



BỘ CÔNG THƯƠNG
TRƯỜNG CAO ĐẲNG KỸ THUẬT CAO THẮNG
KHOA ĐIỆN TỬ TIN HỌC



LƯU VĂN ĐẠI
LẠI NGUYỄN DUY

GIÁO TRÌNH THIẾT KẾ VÀ QUẢN TRỊ WEB



THÀNH PHỐ HỒ CHÍ MINH – 2017
(LƯU HÀNH NỘI BỘ)

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN VỀ THIẾT KẾ WEB	1
1.1. Khái quát về Internet	1
1.2. Phương thức kết nối	3
1.3. Địa chỉ IP và tên miền	5
1.4. World Wide Web và HTML	9
1.5. Các nhà cung cấp dịch vụ Internet	12
1.6. Giới thiệu một số dịch vụ Internet thông dụng	13
1.7. Thiết kế web	17
CHƯƠNG 2. TỔNG QUAN VỀ HTML	24
2.1. Các thẻ định cấu trúc tài liệu	24
2.2. Các thẻ định dạng khối	25
2.3. Các thẻ định dạng danh sách	26
2.4. Các thẻ định dạng ký tự	27
2.5. Các thẻ chèn âm thanh, hình ảnh	33
2.6. Các thẻ định dạng bảng biểu	36
2.7. FORM	37
CHƯƠNG 3. SỬ DỤNG DREAMWEAVER THIẾT KẾ WEBSITE	43
3.1. Tạo thư mục chứa bộ web	43
3.2. Tạo mới một trang web	44
3.3. Lưu một trang web	44
3.4. Định dạng trang web	44
3.5. Xem trang web trên trình duyệt	46
3.6. Tạo bảng trong trang web	46
3.7. Chèn hình vào trang web	49
3.8. Chèn ảnh động flash, video clip vào trang web.	51
3.9. Tạo menu cho trang web.	51
3.10. Tạo liên kết cho trang web.	57
3.11. Cách tạo Template	58
3.12. Tạo trang web mới từ template	60
3.13. Đưa website lên hosting	61
CHƯƠNG 4. CASCADING STYLE SHEETS	65
4.1. CSS là gì?	65

4.2. Lợi ích việc sử dụng CSS	65
4.3. Sử dụng CSS	65
4.4. Cú pháp CSS.....	66
4.5. Thứ tự ưu tiên và tính kế thừa	70
4.6 Các thuộc tính.....	79
CHƯƠNG 5. JAVASCRIPT	126
5.1. JavaScript là gì	126
5.2. Chèn mã Javascript.....	126
5.3. THẺ <NOSCRIPT> VÀ </NOSCRIPT>	127
5.4 Các lệnh xuất thông báo cơ bản.....	127
5.5 Chèn chuỗi vào nội dung trang HTML.....	128
5.6 Biến-Kiểu dữ liệu.....	129
5.7 Các Lệnh	131
5.8 Hàm (FUNCTIONS)	138
5.9 Sự kiện	142
5.10. Các đối tượng trong javascript	146
5.11. Các phần tử của đối tượng form.....	164
5.12. JQuery cơ bản	186
CHƯƠNG 6. QUẢN TRỊ WEBSITE	222
6.1. Đăng kí website miễn phí trên internet	222
6.2. WordPress là gì?.....	224
6.3. WordPress.com và WordPress.org khác nhau như thế nào?	225
6.4. Cài Đặt WordPress.....	226
6.5. Các Chức Năng Chính Trong WordPress	233

CHƯƠNG 1. TỔNG QUAN VỀ THIẾT KẾ WEB

1.1. Khái quát về Internet

1.1.1. Internet là gì?

Internet là một liên mạng máy tính toàn cầu được hình thành từ các mạng nhỏ hơn, liên kết hàng triệu máy tính trên thế giới thông qua cơ sở hạ tầng viễn thông. Internet là mạng của các mạng máy tính. Trong mạng này, các máy tính và thiết bị mạng giao tiếp với nhau bằng một ngôn ngữ thống nhất. Đó là bộ giao thức TCP/IP (Transmission Control Protocol – Internet Protocol).

1.1.2. Lịch sử phát triển

Internet được hình thành từ năm 1969, từ một dự án nghiên cứu của Bộ quốc phòng Mỹ. Lúc đó Internet chỉ liên kết 4 địa điểm: Viện Nghiên cứu Stanford, Trường Đại học tổng hợp California ở Los Angeles, UC – Santa Barbara và Trường Đại học Tổng hợp Utah. Mạng này được biết đến dưới cái tên ARPANET.

ARPANET càng phát triển khi có nhiều máy nối vào – rất nhiều trong số này là từ các cơ quan của Bộ quốc phòng Mỹ hoặc những trường đại học nghiên cứu với các đầu nối vào Bộ quốc phòng. Đây là những giao điểm trên mạng. Trong khi ARPANET đang cố gắng chiếm lĩnh mạng quốc gia thì một nghiên cứu tại Trung tâm nghiên cứu Palo Alto của công ty Xerox đã phát triển một kỹ thuật được sử dụng trong mạng cục bộ là Ethernet.

Theo thời gian, Ethernet trở thành một trong những chuẩn quan trọng để kết nối trong các mạng cục bộ. Cũng trong thời gian này, DARPA (đặt lại tên từ ARPA) chuyển sang hợp nhất TCP/IP (giao thức được sử dụng trong việc truyền thông trên Internet) vào phiên bản hệ điều hành UNIX của trường đại học tổng hợp California ở Berkeley. Với sự hợp nhất như vậy, những trạm làm việc độc lập sử dụng UNIX đã tạo nên một thế mạnh trên thị trường, TCP/IP cũng có thể dễ dàng tích hợp vào phần mềm hệ điều hành. TCP/IP trên Ethernet đã trở thành một cách thức thông dụng để trạm làm việc nối đến trạm khác.

Trong thập kỷ 1980, máy tính cá nhân được sử dụng rộng rãi trong các công ty và trường Đại học trên thế giới. Mạng Ethernet kết nối các máy tính cá nhân (PC) trở thành phổ biến. Các nhà sản xuất phần mềm thương mại cũng đưa ra những chương trình cho phép các máy PC và máy UNIX giao tiếp cùng một ngôn ngữ trên mạng.

Vào giữa thập kỷ 1980, giao thức TCP/IP được dùng trong một số kết nối khu vực – khu vực (liên khu vực) và cũng được sử dụng cho các mạng cục bộ và mạng liên khu vực.

Thuật ngữ "Internet" xuất hiện lần đầu vào khoảng 1974 trong khi mạng vẫn được gọi là ARPANET. Vào thời điểm này, ARPANET (hay Internet) còn ở qui mô rất nhỏ.

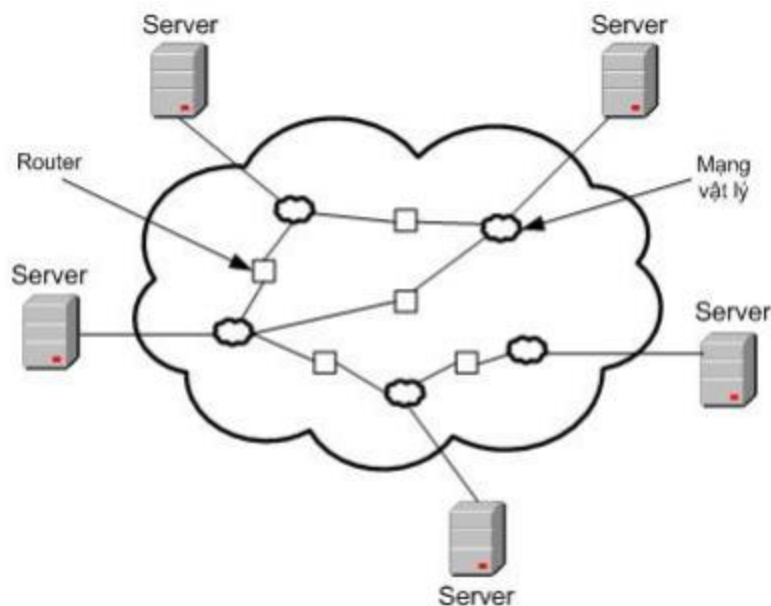
Mốc lịch sử quan trọng của Internet được chọn vào giữa thập kỷ 1980, khi Quỹ khoa học quốc gia Mỹ NSF (National Science Foundation) thành lập mạng liên kết các trung tâm máy tính lớn với nhau gọi là NSFNET. Mạng này chính là mạng Internet. Điểm quan trọng của NSFNET là cho phép mọi người cùng sử dụng. Trước NSFNET, chỉ các nhà khoa học, chuyên gia máy tính và nhân viên các cơ quan Chính phủ được kết nối Internet.

Nhiều doanh nghiệp đã chuyển từ ARPANET sang NSFNET. Chính vì vậy, sau gần 20 năm ARPANET trở nên không còn hiệu quả và đã ngừng hoạt động vào khoảng năm 1990.

Ngày nay, mạng Internet phát triển mạnh mẽ hơn các phương tiện truyền thông truyền thống khác như phát thanh và truyền hình, do sự cải tiến và phát triển không ngừng. Các công nghệ đang áp dụng trên Internet giúp cho Internet trở thành mạng liên kết vô số kho thông tin toàn cầu, có dịch vụ phong phú về nội dung, hình thức. Đó cũng chính là điều thúc đẩy chúng ta nên bắt đầu ngay với hành trình khám phá thế giới mới – thế giới Internet.

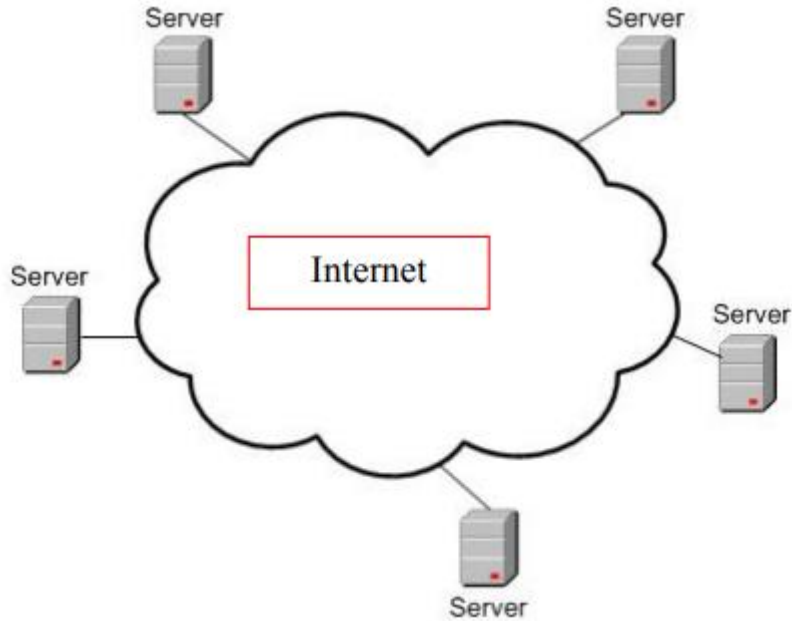
1.1.3. Cấu trúc Internet

Internet là một liên mạng kết nối các mạng nhỏ hơn với nhau. Như vậy, cấu trúc Internet gồm các mạng máy tính được kết nối với nhau thông qua các kết nối viễn thông. Thiết bị dùng để kết nối các mạng máy tính với nhau là cổng nối Internet (Internet Gateway) hoặc Bộ định tuyến (Router).



Hình 1.1. Cấu trúc Internet

Tuy nhiên, đối với người dùng, Internet chỉ là một mạng duy nhất.



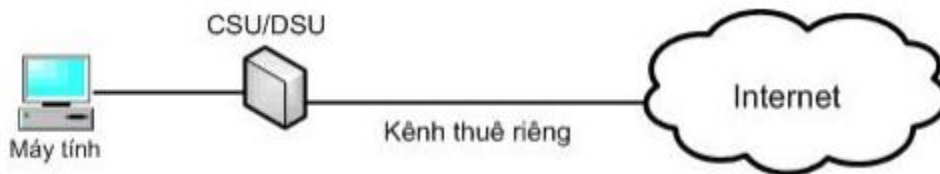
Hình 1.2. Internet dưới góc nhìn của người sử dụng

1.2. Phương thức kết nối

Để có thể sử dụng các dịch vụ Internet, người dùng phải kết nối máy tính của mình với Internet. Có nhiều phương thức kết nối với nhiều tốc độ khác nhau, tùy thuộc vào nhu cầu sử dụng và điều kiện của người sử dụng.

