, cho phép ta gửi các lệnh tới FTP host. Hai là FTP server chạy trên máy chủ ở xa, dùng để xử lý các lệnh FTP của người dùng và tương tác với hệ thống file trên host mà nó đang chạy.

FTP cho phép chúng ta tìm kiếm thông tin trên server bằng các lệnh thông dụng như ls hay dir. Khi người dùng đánh các lệnh này, ftp sẽ chuyển lên cho server, tại server sẽ thực hiện lệnh này và gửi về thông tin danh sách các file tìm được. Người sử dụng sau khi nhận được các thông tin này sẽ gửi yêu cầu về một file nào đó bằng lệnh: `get source_file_name destination_file_name`. Còn khi muốn truyền một file lên máy ở xa, người sử dụng dùng lệnh: `put source_file_name destination_file_name`. Để một lúc có thể tải về hoặc truyền lên máy ở xa nhiều file, người ta có thể dùng các lệnh `mget` và `mput` và sử dụng các ký tự wild cast như trong môi trường DOS. Ví dụ sau sẽ tải các file có tên là *.dat:

```
mget *.dat
```

Sau đây là một ví dụ về một giao dịch truyền file:

```
# ftp ftp.vnd.net kết nối với máy chủ
```

```
Connected to ftp.vnd.net
```

```
220 FTP Server ready.
```

```
name: anonymous gõ user name để login
```

```
331 send your e-mail as password
```

```
Password: password không hiển thị
```

```
230 User guest logged in. Access restricted is apply
```

```
ftp>dir lệnh hiển thị danh sách các file
```

```
sendmail-7.5 tcp-wrapper innd w project.dat
```

```
ftp>get project.dat tải file về local
```

```
ftp>quit thoát ra khỏi dịch vụ 221 Goodbye.
```

Để sử dụng dịch vụ FTP, người sử dụng có thể chạy phần mềm FTP client ví dụ như: WS_FTP hay CUTFTP đây là các chương trình có giao diện đồ họa khá thân thiện với người sử dụng. Chúng ta có thể download các phần mềm này từ Internet để cài lên máy tính của chúng ta.

1.4.3. Dịch vụ mạng thông tin toàn cầu WWW (World Wide Web)

Đây là dịch vụ mới và mạnh nhất trên Internet. WWW được xây dựng dựa trên một kỹ thuật có tên gọi là hypertext (siêu văn bản). Hypertext là kỹ thuật trình bày thông tin trên một trang trong đó có một số từ có thể "nở" ra thành một trang thông tin mới có nội dung đầy đủ hơn. Trên cùng một trang thông tin có thể có nhiều kiểu dữ liệu khác nhau như TEXT, ảnh hay âm thanh. Để xây dựng các trang dữ liệu với các kiểu dữ liệu khác nhau như vậy, WWW sử dụng một ngôn ngữ có tên là HTML (HyperText Markup Language). Ngôn ngữ HTML được xây dựng trên cơ sở ngôn ngữ SGML (Standard General Markup Language). HTML cho phép định dạng các trang thông tin, cho phép thông tin được kết nối với nhau.

Trên các trang thông tin có một số từ có thể "nở" ra, mỗi từ này thực chất đều có một liên kết với các thông tin khác. Để thực hiện việc liên kết các tài nguyên này, WWW sử dụng phương pháp có tên là URL (Universal Resource Locator). Với URL, WWW cũng có thể truy nhập tới các tài nguyên thông tin từ các dịch vụ khác nhau như FTP, Gopher, Wais... trên các server khác nhau.

Người dùng sử dụng một phần mềm Web Browser để xem thông tin trên các máy chủ WWW. Tại server phải có một phần mềm Web server. Phần mềm này thực hiện nhận các yêu cầu từ Web Browser gửi lên và thực hiện yêu cầu đó.

Với sự bùng nổ dịch vụ WWW, dịch vụ này càng ngày càng được mở rộng và đưa thêm nhiều kỹ thuật tiên tiến nhằm tăng khả năng biểu đạt thông tin cho người sử dụng. Một số công nghệ mới được hình thành như Active X, Java cho phép tạo các trang Web động thực sự mở ra một hướng phát triển rất lớn cho dịch vụ này.

1.4.4. Giao thức HTTP (HyperText Transfer Protocol)

HTTP là chữ viết tắt từ HyperText Transfer Protocol (giao thức truyền tải siêu văn bản). Nó là giao thức cơ bản mà World Wide Web sử dụng. HTTP xác định cách các thông điệp (các file văn bản, hình ảnh đồ họa, âm thanh, video, và các file multimedia khác) được định dạng và truyền tải ra sao, và những hành động nào mà các Web server (máy chủ Web) và các trình duyệt Web (browser) phải làm để đáp ứng các lệnh rất đa dạng. Chẳng hạn, khi chúng ta gõ một địa chỉ Web URL vào trình duyệt Web, một lệnh HTTP sẽ được gửi tới Web server để ra lệnh và hướng dẫn nó tìm đúng trang Web được yêu cầu và kéo về mở trên trình duyệt Web. Nói nôm na hơn, HTTP là giao thức truyền tải các file từ một Web

server vào một trình duyệt Web để người dùng có thể xem một trang Web đang hiện diện trên Internet. HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nền tảng cho Internet).

Có một tiêu chuẩn chính khác cũng điều khiển cách thức World Wide Web làm việc là HTML (HyperText Markup Language, ngôn ngữ đánh dấu siêu văn bản), có chức năng quản lý cách thức mà các trang Web được định dạng và hiển thị.

Người ta gọi HTTP là một giao thức “phi trạng thái” (stateless) bởi vì mỗi lệnh đều được thực thi một cách độc lập, lệnh sau không biết bất cứ điều gì về các lệnh đã đến trước mình. Đây chính là một hạn chế, khiếm khuyết của HTTP. Nó là nguyên nhân chính của tình trạng rất khó thực thi các trang Web có khả năng phản ứng thông minh đối với lệnh mà người dùng nạp vào. Và sự hạn chế này đang được các nhà phát triển khắc phục trong các công nghệ mới như ActiveX, Java, JavaScript và cookies.

Phiên bản mới nhất của HTTP là 1.1. So với phiên bản nguyên thủy (HTTP 1.0), phiên bản mới này truyền tải các trang Web nhanh hơn và giảm tình trạng tắc nghẽn giao thông Web.

1.4.5. Giao thức Chat

Internet Relay Chat (IRC - Nói chuyện qua Internet) là phương tiện "thời gian thực", nghĩa là những từ chúng ta gõ vào sẽ xuất hiện gần như tức thời trên màn hình của người nhận và trả lời của họ của xuất hiện trên màn hình của chúng ta như vậy. Thay vì phải chờ vài phút hay vài ngày đối với thông điệp, chúng ta có thể trao đổi tức thời với tốc độ gõ chữ của chúng ta. IRC có thể mang tính cá nhân như e-mail, người lạ không khám phá được nội dung trao đổi của chúng ta, hoặc chúng ta có thể tạo "kênh mở" cho những ai chúng ta muốn cùng tham gia. Cũng không hiếm các kênh IRC có từ 10 người trở lên tham gia hội thoại. Ngoài việc trao đổi lời, người dùng IRC còn có thể gửi file cho nhau như hình ảnh, chương trình, tài liệu hay những thứ khác.

Cũng như các dịch vụ khác của Internet, phạm vi hội thoại trên các kênh IRC là rất rộng, có thể bao gồm cả những chủ đề không phù hợp với trẻ em, vì vậy cần có biện pháp giám sát những trẻ em muốn sử dụng dịch vụ này

1.4.6. URL (Uniform Resource Locator)

URL Là chữ viết tắt của "Uniform Resource Locator", dùng để chỉ tài nguyên trên Internet. Sức mạnh của web là khả năng tạo ra những liên kết siêu văn bản đến các thông tin liên quan. Những thông tin này có thể là những trang web khác, những hình ảnh, âm thanh... Những liên kết này thường được biểu diễn bằng những chữ màu xanh có gạch dưới được gọi là anchor. Các URL có thể được truy xuất thông qua một trình duyệt (Browser) như IE hay Netscape.

URL được dùng để xác định địa chỉ của một tài liệu (hoặc dữ liệu khác) trên World Wide Web. Một địa chỉ đầy đủ sẽ như sau:

http://gocongnghe.net/html/trangtruoc.htm và tuân theo cú pháp sau:

Scheme://host.domain:port/path/filename

Scheme: Là một trong các giao thức Internet, gồm http, ftp, gopher, news (USENET news), nntp (Network News Transfer Protocol), Telnet và WAIS (Wide Area Information Servers), và những giao thức khác. Địa chỉ dưới đây dùng giao thức http:

http://gocongnghe.net/html/html_basic.htm

Domain: xác định tên miền của trang web trên Internet ví dụ như gocongnghe.net

Host: xác định tên miền của host. Nếu được bỏ qua, thì mặc định của host cho http là www.

Port: xác định port number tại host. Số cổng thường được bỏ qua. Số cổng mặc định của http là 80.

Path: xác định đường dẫn trên server. Nếu đường dẫn được bỏ qua, thì tài liệu phải được định vị tại thư mục gốc của trang web.

Filename: xác định tên của tài liệu. Tên mặc định của một tài liệu có thể là default.asp hoặc index.html hoặc một cái gì đó phụ thuộc vào những cài đặt của server.

1.4.7. Web Browser.

Một web browser là một phần mềm ứng dụng để truy xuất, trình diễn và chuyển các nguồn thông tin (information resource) trên mạng hệ thống mạng toàn cầu (World Wide Web). Một nguồn thông tin được nhận dạng bởi một Uniform Resource Identifier (URI) và có thể là một trang web, phim - video, hình ảnh (images) hoặc các mẫu thông tin khác.



Hình 1.1. Trình duyệt web để xem thông tin một website

Mặc dù các trình duyệt với mục đích là để truy cập vào hệ thống mạng toàn cầu, các trình duyệt còn được sử dụng để truy cập các thông tin được cung cấp bởi các web servers (máy chủ web) trong hệ thống mạng riêng hoặc các tài liệu (files) đến các hệ thống file (file system). Hoặc cũng được dùng để tiết kiệm tài nguyên thông tin cho các hệ thống lưu trữ file.

Lịch sử về web browser

Lịch sử của các trình duyệt Web bắt đầu vào cuối thập niên 1980, khi một loạt các công nghệ đặt nền móng cho các trình duyệt Web đầu tiên, WorldWideWeb, do Tim Berners-Lee vào năm 1991. Sự phát triển này dựa trên các công nghệ phần mềm và phần cứng hiện có, ngoài ra cũng mở ra nhiều công nghệ mới bắt đầu từ đây.

Ted Nelson và Douglas Engelbart phát triển khái niệm siêu văn bản trước khi Berners-Lee và CERN khá lâu. Web browser đã trở thành cốt lõi của World Wide Web. Berners-Lee thừa nhận sự đóng góp của Engelbart.

Việc giới thiệu trình duyệt Mosaic NCSA Web vào năm 1993 - một trong những trình duyệt web với giao diện đồ họa đầu tiên - đã dẫn tới sự bùng nổ trong việc dùng web. Marc Andreessen, người lãnh đạo của đội Mosaic tại NCSA, sớm bắt đầu công ty riêng của mình, đặt tên là Netscape, và phát hành các phiên bản Mosaic Netscape Navigator vào năm 1994, và sau đó nhanh chóng trở thành trình duyệt phổ biến nhất thế giới, chiếm 90% của tất cả các sử dụng Web đỉnh cao của nó.



Hình 1.2. Trình duyệt web Netscape

Microsoft đã đáp trả bằng trình duyệt Internet Explorer của mình trong năm 1995 (cũng bị ảnh hưởng nhiều từ Mosaic), và đầu cuộc chiến trình duyệt đầu tiên. Bằng việc kết hợp Internet Explorer với Windows, Microsoft đã có thể tận dụng ưu thế của nó trong thị trường hệ điều hành để quản lý thị trường trình duyệt Web; Internet Explorer sử dụng đạt vị trí trên 95% người dùng vào năm 2002. Internet Explorer chiếm 60% người sử dụng trình duyệt vào Tháng Tư năm 2010 theo thống kê của Net Applications.



Hình 1.3. Trình duyệt web Internet Explorer của Microsoft (IE7)

Opera đầu tiên xuất hiện vào năm 1996; mặc dù nó đã không bao giờ đạt được sử dụng rộng rãi, chiếm 2% thị phần trình duyệt vào tháng tư năm 2010, tuy nhiên Opera chiếm một phần đáng kể của thị trường đang phát triển nhanh là trình duyệt Web điện thoại di động, Opera được cài đặt sẵn trên trên 40.000.000 điện thoại. Opera cũng có sẵn trên một số hệ thống nhúng khác, bao gồm video game console Wii của Nintendo.



Hình 1.4. Trình duyệt web Opera 11

Năm 1998, Netscape triển khai Mozilla Foundation trong một nỗ lực để sản xuất một trình duyệt cạnh tranh bằng cách sử dụng mô hình phần mềm mã nguồn mở. Đó là trình duyệt sau này phát triển thành Firefox, sau đó thì trình duyệt này được phát triển khá tốt trong khi vẫn còn trong giai đoạn beta, ngay sau khi phát hành Firefox 1.0 vào cuối năm 2004, Firefox (mọi phiên bản) đã chiếm 7,4% thị phần sử dụng trình duyệt. Và đến tháng tư năm 2010, Firefox đã có một thị phần 25%.



Hình 1.5. Trình duyệt web Firefox

Safari của Apple đã phát hành phiên bản beta đầu tiên vào tháng một năm 2003; vào tháng 10 năm 2009, đã chi phối thị phần của trình duyệt web dựa trên việc phát triển từ các ứng dụng của Apple, chiếm dưới 5% thị trường trình duyệt toàn bộ vào tháng 4 năm 2010. Nó được xem như là công cụ cho web (còn được gọi là WebKit) để trình diễn các tính năng và là nền tảng được ứng dụng nhiều trên thiết bị di động, bao gồm cả hệ điều hành iPhone, Google Android, Nokia S60 và Palm WebOS.

Vào tháng 9 năm 2008, Google bắt đầu nhảy vào cuộc chiến các trình duyệt bằng trình duyệt web Google Chrome. Đến tháng 4 năm 2010, Google Chrome đã chiếm 7% thị phần.



Hình 1.6. Trình duyệt web Safari của Apple

Chức năng

Mục đích chính của một trình duyệt web là để mang lại nguồn thông tin cho người dùng. Quá trình này bắt đầu khi một người sử dụng nhập vào (URI) hay tạm gọi là một đường dẫn, Ví dụ như <http://www.khkt.net/>. Các tiền tố của URI sẽ xác định cho phương thức truy cập và dữ liệu nhận được sẽ được biên dịch như thế nào. Các loại URI thông dụng nhất bắt đầu với [http:](http://) định một nguồn tài nguyên để được lấy dựa trên Hypertext Transfer Protocol (HTTP) (tôi tạm gọi là giao thức truyền dữ liệu cho các siêu văn bản). Nhiều trình duyệt cũng hỗ trợ một loạt các tiền tố khác, chẳng hạn như [https:](https://) (cũng tương tự như HTTP nhưng dữ liệu được mã hóa và bảo mật hơn) hay [ftp:](ftp://) cho File Transfer Protocol (giao thức truyền tải file), và [file:](file://) cho các tập tin lưu trữ nội bộ. Các trình duyệt web có thể không trực tiếp xử lý mà thường chuyển qua các ứng dụng khác xử lý. Ví dụ, <mailto:> URI thường sẽ được chuyển qua chương trình ứng dụng mail mặc định mà người dùng đang sử dụng, hoặc [news:](news://) sẽ gọi đến các chương trình đọc tin tức.

Tuy nhiên trình duyệt cũng có cơ chế mở và được hỗ trợ bởi các plugins của nó để có thể hiển thị được các loại tài liệu trực tiếp trên web browser. Hầu hết các trình duyệt có thể hiển thị hình ảnh, âm thanh, video, và các tập tin XML, và thường có một số plug-in mặc định để hỗ trợ các ứng dụng Flash và Java applet. Khi gặp phải một tập tin không được hỗ trợ thì tập tin này sẽ được tải về thay vì hiển thị trực tiếp trên web, lúc này trình duyệt sẽ nhắc người dùng để lưu tập tin vào đĩa.

Tính tương tác trong một trang web có thể được ứng dụng bởi javascript mà thường không đòi hỏi sự hỗ trợ thêm của plugin. javascript có thể được sử dụng cùng với các công nghệ khác để cho phép tương tác trực tiếp với máy chủ trang web, ví dụ: tương tác thông qua AJAX.

Các thông tin trên trang web có thể chứa siêu liên kết (hyperlinks) tới các nguồn thông tin khác. Mỗi liên kết có chứa các URI để đi đến các nguồn thông tin khác. Khi một liên kết được nhấp, trình duyệt điều hướng người dùng tới trang thông tin khác.

1.4.8. Web Server

Web Server (máy phục vụ Web): máy tính mà trên đó cài đặt phần mềm phục vụ Web, đôi khi người ta cũng gọi chính phần mềm đó là Web Server.

Tất cả các Web Server đều hiểu và chạy được các file *.htm và *.html, tuy nhiên mỗi Web Server lại phục vụ một số kiểu file chuyên biệt chẳng hạn như IIS của Microsoft dành cho *.asp, *.aspx...; Apache dành cho *.php...; Sun Java System Web Server của SUN dành cho *.jsp...

1.5. Khai thác dịch vụ Internet

Truy cập vào mạng Internet có thể có 2 cách: Truy cập trực tiếp thông qua đường dành riêng (Leased Line) và truy cập gián tiếp thông qua mạng điện thoại công cộng. Việc

đăng ký một đường thuê bao dành riêng chỉ dành cho những cơ quan, đơn vị với mục đích truy cập mạng Internet không chỉ khai thác các tài nguyên, dịch vụ sẵn có trên mạng Internet mà còn sử dụng mạng Internet như là một môi trường kết nối từ xa tới các tài nguyên trên mạng LAN của đơn vị mình. Khi đó người sử dụng có thể xây dựng máy chủ Mail, máy chủ FTP, xây dựng mạng riêng ảo (VPN- Virtual Private Network)...Tất nhiên việc này đòi hỏi tốn kém tiền bạc và công sức. Còn nếu chúng ta chỉ truy cập mạng Internet để khai thác các dịch vụ sẵn có trên mạng thì chúng ta có thể truy cập thông qua mạng điện thoại công cộng. Yêu cầu tối thiểu cho người khai thác dịch vụ Internet với hình thức này chúng ta phải đáp ứng được các yêu cầu sau:

- Phần mềm để kết nối với các máy chủ trên Internet:

Để có thể thực hiện việc kết nối tới máy chủ trên Internet, chúng ta cần có một bộ phần mềm thực hiện giao thức TCP/IP. Phần mềm này sẽ thực hiện chức năng điều khiển modem để kết nối và truyền dữ liệu với máy chủ. Có nhiều phần mềm thực hiện chức năng này, ví dụ Dialup Networking của Windows 3.x và windows 9x, Windows2000, Trumpet Winsock...

Ngày nay việc truy cập này trở nên đơn giản và thuận lợi hơn với phương thức thuê bao kết nối ADSL với phương thức này không chỉ tăng tốc độ truyền nhận dữ liệu mà việc kết nối cũng trở nên đơn giản hơn qua hệ thống modem rounter ADSL thế hệ mới. Nếu chúng ta dùng phương thức thuê bao này thì Modem ADSL sẽ giúp kết nối trực tiếp đến máy chủ mà chúng ta không cần thêm bất cứ phần mềm nào khác để thực hiện kết nối

- Các phần mềm thực hiện TCP/IP thường cho phép chúng ta kết nối theo hai giao thức là SLIP (Serial Line Internet Protocol) và PPP (Point to Point Protocol). Từ Windows98 trở đi có hỗ trợ MPPP(Multi PPP) cho phép một máy tính có thể sử dụng nhiều cổng COM và Modem để tăng tốc độ kết nối. Tất nhiên để thực hiện được điều này thì nhà cung cấp dịch vụ của chúng ta phải hỗ trợ chế độ truy cập MPPP. Hiện nay ở Việt nam chỉ có VDC là hỗ trợ dịch vụ này. Giữa SLIP và PPP có sự khác nhau, về mặt kỹ thuật SLIP là một giao thức nền tảng của mạng và PPP là giao thức ở cấp độ kết nối. Có hai sự khác biệt thực tế: PPP hơi nhanh hơn và có thể xử lý những loại mạng khác như DECnet. Sau khi chúng ta kết nối được với nhà cung cấp dịch vụ thì chúng ta cần phải có phần mềm công cụ để duyệt các trang WEB. Phổ biến hiện nay có Nescape Navigator với các Version 4.x; Internet Explore của Windows, FireFox... Điều cuối cùng, tất nhiên là chúng ta phải thực hiện việc đăng ký sử dụng với nhà cung cấp dịch vụ. ở Việt nam hiện nay đang có các nhà cung cấp dịch vụ nh- VDC, FPT, NETNAM, SAIGON POSTEL. Sau khi đăng ký sử dụng dịch vụ chúng ta sẽ được cung cấp các thông số hoà mạng như :

- User Name: tên đăng ký sử dụng dịch vụ Internet.
- Password: mật khẩu dùng để xác định quyền sử dụng dịch vụ.

- Email Address and Password: Địa chỉ thư điện tử và mã truy nhập địa chỉ thư của chúng ta
- Số điện thoại dùng để truy nhập vào mạng.