1.2.1. Kết nối thông qua kênh thuê riêng

Trong phương thức kết nối này, máy tính hay mạng máy tính của người sử dụng được kết nối trực tiếp tới nhà cung cấp dịch vụ Internet thông qua một kênh thuê riêng do nhà cung cấp dịch vụ Viễn thông cấp.

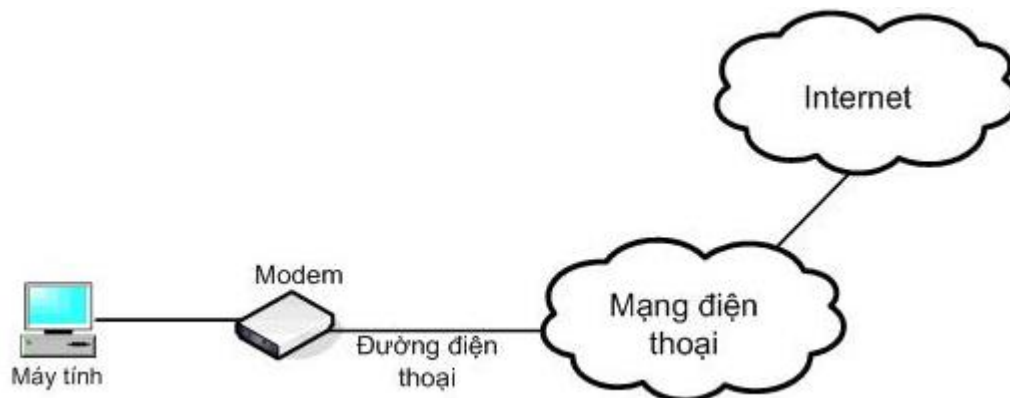


Hình 1.3. Kết nối qua kênh thuê riêng

Đặc điểm của phương thức này là kết nối luôn thường trực, nghĩa là bạn có thể truy nhập Internet bất cứ lúc nào. Tuy nhiên, giá thành sử dụng kết nối này rất cao vì bạn phải trả tiền thuê bao theo tháng chứ không phải trả theo dung lượng sử dụng. Phương thức kết nối này thường được những nhà cung cấp dịch vụ trực tuyến sử dụng.

1.2.2. Kết nối quay số qua mạng điện thoại

Trong phương thức kết nối này, người dùng kết nối với Internet thông qua mạng điện thoại. Để kết nối, người dùng cần có một đường điện thoại và một thiết bị kết nối có tên modem. Máy tính của người dùng kết nối với Modem và modem được kết nối tới đường điện thoại.



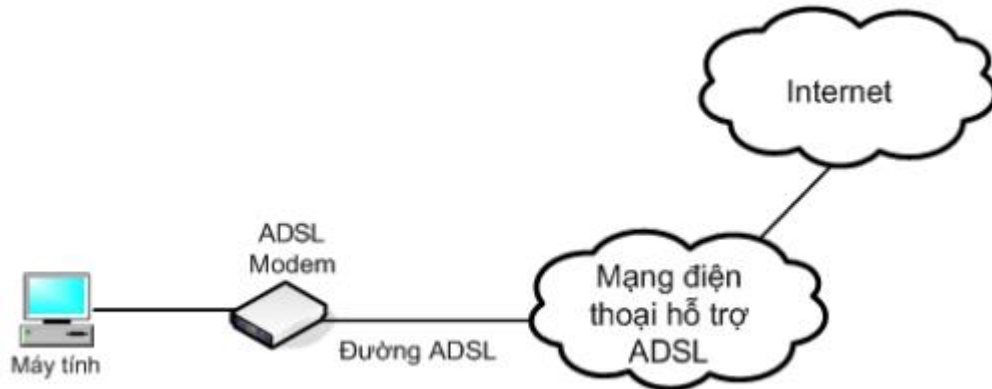
Hình 1.4. Kết nối quay số qua mạng điện thoại

Hiện nay, dịch vụ kết nối này đều được các nhà cung cấp dịch vụ Viễn thông cung cấp. Khi người sử dụng đăng ký, nhà cung cấp sẽ cấp cho họ một tài khoản để truy nhập và số điện thoại cần gọi. Kết nối kiểu này không luôn thường trực. Khi muốn sử dụng dịch vụ, người dùng phải quay số đến số điện thoại do nhà cung cấp dịch vụ cấp. Sau đó nhập tên truy nhập và mật khẩu để đăng nhập. Kiểu kết nối này thường được người dùng cá nhân sử dụng vì giá thành rẻ và dễ lắp đặt.

1.2.3. Kết nối qua ADSL

Kết nối Internet qua ADSL là một dịch vụ mới và đang rất phổ biến. ADSL là công nghệ truy nhập bất đối xứng, tốc độ đường xuống lớn hơn tốc độ đường lên. Đặc điểm này rất phù hợp với truy nhập Internet vì người dùng thường lấy thông tin từ Internet xuống nhiều hơn gửi thông tin lên Internet.

Người dùng có thể đăng ký dịch vụ này ngay trên đường dây điện thoại sẵn có của mình chứ không nhất thiết phải mắc thêm một đường dây mới. Để sử dụng, người dùng cần có ADSL modem. Máy tính của người dùng kết nối tới ADSL modem và modem này được kết nối với đường dây điện thoại đã đăng ký dịch vụ ADSL.



Hình 1.5. Kết nối qua ADSL

Đặc điểm của phương thức này là kết nối mạng cũng luôn thường trực (sau khi kết nối được tự động thực hiện) nhưng người dùng chỉ phải trả tiền cho những thời gian sử dụng. Cụ thể, các nhà cung cấp dịch vụ hiện nay đều tính cước dựa trên dung lượng thông tin người dùng tải xuống và tải lên Internet.

1.3. Địa chỉ IP và tên miền

1.3.1. Địa chỉ IP

Các máy tính trên Internet giao tiếp với nhau sử dụng bộ giao thức TCP/IP. Để các máy tính có thể liên lạc với nhau, mỗi máy tính cần có một địa chỉ liên lạc và địa chỉ này phải là duy nhất. Điều này cũng giống như các thuê bao trong mạng điện thoại di động phải có một số hiệu thuê bao (số máy) và số thuê bao này phải là duy nhất trong mạng.

Bộ giao thức TCP/IP sử dụng địa chỉ IP để đánh địa chỉ cho các máy tính trong mạng. Mỗi địa chỉ IP bao gồm 32 bit, được chia thành 4 nhóm đều nhau, mỗi nhóm 8 bit. Các nhóm này được phân tách với nhau bởi một dấu chấm “.”. Cách biểu diễn địa chỉ IP phổ biến nhất là “thập phân dấu chấm”. Trong cách biểu diễn này, địa chỉ IP được chia thành 4 nhóm, mỗi nhóm là một số thập phân và được phân tách với nhau bởi một dấu chấm.

Cấu trúc địa chỉ IP là: A.B.C.D

Trong đó: A, B, C, D là các số thập phân. Dó mỗi số thập phân này đều được chuyển từ một số nhị phân 8 bit nên giá trị của chúng phải nằm trong khoảng từ 0 đến 255.

Một số ví dụ về địa chỉ IP:

- 10.10.10.10
- 128.3.5.7
- 192.168.10.1

Địa chỉ IP đang được sử dụng hiện tại là (IPv4) có 32 bit. Hiện nay một số quốc gia đã đưa vào sử dụng địa chỉ IPv6 nhằm mở rộng không gian địa chỉ và những ứng dụng mới, IPv6 bao gồm 128 bit dài gấp 4 lần của IPv4. Version IPv4 có khả năng cung cấp $2^{32} = 4294967296$ địa chỉ. Còn Version IPv6 có khả năng cung cấp tới 2^{128} địa chỉ.

Do địa chỉ IP phải là duy nhất nên cần có một tổ chức quản lý việc cấp phát địa chỉ IP. Hiện nay tổ chức phi Chính phủ Inter – NIC – chịu trách nhiệm cung cấp địa chỉ IP để đảm bảo không có máy tính kết nối Internet nào bị trùng địa chỉ: (<http://www.internic.net>).

1.3.2. Tên miền

Với cấu trúc địa chỉ IP như trên, người sử dụng sẽ khó nhớ được địa chỉ IP dẫn đến việc sử dụng dịch vụ từ một máy tính nào đó là rất khó khăn. Để thuận tiện cho người sử dụng, một tên tượng trưng sẽ được sử dụng thay thế cho địa chỉ IP. Tên tượng trưng này được gọi là **tên miền (domain name)**.

Máy chủ Web Server của VNNIC có địa chỉ là 203.162.57.101, tên miền của nó là `www.vnnic.net.vn`. Để truy nhập đến một máy chủ, người dùng có thể dùng địa chỉ IP hoặc tên miền. Tuy nhiên trên thực tế, người dùng không cần biết đến địa chỉ IP mà chỉ cần nhớ tên miền này là truy cập được.

Như vậy, tên miền là một sự nhận dạng vị trí của một máy tính trên mạng Internet. Nói cách khác, tên miền là tên của các mạng lưới, tên của các máy chủ trên mạng Internet. Mỗi địa chỉ dạng chữ này luôn tương ứng với một địa chỉ IP dạng số.

Hệ thống quản lý tên miền (Domain Name System – DNS)

Mỗi máy tính khi kết nối vào mạng Internet thì được gán cho một địa chỉ IP xác định. Địa chỉ IP của mỗi máy là duy nhất. Hệ thống DNS ra đời nhằm giúp chuyển đổi từ địa chỉ IP khó nhớ mà máy sử dụng sang một tên dễ nhớ cho người sử dụng, đồng thời giúp hệ thống Internet ngày càng phát triển.

Hệ thống DNS sử dụng hệ thống cơ sở dữ liệu phân tán và phân cấp hình cây. Vì vậy, việc quản lý sẽ dễ dàng và thuận tiện cho việc chuyển đổi từ tên miền sang địa chỉ IP và ngược lại. Hệ thống DNS giống như mô hình quản lý công dân của một nước. Mỗi công dân sẽ có một tên xác định đồng thời cũng có địa chỉ chứng minh thư để giúp quản lý con người một cách dễ dàng hơn.

Mỗi công dân đều có số căn cước để quản lý, ví dụ: Ông Vũ Hữu Hùng có chứng minh thư: 111166520.

Mỗi một địa chỉ IP tương ứng với tên miền, ví dụ: Trang chủ của nhà cung cấp dịch vụ ISP lớn nhất Việt Nam hiện tại là VDC có tên miền là: **home.vnn.vn** , tương ứng với địa chỉ IP là: 203.162.0.12.

Hoạt động của hệ thống DNS:

Giả sử người sử dụng muốn truy cập vào trang web có địa chỉ là <http://www.yahoo.com/>. Tiến trình hoạt động của DNS như sau:

- Trước hết chương trình trên máy người sử dụng gửi yêu cầu tìm kiếm địa chỉ IP ứng với tên miền `www.yahoo.com` tới máy chủ quản lý tên miền (Name Server) cục bộ thuộc mạng của nó (ISP DNS Server).

- Máy chủ quản lý tên miền cục bộ này kiểm tra trong cơ sở dữ liệu của nó xem có chứa cơ sở dữ liệu chuyển đổi từ tên miền sang địa chỉ IP của tên miền mà người sử dụng yêu cầu không. Trong trường hợp máy chủ quản lý tên miền cục bộ có cơ sở dữ liệu này, nó sẽ gửi trả lại địa chỉ IP của máy có tên miền nói trên (`www.yahoo.com`).

- Trong trường hợp máy chủ quản lý tên miền cục bộ không có cơ sở dữ liệu về tên miền này, nó thường hỏi lên các máy chủ quản lý tên miền ở cấp cao nhất (máy chủ quản lý tên miền làm việc ở mức Root). Máy chủ quản lý tên miền ở mức Root này sẽ trả về cho máy chủ quản lý tên miền cục bộ địa chỉ của máy chủ tên miền quản lý

các tên miền có đuôi .com.

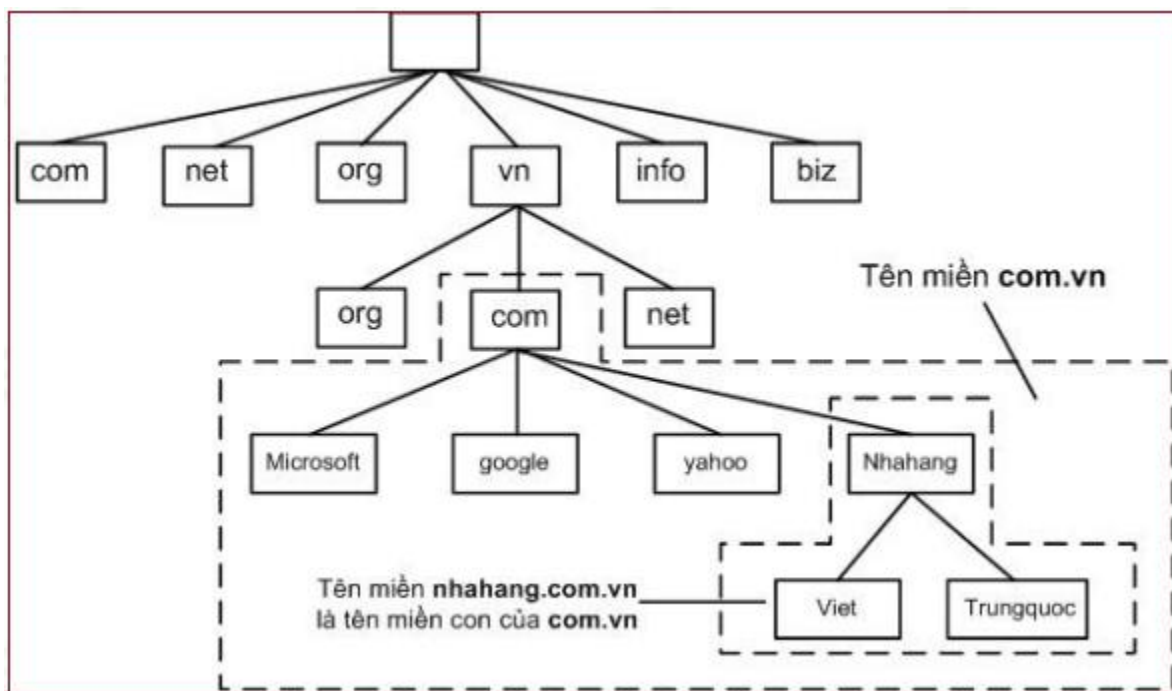
- Máy chủ quản lý tên miền cục bộ gửi yêu cầu đến máy chủ quản lý tên miền có đuôi .com tìm tên miền www.yahoo.com. Máy chủ quản lý tên miền quản lý các tên miền .com sẽ gửi lại địa chỉ của máy chủ quản lý tên miền yahoo.com.

- Máy chủ quản lý tên miền cục bộ sẽ hỏi máy chủ quản lý tên miền yahoo.com này địa chỉ IP của tên miền www.yahoo.com. Do máy chủ quản lý tên miền yahoo.com có cơ sở dữ liệu về tên miền www.yahoo.com nên địa chỉ IP của tên miền này sẽ được gửi trả lại cho máy chủ quản lý tên miền cục bộ.

- Máy chủ tên miền cục bộ chuyển thông tin tìm được đến máy của người sử dụng.

- Máy tính của người dùng sẽ sử dụng địa chỉ IP này để mở một phiên kết nối TCP/IP đến máy chủ chứa trang web có địa chỉ http://www.yahoo.com/.

Tổ chức Hệ thống DNS theo sự phân cấp tên miền trên Internet được cho ở hình dưới đây:



Hình 1.6. Tổ chức của hệ thống tên miền

Cấu tạo tên miền (Domain Name):

Để quản lý các máy đặt tại những vị trí vật lý khác nhau trên hệ thống mạng nhưng thuộc cùng một tổ chức, cùng lĩnh vực hoạt động... người ta nhóm các máy này vào một tên miền (Domain). Trong miền này nếu có những tổ chức nhỏ hơn, lĩnh vực hoạt động hẹp hơn... thì được chia thành các miền con (Sub Domain). Tên miền dùng dấu chấm (.) làm dấu phân cách. Cấu trúc miền và các miền con giống như một cây phân cấp.

Ví dụ www.home.vnn.vn là tên miền máy chủ web của VNNIC. Thành phần thứ nhất 'www' là tên của máy chủ, thành phần thứ hai 'home' thường gọi là tên miền cấp 3 (Third Level Domain Name), thành phần thứ ba 'vnn' gọi là tên miền mức 2 (Second Level

Domain Name) thành phần cuối cùng ‘vn’ là tên miền mức cao nhất (ccTLD – Country Code Top Level Domain Name).

Qui tắc đặt tên miền:

Tên miền nên được đặt đơn giản và có tính chất gợi nhớ, phù hợp với mục đích và phạm vi hoạt động của tổ chức, cá nhân sở hữu tên miền.

Mỗi tên miền được có tối đa 63 ký tự bao gồm cả dấu “.”. Tên miền được đặt bằng các chữ số và chữ cái (a-z A-Z 0-9) và ký tự “-”.

Một tên miền đầy đủ có chiều dài không vượt quá 255 ký tự

Dưới đây là các tên miền thông dụng :

.com (Communication – Dành cho mọi đối tượng, cá nhân, doanh nghiệp)

.net (Network – Dành cho các nhà cung cấp dịch vụ web, net)

.org (Organization – Các tổ chức phi chính phủ hoặc phi lợi nhuận)

.edu (Education – Dành cho các tổ chức giáo dục đào tạo)

.info (Information – Website về lĩnh vực thông tin)

.name (Name - Sử dụng cho trang cá nhân, blog, website cá nhân)

.biz (Business – Dùng cho các trang thương mại)

.gov (Government – Dành cho các tổ chức Chính phủ)

.ws (Website – Sử dụng cho các tổ chức thương mại hoặc cá nhân)

.us (US – Dành cho cá nhân hay công ty Mỹ)

Ngoài ra, mỗi Quốc gia còn có một miền gồm hai ký tự. Ví dụ: “vn” (Việt Nam), “fr” (Pháp), “ca” (Canada)...

Bảng sau là các ký hiệu tên vùng của một số nước trên thế giới:

Tên miền	Ý nghĩa
au	Úc
in	Ấn Độ
cl	Chi Lê
fr	Pháp
us	Hoa Kỳ
za	Nam Phi
uk	Anh

jp	Nhật Bản
es	Tây Ba Nha
de	Đức
ca	Canada
ee	Estonia
hk	Hồng Kông
vn	Việt Nam

1.4. World Wide Web và HTML

1.4.1. World Wide Web

World Wide Web (gọi tắt là Web hay WWW)

- Là một dịch vụ của Internet, cho phép bạn truy nhập tới nguồn thông tin đồ sộ của Internet. Nguồn thông tin này được tổ chức dưới dạng các trang web có sự liên kết chặt chẽ với nhau.
- Mỗi trang web là một tài liệu siêu văn bản. Tài liệu này có thể chứa văn bản, âm thanh, hình ảnh... Được mã hoá đặc biệt, sử dụng ngôn ngữ đánh dấu siêu văn bản – HTML (HyperText Markup Languages). Ngôn ngữ này cho phép tác giả của một tài liệu nhúng các liên kết siêu văn bản (còn được gọi là các siêu liên kết – hyperlink) vào trong tài liệu. Các liên kết siêu văn bản là nền móng của World Wide Web.
- Khi đọc một trang web, có thể nhấp chuột vào một từ hay một hình ảnh được mã hoá như một liên kết siêu văn bản và sẽ lập tức chuyển tới một vị trí khác nằm bên trong tài liệu đó hoặc tới một trang Web khác. Trang thứ hai có thể nằm trên cùng máy tính với trang đầu, hoặc có thể nằm bất kì nơi nào trên Internet.
- Một tập hợp các trang Web có liên quan được gọi là WebSite. Mỗi WebSite được lưu trữ trên một máy phục vụ Web, vốn là các máy chủ Internet lưu trữ hàng ngàn trang Web riêng lẻ. Việc sao chép một trang lên một Web Server được gọi là tải (hoặc nạp) lên (uploading).



Hình 1.7. Hình ảnh của một trang Web

- Web cung cấp thông tin rất đa dạng bao gồm văn bản, hình ảnh, âm thanh, video. Hiện nay các trang Web sử dụng để phân phối tin tức, các dịch vụ giáo dục, thông tin, danh mục sản phẩm, cùng nhiều thứ khác. Các trang Web tương tác cho phép các độc giả tra cứu cơ sở dữ liệu, đặt hàng các sản phẩm và các thông tin, gửi số tiền thanh toán bằng thẻ tín dụng... Máy chủ Web (web server) • Để cung cấp dịch vụ Web cho người sử dụng, chúng ta cần có một máy chủ web đặt tại một địa chỉ nào đó trên Internet. Máy chủ web là một 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... Trình duyệt Web (web browser) Trình duyệt Web là một phần mềm ứng dụng

được cài đặt trên máy tính của người sử dụng. Phần mềm này cho phép người dùng tìm các tài liệu siêu văn bản trên Web rồi mở các tài liệu đó trên máy tính người sử dụng.

Một số trình duyệt Internet thông dụng gồm:

Trình duyệt Internet Explorer (IE)

Đây là trình duyệt Internet phổ biến nhất. Trình duyệt này đã đi kèm với máy tính hệ điều hành Windows của bạn.

- Ưu điểm: Trình duyệt này máy nào cũng có. Nhiều trang web được thiết kế phục vụ cho người dùng trình duyệt này nên việc hiển thị trang là đẹp nhất trong số các trình duyệt.
- Nhược điểm: Tốc độ nạp trang không nhanh lắm, dễ bị những trang web độc hại, mã độc, virus lợi dụng lỗ hổng của trình duyệt này chui vào máy gây mất an toàn cho người sử dụng.

Trình duyệt Firefox

Đây là một trình duyệt mới, có nhiều tính năng hay và hoàn toàn miễn phí.

- Ưu điểm: Hỗ trợ tốt những chuẩn thiết kế web hiện đại. Tốc độ duyệt khá nhanh, hơn Internet Explorer. Đồng thời Firefox cho phép người dùng cài thêm những công cụ bổ sung có sẵn miễn phí mà người dùng có thể download từ trang <http://getfirefox.com>, cũng là trang chủ của trình duyệt Firefox. Khi duyệt web với Firefox sẽ an toàn hơn so với IE, người dùng hạn chế được lừa đảo theo hình thức phishing, mã độc hại, virus từ các trang web.
- Nhược điểm: Vì vẫn còn nhiều trang web được thiết kế sai theo hỗ trợ của trình duyệt IE vì thế mà khi duyệt những trang này với Firefox sẽ hiển thị không được tốt. Ngoài ra, phải cài thêm hỗ trợ dạng Extension / Plug-ins mới có thể xem được một số trang multimedia (xem phim, nghe nhạc).