Câu hỏi và Bài tập chương 1

1. Hãy cho biết các dịch vụ Internet thông dụng và các giao thức được sử dụng cho các dịch vụ này.
2. Tìm hiểu cách phân lớp trong địa chỉ IP. Hãy cho biết các địa chỉ IP của Việt Nam thuộc lớp nào.
3. Tại sao cần phải có sự ánh xạ giữa tên miền và địa chỉ IP? Hãy cho biết tên của dịch vụ thực hiện việc này.
4. Cho biết các tên miền được dùng thông dụng hiện nay.
5. Tìm hiểu một số công ty được ủy quyền cấp tên miền trên Internet. Chỉ ra một lợi ích của mạng Internet mà bạn hay công ty của bạn đã hoặc có thể có được.
6. Trang web là gì? Chương trình được sử dụng để xem các Web được gọi là gì? địa chỉ web là gì và cấu trúc của địa chỉ? Để xem một trang web, ta cần gõ địa chỉ trang web đó vào đâu?
7. Để tìm kiếm thông tin trên Internet, bạn cần làm gì? Chỉ ra một địa chỉ trang web tìm kiếm thông dụng?
8. Dịch vụ thư điện tử được dùng để làm gì? Để sử dụng thư điện tử, trước hết bạn phải làm gì ? Chỉ ra một địa chỉ trang web cho dịch vụ thư điện tử miễn phí và nêu cách tạo lập một hòm thư trên trang web đó?
9. Trình bày một số kỹ thuật tìm kiếm thông tin trên internet?

Chương 2: NGÔN NGỮ HTML

2.1. Giới thiệu chung về HTML

2.1.1. Lịch sử của HTML

Cuộc sống của chúng ta đang thay đổi từng ngày, từ các hoạt động dạo dịch thương mại lớn đến những thủ tục văn bản thông thường việc trao đổi thông tin ngày càng được đơn giản, nhanh chóng và thuận tiện, giúp chúng ta tiết kiệm công sức và tiền của là nhờ vào các dịch vụ web. Có thể nói các dịch vụ trên web đã và đang làm thay đổi cuộc sống của chúng ta.

Vậy trang web là gì? Làm thế nào để xây dựng được các trang web, tạo ra các ứng dụng web để đáp ứng yêu cầu của chúng ta.

Trang web hiểu một cách đơn giản nó là một trang thông tin được tạo bằng ngôn ngữ HTML và được đọc bởi một trình duyệt web.

Như vậy để tạo được các trang web, tạo ra các ứng dụng trên web thì chúng ta phải biết ngôn ngữ HTML và sử dụng ngôn ngữ HTML này để tạo ra các trang web theo yêu cầu của ứng dụng.

Khái niệm siêu văn bản do nhà tin học Ted Nelson đề ra vào năm 1965 như một ước mơ về khả năng của máy tính trong tương lai "Computer Dreams". Ông hy vọng về các máy tính có trí tuệ như con người, biết cách lục tìm các thông tin cần thiết và do đó nảy sinh khái niệm siêu văn bản.

Ý tưởng để thực hiện khái niệm siêu văn bản là của một kỹ sư trẻ người Anh tên là Tim Berners-Lee. Sau khi tốt nghiệp Đại học Oxford (Anh) năm 1976, vào năm 1980, Tim đã viết một chương trình mô phỏng mối liên kết hai chiều bất kỳ trên một đồ thị như kiểu liên kết siêu văn bản. Năm 1989, trong khi làm việc tại Viện nghiên cứu kỹ thuật hạt nhân châu Âu (CERN) tại Berne, Thụy Sĩ, thấy các đồng nghiệp rất vất vả trong việc tra tìm tài liệu Tim đã đưa ra một đề án lưu trữ siêu văn bản trên máy tính sao cho dễ dàng tìm kiếm tài liệu hơn.

Ngôn ngữ HTML đầu tiên do Tim tạo ra có 20 thẻ, lấy ý tưởng từ ngôn ngữ SGML, nhưng điều kì diệu đáng kinh ngạc là ở chỗ 13 trong số 20 thẻ đó vẫn còn được hiển thị trong HTML4

Trải qua nhiều thăng trầm HTML từ một ngôn ngữ đơn giản ứng dụng trong phạm vi nhỏ thì ngày nay HTML được sử dụng trên toàn thế giới với những ứng dụng rộng rãi vượt qua cả sức tưởng tượng của tác giả.

HTML 1

HTML 1 không cho phép truyền đạt cấu trúc trang phức tạp, nhưng vừa đủ để cho phép tạo ra một trang web đơn giản. Vạn sự khởi đầu nan. Phiên bản đầu tiên được tung ra

vào cuối năm 1990 và gần như một năm sau, ngôn ngữ này mới thực sự được ứng dụng. Năm 1993 các chuẩn HTML trở thành nền tảng của Mosaic- Trình duyệt đầu tiên của Internet.

HTML 2

Nhiều công ty lớn đã đánh giá không đúng sức mạnh của ngôn ngữ siêu văn bản , nhưng cuối cùng thì các chuẩn HTML cũng được phổ biến rộng rãi. Bởi vì không có một tổ chức nào đứng ra hỗ trợ cho sự phát triển ngôn ngữ HTML, mọi thứ bị ngừng trệ. Vào tháng 7 năm 1994, HTML 2 được phát hành. HTML 2 là phiên bản cải tiến của HTML. Phiên bản lần này được tạo ra bởi sự nỗ lực rất lớn của những người yêu thích HTML trên khắp thế giới. Những người này đã đảm nhận một nỗ lực khổng lồ khi chú ý đến tất cả đóng góp từ khắp nơi trên thế giới cho phiên bản mới này. Trước tình hình này, cũng trong cùng năm 1994, Tập Đoàn Tài Chính World Wide Web được thành lập với người đứng đầu là Tim Berners-Lee. Năm 1995, thẻ mới là thẻ “bgcolor”(màu nền) hay thẻ “font face”(font chữ?) được đưa vào ứng dụng; Tôi đưa ra những ví dụ về các thẻ này nhằm nhấn mạnh trình độ đã đạt tới của chuẩn HTML vào thời điểm hiện tại...chúng ta có thể nói rằng sự khác biệt giữa phiên bản đầu tiên và phiên bản năm 1995 là rất lớn.

HTML 3

Internet làm cho ngôn ngữ HTML phát triển và chính nó cũng ứng dụng những phát triển của HTML. W3C chấp nhận những phiên bản cải tiến của HTML với các thẻ mới và các chức năng mới. Dave Ragget đã mua về một phiên bản thú vị với rất nhiều thẻ HTML hấp dẫn và phiên bản này đã được cải tiến rất hay. Nhưng vì nó làm chậm đường truyền của các trình duyệt nên phiên bản này đã bị bỏ đi. Phiên bản HTML 3.2 là phiên bản mạnh nhất của sê-ri này và trước khi được tung ra, nó được duyệt bởi W3C và bởi các nhà cung cấp trình duyệt chóp bu là Netscape và Microsoft.

HTML4

Người ta dành cả năm 1997 để phát triển phiên bản HTML4, một bước tiến triển quan trọng trong những phiên bản cũ. HTML4 có những công cụ có giá trị mang lại thêm nhiều đất sáng tạo cho dân thiết kế web: CSS. Ban đầu CSS cũng không được coi là quan trọng lắm, nhưng đến nay, người ta đánh giá nó cũng quan trọng không kém gì bản thân HTML. Một sự kiện quan trọng nữa là sự phát triển của các trình duyệt: Microsoft ứng dụng hầu như tất cả các thẻ và trình duyệt Internet Explorer được người sử dụng yêu thích hơn ,làm lu mờ Netscape. Vào tháng 4 năm 1998 HTML4 đã được chứng nhận bởi W3C và tương lai trở nên sáng lạng hơn. HTML có một “đôi thủ” gọi là XHTML (Extensible HyperText Markup Language- tạm dịch: ngôn ngữ đánh dấu siêu văn bản mở rộng) và từ năm 1998 đến nay, cuộc chiến vẫn diễn ra ác liệt , nhưng cuối cùng có lợi nhất vẫn là người sử dụng Internet.

HTML5

Vào tháng 1 năm 2008, W3C tung ra bản nháp của HTML5 và thể thượng phong có vẻ nghiêng về HTML (so với XHTML). Phần lớn chúng ta đều biết khả năng của bản HTML4.01- phiên bản gần đây nhất, nhưng những tính năng mới của phiên bản thứ 5 này là gì?

Phiên bản lần này sẽ mang lại nhiều cải tiến và chúng ta nên chuẩn bị tinh thần chờ đợi một vài năm để đưa nó vào hoạt động, tức là cho đến lúc các nhà thiết kế Web làm chủ được các chức năng mới của phiên bản lần này. Một trong những thẻ làm người sử dụng trở nên mê mẩn sẽ là thẻ có chức năng cho phép xem videos mà không cần phải cài đặt Flash hay bất cứ plug-in phụ nào. Các cuộc tranh luận sẽ vẫn còn tiếp tục nhưng chắc chắn sẽ là về các chức năng mới của phiên bản lần này.

Những thẻ thú vị là những thẻ cho phép bạn dễ dàng quản lí dữ liệu hơn như là:

Thẻ article

Aside

Nav

Header

Footer

...

Tự tên của các thẻ này đã nói lên chức năng của chúng, những thẻ HTML mới này sẽ là vũ khí cho các chuyên viên thiết kế web tương lai. Số lượng thẻ trong phiên bản lần này có thể sẽ nhiều hơn phiên bản trước ngay cả khi một số thẻ ở phiên bản trước đã được bỏ đi ở bản HTML5 như các thẻ: : basefont, big, font, s, strike, small, b, I,...

Bản HTML5 có nhiều thẻ nhưng cũng có nhiều vấn đề và rất khó để quyết một cách chắc chắn bởi vì mọi việc còn có thể khác hẳn vào ngày mai. Một thực tế chắc chắn đó là: bởi vì phiên bản lần này được nghiên cứu bởi các tập đoàn và các chuyên viên thiết kế, nên kết quả chỉ có thể là bước nhảy vọt đầy ấn tượng trong dòng chảy của sự phát triển Web, Internet và thiết kế. Trong chuẩn mới thực sự hoàn chỉnh của HTML, một nhân tố vô cùng quan trọng sẽ đóng góp vào vai trò ảnh hưởng của HTML 5, đó là: CSS3.

Tin tốt là những phiên bản mới của CSS đi kèm với HTML 5 sẽ cho phép các nhà thiết kế có thể thỏa sức sáng tạo. Hi vọng rằng không lâu nữa HTML 5 sẽ thực sự đi vào sử dụng và chúng ta sẽ được nhìn thấy các Websites thiết kế bằng HTML 5 trong thời gian gần đây.

2.1.2. Khái niệm về HTML

HTML là viết tắt của cụm từ HyperText Markup Language (Ngôn ngữ đánh dấu siêu văn bản), và đây là ngôn ngữ chính được sử dụng cho việc thiết kế các website. Ngôn ngữ này gây ấn tượng cho người ta bởi sự ngắn gọn nhưng lại được hình thành trong một

quá trình khó phân định. Việc hình thành HTML có mối quan hệ mật thiết với sự phát triển của Internet. Ngày nay, HTML rất được khuyến dùng bởi sự xuất hiện của một chuẩn mới, thực tế là một phiên bản cao hơn của ngôn ngữ này như là một sự phát triển tất yếu HTML5.

Markup : HTML là ngôn ngữ của các thẻ đánh dấu - Tag. Các thẻ này xác định cách thức trình bày đoạn văn bản tương ứng trên màn hình.

Language : HTML là một ngôn ngữ tương tự như các ngôn ngữ lập trình, tuy nhiên đơn giản hơn. Nó có cú pháp chặt chẽ để viết các lệnh thực hiện việc trình diễn văn bản. Các từ khoá có ý nghĩa xác định được cộng đồng Internet thừa nhận và sử dụng. Ví dụ b = bold, ul = unordered list, ...

Text : HTML đầu tiên và trước hết là để trình bày văn bản và dựa trên nền tảng là một văn bản. Các thành phần khác như hình ảnh, âm thanh, hoạt hình ... đều phải chèn vào một đoạn văn bản nào đó.

Hyper : HTML cho phép liên kết nhiều trang văn bản rải rác khắp nơi trên Internet. Nó có tác dụng che dấu sự phức tạp của Internet đối với người sử dụng, có thể đọc mà không cần biết đến văn bản đó nằm ở đâu, xây dựng phức tạp như thế nào. Sự phát triển có tính bùng nổ của Internet trong thời gian vừa qua một phần lớn là nhờ vào WWW.

2.1.3. Các đặc điểm chính của HTML

Độc lập với phần cứng và phần mềm

HTML độc lập với phần cứng và phần mềm. Một văn bản HTML được viết bằng một phần mềm soạn thảo cụ thể bất kỳ, trên một máy cụ thể nào đó đều có thể đọc được trên bất kỳ một hệ thống tương thích nào.

Điều này có nghĩa là các tệp tin siêu văn bản có thể được bộ duyệt hiển thị trên MAC hay PC tùy ý mà không phải sửa chữa thay đổi gì. Sở dĩ có được tính chất này là vì các thẻ chỉ diễn đạt yêu cầu cần phải làm gì chứ không cụ thể cần làm như thế nào.

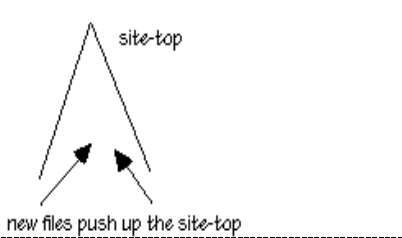
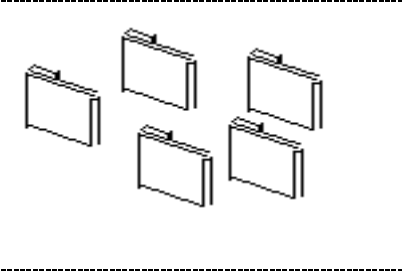
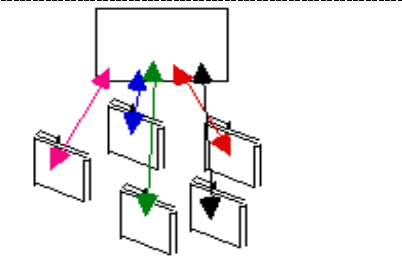
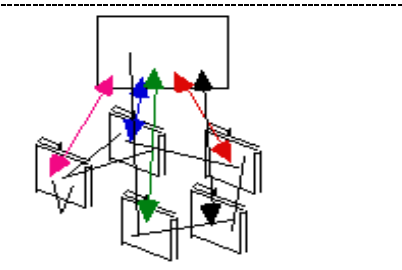
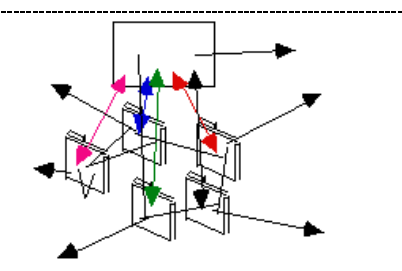
Cũng vì lẽ đó mà chúng ta không thể chắc chắn trang tài liệu siêu văn bản của Chúng ta sẽ hiện lên màn hình chính xác là như thế nào vì còn tùy theo bộ duyệt thể hiện yêu cầu đó ra sao.

Trong thực tiễn, HTML chỉ thực sự độc lập đối với phần cứng, chứ không hoàn toàn độc lập đối với phần mềm. Chỉ phần cốt lõi là chuẩn hoá, còn các phần mở rộng do từng nhà phát triển xây dựng thì không hoàn toàn tương thích nhau.

Độc lập với khái niệm trang và tuần tự các trang

Một tính chất nữa là HTML độc lập với khái niệm trang. Văn bản được trình bày tùy theo kích thước của cửa sổ hiển thị: của sổ rộng bề ngang thì sẽ thu ngắn hơn, của sổ hẹp bề ngang thì sẽ được kéo dài ra để hiển thị cho hết nội dung. Độ dài của văn bản HTML thực sự không bị hạn chế.

Website và trang chủ - homepage

	<p>Một website là một bó các trang web liên kết với nhau và liên kết với các trang ở bên ngoài chẳng chịt như mạng nhện. Hàng triệu Website liên kết với nhau tạo thành World Wide Web – WWW</p>
	<p>Có thể tưởng tượng như một ngọn núi nhỏ các trang web mỗi ngày một cao thêm , được phát triển bằng cách thêm vào nhiều trang web lý thú khác nữa</p>
	<p>Có thể minh họa hình ảnh của một website như trong hình vẽ bên</p>
	<p>Trang chủ hay trang chính - 'home page' có thể hiểu là cửa chính - 'front door' để thâm nhập vào kho thông tin liên kết chằng chịt ấy.</p>
	<p>Vậy home page là trang web mà bộ duyệt sẽ mở ra đầu tiên mỗi khi người dùng bắt đầu thăm website.</p>

2.2. Cấu trúc trang HTML

Cũng giống như việc xây dựng các chương trình bằng các ngôn ngữ lập trình khác. Việc tạo ra một tài liệu HTML cũng phải tuân thủ một cấu trúc chương trình nhất định. Mọi tài liệu HTML đều phải tuân thủ cấu trúc khung như sau:

<HTML>

<HEAD>

<TITLE> Tiêu đề của tài liệu</TITLE>

</HEAD>

<BODY>

Nội dung của tài liệu html được viết ở đây

</BODY>

</HTML>

Một tài liệu HTML gồm 3 phần cơ bản:

Phần HTML: Mọi tài liệu HTML phải bắt đầu bằng thẻ mở HTML <HTML> và kết thúc bằng thẻ đóng HTML </HTML>

<HTML> </HTML>

Thẻ HTML báo cho trình duyệt biết nội dung giữa hai thẻ này là một tài liệu HTML

Phần đầu: Phần đầu bắt đầu bằng thẻ <HEAD> và kết thúc bởi thẻ </HEAD>. Phần này chứa tiêu đề mà được hiển thị trên thanh điều hướng của trang Web. Tiêu đề nằm trong thẻ TITLE, bắt đầu bằng thẻ <TITLE> và kết thúc là thẻ </TITLE>. Tiêu đề là phần khá quan trọng. Các mốc được dùng để đánh dấu một web site. Trình duyệt sử dụng tiêu đề để lưu trữ các mốc này. Do đó, khi người dùng tìm kiếm thông tin, tiêu đề của trang Web cung cấp từ khóa chính yếu cho việc tìm kiếm.

Phần thân: Phần này nằm sau phần tiêu đề. Phần thân bao gồm văn bản, hình ảnh và các liên kết mà Chúng ta muốn hiển thị trên trang web của mình. Phần thân bắt đầu bằng thẻ <BODY> và kết thúc bằng thẻ </BODY>

Ví dụ 2.1: Sử dụng một phần mềm soạn thảo như Notepad rồi gõ đoạn chương trình sau vào và ghi tên file có phần mở rộng là .htm hoặc .html ta sẽ tạo ra một trang web. Sử dụng trình duyệt web để mở trang web trên ta sẽ nhận được nội dung của trang.

<html>

<head>

<title>Ví dụ 2.1</title>

</head>

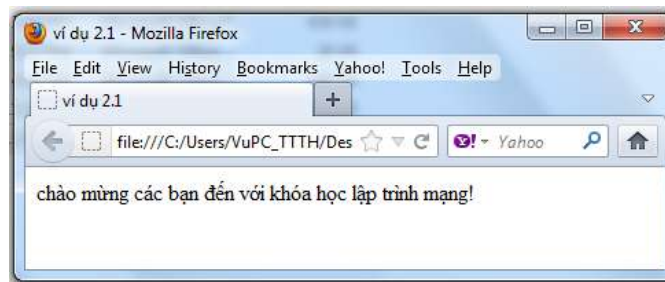
<body>

chào mừng các bạn đến với khóa học lập trình mạng!

</body>

</html>

Mở nội dung của trang web bằng trình duyệt web ta sẽ có kết quả như hình 2.1 dưới đây.



Hình 2.1. Kết quả của ví dụ 2.1

2.3. Thẻ và cấu trúc thẻ

Các lệnh HTML được gọi là các thẻ (Tags). Các thẻ này được dùng để điều khiển nội dung và hình thức trình bày của tài liệu HTML. Mỗi thẻ của HTML được tạo ra bao gồm một từ khóa được bao trong dấu đóng mở ngoặc nhọn như sau: <Từ khóa>

HTML có 2 loại thẻ đó là Thẻ mở (“<>”) và thẻ đóng (“</>”), chỉ ra sự bắt đầu và kết thúc của một lệnh HTML.

Hầu hết các lệnh thẻ hiện bằng một cặp hai thẻ : thẻ mở (<KEYWORD>) và thẻ đóng (</KEYWORD>). Dấu gạch chéo (“/”) kí hiệu thẻ đóng . Lệnh sẽ tác động vào đoạn văn bản nằm giữa hai thẻ.

<KEYWORD> Đoạn văn bản chịu tác động của lệnh</KEYWORD>

Một số thẻ không có cặp, chúng được gọi là các thẻ **rỗng** hay thẻ **đơn**. Chỉ có thẻ mở <KEYWORD> mà thôi.

Ví dụ: Thẻ
 là thẻ có ý nghĩa là xuống dòng thẻ này không có thẻ đóng

Nhiều thẻ có kèm các thuộc tính (attribute), cung cấp thêm các tham số chi tiết hơn cho việc thực hiện lệnh . Các thuộc tính được chia làm hai loại : thuộc tính bắt buộc và thuộc tính không bắt buộc hay tùy chọn .

Một thuộc tính là bắt buộc nếu như phải có nó thì thẻ lệnh mới thực hiện được. Ví dụ, để chèn một hình ảnh vào trang tài liệu ta dùng thẻ (Image). Tuy nhiên, cần chỉ rõ cái ảnh nào sẽ được dán vào đây. Điều này được thiết lập bằng thuộc tính SRC="địa chỉ của tệp ảnh". Thuộc tính SRC là bắt buộc phải có đối với thẻ .