1.4.2. HTML

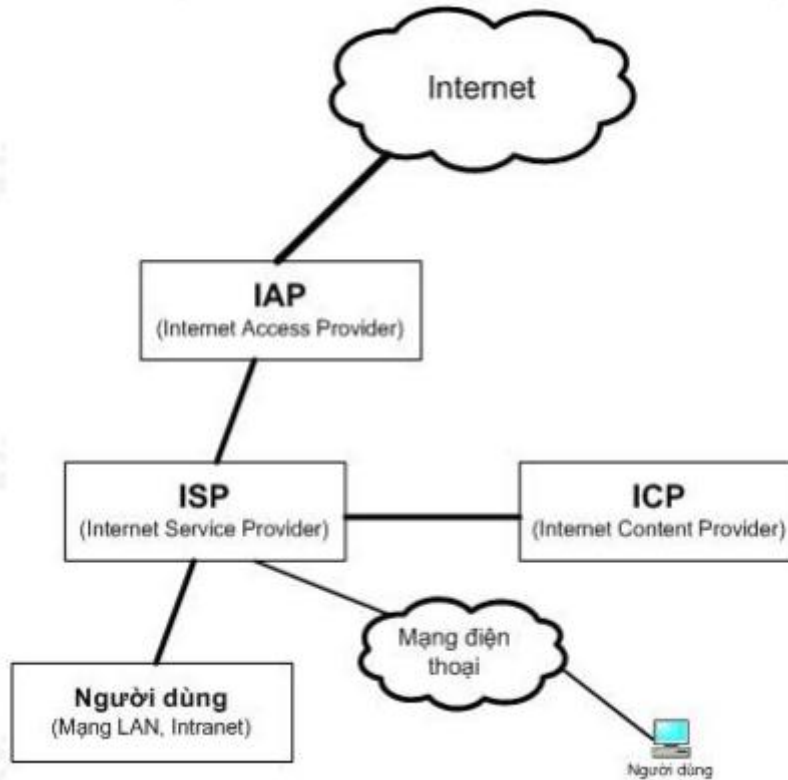
HTML (HyperText Markup Language), nghĩa là "Ngôn ngữ Đánh dấu Siêu văn bản" là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web. Tên của ngôn ngữ này được hình thành từ bốn từ: • Hyper (Siêu): HTML cho phép liên kết nhiều trang văn bản rải rác khắp nơi trên Internet. Nó có tác dụng che dấu sự phức tạp của Internet đối với người sử dụng. Người dùng Internet có thể đọc văn bản mà không cần biết đến văn bản đó đang nằm ở đâu, hệ thống được xây dựng phức tạp như thế nào. • Text (Văn bản): HTML đầu tiên và trước hết là để trình bày văn bản và dựa trên nền tảng là một văn bản. • Language (Ngôn ngữ): HTML là một ngôn ngữ tương tự như các ngôn ngữ lập trình, tuy đơn giản hơn. Nó có cú pháp chặt chẽ để viết các lệnh nhằm thực hiện việc trình diễn văn bản. • Markup (Đánh dấu): HTML là ngôn ngữ của các thẻ (Tag) đánh dấu. Các thẻ này

xác định cách thức trình bày đoạn văn bản tương ứng trên màn hình. Một file HTML là một tệp văn bản bao gồm nhiều thẻ (tag). Những thẻ này nói cho trình duyệt biết nó phải hiển thị trang đó như thế nào. Một file HTML phải có phần mở rộng là .htm hoặc .html và có thể được tạo bởi một trình soạn thảo văn bản đơn giản

1.5. Các nhà cung cấp dịch vụ Internet

Để mọi người có thể khai thác và sử dụng các dịch vụ Internet, cần có các nhà cung cấp dịch vụ Internet. Có thể liệt kê như sau:

- ISP (Internet Service Provider) – Nhà cung cấp dịch vụ Internet. Nhà cung cấp dịch vụ Internet cấp quyền truy cập Internet qua mạng viễn thông và các dịch vụ như: Email, Web, FTP, Telnet, Chat. Để có thể truy nhập và sử dụng các dịch vụ của Internet, người sử dụng phải đăng ký với nhà cung cấp dịch vụ này. ISP được cấp cổng truy cập vào Internet bởi IAP. Hiện tại ở Việt Nam có 18 ISP đăng ký cung cấp dịch vụ, trong đó, một số nhà cung cấp dịch vụ lớn gồm: Tập đoàn Bưu chính Viễn thông Việt Nam (VNPT), Công ty Cổ phần truyền thông (FPT), Tổng công ty Viễn thông Quân đội (Vietel).
- IAP (Internet Access Provider) – Nhà cung cấp dịch vụ đường truyền để kết nối với Internet (còn gọi là IXP – Internet Exchange Provider). Nếu hiểu Internet như một siêu xa lộ thông tin thì IAP là nhà cung cấp phương tiện để đưa người dùng vào xa lộ. Nói cách khác IAP là kết nối người dùng trực tiếp với Internet. IAP có thể thực hiện cả chức năng của ISP nhưng ngược lại thì không. Một IAP thường phục vụ cho nhiều ISP khác nhau. Hiện nay, tại Việt Nam có 7 IAP, bao gồm: Tập đoàn Bưu chính Viễn thông Việt Nam (VNPT), Công ty đầu tư phát triển công nghệ FPT, Tổng công ty viễn thông quân đội (Viettel), Công ty thông tin viễn thông điện lực (ETC), Công ty cổ phần dịch vụ Bưu chính Viễn thông Sài Gòn (SPT), Công ty cổ phần viễn thông Hà Nội (HANOITELECOM), Công ty truyền thông đa phương tiện (VTC).
- ISP dùng riêng ISP dùng riêng được quyền cung cấp đầy đủ dịch vụ Internet. Điều khác nhau duy nhất giữa ISP và ISP dùng riêng là ISP dùng riêng không cung cấp dịch vụ Internet với mục đích kinh doanh. Đây là loại hình dịch vụ Internet của các cơ quan hành chính, các trường đại học hay viện nghiên cứu.
- ICP (Internet Content Provider) – Nhà cung cấp dịch vụ nội dung thông tin Internet. ICP cung cấp các thông tin về: kinh tế, giáo dục, thể thao, chính trị, quân sự (thường xuyên cập nhật thông tin mới theo định kỳ) đưa lên mạng.
- OSP (Online Service Provider) – Nhà cung cấp dịch vụ ứng dụng Internet. OSP cung cấp các dịch vụ trên cơ sở ứng dụng Internet (OSP) như: Mua bán qua mạng, giao dịch ngân hàng, tư vấn, đào tạo, ...



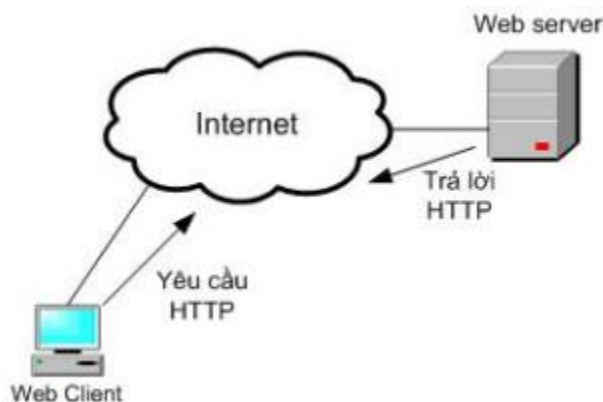
Hình 1.8. Các nhà cung cấp dịch vụ Internet

1.6. Giới thiệu một số dịch vụ Internet thông dụng

1.6.1. World Wide Web

World Wide Web hay Web là một trong những dịch vụ phổ biến nhất của Internet. Dịch vụ này cho phép bạn truy cập đến các trang thông tin siêu văn bản (trang web) được đặt tại nhiều vị trí khác nhau trên Internet. Dịch vụ này hoạt động theo mô hình Khách/Chủ (Client/Server). Trong đó máy chủ web là máy tính trên Internet có chạy phần mềm Web server. Máy chủ web lưu trữ nội dung thông tin (các trang web), nhận và trả lời các yêu cầu từ máy khách web. Máy khách web là máy tính của người dùng có chạy trình duyệt web (như Internet Explorer, Netscape Navigator, Firefox ...). Máy khách web gửi yêu cầu và hiển thị thông tin trả lời từ máy chủ web.

Dịch vụ web sử dụng giao thức HTTP (Hyper Text Transfer Protocol): Giao thức truyền siêu văn bản.



Hình 1.9. Mô hình hoạt động của một dịch vụ Web

Để truy nhập đến một trang web nào đó, người dùng gõ địa chỉ trang web vào thanh địa chỉ của trình duyệt web.

Ví dụ, để truy nhập tới trang web của Chương trình đào tạo cử nhân theo phương thức Elearning (NEU-EDUTOP), bạn gõ: <http://www.neu-edutop.edu.vn>, <http://www.topica.edu.vn>

1.6.2. Dịch vụ thư điện tử

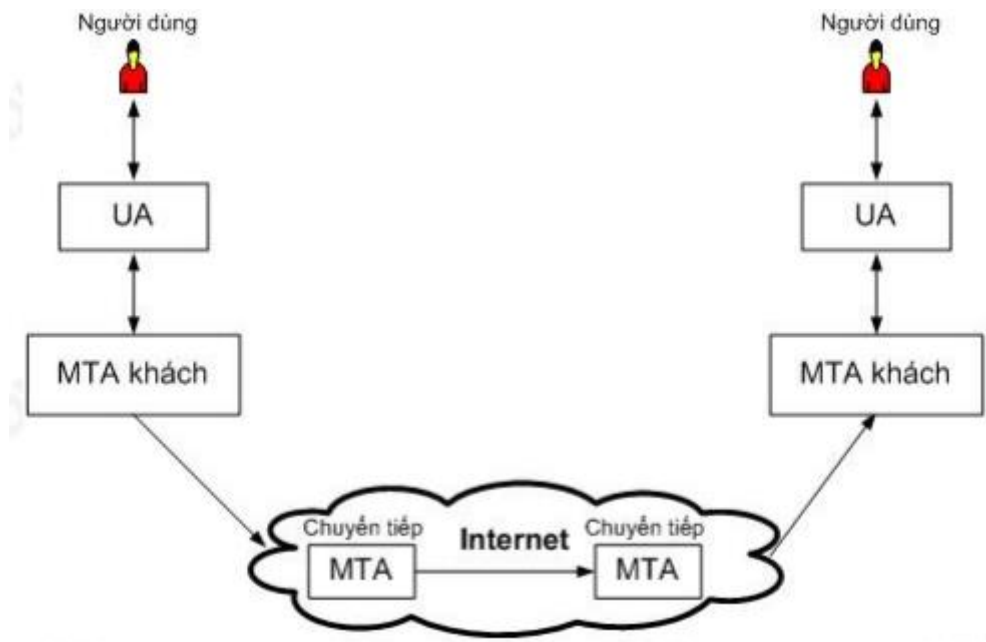
Dịch vụ thư điện tử là một dịch vụ thông dụng nhất của Internet. Nó cho phép bạn gửi một thông điệp tới một hoặc một nhóm người qua mạng Internet. Ngoài việc gửi thông điệp dưới dạng văn bản, bạn còn có thể đính kèm các tệp tin cùng với thông điệp. Dịch vụ thư điện tử được sử dụng phổ biến do có các ưu điểm sau:

- **Tốc độ cao và khả năng chuyển tải trên toàn cầu:** Có thể nói đây là một trong những ưu điểm hàng đầu của hệ thống thư điện tử. Bạn có thể gửi thư cho bất kỳ người nào gần như ngay lập tức. Người nhận cũng có thể nhận thư ở bất kỳ đâu, miễn là nơi đó có kết nối Internet.
- **Giá thành thấp:** Giá thành của việc gửi thông tin bằng thư điện tử gần như không đáng kể bởi bạn chỉ cần trả chi phí cho việc sử dụng Internet là bạn đã có khả năng sử dụng các hệ thống thư điện tử miễn phí trên toàn cầu và từ đó liên lạc đến khắp mọi nơi. Nếu so sánh về mặt giá thành với hệ thống thư tín thông thường, nhất là gửi thư quốc tế thì việc gửi bằng hệ thống thư điện tử rẻ và tiện dụng hơn rất nhiều lần.
- **Linh hoạt về mặt thời gian:** Nếu bạn có người quen ở Mỹ và bạn muốn gọi điện cho người đó lúc 12 giờ trưa, bạn có thể không nhận được câu trả lời (do các cơ quan ở Mỹ đã nghỉ), hoặc có thể bạn sẽ đánh thức họ vào lúc nửa đêm, rất phiền toái. Tuy nhiên, nếu sử dụng thư điện tử thì bạn có thể gửi vào bất cứ lúc nào và người nhận cũng có thể đọc thư vào lúc nào họ muốn. Để có thể sử dụng thư điện tử, mỗi người dùng phải có một tài khoản. Tài

khoản này có thể được đăng ký miễn phí hoặc được các nhà cung cấp dịch vụ cấp. Cấu trúc chung của một địa chỉ thư điện tử như sau: Tênđăngký@tênmiền

Ví dụ địa chỉ thư: nva@topica.edu.vn, nve@neu-edutop.edu.vn Tên đăng ký: nva, nve

Ký tự @ phân tách tên đăng ký và tên miền, ký tự này buộc phải có trong mọi địa chỉ thư điện tử. Tên miền: topica.edu.vn, neu-edutop.edu.vn là địa chỉ website của tổ chức mà người dùng đăng ký dịch vụ thư điện tử. Hệ thống thư điện tử được chia làm hai phần: UA (Mail User Agent) và MTA (Message Transfer Agent). MUA thực chất là một hệ thống làm nhiệm vụ tương tác trực tiếp với người dùng cuối, giúp họ nhận bản tin, soạn thảo bản tin, lưu các bản tin và gửi bản tin. Nhiệm vụ của MTA là định tuyến bản tin và xử lý các bản tin đến từ hệ thống của người dùng sao cho các bản tin đó đến được đúng hệ thống đích.



Hình 1.10. Cấu trúc hệ thống thư điện tử

1.6.3. Telnet

Telnet là một dịch vụ Internet cho phép người dùng ngồi trên một thiết bị đầu cuối có thể thông qua kết nối mạng truy nhập đến một thiết bị từ xa để điều khiển nó bằng câu lệnh như là đang ngồi tại máy ở xa. Một máy trạm có thể thực hiện đồng thời nhiều phiên telnet đến nhiều địa chỉ IP khác nhau. Telnet hoạt động theo phiên, mỗi phiên là một kết nối truyền dữ liệu theo giao thức TCP với cổng 23. Telnet hoạt động theo mô hình khách/chủ (Client/Server), trong đó Client là một phần mềm chạy trên máy của người dùng, phần mềm này sẽ cung cấp giao diện hiển thị để người dùng gõ lệnh điều khiển. Phần Server là dịch vụ chạy trên máy từ xa lắng nghe và xử lý các kết nối và câu lệnh được gửi đến từ máy trạm tại chỗ. Dịch vụ Telnet thường được sử dụng để điều khiển và cấu hình từ xa cho

các thiết bị, chẳng hạn bộ định tuyến (Router) và bộ chuyển mạch (Switch). Để kết nối từ xa đến một thiết bị nào đó, câu lệnh được sử dụng là: Telnet IP_address Trong đó:

- Telnet là tên lệnh.
- IP_address là địa chỉ IP của thiết bị.

1.6.4. Dịch vụ truyền tệp

Dịch vụ truyền tệp (FTP) là một dịch vụ cơ bản và phổ biến cho phép chuyển các tệp dữ liệu giữa các máy tính khác nhau trên mạng. FTP hỗ trợ tất cả các dạng tệp, trên thực tế nó không quan tâm tới dạng tệp cho dù đó là tệp văn bản mã ASCII hay các tệp dữ liệu dạng nhị phân. Với cấu hình của máy phục vụ FTP, có thể quy định quyền truy nhập của người sử dụng với từng thư mục dữ liệu, tệp dữ liệu cũng như giới hạn số lượng người sử dụng có khả năng cùng một lúc có thể truy nhập vào cùng một nơi lưu trữ dữ liệu.

1.6.5. Dịch vụ Gopher

Trước khi Web ra đời, Gopher là dịch vụ rất được ưa chuộng. Gopher là một dịch vụ truyền tệp tương tự như FTP, nhưng nó hỗ trợ người dùng trong việc cung cấp thông tin về tài nguyên. Client Gopher hiển thị một thực đơn, người dùng chỉ việc lựa chọn cái mà mình cần. Kết quả của việc lựa chọn được thể hiện ở một thực đơn khác. Gopher bị giới hạn trong kiểu các dữ liệu. Nó chỉ hiển thị dữ liệu dưới dạng mã ASCII mặc dù có thể chuyển dữ liệu sang dạng nhị phân và hiển thị bằng một phần mềm khác.

1.6.6. Dịch vụ WAIS

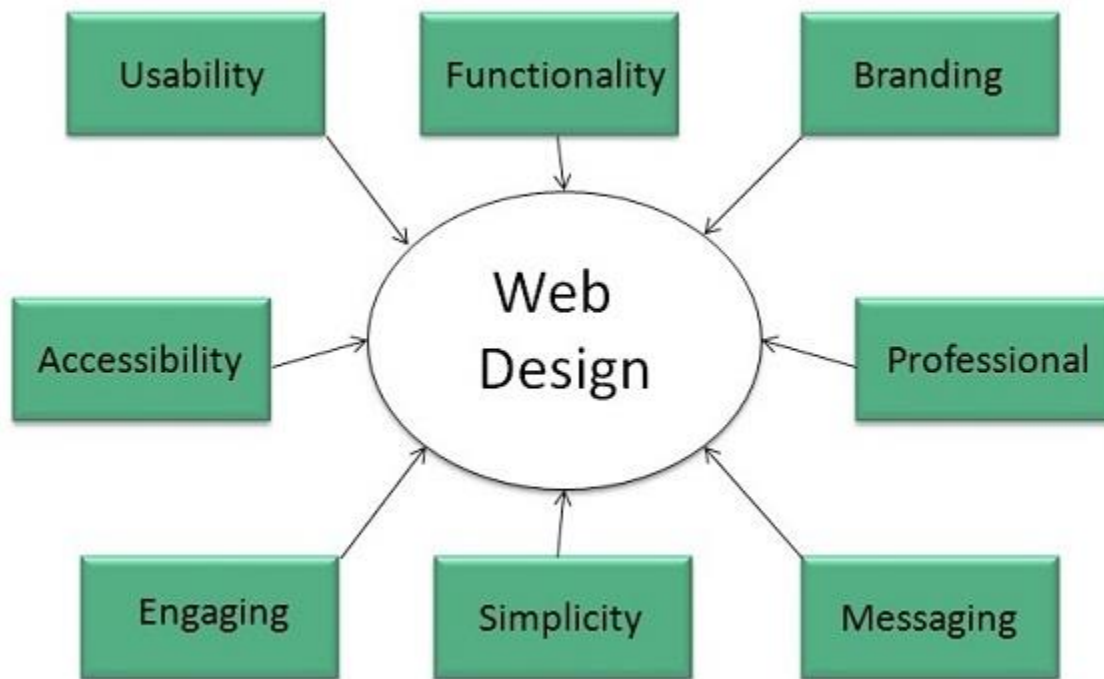
WAIS (Wide Area Information Serves) là một dịch vụ tìm kiếm dữ liệu. WAIS thường xuyên bắt đầu việc tìm kiếm dữ liệu tại thư mục của máy chủ, nơi chứa toàn bộ danh mục của các máy phục vụ khác. Sau đó WAIS thực hiện tìm kiếm tại máy phục vụ thích hợp nhất. WAIS có thể thực hiện công việc của mình với nhiều loại dữ liệu khác nhau như văn bản ASCII, PostScript, GIF, TIFF, điện thư,...

1.6.7. Dịch vụ chat

Chat là hình thức hội thoại trực tiếp trên Internet. Với dịch vụ này hai hay nhiều người có thể cùng trao đổi thông tin trực tiếp qua bàn phím máy tính. Điều đó có nghĩa là bất kỳ câu đánh trên máy của người này đều hiển thị trên màn hình của người đang cùng hội thoại. Có nhiều chương trình hỗ trợ cho phép chat trực tiếp (những người chat đang trực tuyến) hoặc gián tiếp (những người chat đang không trực tuyến) với đối phương. Người sử dụng có thể chat bằng chữ (Text), chat bằng âm thanh (Voice) hoặc bằng hình ảnh (Web-cam)... Ngoài chat trên Internet người sử dụng còn có thể chat với nhau trên mạng LAN.

1.7. Thiết kế web

Thiết kế web liên quan trực tiếp đến khía cạnh hình ảnh của một trang web. Thiết kế web hiệu quả là sự hiệu quả của việc truyền đạt ý tưởng một cách hiệu quả.



Thiết kế web là một phần của phát triển web. Tuy nhiên những định nghĩa này có thể được sử dụng để hoán đổi cho nhau.

Các điểm chính

Kế hoạch thiết kế bao gồm:

- Liệt kê chi tiết về cấu trúc thông tin.
- Lập kế hoạch về cấu trúc của trang.
- Một sơ đồ các trang

1.7.1 Wireframe

Wireframe là các chỉ dẫn hình ảnh xuất hiện trên trang web. Nó giúp xác định rõ cấu trúc của một trang web, liên kết giữa các trang và bố trí các hình ảnh.

Một wireframe bao gồm:

- Các hộp chứa các yếu tố đồ họa sơ cấp.
- Sắp đặt các đầu đề chính và đầu đề phụ.
- Cấu trúc bố trí đơn giản.
- Lời kêu gọi hành động (call to action)
- Các khối văn bản.

Wireframe có thể được tạo ra bởi sử dụng các chương trình như Visio nhưng bạn cũng có thể sử dụng một bút và giấy.

1.7.2. Các công cụ thiết kế web

Dưới đây liệt kê các công cụ mà có thể được sử dụng để thiết kế web hiệu quả:

STT	Miêu tả công cụ
1	<p>Photoshop CC</p> <p>Là một công cụ thiết kế web tuyệt vời được cung cấp bởi Adobe. Phiên bản mới nhất CC 2014 hỗ trợ nhiều tính năng mới như các đối tượng smart, sắp chữ các lớp, các chỉ dẫn smart, tích hợp Typekit, tìm kiếm font và cải thiện luồng công việc.</p>
2	<p>Illustrator CC</p> <p>Cũng là một công cụ thiết kế web với các tính năng mạnh mẽ như các thư viện AutoCad, in đề trắng, tạo góc tự động, các hình ảnh không được nhúng, các công cụ cảm ứng,</p>
3	<p>Coda 2</p> <p>Coda 2 là một công cụ thiết kế và phát triển web mạnh mẽ, nó cung cấp giao diện người dùng tốt hơn, quản lý file, clip, các site, thiết kế tốt hơn và hỗ trợ Mysql tốt hơn.</p>
4	<p>OmniGraffle</p> <p>OmniGraffle được sử dụng phần lớn cho wireframe. Nhược điểm của công cụ này là nó không có nguyên mẫu tương tác và nó chỉ dành riêng cho Mac.</p>
5	<p>Sublime Text</p>

	Là một bộ soạn mã nguồn với giao diện chương trình ứng dụng Python. Chức năng của nó có thể được mở rộng bởi sử dụng các plugin.
6	Pen and Paper Pen and paper có thể được sử dụng để vẽ bề mặt của website.
7	Vim Vim là một công cụ thiết kế web tuyệt vời. Nó hỗ trợ sự tùy chỉnh tự động của mã, đa vùng bộ đệm cho việc lưu các mã bị cắt/sao chép, và ghi lại các hoạt động lặp lại tự động.
8	Imageoptim Cơ bản nó được sử dụng cho các hình ảnh tối ưu trên một website để tải chúng nhanh hơn bởi tìm kiếm các tham số nén tốt nhất và bởi gỡ bỏ các lời bình không cần thiết.
9	Sketch 3 Là công cụ thiết kế web được phát triển riêng cho các thiết kế các giao thức, các website, các icon....
10	Heroku Cũng là một công cụ phát triển web tuyệt vời mà hỗ trợ Ruby, Node.js, Java và PHP.
11	Axure Nó hỗ trợ các công cụ nguyên mẫu, tài liệu hóa và wireframe để tạo thiết kế giao diện web.
12	Hype 2 Hype 2 cung cấp: cách dễ dàng nhất để tạo hiệu ứng và thêm tính tương tác, sức mạnh của HTML5, tính di động, và các tính năng WYSIWYG.
13	Image Alpha Công cụ này giúp giảm kích cỡ file của các tệp PNG 24 bit. Nó thực hiện điều này bởi áp dụng việc nén và chuyển đổi nó tới định dạng PNG8+alpha mà hiệu quả hơn.

14	Hammer Công cụ này phù hợp với các sản phẩm mà không có kế hoạch, chương trình hoặc cho các dự án nhỏ.
15	JPEGmini Lite Nó là một công cụ tối ưu hình ảnh và hỗ trợ hình ảnh mà có độ phân giải lên tới 28 Megapixel.
16	BugHerd Công cụ này giúp quan sát cách các dự án thực hiện và người nào đang làm việc trên đó. Nó cũng giúp nhận diện các vấn đề trong khi phát triển dự án.