Các quy tắc chung

Một số điều cần lưu ý khi soạn thảo siêu văn bản bằng HTML:

- *Nhiều dấu cách liền nhau cũng chỉ có tác dụng như một dấu cách. Bạn phải sử dụng thẻ để thể hiện nhiều dấu giãn cách liền nhau.*
- *Gõ Enter để xuống dòng được xem như một dấu cách , để xuống hàng thì chúng ta phải sử dụng thẻ tương ứng*
- *Có thể viết tên thẻ không phân biệt chữ in thường và in hoa.*
- *Vì các kí tự dấu lớn hơn ">," dấu nhỏ hơn "<" đã được dùng làm thẻ đánh dấu , do đó để hiển thị các kí tự này HTML quy định cách viết: > <*
- *Nói chung quy tắc viết các kí tự đặc biệt trong HTML là tên quy định của kí tự nằm giữa dấu ampersand & và dấu chấm phẩy; ' & tên quy định;*

Có thể chèn các dòng bình luận , chú thích... vào trang mã nguồn bằng cách đặt giữa cặp dấu chú thích <!-- và -->. Trình duyệt sẽ bỏ qua không xét đến phần mã nằm giữa cặp dấu đó:

<!-- Dòng chú thích -->

Các phần tử HTML

Một tài liệu HTML tạo nên từ nhiều thành phần HTML. Một thành phần HTML được đánh dấu bằng một cặp thẻ mở và thẻ đóng.

Các thành phần HTML có thể cấu trúc phân cấp hình cây, thành phần "mẹ" chứa nhiều thành phần "con" khác lồng bên trong nó.

Có thành phần rỗng, chỉ có thẻ mở.

Để soạn thảo tài liệu HTML đúng cú pháp, cần nắm vững cấu trúc của từng thành phần HTML.

Không giống như khi lập trình, nếu bạn mắc lỗi cú pháp HTML sẽ không có một thông báo lỗi nào báo cho biết mà trình duyệt sẽ hiểu lầm và trình bày trang Web không như ý muốn mà thôi.

2.4. Thẻ <META>

Phần đầu cũng chứa phần tử META. Phần tử này cung cấp thông tin về trang web của Chúng ta. Nó gồm tên tác giả, tên phần mềm dùng để viết trang đó, tên công ty, thông tin liên lạc Phần tử META sử dụng sự kết hợp giữa thuộc tính và giá trị.

Ví dụ 2.2, để chỉ Graham Browne là tác giả, người ta sử dụng phần tử META như sau:

```
<META name="Author" content="Graham Browne">
```

Tác giả của tài liệu là "Graham Browne"

Thuộc tính http-equiv có thể được sử dụng để thay thế thuộc tính name. Máy chủ HTTP sử dụng thuộc tính này để tạo ra một đầu đáp ứng HTTP (HTTP response header).

Đầu đáp ứng được truyền đến trình duyệt để nhận dạng dữ liệu. Nếu trình duyệt hiểu được đầu đáp ứng này, nó sẽ tiến hành các hành động đặc biệt đối với đầu đáp ứng đó.

Ví dụ 2.3, <META http-equiv="Expires" content="Mon, 15 Sep 2003 14:25:27 GMT">

sẽ sinh ra một đầu đáp ứng HTTP như sau:

```
Expires: Mon, 15 Sep 2003 14:25:27 GMT
```

Do vậy, nếu tài liệu đã lưu lại, HTTP sẽ biết khi nào truy xuất một bản sao của tài liệu tương ứng.

2.5. Thẻ <Head>

Thẻ Head là thẻ bắt đầu của một tài liệu HTML, cặp thẻ <Head> ...</Head> sẽ quy định phần đầu của một tài liệu HTML, thành phần của Head bao gồm các thông tin chung hay còn được gọi là meta – information về tài liệu.

Theo như chuẩn HTML thì nội dung nằm trong cặp thẻ này sẽ không được hiển thị ra màn hình trình duyệt, tuy nhiên chúng chỉ cho phép một số thẻ nhất định như: <base>, <link>, <meta>, <title>, <style>, và <script>. Trong trường hợp chúng ta chèn nội dung là những thẻ khác chẳng hạn:

```
<head>
<p>This is some text</p>
</head>
```

Thì hầu hết các trình duyệt lại vẫn hiển thị nội dung nằm trong cặp thẻ P.

2.6. Sử dụng ký tự đặc biệt trong tài liệu HTML

Chúng ta có thể chèn các ký tự đặc biệt vào văn bản của tài liệu HTML. Để đảm bảo trình duyệt không nhầm chúng với thẻ HTML, Chúng ta phải gán mã định dạng cho các ký tự đặc biệt này. Bảng sau đây mô tả 5 ký tự đặc biệt hay dùng

Ký tự	Cách viết
<	<
>	>
“	"
Ký tự trắng	
&	&

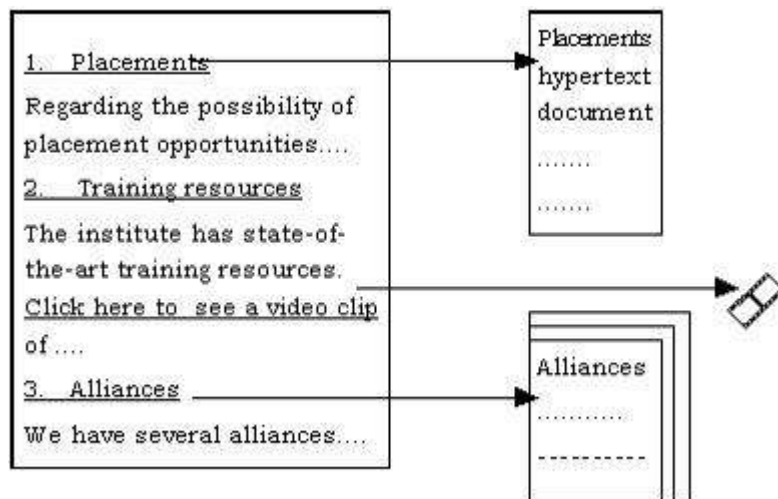
2.7. Sử dụng các siêu liên kết

2.7.1. Giới thiệu siêu liên kết và URL

Các liên kết siêu văn bản trong một tài liệu HTML là để tham chiếu đến một tài liệu khác hay một phần tài liệu khác nằm trong tài liệu đó.

Khả năng chính của HTML là hỗ trợ siêu liên kết. Một siêu liên kết, hay nói ngắn gọn là một liên kết, là sự kết nối đến tài liệu hay file khác (đồ họa, âm thanh, video) hoặc ngay cả đến một phần khác trong cùng tài liệu đó. Khi kích vào siêu liên kết, người dùng được đưa đến địa chỉ URL mà chúng ta chỉ rõ trong liên kết, địa chỉ đó có thể là:

- ✓ Một phần khác trong cùng tài liệu
- ✓ Một tài liệu khác
- ✓ Một phần trong tài liệu khác
- ✓ Các file khác – hình ảnh, âm thanh, trích đoạn video
- ✓ Vị trí hoặc máy chủ khác



Hình 2.2. Mô tả liên kết trên website

Các liên kết có thể là liên kết trong hoặc liên kết ngoài

Liên kết trong là liên kết kết nối đến các phần trong cùng tài liệu hoặc cùng một web site

Liên kết ngoài là liên kết kết nối đến các trang trên các web site khác hoặc máy chủ khác.

Để tạo siêu liên kết, chúng ta cần phải xác định hai thành phần:

1. Địa chỉ đầy đủ hoặc URL của file được kết nối
2. Điểm nóng cung cấp cho liên kết. Điểm nóng này có thể là một dòng văn bản, thậm chí là một ảnh.

Khi người dùng kích vào điểm nóng, trình duyệt đọc địa chỉ được chỉ ra trong URL và “nhảy” đến vị trí mới

Liên kết siêu văn bản gồm có ba phần: giao thức, cái neo và địa chỉ hay URL

Để tạo một mối liên kết thì việc đầu tiên cần làm là thiết lập cái neo. Thẻ tạo neo là <A>... (Anchor). Thẻ này có nhiều thuộc tính bắt buộc phải xác định rõ, ta sẽ tìm hiểu kỹ hơn ở phần sau.

URL

URL là một chuỗi cung cấp địa chỉ Internet của website hay tài nguyên trên World Wide Web. Để liên kết đến một tài liệu khác, ta cần phải biết địa chỉ URL của tài liệu đích. URL Là chữ viết tắt của "Uniform Resource Locator", dùng để chỉ tài nguyên trên Internet. Sức mạnh của web là khả năng tạo ra những liên kết siêu văn bản đến các thông tin liên quan. Những thông tin này có khi là những trang web khác, những hình ảnh, âm thanh... Những liên kết này thường được biểu diễn bằng những chữ màu xanh có gạch dưới được gọi là anchor. Các URL có thể được truy xuất thông qua một trình duyệt (Browser) như IE hay Netscape.

URL được dùng để xác định địa chỉ của một tài liệu (hoặc dữ liệu khác) trên World Wide Web. Một địa chỉ đầy đủ sẽ như sau:

http://gocongnghe.net/html/trangtruoc.htm và tuân theo cú pháp sau:

Scheme://host.domain:port/path/filename

Cụ thể cú pháp của từng thành phần Chúng ta có thể tham khảo trong Chương 1

URL cũng có thể tham chiếu đến một vị trí trong một tài nguyên. Ví dụ, Chúng ta có thể tạo liên kết đến một chủ đề trong cùng một tài liệu. Trong trường hợp đó, định danh đoạn được sử dụng ở phần cuối của URL

Định dạng là, Scheme://host.domain:port/path/filename #định danh đoạn

Có hai dạng URL:

URL tuyệt đối – là địa chỉ Internet đầy đủ của trang hoặc file, bao gồm giao thức, vị trí mạng, đường dẫn tùy chọn và tên file. Ví dụ, http://www.microsoft.com là một địa chỉ URL tuyệt đối.

URL tương đối – là một URL có một hoặc nhiều phần bị thiếu. Trình duyệt lấy thông tin bị thiếu từ trang chứa URL đó. Ví dụ, nếu giao thức bị thiếu, trình duyệt sử dụng giao thức của trang hiện thời.

2.7.2. Sử dụng siêu liên kết

Để tạo liên kết cho một tài liệu chúng ta sử dụng thẻ A (Anchor) cú pháp như sau:

` Siêu văn bản `

Trong đó thuộc tính,

- ✓ **Name** – Là một tên do người dùng định nghĩa, được dùng để đánh dấu điểm neo, sử dụng cho việc liên kết từ một nơi nào đó đến vị trí này.
- ✓ **Href** – chỉ ra địa chỉ url của tài liệu sẽ được liên kết đến
- ✓ **Siêu văn bản** – Đây là văn bản hay hình ảnh mà người dùng cần nhấp vào để kích hoạt liên kết
- ✓ **Target** – Chỉ ra nơi sẽ hiển thị tài liệu được liên kết đến, Windows ở đây có thể là tên một cửa sổ hoặc có thể là `_blank` nếu cửa sổ đó là cửa sổ mới, `_parent` cửa sổ cha của cửa sổ hiện hành, `_self` cửa sổ hiện hành, `_top` cửa sổ trên cùng Theo mặc định tài liệu mới sẽ liên kết đến sẽ được mở ở cửa sổ hiện hành.

Liên kết đến tài liệu khác

Để liên kết từ một tài liệu này đến một tài liệu khác ta thực hiện bằng cách khai báo thuộc tính Href chỉ đến url của tài liệu cần liên kết đến.

Ví dụ 2.4: Giả sử có hai tài liệu HTML trên đĩa cứng cục bộ, Doc1.htm và Doc2.htm. Để tạo một liên kết từ Doc1.html đến Doc2.htm và ngược lại ta làm như sau:

Doc1.htm ta viết như sau:

```
<HTML>
  <HEAD>
    <TITLE> Tạo liên kết</TITLE>
  </HEAD>
  <BODY>
```

`<P> Tạo liên kết trong tài liệu.`

`kích chuột vào đây để mở tài liệu 2`

`</BODY>`

`</HTML>`

Doc2.htm ta viết như sau:

`<HTML>`

`<HEAD>`

`<TITLE> Tài liệu 2</TITLE>`

`</HEAD>`

`<BODY>`

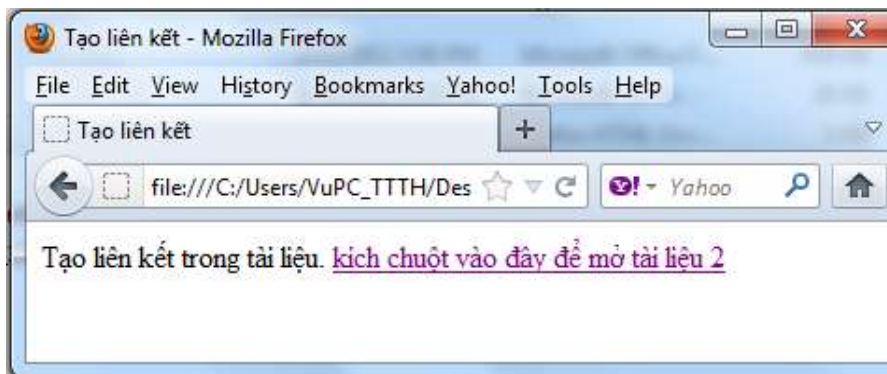
`<P> Đây là tài liệu 2 trang này sẽ hiển thị khi kích chuột vào liên kết trong tài liệu 1
`

`Trở lại tài liệu 1`

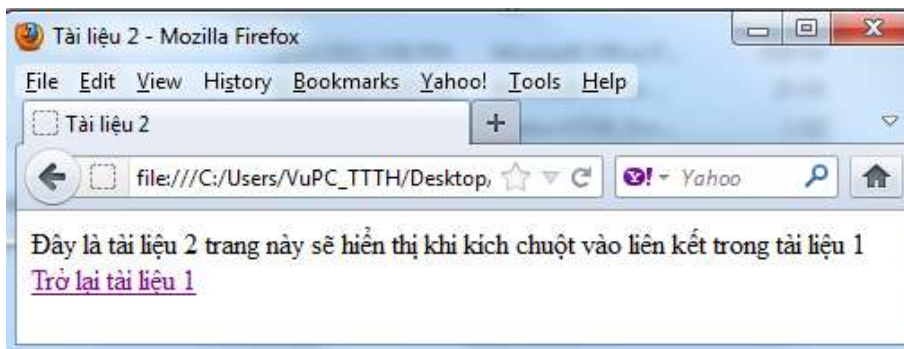
`</BODY>`

`</HTML>`

Ghi chương trình và chạy ta sẽ được như sau:



Hình 2.3. Kết quả chạy file doc1.htm



Hình 2.4. Kết quả khi kích liên kết – File doc2.htm được mở

Liên kết đến các phần trong cùng một tài liệu

Thẻ neo `<A>` (anchor) cũng được sử dụng để người dùng “nhảy” đến những phần khác nhau của một tài liệu. Ví dụ, Chúng ta có thể hiển thị nội dung của trang Web như

một loạt các liên kết. Khi người dùng kích vào một đề tài nào đó thì các chi tiết nằm ở một phần khác của tài liệu được hiển thị.

Kiểu liên kết này gọi là “named anchor” bởi vì thuộc tính NAME được sử dụng để tạo các liên kết này.

```
<A NAME = “marker”>Tên chủ đề</A>
```

Chúng ta không phải sử dụng bất kỳ văn bản nào để đánh dấu điểm neo

Để dùng, ta sử dụng dấu # (marker) trong đường dẫn url của thuộc tính HREF như sau

```
<A HREF= “#marker”>Tên chủ đề</A>
```

Dấu # ở trước tên của siêu liên kết để báo cho trình duyệt biết rằng liên kết này liên kết đến một điểm được đặt tên trong tài liệu. Khi không có tài liệu nào được chỉ ra trước ký tự #, trình duyệt hiểu rằng liên kết này nằm trong cùng tài liệu.

Ví dụ 2.5: Giả sử chúng ta tạo ra 3 mục lục ở đầu trang gồm: **Internet, Introduction to HTML, Unity and Variety** khi ta kích chuột vào từng mục thì nó sẽ nhảy đến nội dung của từng mục đó trong trang.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Using Links</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<A HREF = #Internet>Internet</A><BR><BR>
```

```
<A HREF = #HTML>Introduction to HTML</A><BR><BR>
```

```
<A HREF = #Consistency>Unity and Variety</A><BR><BR>
```

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

```
<P> The Internet is a network of networks. That is, computer networks are linked to other networks, spanning countries and today the globe. The TCP/IP transfer protocol provides the bindings that connect all these computers the world over.
```

```
</P>
```

```
<A name = HTML>Introduction to HTML</A>
```

```
<BR>
```

```
<P> Hyper Text Markup Language is the standard language that the Web uses for creating and recognizing documents. Although not a subset of, it is loosely related to the Standard Generalized Markup Language (SGML). SGML is a method for representing document formatting languages.
```

</P>

Unity and Variety

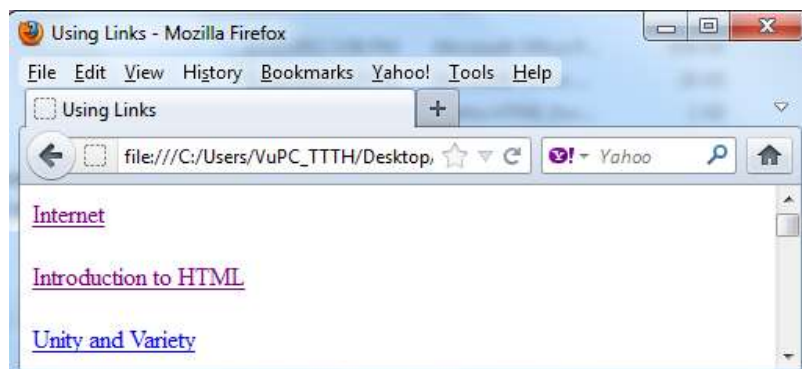
<P> A basic rule is that of unity and variety. That is, everything should fit together as a unit, but at the same time there is enough variety to keep things interesting. Consistency creates and reinforces the unique identity of a site. Colors, fonts, column layout and other design elements should be consistent throughout every section of the site.

</P>

</BODY>

</HTML>

Khi chạy chương trình ta được kết quả:



Hình 2.5. Kết quả chạy ví dụ 2.5

Khi kích chuột vào Internet thì nó sẽ nhảy đến mục tương ứng như sau:



Hình 2.6. Kết quả khi kích vào liên kết

Liên kết đến một điểm xác định ở một tài liệu khác

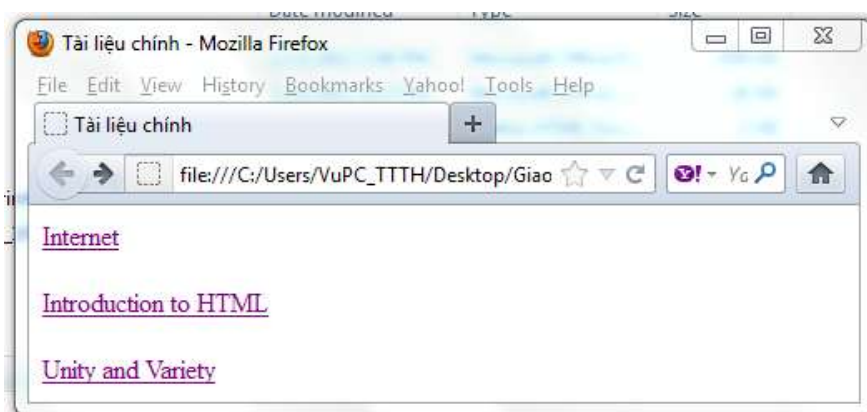
Bây giờ chúng ta đã biết cách sử dụng các vạch dấu trong cùng một tài liệu, hãy thử “nhảy” đến một vị trí trên một tài liệu khác.

Để “nhảy” đến một điểm trên tài liệu khác, chúng ta cần phải chỉ ra tên của tài liệu khi chúng ta tạo vạch dấu. Trước tiên trình duyệt sẽ đọc tên tài liệu và mở tài liệu đó. Sau đó nó sẽ đọc vạch dấu và di chuyển đến điểm được đánh dấu.

Ví dụ 2.6 tạo file htm như sau:

```
<HTML>
  <HEAD>
    <TITLE>Tài liệu chính</TITLE>
  </HEAD>
  <BODY>
    <A HREF= vidu2.5.htm#Internet>Internet</A>
    <BR>
    <BR>
    <A HREF= vidu2.5.htm #HTML>Introduction to HTML</A>
    <BR>
    <BR>
    <A HREF= vidu2.5.htm #Consistency>Unity and Variety</A>
  </BODY>
</HTML>
```

Khi chạy nó ta được kết quả sau



Hình 2.7. Kết quả chạy ví dụ 2.6

Khi kích chuột vào mục Internet ở trên nó sẽ nhảy đến vidu2.5.htm mục Internet.



Hình 2.8. Kết quả khi kích vào liên kết từ trang ví dụ 2.6

Sử dụng E-mail

Nếu muốn người sử dụng gửi được e-mail, chúng ta có thể đưa một đặc tính vào trong trang Web và cho phép họ gửi e-mail từ trình duyệt. Tất cả những gì chúng ta cần làm là chèn giá trị mailto vào trong thẻ liên kết.

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

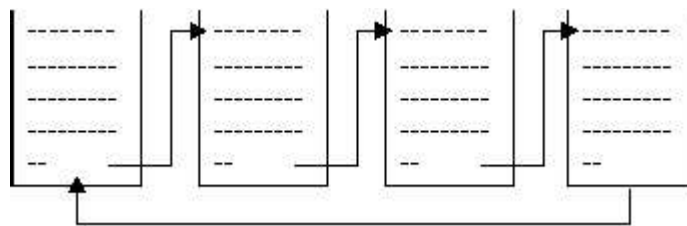
2.7.3. Điều hướng quanh Web site

Dù web site có lời cuốn đến đâu, nếu nó không có một lược đồ điều hướng đơn giản thì người dùng cũng sẽ mất phương hướng ngay. Một số trong số đó có thể sẽ không bao giờ quay lại. Lược đồ điều hướng trong mỗi site là khác nhau. Tuy nhiên có một số nguyên tắc cơ bản cần nhớ:

- ✓ Xác định nội dung của web site
- ✓ Tạo một lược đồ điều hướng để giúp người dùng đi đến phần cần đến một cách nhanh chóng
- ✓ Cung cấp các chức năng tìm kiếm thông tin

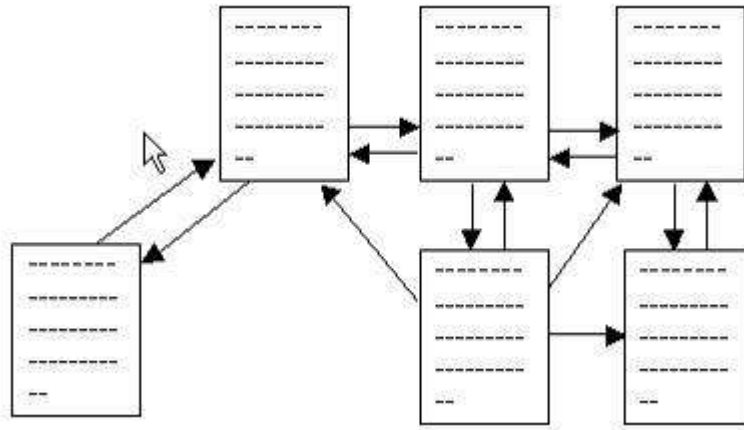
Có nhiều cách để tổ chức một web site

Trình bày tuyến tính: Cách này theo cấu trúc tuyến tính. Có một chuỗi liên kết liên tục giữa các trang. Mỗi trang có liên kết đến trang trước và trang sau. Trang cuối có liên kết đến trang đầu. Định dạng này được dùng khi chúng ta muốn trình bày thông tin liên tục. Ví dụ, các chương trong cuốn sách hoặc các slide của một bài trình bày mà yêu cầu người dùng đọc thông tin theo trình tự liên tiếp nhau.



Hình 2.9. Trình bày tuyến tính

Trình bày theo phân cấp: Đây là cách trình bày thông thường nhất được sử dụng trong thiết kế Web. Trang chủ liên kết với nhiều trang khác. Người dùng có thể chọn một liên kết và “nhảy” đến trang cần đến. Mỗi trang đều có liên kết về trang chủ.



Hình 2.10. Trình bày theo phân cấp

Bản đồ ảnh: Một số người thấy cách trình bày trực quan thì dễ hiểu hơn. Vì vậy, bản đồ ảnh hay bản đồ site được đưa vào các trang chủ. Khi người dùng kích vào các điểm nóng thì tài liệu liên quan được hiển thị.

Khi đã quyết định một lược đồ điều hướng, đây là lúc để thiết kế tài liệu. Có một số nguyên tắc cần ghi nhớ:

- ✓ Siêu liên kết nên rõ ràng. Từ nên được gạch chân và có màu xanh, bởi vì trên Web, màu xanh được quy ước là siêu liên kết. Siêu liên kết thường xuất hiện ở cuối trang. Nếu chúng xuất hiện ở giữa đoạn văn bản, có thể làm người đọc không tập trung vào đề tài.

- ✓ Các trang con – nên có một thanh menu hoặc thanh điều hướng, nếu không người đọc có thể không thấy các trang chính. Nếu người dùng kích vào một biểu tượng bất kỳ trên thanh điều hướng thì họ sẽ nhảy từ trang hiện thời đến trang mới. Luôn có một bảng mục lục để người dùng nhảy sang một đề tài nào đó. Bảng mục lục (TOC) được hiển thị ở cuối trang. Người dùng có thể chọn một đề tài nào đó để nhảy đến trang cần đến.

- ✓ Luôn có nút “Back” và “Forward” trong trang để người dùng có thể điều hướng đến trang cần đến. Trình duyệt có cung cấp các chọn lựa này, nhưng nó sử dụng các trang trong danh sách history mà có thể không phải là các trang trong Web site.

2.8. Các thẻ mức đoạn

Trong phần này chúng ta sẽ đi tìm hiểu các thẻ dùng để trình bày một đoạn văn bản trong một tài liệu HTML. Có nhiều thẻ được sử dụng nhưng trong tài liệu này chỉ trình bày một số thẻ trình bày chính: định dạng phần địa chỉ (<ADDRESS>), đoạn văn bản (<P>), xuống dòng (
), căn chính giữa (<CENTER>), đường kẻ ngang (<HR>), đoạn văn bản đã định dạng sẵn (<PRE>), trích dẫn nguồn tài liệu (<BLOCKQUOTE>)

2.8.1. Thẻ P

Để phân đoạn và định dạng đoạn văn bản trong ngôn ngữ HTML người ta sử dụng thẻ P(Paragraph)

Cú pháp của thẻ như sau:

<p align = “kiểu căn lề” > đoạn văn bản </p>

Trong đó:

Align là thuộc tính dùng để định dạng căn lề cho đoạn văn bản, kiểu căn lề là một trong các kiểu sau:

- Justify là căn lề hai bên
- Left là căn lề trái
- Center là căn lề giữa
- Right là căn lề phải

2.8.2. Các thẻ định dạng đề mục

HTML cung cấp 6 thẻ dùng để định dạng các đề mục cho một văn bản hiển thị lên trên trang web theo cấp độ từ lớn đến bé tương ứng với các thẻ cho trong bảng sau:

Tên loại đề mục	Thẻ và cú pháp của thẻ	Chú thích
Đề mục mức 1	<H1 align = “kiểu căn lề”>...</H1>	Mức cao nhất
Đề mục mức 2	<H2 align = “kiểu căn lề”>...</H2>	
Đề mục mức 3	<H3 align = “kiểu căn lề”>...</H3>	
Đề mục mức 4	<H4 align = “kiểu căn lề”>...</H4>	
Đề mục mức 5	<H5 align = “kiểu căn lề”>...</H5>	
Đề mục mức 6	<H6 align = “kiểu căn lề”>...</H6>	Mức thấp nhất

Cũng giống như thẻ P ở mục trên các thẻ định dạng đề mục cũng có thuộc tính align dùng để căn lề cho các đề mục. Trong đó kiểu căn lề là một trong các kiểu: Justify, left, right, hoặc center.

Ví dụ 2.7: tạo một trang web như sau

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Các thẻ định dạng đề mục</title>
</head>
<body>
<H1>Đây là đề mục cấp 1</H1>
<H2>Đây là đề mục cấp 2</H2>
<H3>Đây là đề mục cấp 3</H3>
<H4>Đây là đề mục cấp 4</H4>
<H5>Đây là đề mục cấp 5</H5>
<H6>Đây là đề mục cấp 6</H6>
</body>
```

</html>

Khi chạy ta được kết quả như sau:



Hình 2.11. Kết quả chạy file ví dụ 2.7

2.8.3. Thẻ xuống dòng BR

Nếu chúng ta không chèn vào một dòng trắng mà đơn giản chỉ là xuống dòng thì chúng ta có thể sử dụng thẻ
 (Break) cú pháp của thẻ như sau:

Thẻ
 là một thẻ đặc biệt chỉ có thẻ mở và không có thẻ đóng, ý nghĩa của thẻ là xuống dòng về đầu dòng.

2.8.4. Thẻ ADDRESS

Thẻ địa chỉ - Address thẻ này dùng để định dạng một đoạn văn bản là địa chỉ, danh thiếp và tác giả, thường đặt ở đầu hay cuối tài liệu. khi dùng thẻ này phong chữ của toàn đoạn sẽ in nghiêng cú pháp như sau:

<ADDRESS> văn bản </ADDRESS>

Ví dụ 2.8:

<html>

<head>

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

<title>Các thẻ định dạng đề mục</title>

</head>

<body>

<address>

NAMHA GARMENT STOCK COMPANY

Địa chỉ: 510 Truong Chinh Str., Nam dinh, Vietnam.

Điện thoại: 84-0350-3649563/3649329/3635324/3649326

Fax: 84-0350-3644767

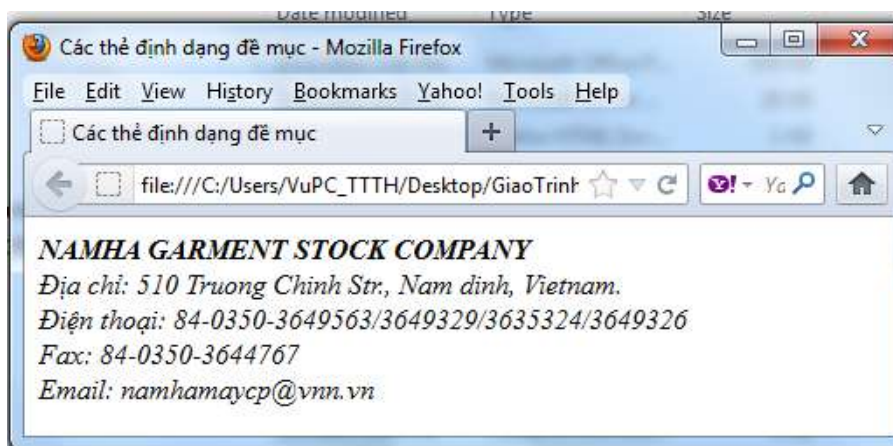
Email: namhamaycp@vnn.vn

</address>

</body>

</html>

Kết quả thu được:



Hình 2.12. Kết quả chạy ví dụ 2.8

2.8.5. Thẻ BLOCKQUOTE

Để định dạng đoạn văn bản căn lề lùi vào cả hai bên, lề trái và lề phải HTML hỗ trợ một thẻ gọi là thẻ BLOCKQUOTE cú pháp của thẻ như sau:

<BLOCKQUOTE> Đoạn văn bản</BLOCKQUOTE>

2.8.6. Thẻ PRE

Thẻ định kiểu ký tự định dạng trước Pre là một thẻ của HTML cho phép chúng ta có thể bảo toàn các dấu cách và dấu xuống dòng trong trang văn bản HTML y như khi chúng ta gõ vào trong quá trình soạn thảo. Điều đó có nghĩa là chúng ta không nên sử dụng các thẻ
 cũng như <P> trong đoạn ký tự định dạng trước.

Browser sử dụng phông chữ bề rộng cố định (fixed-width font) để hiển thị nội dung trong phần ký tự định dạng trước. Thông thường, đó là phông hệ thống đang được sử dụng trong máy tính. Kiểu ký tự định dạng trước rất có ích trong trường hợp dấu cách là không thể thiếu được, chẳng hạn như khi thể hiện một đoạn mã chương trình.

Giới hạn đoạn văn bản đã được định dạng sẵn (pre-formatted) cần được thể hiện bằng phông chữ có độ rộng ký tự không đổi (do phải thẳng cột). Nếu không có thuộc tính WIDTH đi cùng thì bề rộng mặc định là 80 ký tự/dòng. Bề rộng 40,80 và 132 được thể hiện tối ưu, còn các bề rộng khác có thể được làm tròn.

Cú pháp: **<PRE width = n>đoạn văn bản</PRE>**

2.9. Thẻ khối SPAN, DIV

Có những trường hợp chúng ta muốn chia văn bản trong một trang web thành những khối thông tin logic. Chúng ta cũng có thể áp dụng những thuộc tính thông thường cho toàn bộ khối. Thẻ DIV và SPAN được sử dụng để nhóm nội dung lại với nhau.

Thẻ DIV dùng để chia tài liệu thành các thành phần có liên quan với nhau. Thẻ SPAN dùng để chỉ một khoảng các ký tự. Thẻ SPAN dùng để định nghĩa nội dung trong dòng (inline) còn thẻ DIV dùng để định nghĩa nội dung mức khối (block-level)

Cú pháp của các thẻ như sau:

<DIV> Nội dung </DIV>

** Văn bản **

2.10. Các thẻ định dạng ký tự thường dùng

Với HTML chúng ta không chỉ định dạng cho từng đoạn văn bản hay các đoạn văn bản khác nhau mà HTML còn cung cấp một loạt các thẻ cho phép chúng ta định dạng đến từng ký tự cho văn bản sau đây là một số thẻ định dạng ký tự thường dùng.

STT	Tên thẻ	Cú pháp	Ý nghĩa
1	Bold	 ... 	Văn bản nằm giữa cặp thẻ này sẽ In đậm
2	Italics	<I> ... </I>	Văn bản nằm giữa cặp thẻ này sẽ In nghiêng
3	Underline	<U> ... </U>	Văn bản nằm giữa cặp thẻ này sẽ Gạch chân
4	Emphasis	 ... 	Văn bản nằm giữa cặp thẻ này sẽ In nghiêng
5	Strong	...	Văn bản nằm giữa cặp thẻ này sẽ Chữ đậm
6	Sup	<SUP> ... </SUP>	Văn bản nằm giữa cặp thẻ này sẽ định dạng chỉ số trên
7	Sub	<SUB> ... </SUB>	Văn bản nằm giữa cặp thẻ này sẽ định dạng là chỉ số dưới
8	Hr	<HR WIDTH=n% SIZE=n ALIGN=LEFT RIGHT NOSHADE>	Tạo ra đường kẻ nằm ngang: với thuộc tính width là độ rộng của đường, Size là cơ hay bề giấy của đường, align là căn lề cho đường.
9	Left	<LEFT> ...</LEFT>	Văn bản nằm giữa cặp thẻ này sẽ căn lề bên trái
10	Right	<RIGHT>...</RIGHT>	Văn bản nằm giữa cặp thẻ này sẽ căn lề

			bên phải
11	Center	<CENTER>...</CENTER>	Văn bản nằm giữa cặp thẻ này sẽ căn lề giữa
12	Strick	<S>...</S>	Văn bản nằm giữa cặp thẻ này sẽ có đường gạch ngang

2.11. Danh sách

Việc liệt kê danh sách sẽ làm cho văn bản trở nên sáng sủa và dễ theo dõi. Trong HTML có một số thẻ cho phép chúng ta tạo ra danh sách cho một tài liệu như danh sách đánh số thứ tự, danh sách không đánh số thứ tự, danh sách định nghĩa. Sau đây chúng ta sẽ đi tìm hiểu các thẻ cho phép chúng ta định dạng các loại danh sách này.

2.11.1. Danh sách không thứ tự

Danh sách không có thứ tự thường được liệt kê các mục dưới dạng có các bullet ở đầu như dấu -, dấu +, hình tròn, hình vuông hay một hình ảnh hoặc một ký hiệu biểu tượng nào đó.

Trong HTML có thẻ UL cho phép chúng ta định dạng các mục trong một tài liệu là một danh sách không có thứ tự. Thẻ này có định dạng 3 kiểu bullet cơ bản là kiểu dấu chấm tròn, hình tròn và hình vuông ở đầu mỗi mục trong danh sách

Cú pháp như sau:

```
<UL TYPE=DISC|CIRCLE|SQUARE >
```

```
<LI> Mục thứ 1
```

```
<LI> Mục thứ 2
```

```
.....
```

```
</UL>
```

Trong đó:

- Cặp thẻ ... đánh dấu bắt đầu và kết thúc danh sách. Nó có tác dụng lùi lề trái danh sách vào sâu hơn.

- Các thẻ có tác dụng thể hiện bắt đầu một phần tử mới trong danh sách đây là thẻ đặc biệt chỉ có thẻ mở mà không có thẻ đóng

- Thuộc tính Type chỉ ra kiểu bullet sẽ được sử dụng để đánh cho danh sách, theo mặc định kiểu bullet là chấm tròn (Disc)

Ví dụ 2.9:

```
<html>
```

```
<head>
```

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

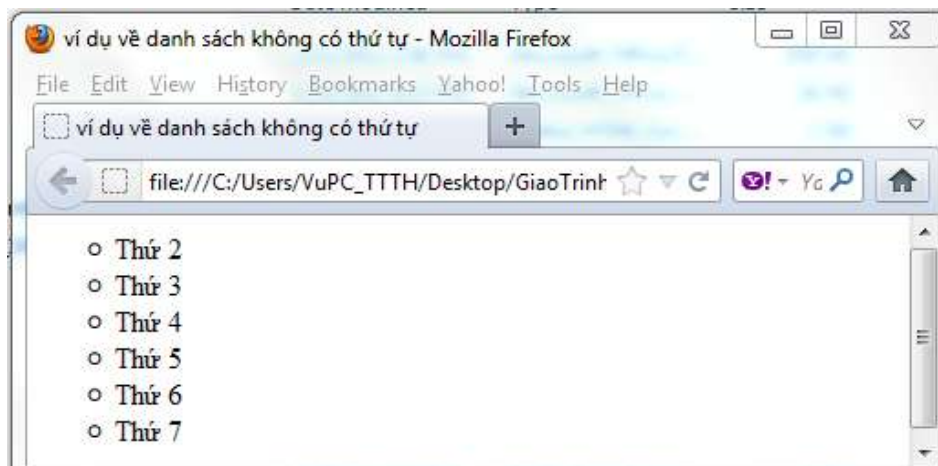
```
<title>ví dụ về danh sách không có thứ tự</title>
```

```

</head>
<body>
  <ul type=circle>
    <li>Thứ 2
    <li>Thứ 3
    <li>Thứ 4
    <li>Thứ 5
    <li>Thứ 6
    <li>Thứ 7
  </ul>
</body>
</html>

```

Chạy chương trình ta được kết quả sau:



Hình 2.13. Kết quả chạy của ví dụ 2.9

2.11.2. Danh sách có thứ tự

Thẻ trong HTML cho phép chúng ta định dạng các mục trong một văn bản là một danh sách có thứ tự. Tức là danh sách mà các phần tử của nó sẽ được đánh thứ tự chẳng hạn như 1, 2, 3... hay a, b, c....

Cú pháp:

```

<OL type = 1|a|A|i|I>
  <LI> Mục thứ 1
  <LI> Mục thứ 2
  .....
</OL>

```

Trong đó:

- Cặp thẻ ... đánh dấu bắt đầu và kết thúc danh sách. Nó có tác dụng lùi lề trái danh sách vào sâu hơn.

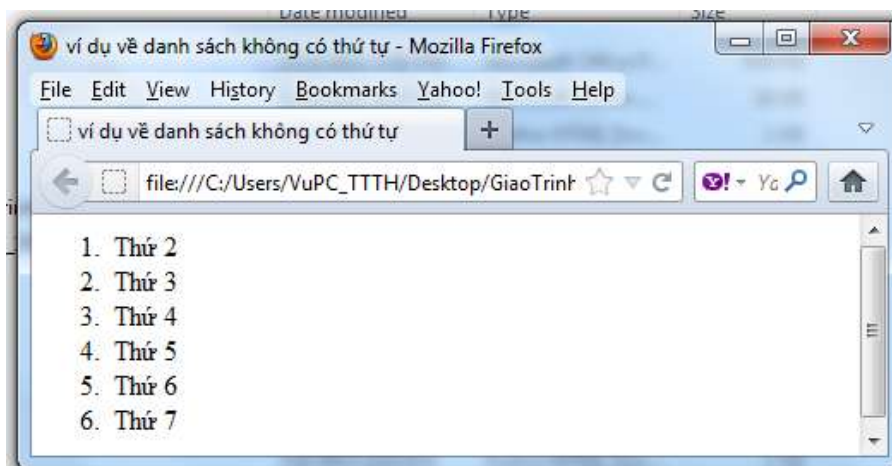
- Các thẻ có tác dụng thể hiện bắt đầu một phần tử mới trong danh sách đây là thẻ đặc biệt chỉ có thẻ mở mà không có thẻ đóng.

- Thuộc tính Type chỉ ra kiểu thứ tự sẽ được sử dụng để đánh cho danh sách, theo mặc định kiểu sẽ là kiểu số.