1.7.3. Phân tích trang web

Một trang web chứa các thành phần sau:

Khối

Container - Khối chứa có thể ở trong thẻ thân của trang, một thẻ div. Nếu không có khối chứa, sẽ không có vị trí để đặt các nội dung của trang web.

Logo

Logo liên quan tới việc nhận diện một website và được sử dụng trong các công ty khác nhau để tiếp thị các thẻ kinh doanh, các thư kinh doanh, các broucher,

Điều hướng

Hệ thống điều hướng của trang nên là dễ dàng để tìm ra và sử dụng. Thường thì điều hướng được đặt ở bên phải tại trên cùng của trang.

Nội dung

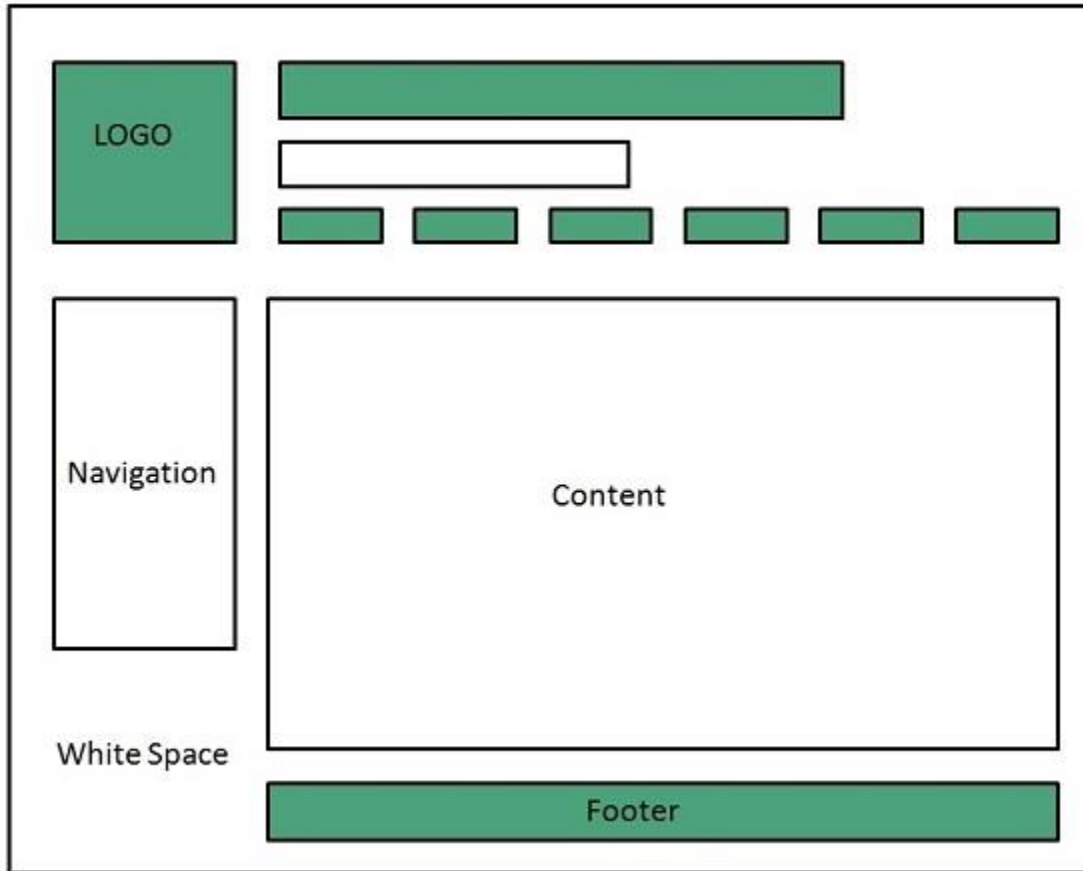
Nội dung trên trang website nên liên quan tới mục đích của trang.

Chân trang (footer)

Footer được đặt ở dưới cùng của trang. Nó được sử dụng để chứa đựng bản quyền tác giả, thông tin liên hệ và thông tin hợp pháp cũng như một số liên kết tới các khu vực chính trong site.

Khoảng trống trắng

Nó cũng được gọi là **không gian phủ định** và có mặt ở bất cứ khu vực nào của trang mà không được bao phủ bởi văn bản hoặc minh họa.



1.7.4. Các lỗi thiết kế web

Bạn cũng nên biết và ghi nhớ về các lỗi thường gặp trong quá trình thiết kế web:

- Website không làm việc trên bất kỳ trình duyệt nào khác ngoài trên IE.
- Sử dụng công nghệ cắt góc.
- Âm thanh và video bắt đầu một cách tự động.
- Điều hướng bị ẩn...
- 100% nội dung lèo lẹt.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 1

1. Hãy trình bày định nghĩa về Internet và những mốc phát triển chính của Internet.
2. Hãy trình bày tác dụng của địa chỉ IP, cấu trúc của địa chỉ IP và liệt kê 5 địa chỉ IP mà bạn biết.
3. Hãy nêu 3 phương pháp kết nối Internet thông dụng và trình bày ưu, nhược điểm của từng phương pháp.
4. Hãy nêu tên 5 nhà cung cấp dịch vụ Internet tại Việt Nam mà bạn biết.
5. Tên miền là gì? Hãy cho ví dụ.
6. Nêu quy tắc đặt tên miền và liệt kê các tên miền thông dụng.
7. Thế nào là tên miền mức cao nhất, mức hai, mức ba? Cho ví dụ.
8. Nêu sự tương ứng giữa tên miền và địa chỉ IP. Cho ví dụ.
9. Thế nào là một Website?
10. HTTP là gì?
11. Internet được ra đời vào năm nào?
 - a) 1968
 - b) 1969
 - c) 1974
 - d) 1997
12. Mạng nào là mạng tiền thân của Internet?
 - a) LAN
 - b) B, WAN
 - c) MAN

d) ARPANET

13. Bộ giao thức nào được sử dụng trên Internet?

a) H.323

b) TCP/IP

c) AppleTalk

d) IPX/SPX

14. ISP là viết tắt của cụm từ tiếng Anh nào?

a) Internet Service Protocol

b) Internet Service Provider

c) Internet Search Provider

d) Important Service Provider

15. IAP là viết tắt của cụm từ tiếng Anh nào?

a) Internet Access Protocol

b) Internet Application Provider

c) Internet Access Provider

d) Internet Application Protocol

CHƯƠNG 2. TỔNG QUAN VỀ HTML

2.1. Các thẻ định cấu trúc tài liệu

2.1.1 HTML

Cặp thẻ này được sử dụng để xác nhận một tài liệu là tài liệu HTML, tức là nó có sử dụng các thẻ HTML để trình bày. Toàn bộ nội dung của tài liệu được đặt giữa cặp thẻ này.

Cú pháp:

```
<HTML>  
... Toàn bộ nội của tài liệu được đặt ở đây  
</HTML>
```

Trình duyệt sẽ xem các tài liệu không sử dụng thẻ **<HTML>** như những tệp tin văn bản bình thường.

2.1.2 HEAD

Thẻ **HEAD** được dùng để xác định phần mở đầu cho tài liệu.

Cú pháp:

```
<HEAD>  
... Phần mở đầu (HEADER) của tài liệu được đặt ở đây  
</HEAD>
```

2.1.3 TITLE

Cặp thẻ này chỉ có thể sử dụng trong phần mở đầu của tài liệu, tức là nó phải nằm trong thẻ phạm vi giới hạn bởi cặp thẻ **<HEAD>**.

Cú pháp:

```
<TITLE>Tiêu đề của tài liệu</TITLE>
```

2.1.4 BODY

Thẻ này được sử dụng để xác định phần nội dung chính của tài liệu - phần thân (body) của tài liệu. Trong phần thân có thể chứa các thông tin định dạng nhất định để đặt ảnh nền cho tài liệu, màu nền, màu văn bản siêu liên kết, đặt lề cho trang tài liệu... Những thông tin này được đặt ở phần tham số của thẻ.

```
<BODY>  
... phần nội dung của tài liệu được đặt ở đây  
</BODY>
```

Cú pháp:

Trên đây là cú pháp cơ bản của thẻ **BODY**, tuy nhiên bắt đầu từ HTML 3.2 thì có nhiều thuộc tính được sử dụng trong thẻ **BODY**. Sau đây là các thuộc tính chính:

BACKGROUND=	Đặt một ảnh nào đó làm ảnh nền (background) cho văn bản. Giá trị của tham số này (phần sau dấu bằng) là URL của file ảnh. Nếu kích thước ảnh nhỏ hơn cửa sổ trình duyệt thì toàn bộ màn hình cửa sổ trình duyệt sẽ được lát kín bằng nhiều ảnh.
BGCOLOR=	Đặt màu nền cho trang khi hiển thị. Nếu cả hai tham số BACKGROUND và BGCOLOR cùng có giá trị thì trình duyệt sẽ hiển thị màu nền trước, sau đó mới tải ảnh lên phía trên.
TEXT=	Xác định màu chữ của văn bản, kể cả các đề mục.
ALINK=,VLINK=,LINK=	Xác định màu sắc cho các siêu liên kết trong văn bản. Tương ứng, alink (<i>active link</i>) là liên kết đang được kích hoạt - tức là khi đã được kích chuột lên; vlink (<i>visited link</i>) chỉ liên kết đã từng được kích hoạt;

Như vậy một tài liệu HTML cơ bản có cấu trúc như sau:

```
<HTML>
<HEAD>
<TITLE>Tiêu đề của tài liệu</TITLE>
</HEAD>
<BODY Các tham số nếu có>
... Nội dung của tài liệu
</BODY>
</HTML>
```

2.2. Các thẻ định dạng khối

2.2.1. THẺ P

Thẻ **<P>** được sử dụng để định dạng một đoạn văn bản.

Cú pháp:

```
<P>Nội dung đoạn văn bản</P>
```

2.2.2. CÁC THẺ ĐỊNH DẠNG ĐỀ MỤC H1/H2/H3/H4/H5/H6

HTML hỗ trợ 6 mức đề mục. Chú ý rằng đề mục chỉ là các chỉ dẫn định dạng về mặt logic, tức là mỗi trình duyệt sẽ thể hiện đề mục dưới một khuôn dạng thích hợp. Có thể ở trình duyệt này là font chữ 14 point nhưng sang trình duyệt khác là font chữ 20 point. Đề mục cấp 1 là cao nhất và giảm dần đến cấp 6. Thông thường văn bản ở đề mục cấp 5 hay cấp 6 thường có kích thước nhỏ hơn văn bản thông thường.

Dưới đây là các thẻ dùng để định dạng văn bản ở dạng đề mục:

<code><H1> ... </H1></code>	Định dạng đề mục cấp 1
<code><H2> ... </H2></code>	Định dạng đề mục cấp 2
<code><H3> ... </H3></code>	Định dạng đề mục cấp 3
<code><H4> ... </H4></code>	Định dạng đề mục cấp 4
<code><H5> ... </H5></code>	Định dạng đề mục cấp 5
<code><H6> ... </H6></code>	Định dạng đề mục cấp 6

2.2.3 THẺ XUỐNG DÒNG BR

Thẻ này không có thẻ kết thúc tương ứng (`</BR>`), nó có tác dụng chuyển sang dòng mới. Lưu ý, nội dung văn bản trong tài liệu HTML sẽ được trình duyệt Web thể hiện liên tục, các khoảng trắng liền nhau, các ký tự tab, ký tự xuống dòng đều được coi như một khoảng trắng. Để xuống dòng trong tài liệu, bạn phải sử dụng thẻ `
`

2.2.4 THẺ PRE

Để giới hạn đoạn văn bản đã được định dạng sẵn bạn có thể sử dụng thẻ `<PRE>`. Văn bản ở giữa hai thẻ này sẽ được thể hiện giống hệt như khi chúng được đánh vào, ví dụ dấu xuống dòng trong đoạn văn bản giới hạn bởi thẻ `<PRE>` sẽ có ý nghĩa chuyển sang dòng mới (trình duyệt sẽ không coi chúng như dấu cách)

Cú pháp:

```
<PRE>Văn bản đã được định dạng</PRE>
```

2.3. Các thẻ định dạng danh sách

Cú pháp:

```
<UL>
<LI> Mục thứ nhất
<LI> Mục thứ hai
</UL>
```

Có 4 kiểu danh sách:

- Danh sách không sắp xếp (hay không đánh số) ``
- Danh sách có sắp xếp (hay có đánh số) ``, mỗi mục trong danh sách được sắp xếp thứ tự.
- Danh sách thực đơn `<MENU>`
- Danh sách phân cấp `<DIR>`

Với nhiều trình duyệt, danh sách phân cấp và danh sách thực đơn giống danh sách không đánh số, có thể dùng lẫn với nhau. Với thẻ OL ta có cú pháp sau:

```
<OL TYPE=1/a/A/i/I>
  <LI>Mục thu nhất
  <LI>Mục thu hai
  <LI>Mục thu ba
</OL>
```

trong đó:

TYPE =1	Các mục được sắp xếp theo thứ tự 1, 2, 3...
=a	Các mục được sắp xếp theo thứ tự a, b, c...
=A	Các mục được sắp xếp theo thứ tự A, B, C...
=i	Các mục được sắp xếp theo thứ tự i, ii, iii...
=I	Các mục được sắp xếp theo thứ tự I, II, III...

Ngoài ra còn thuộc tính **START**= xác định giá trị khởi đầu cho danh sách.

Thẻ **** có thuộc tính **TYPE**= xác định ký hiệu đầu dòng (bullet) đứng trước mỗi mục trong danh sách. Thuộc tính này có thể nhận các giá trị : *disc* (chấm tròn đậm); *circle* (vòng tròn); *square* (hình vuông).

2.4. Các thẻ định dạng ký tự

2.4.1. Các thẻ định dạng in ký tự

Sau đây là các thẻ được sử dụng để quy định các thuộc tính như in nghiêng, in đậm, gạch chân... cho các ký tự, văn bản khi được thể hiện trên trình duyệt.

 ... 	In chữ đậm
 ... 	
<I> ... </I>	In chữ nghiêng
 ... 	
<U> ... </U>	In chữ gạch chân
<DFN>	Đánh dấu đoạn văn bản giữa hai thẻ này là định nghĩa của một từ. Chúng thường được in nghiêng hoặc thể hiện qua một kiểu đặc biệt nào đó.
<S> ... </S>	In chữ bị gạch ngang.
<STRIKE> ... </STRIKE>	
<BIG> ... </BIG>	In chữ lớn hơn bình thường bằng cách tăng kích thước font hiện thời lên một. Việc sử dụng các thẻ <BIG> lồng nhau tạo ra hiệu ứng chữ tăng dần. Tuy nhiên đối với mỗi trình duyệt có giới hạn về kích

	thước đối với mỗi font chữ, vượt quá giới hạn này, các thẻ <BIG> sẽ không có ý nghĩa.
<SMALL> ... </SMALL>	In chữ nhỏ hơn bình thường bằng cách giảm kích thước font hiện thời đi một. Việc sử dụng các thẻ <SMALL> lồng nhau tạo ra hiệu ứng chữ giảm dần. Tuy nhiên đối với mỗi trình duyệt có giới hạn về kích thước đối với mỗi font chữ, vượt quá giới hạn này, các thẻ <SMALL> sẽ không có ý nghĩa.
<SUP> ... </SUP>	Định dạng chỉ số trên (SuperScript)
<SUB> ... </SUB>	Định dạng chỉ số dưới (SubScript)
<BASEFONT>	Định nghĩa kích thước font chữ được sử dụng cho đến hết văn bản. Thẻ này chỉ có một tham số size= xác định cỡ chữ. Thẻ <BASEFONT> không có thẻ kết thúc.
 ... 	Chọn kiểu chữ hiển thị. Trong thẻ này có thể đặt hai tham số size= hoặc color= xác định cỡ chữ và màu sắc đoạn văn bản nằm giữa hai thẻ. Kích thước có thể là tuyệt đối (nhận giá trị từ 1 đến 7) hoặc tương đối (+2,-4...) so với font chữ hiện tại.

2.4.2. Căn lề văn bản trong trang web

Trong trình bày trang Web của mình các bạn luôn phải chú ý đến việc căn lề các văn bản để trang Web có được một bố cục đẹp. Một số các thẻ định dạng như **P**, **Hn**, **IMG**... đều có tham số **ALIGN** cho phép bạn căn lề các văn bản nằm trong phạm vi giới hạn bởi của các thẻ đó.

Các giá trị cho tham số **ALIGN**:

LEFT	Căn lề trái
CENTER	Căn giữa trang
RIGHT	Căn lề phải

Ngoài ra, chúng ta có thể sử dụng thẻ **CENTER** để căn giữa trang một khối văn bản.

Cú pháp:

```
<CENTER>Văn bản sẽ được căn giữa trang</CENTER>
```

2.4.3. Các ký tự đặc biệt

Ký tự & được sử dụng để chỉ chuỗi ký tự đi sau được xem là một thực thể duy nhất. Ký tự ; được sử dụng để tách các ký tự trong một từ.

Ký tự	Mã ASCII	Tên chuỗi
<	<	<
>	>	>
&	&	&

2.4.4. Sử dụng màu sắc trong thiết kế các trang web

Một màu được tổng hợp từ ba thành phần màu chính, đó là: Đỏ (Red), Xanh lá cây (Green), Xanh nước biển (Blue). Trong HTML một giá trị màu là một số nguyên dạng hexa (hệ đếm cơ số 16) có định dạng như sau:

#RRGGBB

trong đó:

RR - là giá trị màu Đỏ.

GG - là giá trị màu Xanh lá cây.

BB - là giá trị màu Xanh nước biển.

Màu sắc có thể được xác định qua thuộc tính bgcolor= hay color=. Sau dấu bằng có thể là giá trị RGB hay tên tiếng Anh của màu. Với tên tiếng Anh, ta chỉ có thể chỉ ra 16 màu trong khi với giá trị RGB ta có thể chỉ tới 256 màu.

Sau đây là một số giá trị màu cơ bản:

Màu sắc	Giá trị	Tên tiếng Anh
Đỏ	#FF0000	RED
Đỏ sẫm	#8B0000	DARKRED
Xanh lá cây	#00FF00	GREEN
Xanh nhạt	#90EE90	LIGHTGREEN
Xanh nước biển	#0000FF	BLUE
Vàng	#FFFF00	YELLOW
Vàng nhạt	#FFFFE0	LIGHTYELLOW
Trắng	#FFFFFF	WHITE
Đen	#000000	BLACK
Xám	#808080	GRAY
Nâu	#A52A2A	BROWN
Tím	#FF00FF	MAGENTA
Tím nhạt	#EE82EE	VIOLET
Hồng	#FFC0CB	PINK
Da cam	#FFA500	ORANGE
Màu đồng phục hải quân	#000080	NAVY
	#4169E1	ROYALBLUE
	#7FFFD4	AQUAMARINE

Cú pháp:

```
<BODY  
  LINK           = color  
  ALINK          = color  
  VLINK         = color  
  BACKGROUND    = url  
  BGCOLOR       = color  
  TEXT          = color  
  TOPMARGIN     = pixels  
  RIGHTMARGIN  = pixels  
  LEFTMARGIN   = pixels  
>  
.... phần nội dung của tài liệu được đặt ở đây  
</BODY>
```

Sau đây là ý nghĩa các tham số của thẻ BODY:

Các tham số	ý nghĩa
LINK	Chỉ định màu của văn bản siêu liên kết
ALINK	Chỉ định màu của văn bản siêu liên kết đang chọn
VLINK	Chỉ định màu của văn bản siêu liên kết đã từng mở
BACKGROUND	Chỉ định địa chỉ của ảnh dùng làm nền
BGCOLOR	Chỉ định màu nền
TEXT	Chỉ định màu của văn bản trong tài liệu
SCROLL	YES/NO - Xác định có hay không thanh cuộn
TOPMARGIN	Lề trên
RIGHTMARGIN	Lề phải
LEFTMARGIN	Lề trái

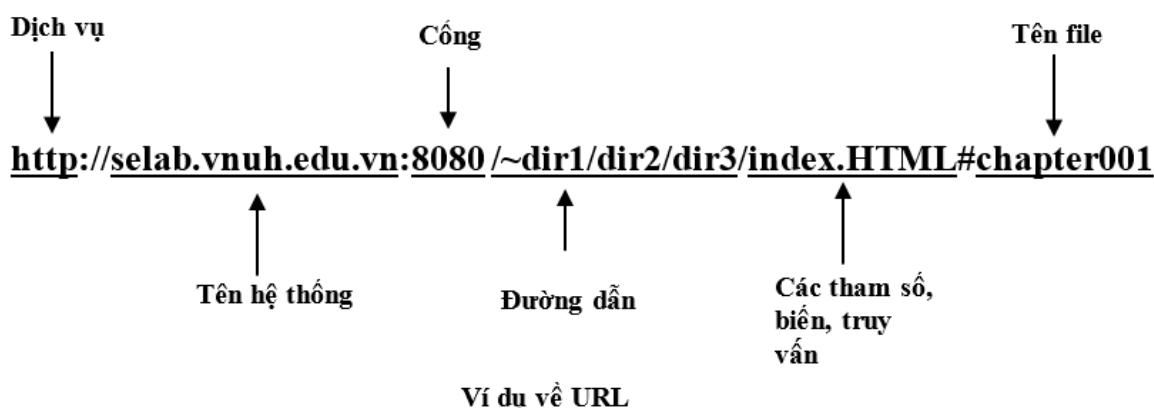
2.4.5. Chọn kiểu chữ cho văn bản

Cú pháp:

```
<FONT  
  FACE           = font-name  
  COLOR         = color  
  SIZE          = n >  
...  
</FONT>
```

2.4.6. Khái niệm văn bản siêu liên kết

Văn bản siêu liên kết hay còn gọi là siêu văn bản là một từ, một cụm từ hay một câu trên trang Web được dùng để liên kết tới một trang Web khác. Siêu văn bản là môi trường trong đó chứa các liên kết (link) của các thông tin. Do WWW cấu thành từ nhiều hệ thống khác nhau, cần phải có một quy tắc đặt tên thống nhất cho tất cả các văn bản trên Web. Quy tắc đặt tên đó là URL (Universal Resource Locator).



Các thành phần của URL được minh họa ở hình trên.

- **Dịch vụ**: Là thành phần bắt buộc của URL. Nó xác định cách thức trình duyệt của máy khách liên lạc với máy phục vụ như thế nào để nhận dữ liệu. Có nhiều dịch vụ như **http, wais, ftp, gopher, telnet**.
- **Tên hệ thống**: Là thành phần bắt buộc của URL. Có thể là tên miền đầy đủ của máy phục vụ hoặc chỉ là một phần tên đầy đủ – trường hợp này xảy ra khi văn bản được yêu cầu vẫn nằm trên miền của bạn. Tuy nhiên nên sử dụng đường dẫn đầy đủ.
- **Cổng**: Không là thành phần bắt buộc của URL. Cổng là địa chỉ socket của mạng dành cho một giao thức cụ thể. Giao thức http ngầm định nối với cổng 8080.
- **Đường dẫn thư mục**: Là thành phần bắt buộc của URL. Phải chỉ ra đường dẫn tới file yêu cầu khi kết nối với bất kỳ hệ thống nào. Có thể đường dẫn trong URL khác với đường dẫn thực sự trong hệ thống máy phục vụ. Tuy nhiên có thể rút gọn đường dẫn bằng cách đặt biệt danh (alias). Các thư mục trong đường dẫn cách nhau bởi dấu gạch chéo (/).
- **Tên file**: Không là thành phần bắt buộc của URL. Thông thường máy phục vụ được cấu hình sao cho nếu không chỉ ra tên file thì sẽ trả về file ngầm định trên thư mục được yêu cầu. File này thường có tên là index.html, index.htm, default.html hay default.htm. Nếu cũng không có các file này thì thường kết quả trả về là danh sách liệt kê các file hay thư mục con trong thư mục được yêu cầu.
- **Các tham số**: Không là thành phần bắt buộc của URL. Nếu URL là yêu cầu tìm kiếm trên một cơ sở dữ liệu thì truy vấn sẽ gắn vào URL, đó chính là đoạn mã đằng sau dấu chấm hỏi (?). URL cũng có thể trả lại thông tin được thu thập từ form. Trong trường hợp dấu thăng (#) xuất hiện đoạn mã đằng sau là tên của một vị trí (location) trong file được chỉ ra.