Ví dụ 2.10:

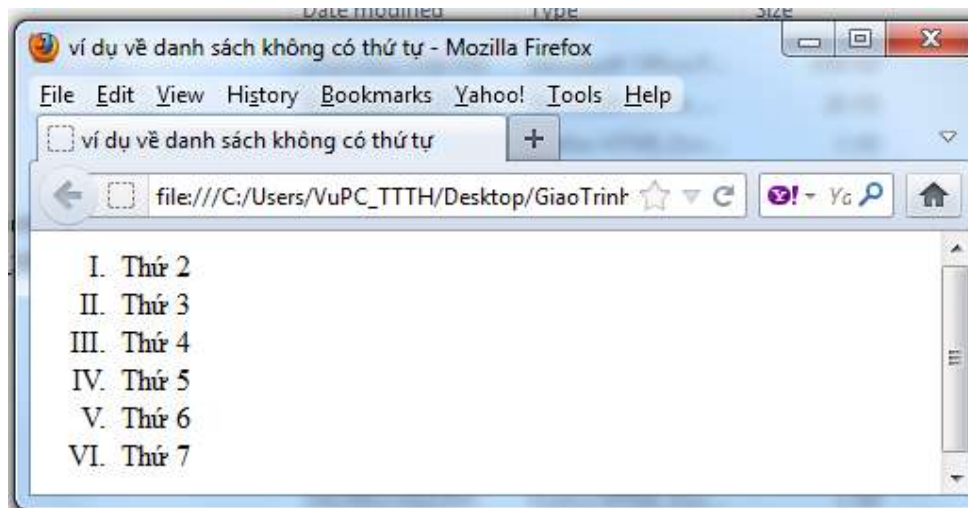
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>ví dụ về danh sách không có thứ tự</title>
</head>
<body>
  <OL type=1>
    <li>Thứ 2
    <li>Thứ 3
    <li>Thứ 4
    <li>Thứ 5
    <li>Thứ 6
    <li>Thứ 7
  </OL>
</body>
</html>
```

Kết quả chạy chương trình ta được danh sách có thứ tự như sau:



Hình 2.14. Kết quả chạy ví dụ 2.10

Nếu ta thay thuộc tính Type = I thì ta sẽ được kết quả như sau:



Hình 2.15. Kết quả chạy file ví dụ 2.10 với type =I

2.11.3. Danh sách định nghĩa

Danh sách các định nghĩa trông giống như một bảng từ vựng hay bảng giải thích các thuật ngữ lùi vào trong.

Cú pháp:

<DL>

<DT> Mục thứ 1

<DL> Mục thứ 2

...

</DL>

Trong đó

- Cặp thẻ **<DL> ...</DL>** đánh dấu bắt đầu và kết thúc danh sách. Nó có tác dụng lùi lề trái danh sách vào sâu hơn.

- Các thẻ **<DT>** có tác dụng thể hiện bắt đầu một phần tử mới trong danh sách đây là thẻ đặc biệt chỉ có thẻ mở mà không có thẻ đóng

Ví dụ 2.11:

<html>

<head>

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

<title>ví dụ về danh sách không có thứ tự</title>

</head>

<body>

<DL>

<DT>Thứ 2

<DL>Thứ 3

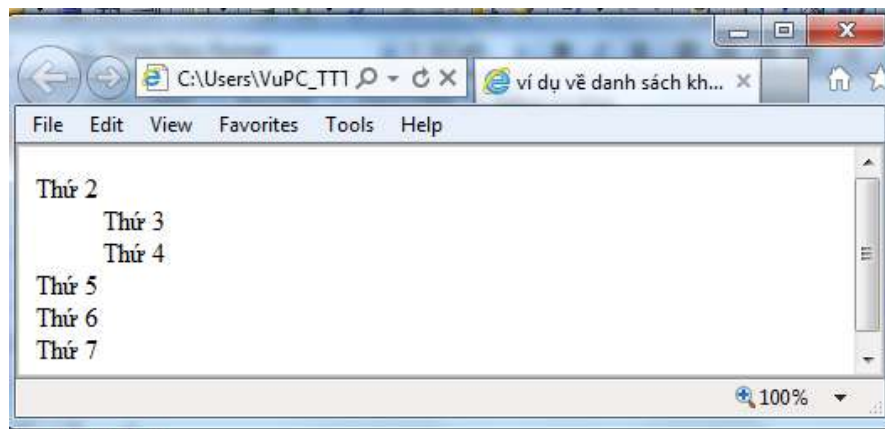
<DT>Thứ 4


```

</DL>Thứ 5
<DT>Thứ 6
<DT>Thứ 7
</DL>
</body>
</html>

```

Chạy chương trình ta thu được kết quả:



Hình 2.16. Kết quả chạy file ví dụ 2.11

2.12. Sử dụng font

Thẻ dùng để định dạng hiển thị văn bản trên trang web. Chúng ta cũng có thể chỉ định các thuộc tính như kích thước, màu sắc, kiểu...

Cú pháp:

```
<Font face = "Time New Roman" size = 4 color =#0000FF>..... </Font>
```

Trong đó:

- Thuộc tính face dùng để chỉ loại phông chữ sẽ được sử dụng để định dạng cho văn bản giá trị của nó là một phông chữ nào đó như: Arial, Time New Roman, Verdana...
- Thuộc tính Size là để chỉ định cỡ chữ giá trị của nó có thể là 1, 2, 3, 4, 5, ...
- Color dùng để chỉ định màu cho chữ, màu có thể là tên màu chẳng hạn như red, blue, black .. cũng có thể là mã màu trong trường hợp là mã màu thì phải có dấu # đứng trước mã màu.

2.13. Sử dụng màu sắc

Trong thiết kế web việc sử dụng màu cho một trang web là rất cần thiết, nó giúp cho trang web trở nên sinh động hơn nếu người thiết kế biết sử dụng màu cho trang web và cho các nội dung cho trang web như màu nền, màu chữ ...

Để sử dụng màu chúng ta chỉ cần gán giá trị màu cho thuộc tính color, bgcolor, bordercolor...trong một số thẻ của HTML. Chúng ta có thể gán giá trị màu bằng một trong

2 cách là sử dụng tên màu, hay sử dụng mã màu. Trong trường hợp sử dụng mã màu nó là một mã hecxa tổ hợp từ 3 màu có bản Red, Green và Blue có dấu # ở trước có cấu trúc sau:

#RRGGBB

Số hecxa 00 chỉ 0% của màu trong khi đó số hecxa FF chỉ 100% của màu. Giá trị cuối cùng là một mã sáu chữ số chỉ màu.

Mã thập lục phân	Màu
#FF0000	Red
#00FF00	Green
#0000FF	Blue
#000000	Black
#FFFFFF	White

2.14. Sử dụng hình ảnh trong tài liệu HTML

2.14.1. Chèn hình ảnh vào tài liệu HTML

Việc sử dụng hình ảnh trong trang web là hết sức cần thiết. Ảnh làm cho trang web của chúng ta trở nên sinh động và trực quan hơn.

Ảnh có thể là biểu tượng, bullet, ảnh, logo công ty và nhiều cái khác.

Ngày nay có nhiều định dạng đồ họa đang được sử dụng như: *.bmp, *.gif, *.jpeg, *.jpg,.... Tuy nhiên trình duyệt web thường chỉ đọc được một vài định dạng ảnh như: Ảnh GIF (Graphics Interchange Format) (.GIF), Ảnh JPEG (Joint Photographic Expert Group) (.JPEG), PNG (Portable Network Graphics)...

Để chèn hình ảnh vào trang web HTML có một thẻ gọi là thẻ Image cú pháp của thẻ như sau:

```
<IMG SRC="đường dẫn đến ảnh cần gọi" BORDER="chiều rộng đường viền"  
ALT="Dòng văn bản thông báo" WIDTH="chiều rộng ảnh" HEIGHT="chiều cao ảnh"  
ALIGN="vị trí của ảnh">
```

Trong đó các thuộc tính:

- ALIGN="left|right|top|texttop|middle|absmiddle|baseline|bottom": Thuộc tính này dùng để căn lề của hình ảnh với các dòng văn bản trong web.

- ALT="Alternative Text": Thuộc tính này cho hiển thị một dòng văn bản thay thế cho tập tin hình ảnh trong trường hợp trình duyệt Web đang ở trong chế độ không hiển thị được hình ảnh. Dòng văn bản này cũng hiển thị theo dạng ToolTip khi dờ chuột đến hình.

- SRC="URL of image": Chỉ định địa chỉ tập tin hình chèn vào trang Web.

- WIDTH="value"/ HEIGHT="value": Chỉ định chiều rộng và chiều cao của ảnh sẽ được hiển thị lên trên trình duyệt web.

- BORDER="value": Chỉ định cho hiển thị đường viền bao quanh hình ảnh. Giá trị value là số nguyên dương.

- VSPACE="value" HSPACE="value": Quy định khoảng trống giữa hình và đoạn văn bản. VSPACE cho trên và dưới hình, HSPACE cho trái và phải hình. Giá trị value được tính theo đơn vị pixel.

2.14.2. Chèn ảnh động vào tài liệu HTML

Đôi khi ảnh động sẽ làm cho tài liệu HTML của ta trở nên trực quan và sinh động hơn so với ảnh tĩnh. Chúng ta cũng có thể chèn ảnh động vào website.

Để chèn ảnh động vào website chúng ta sử dụng thẻ Image như trong phần trên, tuy nhiên thuộc tính src được thay bằng thuộc tính Dynsrc:

```
<IMG DYN SRC="đường dẫn đến ảnh cần gọi" BORDER="chiều rộng đường viền"  
ALT="Dòng văn bản thông báo" WIDTH="chiều rộng ảnh" HEIGHT="chiều cao ảnh"  
ALIGN="vị trí của ảnh">
```

2.14.3. Chèn âm thanh vào tài liệu HTML

Để chèn âm thanh vào một trang web ta sử dụng thẻ BGSound trong ngôn ngữ HTML. Cú pháp của thẻ như sau:

```
<BGSOUND src =url loop =số lần lặp| infinite >
```

Trong đó:

- Src là thuộc tính chỉ định file âm thanh sẽ được chơi trên trang web, url là đường dẫn đến file âm thanh đó. Ngày nay hầu hết các trình duyệt web đều hỗ trợ một số định dạng âm thanh như MIDI, . WAV, hay .AU. Một số trình duyệt web IE hay FireFox thì có thể hỗ trợ một số định dạng khác như .mp3 ...

- Loop là thuộc tính dùng để chỉ số lần sẽ chơi lại bản nhạc này, trong trường hợp ta đặt loop = -1 hoặc infinite thì sẽ lặp lại vô hạn lần (lặp lại cho đến khi tắt trang web đó đi)

Ví dụ 2.12:

```
html>  
  <head>  
  <title>Chèn âm thanh vào trang web</title>  
  </head>  
  <body>  
    <bgsound src =10minutes1.mp3 loop =2>  
  </body>  
</html>
```

Thẻ BGSound cho phép chèn âm thanh vào một trang web tuy nhiên không có trình điều khiển để điều khiển việc chơi bản nhạc này chẳng hạn như việc tăng hoặc giảm volume hay tạm dừng (stop) và chơi lại (play). Để chèn âm thanh vào trang web có chứa

thanh công cụ để điều khiển việc chơi nhạc trên trang web ta sử dụng một thẻ khác của HTML đó là thẻ Embed chi tiết về thẻ này và cách sử dụng thẻ ta sẽ xét ở phần sau chèn video vào tài liệu HTML.

2.14.4. Chèn Video vào tài liệu HTML

Để chèn video vào tài liệu HTML ta sử dụng thẻ embed như sau:

Cú pháp:

```
<EMBED src = url height = n-pixel width = n-pixel >
```

Trong đó:

- src là thuộc tính dùng để chỉ định file video sẽ được chèn vào trang web, url là đường dẫn đến file hình ảnh đó. Hầu hết các trình duyệt web hiện tại đều hỗ trợ một số định dạng video như: .avi, .asf, .ram, .mpg tuy nhiên một số trình duyệt như IE, FireFox còn có thể hỗ trợ một số định dạng khác như .mp4 chẳng hạn.

- height thuộc tính chỉ định chiều cao của video, giá trị của nó là một số pixel trên màn hình.

- Width là thuộc tính chỉ định chiều rộng của video, giá trị cũng được thể hiện bằng số pixel trên màn hình.

Ví dụ 2.13:

```
<html>
  <head>
    <title>Chèn video vào trang web</title>
  </head>
  <body>
    <embed src =Step_LastThingOnMyMind.mpg width=200 height =200>
  </body>
</html>
```

Chạy chương trình kết quả như sau :



Hình 2.17. Kết quả chạy ví dụ 2.13

2.14.5. Chèn các Java Applets

Ngoài khả năng chèn hình ảnh, âm thanh vào trang web. Chúng ta còn có thể chèn một chương trình Java Applets vào trang web. Chương trình Java Applets là một chương trình Java nhỏ có thể chạy trên internet. Chương trình này có thể tạo ra các ứng dụng multimedia trên web, các plus in hay các chương trình đồ họa trên web. Để chèn Java Applets vào trang web chúng ta sử dụng thẻ Applet cú pháp như sau:

Cú pháp :

```
<APPLET code = "Tên lớp applet" width =n-pixel height = n-pixel>  
    <param name = "Tên biến1" value = "Giá trị">  
    <param name = "Tên biến2" value = "Giá trị">  
    ....  
</APPLET>
```

Trong đó :

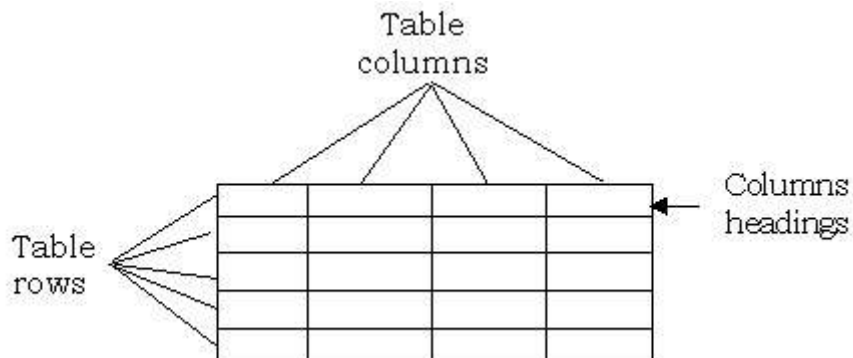
- Thuộc tính code để chỉ định lớp applet muốn chèn vào trang web
- Width, height thuộc tính quy định chiều rộng và cao của applet sẽ hiển thị trong trang web, được tính theo số pixel màn hình.
- param là một thẻ mở mà không có thẻ đóng được nằm trong cặp thẻ applet để chỉ định các tham biến sẽ được truyền giá trị vào trong lớp applet. Thuộc tính name chỉ tên biến cần truyền, Value thuộc tính quy định giá trị sẽ truyền vào biến đó.

Ví dụ 2.14: Chèn vào trang web một applet trong lớp clock1 là một cái đồng hồ được lập trình bằng java Applet

```
HTML>  
<HEAD>  
    <TITLE>Chèn Applet vào trang web</TITLE>  
</HEAD>  
<BODY>  
    <APPLET codebase="classes" code="Clock1.class" width=350 height=200>  
</APPLET>  
</BODY>  
</HTML>
```

2.15. Cách tạo bảng

Chúng ta có thể sử dụng bảng để hiển thị dữ liệu dưới dạng các hàng và các cột. Bảng giúp cho chúng ta điều khiển, xác định và sắp xếp vị trí của văn bản và hình ảnh trên trang web, thay vì giao tất cả các việc đó cho trình duyệt.



2.15.1. Thẻ dùng để tạo bảng

Thẻ <TABLE> được dùng để tạo bảng trong tài liệu HTML. Các thuộc tính của phần tử <TABLE> được áp dụng cho bảng, nhưng không cho dữ liệu hiển thị trên bảng. Đơn vị cơ bản của bảng là một ô và được định nghĩa bằng thẻ <TD>.

Cú pháp tạo ra cấu trúc chung của bảng như sau :

```

<TABLE>
  <TR>
    <TH> Tên cột 1 </TH>
    <TH> Tên cột 2 </TH>
    ...
  </TR>
  <TR>
    <TD> Ô 1 dòng 1 </TD>
    <TD> Ô 2 dòng 1 </TD>
    ...
  </TR>
  .....
</TABLE>

```

Trong đó :

- Cặp thẻ <TABLE>...</TABLE> tạo ra một bảng
- Cặp thẻ <TR>...</TR> tạo ra hàng trong bảng
- Cặp thẻ <TH>...</TH> tạo ra một tiêu đề trong một hàng của bảng
- Cặp thẻ <TD>...</TD> tạo ra một ô trong một hàng của bảng

Như vậy để chèn một bảng vào trang web ta phải sử dụng thẻ table thẻ này có một số thuộc tính như sau:

```

<TABLE
  Boder =n

```

Bordercolor = color
Bgcolor=color
Background = url
Cellspacing=n
Cellpadding=n
Align = Left | Center | Right
Width = n
Height = n
>

Trong đó:

- Border thuộc tính quy định đường viền, n là độ dày của đường viền
- Bordercolor là thuộc tính quy định màu của đường viền
- Bgcolor là thuộc tính quy định màu nền của bảng
- Background là thuộc tính quy định ảnh nền của bảng, url là đường dẫn đến file hình ảnh làm nền
- Cellspacing thuộc tính quy định khoảng đường biên trong bảng được tính bằng pixel
- Cellpadding thuộc tính quy định khoảng cách từ lề đến dữ liệu trong ô
- Align thuộc tính dùng để căn lề cho dữ liệu trong bảng, nhận một trong các giá trị left căn lề trái, right căn lề phải, center căn lề giữa
- Width thuộc tính quy định chiều rộng của bảng tính bằng pixel
- Height thuộc tính quy định chiều cao của bảng tính bằng pixel

2.15.2. Chèn hàng và cột

Thẻ <TR> được sử dụng để chèn một hàng vào bảng thẻ này có một số thuộc tính như sau :

<TR
Bordercolor = color
Bgcolor=color
Background = url
Align = Left | Center | Right
Width = n
Height = n
>

Các thuộc tính tương tự như các thuộc tính của thẻ table nhưng có tác dụng trong một hàng
Để chèn cột vào bảng, một ô trong một hàng ta sử dụng thẻ <TD> như trên thẻ này có một số thuộc tính quan trọng sau :

<TD

Bordercolor = color

Bgcolor=color

Background = url

Align = Left | Center | Right

valign= top| Middle| Center|Bottom

Colspan = n

Rowspan =n

>

Trong đó :

- Các thuộc tính Bordercolor, Bgcolor, Background, Align tương tự như các thẻ trên
- Thuộc tính valign quy định căn lề theo trên, giữa và cuối
- Colspan là thuộc tính dùng để trộn các ô theo cột, n là số ô trộn
- Rowspan thuộc tính quy định trộn các ô theo hàng, n là số ô trộn

2.15.3. Xóa hàng và cột

Để xóa một hàng, xóa thẻ <TR> tương ứng cùng với tất cả các thẻ <TD> bên trong nó từ một file mã nguồn HTML. Tương tự, để xóa các cột xóa các thẻ <TD> tương ứng theo một cột trên tất cả các thẻ <TR>.

2.15.4. Trộn ô

Trộn ô là sự kết hợp các ô liền nhau theo cột hay theo hàng, để làm việc này ta sử dụng thuộc tính colspan và rowspan trong thẻ TD.

Ví dụ 2.15:

```
<html>
<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Ví dụ về tạo bảng</title>
</head>
<body>
<table border="1" width="47%">
  <tr>
    <td width="64" rowspan="2">
      <p align="center">Thứ</p>
    <td width="72" rowspan="2">
      <p align="center">Buổi</p>
    </td>
  </tr>
  <tr>
    <td>
    </td>
  </tr>
</table>
```




```

        <td colspan="4">
        <p align="center">Tháng</td>
    </tr>
    <tr>
        <td height="24" align="center">
        <p align="center">Tuần 1</td>
        <td height="24" align="center">Tuần 2</td>
        <td height="24" align="center">Tuần 3</td>
        <td height="24" align="center">Tuần 4</td>
    </tr>
    <tr>
        <td width="64" rowspan="3">
        <p align="center">2</td>
        <td width="72" align="left">Sáng</td>
        <td> &nbsp;</td>
        <td> &nbsp;</td>
        <td> &nbsp;</td>
        <td> &nbsp;</td>
    </tr>
    <tr>
        <td width="72" align="left">Chiều</td>
        <td> &nbsp;</td>
        <td> &nbsp;</td>
        <td> &nbsp;</td>
        <td> &nbsp;</td>
    </tr>
    <tr>
        <td width="72" align="left">Tối</td>
        <td> &nbsp;</td>
        <td> &nbsp;</td>
        <td> &nbsp;</td>
        <td> &nbsp;</td>
    </tr>
</table>
</body>
</html>

```

Chạy chương trình ta được kết quả:



The screenshot shows a Mozilla Firefox browser window titled "Ví dụ về tạo bảng - Mozilla Firefox". The address bar shows the file path: file:///C:/Users/VuPC_TTTH/L. The main content area displays a table with the following structure:

Thứ	Buổi	Tháng			
		Tuần 1	Tuần 2	Tuần 3	Tuần 4
2	Sáng				
	Chiều				
	Tối				

Hình 2.18. Kết quả chạy ví dụ 2.15

2.15.5. Định dạng cho ô

Để định dạng cho các ô trong bảng ta sử dụng các thuộc tính của thẻ <TD> hay thẻ <Table> . Chẳng hạn có thể chỉ định được vị trí của văn bản bên trong mỗi ô của bảng. Những thuộc tính canh lề ngang (ALIGN) và canh lề dọc (VALIGN) được sử dụng để điều khiển việc canh lề trong các ô của bảng. Thuộc tính ALIGN có thể có các giá trị: trái, phải, giữa, đều hai bên. Thuộc tính VALIGN có thể có các giá trị: trên, giữa và dưới.

Ngoài việc xác định vị trí nội dung của các ô trong bảng, chúng ta cũng có thể thay đổi kích thước, màu sắc của bảng. Chúng ta có thể xác định chiều rộng của bảng. Đó là vùng trên màn hình mà bảng sẽ mở rộng ra. Thuộc tính WIDTH của thẻ TABLE sẽ thật sự cần thiết. Giá trị của thuộc tính WIDTH có thể là một định dạng phần trăm hay thậm chí là các điểm. Chẳng hạn, <TABLE WIDTH = 50%> . Điều này sẽ kéo rộng bảng đến 50% của màn hình. Nếu chúng ta muốn thêm một vài màu sắc cho bảng, chúng ta sử dụng thuộc tính BGCOLOR. Giá trị này có thể là tên của một màu hay là giá trị ở hệ thập lục phân.

2.16. Lớp (Layer)

2.16.1. Khái niệm lớp

Khi đặt vị trí cho một phần tử trên trang Web, chúng ta xác định tọa độ x và y. Tuy nhiên, chúng ta cũng có thể xác định tọa độ z. Như thế, trục tọa độ z là một đường ảo từ phía trước đến phía sau màn hình. Vì vậy, một phần tử có thể được đặt ở trên hay dưới phần tử khác trong lớp. Tọa độ z xác định thứ tự trong đó các phần tử được hiển thị.

Khái niệm về lớp có thể được đưa ra đối với một tài liệu HTML. Một lớp giống như một frame mà nó có thể chứa nội dung và được chỉ định vị trí ở nơi được yêu cầu. Trong phần tiếp theo sẽ thảo luận kỹ hơn về lớp.

Lớp là một phần tử có thể được thêm vào một trang Web. Chúng ta có thể xác định chính xác vị trí của một lớp trong trang Web. Netscape cung cấp một phần tử <LAYER> mà nó có thể được sử dụng để xác định vị trí nội dung trên trang Web.

Lớp có tập thuộc tính riêng của nó có thể được truy xuất và lập trình. Các lớp có thể được lập trình thông qua script (tức là đoạn mã của một chương trình nhỏ) để tạo ra những trang động hay những trang có khả năng tương tác với nhau. Vì vậy, chúng ta có thể xác định vị trí các phần tử trên trang mà ta muốn thay vì phải di chuyển nó đến phần chức năng (functionality) được yêu cầu bởi mỗi trình duyệt. Thêm vào đó, chúng ta có thể kết nối các văn bản và hình ảnh lại với nhau để tạo ra một trang web sinh động và dễ đọc. Các hiệu ứng đặc biệt có thể được thêm vào trên một trang web mà không cần phải tăng kích thước của file lên nhiều lắm. Chúng ta có thể thay đổi các phần của trang mà không phải thay đổi lại, thiết kế lại trang chủ, như thế, chúng ta có thể thay những nội dung của một tầng đặc biệt mà không phải hiệu chỉnh trang chủ.

Cả Internet Explorer và Netscape Navigator đều cung cấp công nghệ lớp và mỗi nhà cung cấp đều có một cách riêng. Đối với Internet Explorer hỗ trợ mô hình CSS (Cascading Style Sheet – Mô hình thác nước). Khi chúng sử dụng những kiểu Bảng kiểu như thế này thì chúng ta có thể nhóm các thành phần của trang lại với nhau trong một Bảng kiểu hình thác nước. Bảng kiểu loại này có thể được đưa ra để chọn một hay tất cả các phần tử trong một tài liệu HTML. Những phần tử có thể được điều khiển thông qua VBScript hay JavaScript. Để tạo ra một tầng tài liệu chúng ta sẽ phải thay đổi thuộc tính của kiểu Bảng kiểu này. Netscape Navigator cũng hỗ trợ việc sử dụng CSS, nhưng theo một cách khác. Chúng ta phải sử dụng thẻ <LAYER> do Netscape hỗ trợ.

2.16.2. Sử dụng các lớp

Lớp được bàn ở đây như là một thực thể đơn của nội dung. Nó cũng được chỉ ra giống như bất kỳ một kiểu nào khác mà nó chỉ đến những phần tử bên trong một trang HTML. Vì vậy, nếu chúng ta muốn ứng dụng kiểu lớp cho tất cả những phần tử bên trong lớp, chúng ta có thể sử dụng những phần tử như DIV hay SPAN.

Ví dụ 2.16:

```
<HTML>
<HEAD>
<TITLE>Grouping elements</TITLE>
</HEAD>
<BODY BGCOLOR=lavender>
  <LAYER id = layer1 bgcolor=aqua>
    <DIV><font color = red>
      <H1>Layer 1</H1>
```

```

<P>This is a paragraph element within division 1</P>
<P>Content for layer 1.</P></FONT>
</DIV>
<P>Outside the division</P>
<DIV> <font color=blue>
<H2>Division 2</H2>
<P>This is a paragraph element within division 2</P>
<P>Content for layer 1.</P></FONT>
</DIV>
</LAYER>
</BODY>
</HTML>

```

2.17. Giới thiệu biểu mẫu

Để mở rộng khả năng phục vụ người sử dụng , dịch vụ Web cần phải sử dụng các ứng dụng khác bên ngoài . Ví dụ NSD (Người sử dụng) yêu cầu tìm kiếm trong cơ sở dữ liệu, lấy các thông tin tức thời, luôn được cập nhật.... Để làm được điều này dịch vụ Web phải chuyển yêu cầu của NSD đến một ứng dụng khác . Ứng dụng này sẽ thực hiện yêu cầu và trả lại kết quả cho Web server để chuyển tiếp đến NSD.

Form là một cách để chuyển dữ liệu từ NSD đến cho Web Server xử lý. Forms được sử dụng rộng rãi trên WWW . Các forms rất tiện lợi cho người dùng điền các yêu cầu tìm kiếm, các biểu mẫu điều tra, nhập dữ liệu đầu vào cho các ứng dụng..

Có nhiều thành phần khác nhau (gọi là các điều khiển - control) trong một form. Tùy theo yêu cầu giao tiếp với NSD cần chọn thành phần thích hợp nhất.

Khi tạo form cũng cần phải chỉ rõ cho máy chủ dịch vụ biết cách xử lý form. Có nhiều loại chương trình ứng dụng khác nhau trong máy chủ dịch vụ để làm việc này: các chương trình CGI, ISAPI, các script ASP, JSP, Java Bean, Servlet...

Tạo form là khâu đầu tiên trong việc xây dựng giao tiếp giữa NSD với các ứng dụng Internet / Intranet

Web form hay Form HTML là một phần của tài liệu, nó chứa các phần tử đặc biệt gọi là các điều khiển. Các điều khiển được sử dụng để nhập thông tin từ người dùng và cung cấp một số tương tác. Dữ liệu do người dùng nhập vào có thể được xác nhận hợp lệ nhờ các kịch bản phía máy khách (client-side scripts) và được chuyển đến máy chủ để xử lý thêm.

2.17.1. Sử dụng biểu mẫu

Việc sử dụng biểu mẫu trên World Wide Web là khá nhiều và liên tục tăng lên. Sau đây là một số cách sử dụng thông thường:

Thu thập tên, địa chỉ, số điện thoại, địa chỉ e-mail và các thông tin khác để người dùng đăng ký cho một dịch vụ hay một sự kiện nào đó.

Thu thập thông tin dùng để đăng ký mua một mặt hàng nào đó, ví dụ, khi muốn mua một cuốn sách trên Internet, ta phải điền tên, địa chỉ gửi thư, 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 một dịch vụ nào đấy 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 dịch vụ.

Cung cấp công cụ tìm kiếm cho web site. Các site cung cấp nhiều thông tin khác nhau thường cung cấp cho người dùng hộp tìm kiếm để cho phép họ tìm kiếm thông tin nhanh hơn.

2.17.2. Phần tử FORM

Phần tử <FORM> được sử dụng để tạo một vùng trên trang như một biểu mẫu. Nó chỉ ra cách bố trí của biểu mẫu.

Cú pháp:

<Form action =url name = tên form method = get|post>

Các phần điều khiển

</Form>

Trong đó các thuộc tính:

- Action chỉ định địa chỉ mà form đó sẽ gửi thông tin đến, url là đường dẫn đến địa chỉ trang xử lý thông tin.

- Name chỉ định tên của form

- Method chỉ định phương thức truyền, một trong 2 phương thức sẽ được chỉ định là post hoặc get. Theo mặc định phương thức post sẽ được chỉ định.

2.17.3. Các phần tử nhập của HTML

Khi tạo một biểu mẫu (web form), ta có thể tạo các phần tử điều khiển cho form – Form Control. Các điều khiển này giúp nhận các thông tin của người dùng như các textbox cho phép nhập thông tin, button là các nút điều khiển.... Các điều khiển này được sử dụng với phần tử <FORM>. Mỗi một phần tử được tạo ra bằng một thẻ <input> cú pháp như sau:

Cú pháp:

<input

type = button| checkbox| file| hidden| image| password| radio| reset| submit|text
name = "tên phân tử"
value = "Giá trị"
checked

>

Trong đó các thuộc tính:

- Type chỉ định kiểu của đối tượng điều khiển trên form, có thể nhận một trong các đối tượng điều khiển như:

- ✓ Button là một nút điều khiển
- ✓ Checkbox là một hộp lựa chọn kiểu ô vuông, cho phép người dùng lựa chọn hoặc không lựa chọn một mục nào đó, kiểu checkbox là một kiểu lựa chọn nhiều trong nhiều.
- ✓ File là một đối tượng điều khiển cho phép người dùng lựa chọn một file dữ liệu
- ✓ Hidden là một đối tượng ẩn
- ✓ Image là một đối tượng điều khiển dạng ảnh
- ✓ Password là một đối tượng cho phép người dùng nhập vào mật khẩu
- ✓ Radio là một đối tượng lựa chọn dạng chấm tròn, đây là dạng lựa chọn một trong nhiều.
- ✓ Reset là một nút điều khiển cho phép thiết lập lại các giá trị trên các điều khiển khác của form theo mặc định
- ✓ Submit là một nút điều khiển cho phép gửi tin đi khi người dùng kích hoạt
- ✓ Text là một điều khiển dưới dạng một hộp chữ nhật cho phép người dùng nhập văn bản vào

- Name quy định tên của đối tượng điều khiển đó, thuộc tính này rất quan trọng trong quá trình lập trình. Trong khi lập trình chúng ta chỉ lấy được dữ liệu của các đối tượng gửi về thông qua tên của chúng.

- Value giá trị được gán cho đối tượng điều khiển đó, giá trị này sẽ được hiển thị lên màn hình nếu đối tượng có dạng button, submit, text, password, hay reset.

- Checked thiết lập đối tượng sẽ được lựa chọn, thuộc tính này chỉ có giá trị với các đối tượng radio hoặc checkbox

- Src chỉ ra địa chỉ của file ảnh, thuộc tính này chỉ có giá trị với đối tượng có kiểu image

Ngoài các đối tượng điều khiển trên HTML các có các đối tượng điều khiển khác như.

Đối tượng điều khiển Vùng văn bản TextArea Loại này tạo ra một điều khiển nhập văn bản trên nhiều dòng so với hộp văn bản nhập một dòng. Ta phải xác định kích thước của textarea. Ta cũng phải xác định số dòng, số cột trong textarea.

Cú pháp như sau:

```
<textarea cols =n rows =n name = Tên >
```

..

```
</textarea>
```

Trong đó các thuộc tính:

- Cols chỉ định số cột của vùng văn bản
- rows là số hàng của vùng văn bản
- Name tên của đối tượng điều khiển

Đối tượng điều khiển select được sử dụng để hiển thị một danh sách các lựa chọn cho người dùng. Mỗi lựa chọn được biểu diễn bởi phần tử OPTION. Một phần tử SELECT phải chứa ít nhất một phần tử OPTION

Cú pháp:

```
<select size="1" name="D1">
```

```
    <option>...</option>
```

```
    <option>...</option>
```

.....

```
</select>
```

Trong đó:

- Name là thuộc tính đặt tên cho đối tượng
- Size là độ cao của đối tượng
- <Option> là thẻ dùng để tạo ra một lựa chọn

2.17.4. Tạo biểu mẫu

Chúng ta đã thảo luận về phần tử <FORM> và các điều khiển có thể thêm vào biểu mẫu để nhận thông tin người dùng. Trong phần này, chúng ta sẽ tạo một số biểu mẫu để nhận thông tin. Các nút RESET và SUBMIT thực hiện các công việc cần thiết.

Ví dụ 2.17:

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Language" content="en-us">
```

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

```
<title>Đăng nhập người dùng</title>
```

```
</head>
```

```
<body>
```

```
<form action = "" name = tenform method = get/post>
```

Đăng nhập người dùng

Tên truy nhập <input type="text" name="T1" size="23">

Mật khẩu

```
<input type="password" name="T2" size="23"><br>
```

```
<input type="submit" value="Login" name="B1"><input type="reset"
```

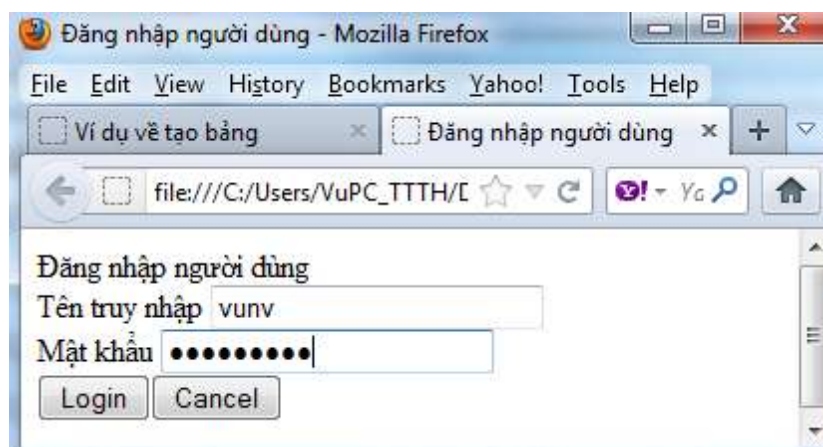
```
value="Cancel" name="B2"><br>
```

```
</form>
```

```
</body>
```

```
</html>
```

Khi chạy ta có kết quả sau:



Hình 2.19. Kết quả chạy của ví dụ 2.17

Ví dụ 2.18: tạo file application form

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Language" content="en-us">
```

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

```
<title>Application Form</title>
```

```
</head>
```

```
<body>
```

```
<form method="POST" action="Application.asp">
```

```
<p>Application Form</p>
```

```
<hr width = 400 align =left>
```

```
<p>Tên <input type="text" name="T1" size="36"></p>
```

```
<p>Lĩnh vực hoạt động</p>
```

```
<p><input type="radio" value="V1" checked name="R1">Web Designer
```

```
<input type="radio" name="R1" value="V2">Web Administrator
```

```
<input type="radio" name="R1" value="V3">Web Developer</p>
```



```

<p>Kinh nghiệp <select size="1" name="D1">
<option>None</option>
<option>1 năm</option>
<option>2 năm</option>
</select></p>
<p>&nbsp;Ghi chú:</p>
<p><textarea rows="5" name="S1" cols="42"></textarea></p>
<p><input type="checkbox" name="C1" value="ON"> Tôi chấp nhận và gửi đi</p>
<p><input type="submit" value="ok" name="B1"><input type="reset"
value="Reset" name="B2"></p>
</form>
</body>
</html>

```

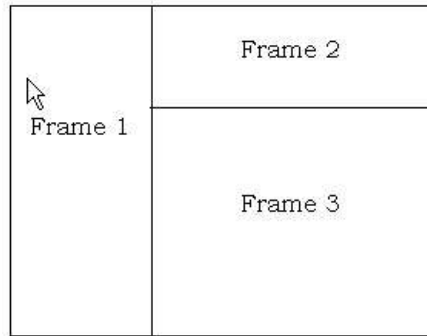
Chạy chương trình ta thu được kết quả sau:



Hình 2.20. Kết quả chạy ví dụ 2.18

2.18. Khung (Frame)

Khung chia một cửa sổ trình duyệt thành nhiều vùng riêng biệt, mà mỗi vùng có thể hiển thị một trang web riêng biệt có thể cuộn được. Mỗi khung là một cửa sổ trong cửa sổ chính. Ví dụ, ta có thể sử dụng ba khung trong trang Web, một làm biểu ngữ (banner), một làm menu điều hướng và một để hiển thị dữ liệu. Mỗi khung có thể được tạo, thay đổi và cuộn độc lập nhau.



Hình 2.21: Khung trong trang Web

Trong thiết kế web người ta phải thiết kế bố cục cho website đó. Thông thường các website được chia thành nhiều phần khác nhau, mỗi phần sẽ hiển thị một nội dung độc lập nhau chẳng hạn như: phần đầu thường được hiển thị nội dung là một logo, banner, phần bên trái là menu, phần chính giữa hiển thị nội dung của các trang. Do vậy việc sử dụng khung chia màn hình thành nhiều phần khác nhau mỗi phần sẽ hiển thị một nội dung độc lập nhau sẽ làm cho việc thiết kế màn hình đơn giản và thuận lợi hơn.

Để tạo khung trong trang web ta sử dụng thẻ Frame theo cú pháp sau:

```
<frameset rows="...,*" | cols = " ...,*" >
    <frame name="contents" src = url scrolling = yes|no|Auto Frameborder = n>
    <frame name="contents" src = url scrolling = yes|no|Auto Frameborder = n>
    .....
</frameset>
```

Trong đó:

- <Frameset> là thẻ dùng để định nghĩa một tập các frame sẽ được tạo ra
- Rows thuộc tính xác định việc chia màn hình theo chiều ngang, số lượng khung chia theo chiều ngang được xác định bằng độ rộng của khung tính theo pixel hoặc % màn hình. Chúng ta có thể chia thành nhiều khung mỗi khung cách nhau một dấu phẩy.
- Cols thuộc tính xác định việc chia màn hình theo chiều dọc, số lượng khung chia theo chiều dọc được xác định bằng độ rộng của khung tính theo pixel hoặc % màn hình. Chúng ta có thể chia thành nhiều khung mỗi khung cách nhau một dấu phẩy.
- <Frame> thẻ để định nghĩa một khung trong tập các khung được chia có các thuộc tính Name tên của khung, src là chỉ định trang web sẽ được hiển thị trên khung, url là địa chỉ của trang web đó, Scrolling thuộc tính quy định thanh cuộn cho cửa sổ, nó nhận một trong các giá trị yes, no hoặc auto, Frameborder thuộc tính quy định đường viền của khung.

Ví dụ 2.19: tạo Frame

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

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>New Page 3</title>
</head>
<frameset rows = "30%,*" border =0 >
  <frame src= banner.htm name="banner" border =0 scrolling =no>
  <frameset cols = "15%,60%,*" border =0>
    <frame src=menu.htm name = "menu" border =0>
    <frame src = trangchu.htm name = "main" border =0>
    <frame src = dangnhap.htm name = "dangnhap" border = 0>
  </frameset>
</frameset>
</html>

```

Khi chạy chương trình ta thu được kết quả sau:



Hình 2.22. Kết quả chạy ví dụ 2.19

2.19. Phần tử IFRAME

Phần tử IFRAME được sử dụng để tạo khung trên dòng (inline frame) hay khung nổi (floating frame). Ta có thể tạo và chèn một khung vào một khối văn bản. Khi trình duyệt không hỗ trợ khung thì nội dung nằm trong thẻ IFRAME bị trả lại. Khung trên dòng (inline frame) không thể thay đổi kích thước.

Để tạo Iframe ta thực hiện theo cú pháp sau:

```

<iframe
  name = "Tên frame"
  src = url
  frameborder = no|yes
  scrolling =yes |no|Auto
  height = n
  width = n>
</iframe>

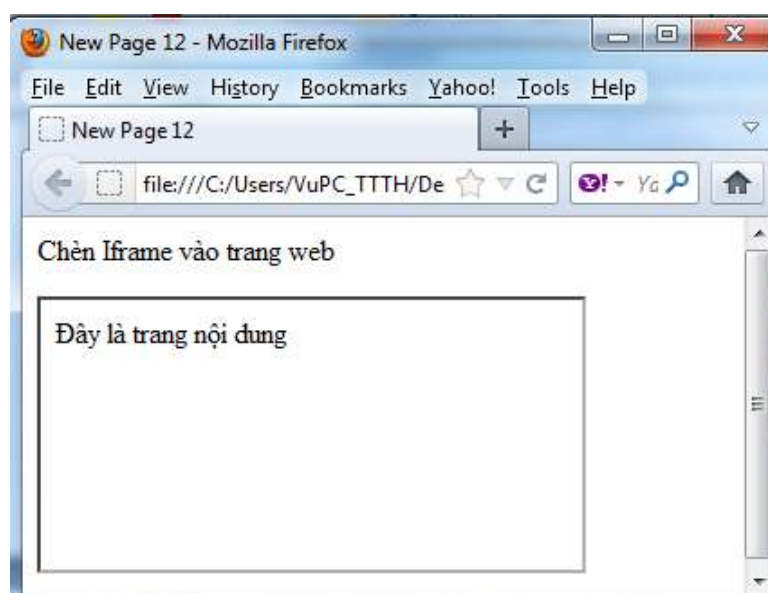
```

Trong đó các thuộc tính tương tự như thuộc tính của frame mà chúng ta đã biết.

Ví dụ 2.20:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>New Page 12</title>
</head>
<body>
<p>Chèn Iframe vào trang web </p>
<p><iframe name="I1" src="Noidung.htm" >
</iframe></p>
</body>
</html>
```

Chạy ví dụ ta thu được kết quả sau:



Hình 2.23. Kết quả chạy ví dụ 2.20

2.20. Định dạng bằng CSS

2.20.1. Khái niệm CSS

Trong lĩnh vực xây dựng, chúng ta có trang trí nội thất; trong lĩnh vực thẩm mỹ - làm đẹp, chúng ta có kỹ thuật make-up; còn trong lĩnh vực thiết kế web chúng ta có CSS. CSS (Cascading Style Sheets mà tạm dịch là tờ mẫu theo Style Sheets) là một ngôn ngữ quy định cách trình bày cho các tài liệu viết bằng HTML, XHTML, XML, SVG, hay UML

Nếu đã từng học qua HTML thì cũng biết HTML cũng hỗ trợ một số thuộc tính định dạng cơ bản cho text, picture, table, nhưng nó không thật sự phong phú và chính xác như nhau trên mọi hệ thống. CSS cung cấp cho hàng trăm thuộc tính trình bày dành cho các đối tượng với sự sáng tạo cao trong kết hợp các thuộc tính giúp mang lại hiệu quả. Ngoài ra, hiện tại CSS đã được hỗ trợ bởi tất cả các trình duyệt, nên hoàn toàn có thể tự tin trang web

của mình có thể hiển thị hầu như “như nhau” dù trên một hệ thống sử dụng Windows, Linux hay trên một máy Mac miễn là đang sử dụng một phiên bản trình duyệt mới nhất.

Sử dụng các mã định dạng trực tiếp trong HTML tốn hao nhiều thời gian thiết kế cũng như dung lượng lưu trữ trên đĩa cứng. Trong khi đó CSS đưa ra phương thức “tờ mẫu ngoại” giúp áp dụng một khuôn mẫu chuẩn từ một file CSS ở ngoài. Nó thật sự có hiệu quả đồng bộ khi tạo một website có hàng trăm trang hay cả khi muốn thay đổi một thuộc tính trình bày nào đó. Hãy thử tưởng tượng có một website với hàng trăm trang và muốn thay đổi font chữ hay màu chữ cho một thành phần nào đó. Đó thật sự sẽ là một công việc buồn chán và tốn nhiều thời gian. Nhưng với việc sử dụng CSS việc đó là hoàn toàn đơn giản cũng như là có một trò ma thuật nào đó.

Ngoài ra, CSS còn cho phép áp đặt những kiểu trình bày thích hợp hơn cho các phương tiện khác nhau như màn hình máy tính, máy in, điện thoại,...

CSS được cập nhật liên tục mang lại các trình bày phức tạp và tinh vi hơn.

2.20.2. Các loại style, mức độ ưu tiên style và các sử dụng

Cú pháp tạo một style

Để tạo một style ta sử dụng cú pháp cơ bản sau:

Selector { property:value;... }

Trong đó:

- *Selector*: Các đối tượng mà chúng ta sẽ áp dụng các thuộc tính trình bày. Nó là các thẻ HTML, class hay id

Trong CSS ngoài viết tên selector theo tên tag, class, id. Chúng ta còn có thể viết tên selector theo phân cấp như để chỉ các ảnh ở trong #entry, chúng ta viết selector là #entry img, như vậy thì các thuộc tính chỉ định sẽ chỉ áp dụng riêng cho các ảnh nằm trong #entry.

Khi viết tên cho class, đôi khi sẽ có nhiều thành phần có cùng class đó, ví dụ như thẻ img và thẻ a cùng có class tên visitors nhưng đây lại là hai đối tượng khác nhau, 1 cái là ảnh của người thăm, 1 cái là liên kết tới trang người thăm. Nên nếu khi viết CSS ta ghi là .visitors { width:50 } thì sẽ ảnh hưởng tới cả hai thành phần. Nên trong trường hợp này, nếu có ý dùng CSS đó chỉ riêng phần ảnh thì chỉ nên ghi là img .visitors thôi.

- *Property*: Chính là các thuộc tính quy định cách trình bày. Ví dụ: background-color, font-family, color, padding, margin,...

Mỗi thuộc tính CSS phải được gán một giá trị. Nếu có nhiều hơn một thuộc tính cho một selector thì chúng ta phải dùng một dấu ; (chấm phẩy) để phân cách các thuộc tính. Tất cả các thuộc tính trong một selector sẽ được đặt trong một cặp ngoặc nhọn sau selector.

Ví dụ 2.21: `body { background:#FFF; color:#FF0000; font-size:14pt }`

Các loại style

Chúng ta có ba cách khác nhau để tạo CSS vào trong một tài liệu HTML

Loại 1: CSS nội tuyến

Đây là một phương pháp nguyên thủy nhất để nhúng CSS vào một tài liệu HTML bằng cách nhúng vào từng thẻ HTML muốn áp dụng. Và dĩ nhiên trong trường hợp này chúng ta sẽ không cần selector trong cú pháp.

Ví dụ 2.22:

```
<html>
<head>
<title>Ví dụ</title>
</head>
<body style="background-color:#FFF;">
<p style="color:green">^_^ Welcome To Wall's Blog ^_^</p>
</body>
</html>
```

Loại 2: CSS Bên trong (thẻ style)

Thật ra nếu nhìn kỹ chúng ta cũng nhận ra đây chỉ là một phương cách thay thế cách thứ nhất bằng cách rút tất cả các thuộc tính CSS vào trong thẻ style (để tiện cho công tác bảo trì, sửa chữa).

Ví dụ 2.23:

```
<html>
<head>
<title>Ví dụ</title>
<style type="text/css">
  body { background-color:#FFF }
  p { color:#00FF00 }
</style>
</head>
<body>
  <p>^_^ Welcome To Wall's Blog ^_^</p>
</body>
</html>
```

Loại 3: CSS ngoại tuyến (liên kết với một file CSS bên ngoài)

Tương tự như cách 2 nhưng thay vì đặt tất cả các mã CSS trong thẻ style chúng ta sẽ đưa chúng vào trong một file CSS (có phần mở rộng .css) bên ngoài và liên kết nó vào trang web bằng thuộc tính href trong thẻ link.

Đây là cách làm được khuyến cáo, nó đặc biệt hữu ích cho việc đồng bộ hay bảo trì một website lớn sử dụng cùng một kiểu mẫu. Các ví dụ trong sách này cũng được trình bày theo kiểu này.

Ví dụ 2.24:

```
<html>
<head>
<title>Ví dụ</title>
  <link rel="stylesheet" type="text/css" href="style.css" />
</head>
  <body>
    <p>^_^ Welcome To Wall's Blog ^_^</p>
  </body>
</html>
```

Sau đó hãy tạo một file style.css với nội dung:

```
body {
background-color:#FFF
}
p {
color:#00FF00
}
```

Thứ tự ưu tiên:

Trước khi thực thi CSS cho một trang web. Trình duyệt sẽ đọc toàn bộ CSS mà trang web có thể được áp dụng, bao gồm: CSS mặc định của trình duyệt, file CSS bên ngoài liên kết vào trang web, CSS nhúng trong thẻ <style> và các CSS nội tuyến. Sau đó, trình duyệt sẽ tổng hợp toàn bộ CSS này vào một CSS ảo, và nếu có các thuộc tính CSS giống nhau thì thuộc tính CSS nào nằm sau sẽ được ưu tiên sử dụng. Theo nguyên tắc đó trình duyệt sẽ ưu tiên cho các CSS nội tuyến > CSS bên trong > CSS bên ngoài > CSS mặc định của trình duyệt.

2.20.3. Các khai báo sử dụng CSS

1). Style áp dụng cho thẻ cụ thể

Màu nền (thuộc tính background-color):

Thuộc tính background-color giúp định màu nền cho một thành phần trên trang web. Các giá trị mã màu của background-color cũng giống như color nhưng có thêm giá trị transparent để tạo nền trong suốt.

Ví dụ 2.25: sau đây sẽ chỉ cho chúng ta biết cách sử dụng thuộc tính background-color để định màu nền cho cả trang web, các thành phần h1, h2 lần lượt là xanh lơ, đỏ và cam.

```
Body {  
    background-color:cyan  
}  
H1{  
    background-color:red  
}  
H2{  
    background-color:orange  
}
```

Ảnh nền (thuộc tính background-image):

Việc sử dụng ảnh nền giúp trang web trông sinh động và bắt mắt hơn. Để chèn ảnh nền vào một thành phần trên trang web chúng ta sử dụng thuộc tính background-image.

Bây giờ chúng ta sẽ cùng làm một ví dụ minh họa để xem thuộc tính background-image sẽ hoạt động ra sao. Đầu tiên hãy tìm một tấm ảnh mà thích, ở đây sẽ lấy tấm ảnh logo của blog. Sau đó, chúng ta sẽ viết CSS để đặt logo này làm ảnh nền trang web như sau:

```
body  
{  
    background-image:url(logo.png)  
}
```

Thuộc tính font-family

Thuộc tính font-family có công dụng định nghĩa một danh sách ưu tiên các font sẽ được dùng để hiển thị một thành phần trang web. Theo đó, thì font đầu tiên được liệt kê trong danh sách sẽ được dùng để hiển thị trang web. Nếu như trên máy tính truy cập chưa cài đặt font này thì font thứ hai trong danh sách sẽ được ưu tiên...cho đến khi có một font phù hợp.

Có hai loại tên font được dùng để chỉ định trong font-family: family-names và generic families.

+ Family-names: Tên cụ thể của một font. Ví dụ: Arial, Verdana, Tohama,...

+ Generic families: Tên của một họ gồm nhiều font. Ví dụ: sans-serif, serif,...

Khi lên danh sách font dùng để hiển thị một trang web sẽ chọn những font mong muốn trang web sẽ được hiển thị để đặt ở các vị trí ưu tiên. Tuy nhiên, có thể những font này sẽ không thông dụng lắm nên cũng cần chỉ định thêm một số font thông dụng dự phần

như Arial, Tohama hay Times New Roman và cũng được đề nghị đặt vào danh sách font của mình một generic families (thường thì nó sẽ có độ ưu tiên thấp nhất). Thực hiện theo cách này thì sẽ đảm bảo trang web của có thể hiển thị tốt trên bất kỳ hệ thống nào.

Ví dụ 2.26: sau chúng ta sẽ viết CSS để quy định font chữ dùng cho cả trang web là Times New Roman, Tohama, sans-serif, và font chữ dùng để hiển thị các tiêu đề h1, h2, h3 sẽ là Arial, Verdana và các font họ serif.

```
body { font-family: "Times New Roman", Tohama, sans-serif }
```

```
h1, h2, h3 { font-family: arial, verdana, serif }
```

Mở trang web trong trình duyệt và kiểm tra kết quả. Chúng ta thấy phần tiêu đề sẽ được ưu tiên hiển thị bằng font Arial, nếu trên máy không có font này thì font Verdana sẽ được ưu tiên và kế đó sẽ là các font thuộc họ serif.

Thuộc tính font-style:

Thuộc tính font-style định nghĩa việc áp dụng các kiểu in thường (normal), in nghiêng (italic) hay xiên (oblique) lên các thành phần trang web. Trong ví dụ 2.27 bên dưới chúng ta sẽ thử thực hiện áp dụng kiểu in nghiêng cho thành phần h1 và kiểu xiên cho h2.

Ví dụ 2.27:

```
H1{
```

```
    font-style:italic;
```

```
}
```

```
H2{
```

```
    font-style:oblique;
```

```
}
```

Thuộc tính font-variant:

Thuộc tính font-variant được dùng để chọn giữa chế độ bình thường và small-caps của một font chữ.

Một font small-caps là một font sử dụng chữ in hoa có kích cỡ nhỏ hơn in hoa chuẩn để thay thế những chữ in thường. Nếu như font chữ dùng để hiển thị không có sẵn font small-caps thì trình duyệt sẽ hiện chữ in hoa để thay thế. Trong ví dụ 2.28 sau chúng ta sẽ sử dụng kiểu small-caps cho phần h1

Ví dụ 2.28 :

```
h1 {
```

```
    font-variant:small-caps
```

```
}
```

Thuộc tính font-weight:

Thuộc tính font-weight mô tả cách thức thể hiện của font chữ là ở dạng bình thường (normal) hay in đậm (bold). Ngoài ra, một số trình duyệt cũng hỗ trợ mô tả độ in đậm bằng các con số từ 100 – 900. Thử in đậm phần p:

```
p {  
    font-weight:bold  
}
```

Thuộc tính font-size:

Kích thước của một font được định bởi thuộc tính font-size. Thuộc tính này nhận các giá trị đơn vị đo hỗ trợ bởi CSS bên cạnh các giá trị xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger. Tùy theo mục đích sử dụng của website có thể lựa chọn những đơn vị phù hợp. Ví dụ trang web của phục vụ chủ yếu là những người già, thị lực kém hay những người dùng sử dụng các màn hình máy tính kém chất lượng thì có thể cân nhắc sử dụng các đơn vị qui đổi như em hay %. Như vậy sẽ đảm bảo font chữ trên trang web của luôn ở kích thước phù hợp.

Thuộc tính text-align :

Thuộc tính text-align giúp thêm các canh chỉnh văn bản cho các thành phần trong trang web.

Cũng tương tự như các lựa chọn canh chỉnh văn bản trong các trình soạn thảo văn bản thông dụng như MS Word, thuộc tính này có tất cả 4 giá trị : left (canh trái – mặc định), right (canh phải), center (canh giữa) và justify (canh đều).

Trong ví dụ 2.29 sau chúng ta sẽ thực hiện canh phải các thành phần h1, h2 và canh đều đối với thành phần <p>

Ví dụ 2.29:

```
h1, h2 {  
    text-align:right  
}  
p {  
    text-align:justify  
}
```

Thuộc tính letter-spacing:

Thuộc tính letter-spacing được dùng để định khoảng cách giữa các ký tự trong một đoạn văn bản.

Muốn định khoảng cách giữa các ký tự trong thành phần h1, h2 là 7px và thành phần <p> là 5px chúng ta sẽ viết CSS sau:

Ví dụ 2.30

```
h1, h2 {  
    letter-spacing:7px  
}  
p { letter-spacing:5px }
```

Thuộc tính text-decoration:

Thuộc tính text-decoration giúp thêm các hiệu ứng gạch chân (underline), gạch xuyên (line-through), gạch đầu (overline), và một hiệu ứng đặc biệt là văn bản nhấp nháy (blink).

Pseudo-classes For Links

Một thành phần rất quan trọng trong mọi website chính là liên kết. Cũng như một đối tượng văn bản thông thường, chúng ta hoàn toàn có thể áp dụng các thuộc tính định dạng đã học ở 2 bài trước như định font chữ, gạch chân, màu chữ cho một liên kết. Hơn nữa, CSS còn cung cấp một điều khiển đặc biệt được gọi là pseudo-classes. Pseudo-classes cho phép xác định các hiệu ứng định dạng cho một đối tượng liên kết ở một trạng thái xác định như khi liên kết chưa được thăm (a:link), khi rê chuột lên liên kết (a:hover), khi liên kết được thăm (a:visited) hay khi liên kết đang được kích hoạt – đang giữ nhấn chuột (a:active). Với điều khiển pseudo-classes cùng với các thuộc tính CSS đã học chắc chắn sẽ mang lại rất nhiều ý tưởng về trang trí liên kết cho trang web.

Sau đây chúng ta sẽ tiến hành một số ví dụ để tìm hiểu thêm về các khả năng trang trí cho một liên kết dựa trên pseudo-classes.

Ví dụ này chúng ta sẽ áp dụng 3 màu sắc khác nhau cho từng trạng thái liên kết: các liên kết chưa thăm có màu xanh lá; các liên kết mouse over sẽ có màu đỏ tươi; các liên kết đã thăm sẽ có màu đỏ.

Ví dụ 2.31:

```
a:link {  
    color:#00FF00  
}  
a:hover {  
    color:#FF00FF  
}  
a:visited {  
    color:#FF0000  
}
```

2). Tạo và sử dụng lớp (class)

Như chúng ta đã biết mỗi lệnh trong CSS sẽ định dạng cho một hoặc nhiều thành phần HTML nào đó trong trang web (các thẻ trong HTML). Nhưng có khi trong thiết kế

chúng ta mong muốn một số thành phần nào đó sẽ có những định dạng khác như khác về màu sắc, đường viền, size chữ ...

Ví dụ 2.32 : ta viết CSS sau:

```
p {color: black; font-size: 12px}
```

Khi đó tất cả các thành phần p sẽ có màu đen, kích cỡ size 12px. Nhưng chúng ta muốn một số thành phần p có màu xanh, một số thành phần p khác lại có màu vàng.

Class và ID được gắn thêm vào thẻ HTML giúp ta có thêm lựa chọn để giải quyết vấn đề trên. Lúc này class và id trở thành các CSS selector.

Class hay còn gọi là lớp dùng để chỉ một lớp các đối tượng có chung các thuộc tính. Các đối tượng trong cùng một class có tính thừa kế lẫn nhau, chính vì vậy trong một trang HTML của Chúng ta có thể có rất nhiều các đối tượng có thể thuộc một class nào đó. Ví dụ sau đây sẽ minh chứng cho về tính thừa kế và việc định nghĩa nhiều đối tượng cùng thuộc một class nào đó.

Để tạo và áp dụng Class đơn giản chúng ta thêm thuộc tính Class = “Tên class” vào thẻ tương ứng

Để định nghĩa class chúng ta sử dụng (.Tên class) như là một selector.

Ví dụ 2.32:

Ta sẽ định dạng cho một thành phần p có màu xanh, và một thành phần p có màu vàng. Và một thành phần p được hiển thị bình thường như sau:

```
<p>Đoạn văn bản hiển thị bình thường màu đen</p>
```

```
<p class="blue">Đoạn văn bản hiển thị màu xanh</p>
```

```
<p class="yellow">Đoạn văn bản hiển thị màu vàng</p>
```

Như vậy ở trên chúng ta cần phải định nghĩa 2 lớp đó là class blue và yellow. Để định nghĩa 2 lớp này ta làm như sau:

```
.blue{ color: blue}
```

```
.yellow{ color: yellow}
```

3). Tạo và sử dụng định danh (id)

id hay là index(chỉ số) được dùng để xác định một đối tượng duy nhất nào đó. Do đó trong một trang HTML mỗi đối tượng chỉ có duy nhất một id, nếu có 2 đối tượng trở lên có id giống nhau trong cùng một trang thì điều đó có nghĩa là trang đó của chúng ta không validate HTML.

Cách sử dụng id hoàn toàn giống class, ta chỉ thay dấu . bằng dấu # trước tên id khi định nghĩa.

Ví dụ 2.33:

```
<p id="red">Đoạn văn bản hiển thị màu đỏ</p>
```

```
#red{ color: red}
```

Sự khác biệt lớn nhất giữa class và Id là ID là duy nhất, còn class thì có nhiều. Trong một trang web không thể tồn tại hai id có cùng tên. Nhưng có thể sử dụng nhiều css có cùng tên cho các thành phần khác nhau.

Điểm khác biệt thứ 2 giữa class và id là có thể sử dụng nhiều class cho một thành phần nào đó, còn id thì chỉ một. Đoạn mã sau cho thấy cách sử dụng 2 class có tên bor và top cho một thành phần p

2.20.4. Định dạng bằng CSS

Như phần trên đã giới thiệu, css có thể giúp chúng ta trình bày một tài liệu HTML sinh động, đẹp mắt, với các thuộc tính phong phú và đa dạng của css có thể giúp chúng ta định dạng, căn chỉnh các tài liệu trên trang web một cách đơn giản và chuyên nghiệp và có thể áp dụng cho toàn bộ website. Phần trên chúng ta đã nghiên cứu các thuộc tính hay dùng trong css giúp định dạng và căn chỉnh một tài liệu html. Trong phần này chúng ta sẽ tìm hiểu một số ví dụ cụ thể dùng css định dạng các tài liệu trong html như định dạng văn bản, định dạng hình ảnh, ...

Định dạng văn bản

Các đoạn văn trong các văn bản trên web cần phải được trình bày thống nhất như màu chữ, cỡ chữ, kiểu chữ, căn lề, khoảng cách dòng ...

Ví dụ 2.34: sau đây sẽ tạo ra một css giúp căn chỉnh các đoạn văn trong tài liệu HTML

```
p { font-family: Times New Roman;
    font-size: 14pt;
    text-align: justify;
    line-height: 150%;
    text-indent: 6;
    margin-top: 6 ;
    color:#008080
}
```

Định dạng hình ảnh

Ví dụ 2.35: sau đây sẽ định dạng kích thước của các hình ảnh khi chèn vào trang web

```
img {
    clip:rect(50px 218px 155px 82px);
}
```

Tạo đường viền

Ví dụ 2.36: này sẽ định dạng đường viền có bo góc

```
.rounded_corner {
-moz-border-radius:10px;
```

```
-webkit-border-radius:10px;  
width:400px;  
height:100px;  
background-color:#000;  
}
```

Định dạng liên kết

Ví dụ 2.37: này sẽ định dạng cho văn bản liên kết trên web

```
a {  
border:1px solid #000;  
font-size:14px  
}  
a:link {  
color:#00FF00;  
}  
a:hover {  
background-color:#00BFF3;  
color:#FF00FF;  
font-size:1.2em;  
text-decoration:blink  
}  
a:visited {  
background-color:#FFF568;  
color:#FF0000;  
text-decoration:none  
}  
a:active { color:#662D91; font-variant:small-caps }
```

2.21. Công cụ trong thiết kế Web

2.21.1. Tổng quan về các loại công cụ

Ngày nay có rất nhiều công cụ hỗ trợ chúng ta trong việc thiết kế và xây dựng website. Nhờ có các công cụ này mà việc tạo một website đã trở nên trực quan hơn, đơn giản hơn, và trang web trở nên đẹp hơn.

Hiện có nhiều loại công cụ khác nhau được giới thiệu với nhiều loại tính năng khác nhau với những mức độ chuyên nghiệp khác nhau. Việc lựa chọn công cụ nào là tùy thuộc vào khả năng và trình độ của người thiết kế tuy nhiên chúng ta thấy hầu hết các công cụ thuộc vào một trong các nhóm công cụ sau:

- Công cụ soạn thảo
- Công cụ tạo và xử lý ảnh tĩnh
- Công cụ tạo và xử lý ảnh động
- Công cụ download, upload trang web

2.21.2. Công cụ soạn thảo

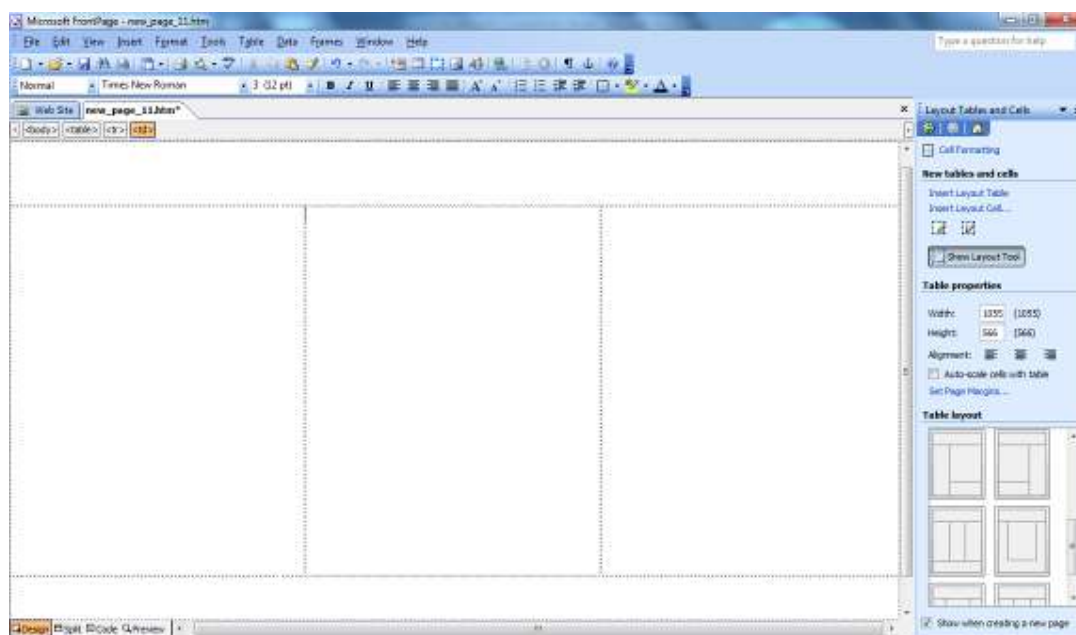
Các công cụ soạn thảo giúp cho chúng ta có thể dễ dàng tạo ra một trang web bằng việc kích chuột và lựa chọn các đối tượng, chương trình sẽ tự động giúp sinh mã nguồn HTML để tạo ra các trang web tương ứng. Các chương trình này còn giúp chúng ta dễ dàng tạo bố cục các trang một cách trực quan. Ngoài ra các chương trình còn hỗ trợ việc viết các mã nguồn html để tạo các trang web hay lập trình bằng các ngôn ngữ khác như javascript, vbscript ...

Hiện có 2 công cụ được nhiều người lựa chọn sử dụng đó là Microsoft FrontPage và Adobe Dreamweaver

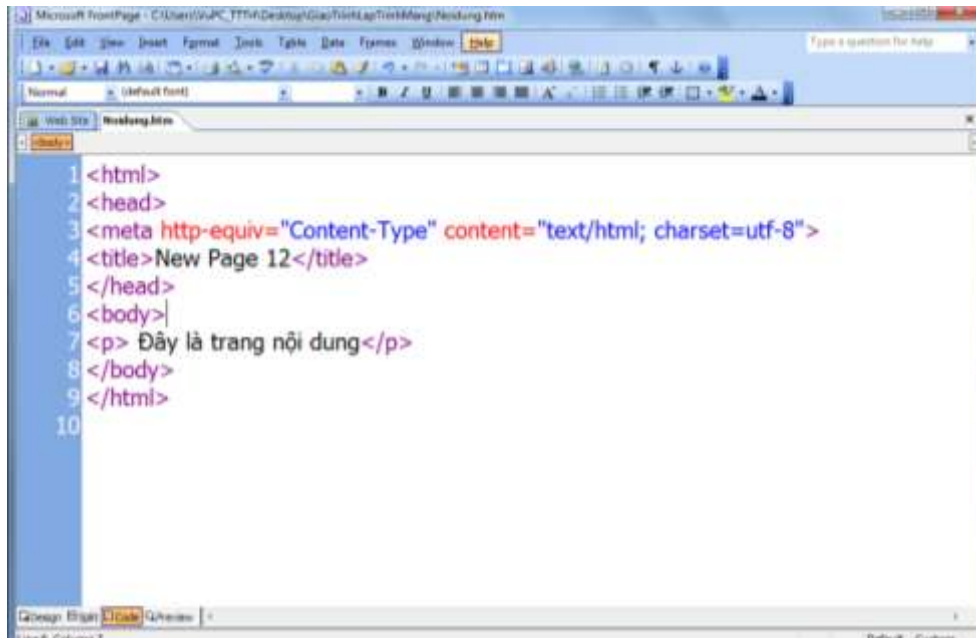
Microsoft FrontPage

Microsoft Frontpage là một phần mềm thiết kế web chuyên nghiệp của hãng phần mềm Microsoft. Phần mềm này được đi kèm với bộ Office, bao gồm những tính năng vượt trội nhưng lại rất dễ sử dụng.

Với Frontpage chúng ta có thể dễ dàng tạo một trang web đơn, một web site, hoặc dùng để cập nhật cho những trang web đã có sẵn mà không cần có những kiến thức như là Java, lập trình hoặc thậm chí ngôn ngữ HTML. Vì Frontpage biến việc thiết kế web đơn giản như khi ta soạn thảo một văn bản trong Microsoft Word vậy.



Hình 2.24. Microsoft Frontpage sử dụng trong thiết kế

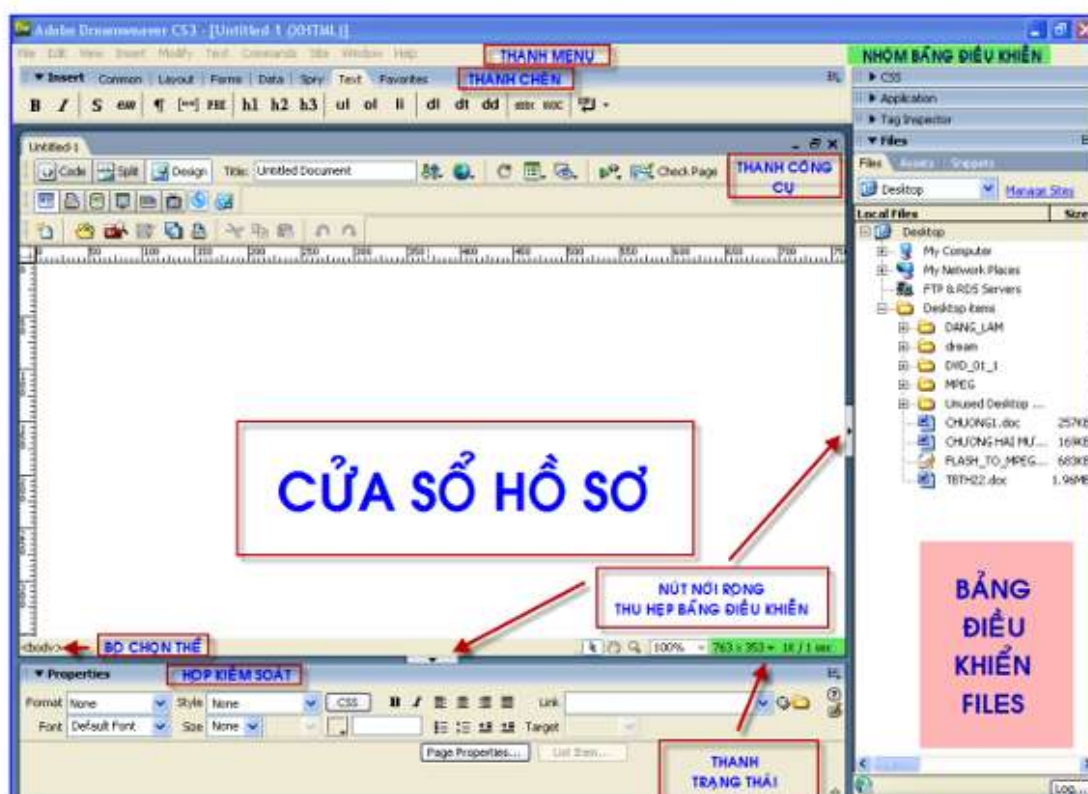


Hình 2.25. Microsoft Fontpage soạn thảo mã nguồn

Dreamweaver

Là một phần mềm thiết kế Web chuyên nghiệp, Tương thích với nhiều đối tượng nhúng (Flash, Shockwave, Active X, ...), Hỗ trợ các công cụ thiết kế trang Web động rất hiệu quả.

Cũng như Frontpage, Dreamweaver cho phép chúng ta dễ dàng tạo và thiết kế một trang web hay website bằng cả 2 hình thức thiết kế trực quan như word và vết bằng lệnh của ngôn ngữ lập trình như HTML, javascript, vbscript, php, asp...



Hình 2.26. Màn hình soạn thảo trang web của DreamWeaver

2.21.3. Công cụ tạo và xử lý ảnh tĩnh

Hình ảnh là một phần không thể thiếu được trong trang web, hình ảnh giúp ta mô tả các nội dung một cách sinh động, đơn giản và dễ hiểu hơn.

Ngày nay có nhiều công cụ giúp ta xử lý hình ảnh như, lắp ghép, tạo và chỉnh sửa ảnh, một trong những công cụ dễ làm, chuyên nghiệp được nhiều người lựa chọn đó là phần mềm Photoshop.

Adobe Photoshop là một phần mềm xử lý ảnh chuyên nghiệp. Photoshop cho phép người sử dụng tút sửa ảnh (retouching), ghép ảnh (composing), phục chế ảnh (restoration), tô màu tranh ảnh (painting)... một cách dễ dàng và hiệu quả.

Phần mềm này là một công cụ không thể thiếu của các nhiếp ảnh gia, các nhà thiết kế đồ họa, thiết kế web và biên tập video.

Ngoài ra Adobe Photoshop CS4 còn phối hợp rất tốt với các phần mềm khác của hãng Adobe như:

- Phần mềm vẽ trang trí và minh họa Adobe Illustrator.
- Phần mềm sắp chữ và trình bày Adobe InDesign.
- Phần mềm tạo hình ảnh động Adobe Flash
- Phần mềm thiết kế trang web Adobe Dreamweaver

2.21.4. Công cụ Flash, tạo và xử lý ảnh động

Ảnh động là không thể thiếu được trên trang web chẳng hạn như banner, logo, hay các đoạn phim trên web. Việc đưa ảnh động vào web giúp cho trang web sinh động hơn, đẹp hơn và dễ chuyển tải các nội dung hơn, có thể thực hiện các mô phỏng trực quan sinh động.

Flash là một phần mềm cho phép tạo và xử lý ảnh động cho trang web tốt nhất hiện nay. Adobe Flash (trước đây là Macromedia Flash và trước đó FutureSplash), hay còn gọi một cách đơn giản là Flash, được dùng để chỉ chương trình sáng tạo đa phương tiện (multimedia) lẫn phần mềm dùng để hiển thị chúng Macromedia Flash Player. Chương trình điện toán này được viết và phân phối bởi Adobe Systems (công ty đã mua Macromedia). Flash dùng kỹ thuật đồ họa vectơ và đồ họa điểm (raster graphics). Ngoài ra Flash còn có một ngôn ngữ văn lệnh riêng gọi là ActionScript và có khả năng truyền và tải luồng âm thanh hoặc hình ảnh. Đúng ra thì từ Macromedia Flash nên được dùng để chỉ chương trình tạo ra các tập tin Flash. Còn từ Flash Player nên được dành để chỉ ứng dụng có nhiệm vụ thi hành hay hiển thị các tập tin Flash đó. Tuy vậy, chữ Flash được dùng để chỉ cả hai chương trình nói trên.

Flash là công cụ để phát triển các ứng dụng như thiết kế các phần mềm mô phỏng. Sử dụng ngôn ngữ lập trình ActionScript để tạo các tương tác, các hoạt cảnh trong phim. Điểm mạnh của Flash là có thể nhúng các file âm thanh, hình ảnh động. Người lập trình có thể chủ động lập các điều hướng cho chương trình. Flash cũng có thể xuất bản đa dạng các file kiểu html, exe, jpg,...để phù hợp với các ứng dụng của người sử dụng như trên Web, CD,

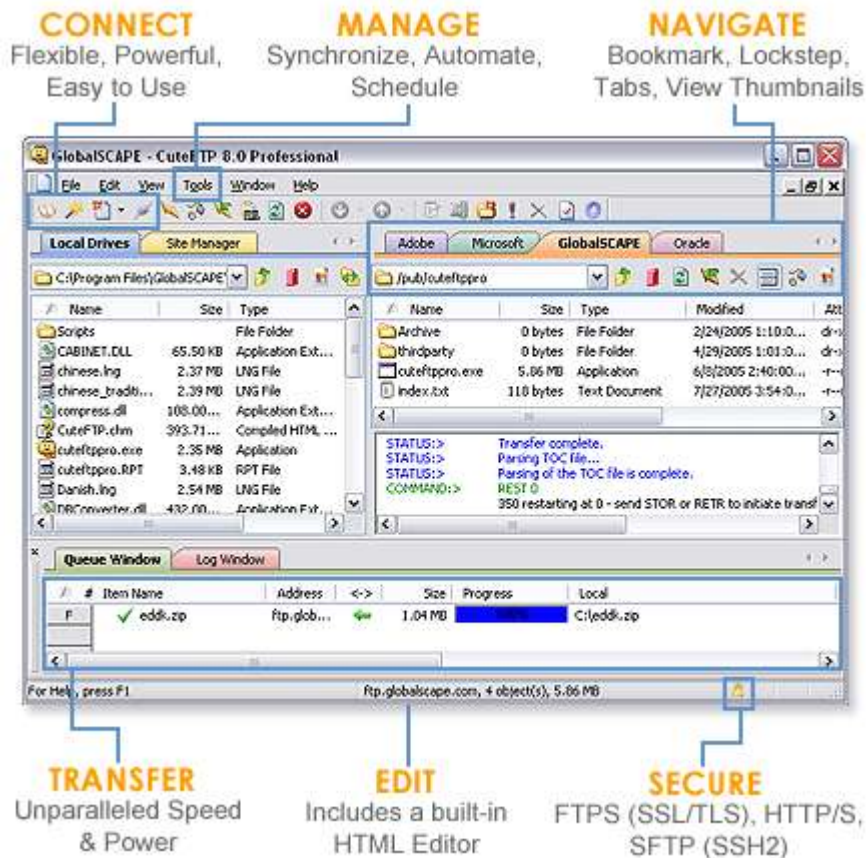
...

Các tập tin Flash, thường thường mang phần mở rộng là .swf và có thể hiện thị bởi các chương trình duyệt trang Web hay ứng dụng Flash Player. Các tập tin Flash thường là hoạt họa, quảng cáo hay các thành phần trang trí của các trang Web. Gần đây Flash còn được sử dụng để tạo ra các ứng dụng Internet phong phú. Với một kích thước tương tự, một tập tin Flash có thể chứa nhiều thông tin hơn là một tập tin hình dạng GIF hay dạng JPEG.

2.21.5. Công cụ Download, Upload trang Web

Với một người làm về web thì việc trao đổi thông tin trên mạng là cần thiết chẳng hạn như phải upload toàn dữ liệu lên web, hay Download tài liệu về. Hiện có nhiều công cụ giúp chúng ta thực hiện việc truyền file. Một trong những công cụ đó là CuteFTP Professional.

CuteFTP Professional là một phần mềm để chuyển files qua nhiều giao thức internet giữa máy tính của Chúng ta tới các máy tính bất kỳ nơi nào trên internet một cách an toàn và hiệu quả. Chức năng Connection Wizard tích hợp trong phần mềm sẽ hướng dẫn kết nối tới một site ftp trong vài giây và giao diện thân thiện của phần mềm sẽ giúp Chúng ta chuyển file dễ dàng ngay khi Chúng ta là người mới sử dụng. Kể cả việc xuất một trang web, download những hình ảnh kỹ thuật số, phần mềm, file nhạc mới nhất hoặc chuyển file có dung lượng lớn giữa các văn phòng chi nhánh.



Hình 2.27. Giao diện của chương trình CuteFTP Professional

CuteFTP Professional cung cấp các công cụ đơn giản mà mạnh mẽ để giải quyết những thách thức phức tạp trong quản lý dữ liệu và giúp hoàn tất các yêu cầu về HIPAA, GLBA về Sarbanes-Oxley.

- CuteFTP Professional được giới chuyên môn đánh giá cao nhất về độ an toàn và tiện ích nó mang lại khi vận chuyển files. Được bình chọn đứng thứ 2 trong toptenreviews 2007...

- Tương thích với Windows 98/ME/NT/2000/XP
- Tương thích với Windows 2003 & IIS6
- Dễ dàng với RFC 959, 2228, 2389, 2289 & hơn nữa
- Dễ dàng với OpenSSL Libraries 0.9.7d
- Dễ dàng với ZLIB Libraries 1.2.1
- Dễ dàng với với những phương thức chuyển thông tin mới nhất (Secure Shell)
- Khởi động nhanh (Delay Loaded Components)

Câu hỏi và Bài tập chương 2

1. Tiêu đề trong trang Web có mấy mức chính?
2. Nêu các thuộc tính của thẻ Paragraph (<P>). Hãy tạo một trang web đơn giản trong đó có sử dụng tiêu đề mức 2 và thẻ Paragraph (<P>).
3. Tạo một trang web đơn giản tự giới thiệu về mình, chúng ta bao nhiêu tuổi, chúng ta làm gì, sở thích của bạn.
4. Áp dụng những gì đã học ở trên, hãy tạo một danh sách như sau:
 - Top level 1
 - Top level 2

1. Second Level

- Fact 1
- Fact 2

- Top level 3

5. Áp dụng những gì đã học ở trên, hãy tạo bảng giống như sau:

Tổng kết năm học 2003 - 2004			
Thống kê	HS giỏi	HS khá	HS trung bình
Lớp 1A	35	4	0
Lớp 1B	20	10	12
Lớp 1C	5	25	20

6. Hãy cho biết tại sao các ảnh dưới đây có thể được nhúng vào trong một tài liệu HTML? Viết đoạn code được dùng để nhúng các ảnh đó.



7. Tạo một trang web và chèn các ảnh sao cho ảnh nằm ở phía bên trái, và các đoạn văn bản nằm bên phải.
8. Tạo một trang web và chèn các ảnh sao cho ảnh được căn giữa và các đoạn văn bản nằm ở trên hoặc ở bên dưới.

9. Tạo một trang khác giới thiệu chi tiết hơn về mình và có liên kết tới trang giới thiệu chính của mình.

10. Chèn 2 ảnh vào trong trang giới thiệu chính sao cho khi click vào ảnh đó thì trang giới thiệu chi tiết được mở ra. Yêu cầu: ảnh thứ nhất sử dụng đường dẫn tuyệt đối, ảnh thứ hai sử dụng đường dẫn tương đối tới trang cần liên kết.

11. Tạo một trang thứ ba trong đó có chèn một đoạn Video clips ở giữa trang.

12. Sửa trang giới thiệu tóm tắt về mình sao cho nền của trang đó là một ảnh bất kỳ, trang giới thiệu chi tiết có nền màu xanh nước biển nhạt.

13. Dùng thẻ IFRAME để nhúng trang thứ 3 vào trong trang giới thiệu tóm tắt (trang chủ).

14. Tạo ra một trang web mới dùng để đăng nhập có giao diện như ở hình dưới (sử dụng FORM).

Đăng nhập

Tên người dùng:

Mật khẩu:

[Xin bạn điền vào form](#)

15. Tạo một file css để đặt mặc định kiểu font và màu cho các phần giới thiệu của bạn.

16. Hãy tạo ra trang web cho phép đăng ký thông tin người dùng trên mạng theo mẫu sau:

không thể truy cập tài khoản của tôi

Tên tài Họ Tên

Giới tính

Ngày sinh Ngày Tháng Năm

Tôi sống tại

Chọn tên truy nhập và mật khẩu

Tên truy nhập Yahoo! và Email @ yahoo.com.vn

Mật khẩu

Đánh lại mật khẩu

Trong trường hợp bạn quên tên truy nhập hoặc mật khẩu...

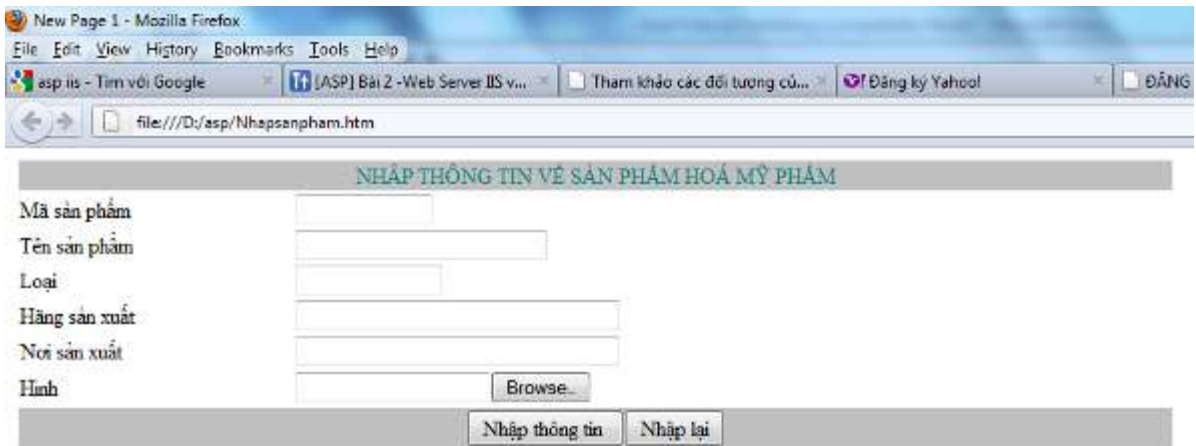
Câu hỏi bí mật 1

Câu trả lời của bạn

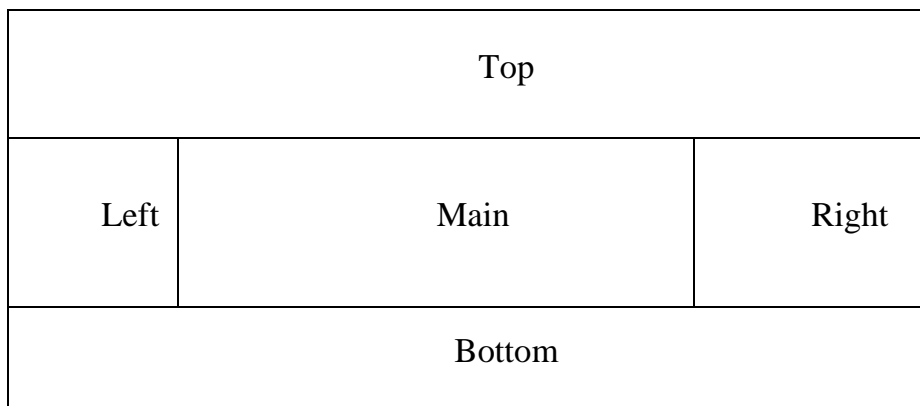
Câu hỏi bí mật 2

Câu trả lời của bạn

17. Hãy tạo trang web cho phép nhập thông tin về hàng hóa như sau:



18. Hãy thiết kế một website giải trí có bố cục như sau



Trong đó:

- Phần top hiển thị nội dung trang banner.htm
- Phần left hiển thị nội dung trang login.htm
- Phần Right hiển thị nội dung trang menuright.htm
- Phần Bottom hiển thị nội dung trang address.htm
- Phần main hiển thị các trang khi liên kết đến