Để tạo ra một siêu văn bản chúng ta sử dụng thẻ <A>.

Cú pháp:

```

<A
    HREF      = url
    NAME      = name
    TABINDEX  = n
    TITLE     = title
    TARGET    = _blank / _self
>
... siêu văn bản
</A>

```

Ý nghĩa các tham số:

HREF	Địa chỉ của trang Web được liên kết, là một URL nào đó.
NAME	Đặt tên cho vị trí đặt thẻ.
TABLEINDEX	Thứ tự di chuyển khi ấn phím Tab
TITLE	Văn bản hiển thị khi di chuột trên siêu liên kết.
TARGET	Mở trang Web được liên trong một cửa sổ mới (<i>_blank</i>) hoặc trong cửa sổ hiện tại (<i>_self</i>), trong một frame (tên frame).

2.4.7. Địa chỉ tương đối

URL được trình bày ở trên là URL tuyệt đối. Ngoài ra còn có URL tương đối hay còn gọi là URL không đầy đủ. Địa chỉ tương đối sử dụng sự khác biệt tương đối giữa văn bản hiện thời và văn bản cần tham chiếu tới. Các thành phần trong URL được ngăn cách bằng ký tự ngăn cách (ký tự gạch chéo /). Để tạo ra URL tương đối, đầu tiên phải sử dụng ký tự ngăn cách. URL đầy đủ hiện tại sẽ được sử dụng để tạo nên URL đầy đủ mới. Nguyên tắc là các thành phần bên trái dấu ngăn cách của URL hiện tại được giữ nguyên, các thành phần bên phải được thay thế bằng thành phần URL tương đối. Chú ý rằng trình duyệt không gửi URL tương đối, nó bổ sung vào URL cơ sở đã xác định trước thành phần URL tương đối xác định sau thuộc tính href=. Ký tự đầu tiên sau dấu bằng sẽ xác định các thành phần nào của URL hiện tại sẽ tham gia để tạo nên URL mới.

Ví dụ, nếu URL đầy đủ là: <http://it-department.vnuh.edu.vn/HTML/index.htm> thì:

- **Dấu hai chấm (:)** chỉ dịch vụ giữ nguyên nhưng thay đổi phần còn lại. Ví dụ [://www.fpt.com/](http://www.fpt.com/) sẽ tải trang chủ của máy phục vụ www.fpt.com với cùng dịch vụ http.
- **Dấu gạch chéo (/)** chỉ dịch vụ và máy phục vụ giữ nguyên nhưng toàn bộ đường dẫn thay đổi. Ví dụ [/Javascript/index.htm](http://www.it-department.vnuh.edu.vn/Javascript/index.htm) sẽ tải file [index.htm](http://www.it-department.vnuh.edu.vn/Javascript/index.htm) của thư mục **Javascript** trên máy phục vụ www.it-department.vnuh.edu.vn.
- **Không có dấu phân cách** chỉ có tên file là thay đổi. Ví dụ [index1.htm](http://www.it-department.vnuh.edu.vn/index1.htm) sẽ tải file [index1.htm](http://www.it-department.vnuh.edu.vn/index1.htm) ở trong thư mục **HTML** của máy phục vụ www.it-department.vnuh.edu.vn.
- **Dấu thăng (#)**: chỉ dịch vụ, máy phục vụ, đường dẫn và cả tên file giữ nguyên, chỉ thay đổi vị trí trong file.

Do đường dẫn được xem là đơn vị độc lập nên có thể sử dụng phương pháp đường dẫn tương đối như trong UNIX hay MS-DOS (tức là `.` chỉ thư mục hiện tại còn `..` chỉ thư mục cha của thư mục hiện tại).

URL cơ sở có thể được xác định bằng thẻ `<BASE>`.

2.4.8. Kết nối mailto

Nếu đặt thuộc tính `href=` của thẻ `<a>` giá trị `mailto:address@domain` thì khi kích hoạt kết nối sẽ kích hoạt chức năng thư điện tử của trình duyệt.

```
<ADDRESS>
Trang WEB này được
<A href="mailto:webmaster@vnuh.edu.vn" >
WEBMASTER
<\A> bảo trì
<\ADDRESS>
```

2.4.9. Vẽ một đường thẳng nằm ngang

Cú pháp:

```
<HR
    ALIGN          = LEFT / CENTER / RIGHT
    COLOR          = color
    NOSHADE
    SIZE           = n
    WIDTH          = width
>
```

Ý nghĩa các tham số:

ALIGN	Căn lề (căn trái, căn phải, căn giữa)
COLOR	Đặt màu cho đường thẳng
NOSHADE	Không có bóng
SIZE	Độ dày của đường thẳng
WIDTH	Chiều dài (tính theo pixel hoặc % của bề rộng cửa sổ trình duyệt).

Thẻ này giống như thẻ **BR**, nó cũng không có thẻ kết thúc tương ứng.

2.5. Các thẻ chèn âm thanh, hình ảnh

2.5.1. Giới thiệu

Liên kết với file đa phương tiện cũng tương tự như liên kết bình thường. Tuy vậy phải đặt tên đúng cho file đa phương tiện. Phần mở rộng của file phải cho biết kiểu của file.

KIỂU	PHÂN MỞ RỘNG	MÔ TẢ
Image/GIF	.gif	Viết tắt của Graphics Interchange Format. Khuôn dạng này xuất hiện khi mọi người có nhu cầu trao đổi ảnh trên nhiều hệ thống khác nhau. Nó được sử dụng trên tất cả các hệ thống hỗ trợ giao diện đồ họa. Định dạng GIF là định dạng chuẩn cho mọi trình duyệt WEB. Nhược điểm của nó là chỉ thể hiện được 256 màu. Mở rộng của chuẩn này là GIF89, được thêm nhiều chức năng cho các ứng dụng đặc biệt như làm ảnh nền trong suốt - tức là ảnh có thể nổi bằng cách làm màu nền giống với màu nền của trình duyệt.
Image/JPEG	.jpeg	Viết tắt của Joint Photographic Expert Group. Là khuôn dạng ảnh khác nhưng có thêm khả năng nén. Ưu điểm nổi bật của khuôn dạng này là lưu trữ được hàng triệu màu và độ nén cao nên kích thước file ảnh nhỏ hơn và thời gian download nhanh hơn. Nó là cơ sở cho khuôn dạng MPEG. Tất cả các trình duyệt đều có khả năng xem ảnh JPEG.
Image/TIFF	.tiff	Viết tắt của Tagged Image File Format. Được Microsoft thiết kế để quét ảnh từ máy quét cũng như tạo các ấn phẩm.
Text/HTML	.HTML, .htm	
PostScript	.eps, .ps	Được tạo ra để hiển thị và in các văn bản có chất lượng cao.
Adobe Acrobat	.pdf	Viết tắt của Portable Document Format. Acrobat cũng sử dụng các siêu liên kết ngay trong văn bản cũng giống như HTML . Từ phiên bản 2.0, các sản phẩm của Acrobat cho phép liên kết giữa nhiều văn bản. Ưu điểm lớn nhất của nó là khả năng WYSISYG.
Video/MPEG	.mpeg	Viết tắt của Motion Picture Expert Group, là định dạng dành cho các loại phim (video). Đây là khuôn dạng thông dụng nhất dành cho phim trên WEB.
Video/AVI	.avi	Là khuôn dạng phim do Microsoft đưa ra.
Video/QuickTime	.mov	Do Apple Computer đưa ra, chuẩn video này được cho là có nhiều ưu điểm hơn MPEG và AVI. Mặc dù đã được tích hợp vào nhiều trình duyệt nhưng vẫn chưa phổ biến bằng hai loại định dạng trên.
Sound/AU	.au	
Sound/MIDI	.mid	Là khuôn dạng dành cho âm nhạc điện tử hết sức thông dụng được nhiều trình duyệt trên các hệ thống khác nhau hỗ trợ. File Midi được tổng hợp số hoá trực tiếp từ máy tính.
Sound/RealAudio	.ram	Định dạng audio theo dòng. Một bất tiện khi sử dụng các định dạng khác là file âm thanh thường có kích thước lớn - do vậy thời gian tải xuống lâu, Trái lại audio dòng bắt đầu chơi ngay khi tải được một phần file trong khi vẫn tải về các phần khác. Mặc dù file theo định dạng này không nhỏ hơn so với các định dạng khác song chính khả năng

		dòng đã khiến định dạng này phù hợp với khả năng chơi ngay lập tức.
VRML	.vrm	Viết tắt của Virtual Reality Modeling Language. Các file theo định dạng này cũng giống như HTML . Tuy nhiên do trình duyệt có thể hiển thị được cửa sổ 3 chiều nên người xem có thể cảm nhận được cảm giác ba chiều.

2.5.2. Đưa âm thanh vào một tài liệu HTML

Cú pháp:

```
<BGSOUND
  SRC = url
  LOOP = n
>
```

Thẻ này không có thẻ kết thúc tương ứng (</BGSOUND>). Để chơi lặp lại vô hạn cần chỉ định **LOOP = -1** hoặc **LOOP = INFINITE**. Thẻ **BGSOUND** phải được đặt trong phần mở đầu (tức là nằm trong cặp thẻ **HEAD**).

2.5.3. Chèn một hình ảnh, một đoạn video vào tài liệu HTML

Để chèn một file ảnh (.jpg, .gif, .bmp) hoặc video (.mpg, .avi) vào tài liệu HTML, bạn có thể sử dụng thẻ **IMG**.

Cú pháp:

```
<IMG
ALIGN      = TOP/MIDDLE/BOTTOM
ALT        = text
BORDER     = n
SRC        = url
WIDTH     = width
HEIGHT    = height
HSPACE    = vspace
VSPACE    = hspace
TITLE     = title
DYN SRC    = url
START     = FILEOPEN/MOUSEOVER
LOOP      = n
>
```

Trong đó:

ALIGN = TOP/ MIDDLE/ BOTTOM/ LEFT/ RIGHT	Căn hàng văn bản bao quanh ảnh
ALT = text	Chỉ định văn bản sẽ được hiển thị nếu chức năng show picture của browser bị tắt đi hay hiển thị thay thế cho ảnh trên những trình duyệt không có khả năng hiển thị

đồ họa. Văn bản này còn được gọi là nhãn của ảnh. Đối với trình duyệt có khả năng hỗ trợ đồ họa, dòng văn bản này sẽ hiện lên khi di chuột qua ảnh hay được hiển thị trong vùng của ảnh nếu ảnh chưa được tải về hết. Chú ý phải đặt văn bản trong hai dấu nháy kép nếu trong văn bản chứa dấu cách hay các ký tự đặc biệt - trong trường hợp ngược lại có thể bỏ dấu nháy kép.

BORDER = n	Đặt kích thước đường viền được vẽ quanh ảnh (tính theo pixel).
SRC = url	Địa chỉ của file ảnh cần chèn vào tài liệu.
WIDTH/HEIGHT	Chỉ định kích thước của ảnh được hiển thị.
HSPACE/VSPACE	Chỉ định khoảng trống xung quanh hình ảnh (tính theo pixel) theo bốn phía trên, dưới, trái, phải.
TITLE = title	Văn bản sẽ hiển thị khi con chuột trỏ trên ảnh
DYNSRC = url	Địa chỉ của file video.
START = FILEOPEN/MOUSEOVER	Chỉ định file video sẽ được chơi khi tài liệu được mở hay khi trỏ con chuột vào nó. Có thể kết hợp cả hai giá trị này nhưng phải phân cách chúng bởi dấu phẩy.
LOOP = n/INFINITE	Chỉ định số lần chơi. Nếu LOOP = INFINITE thì file video sẽ được chơi vô hạn lần.

2.6. Các thẻ định dạng bảng biểu

Sau đây là các thẻ tạo bảng chính:

<TABLE> ... </TABLE>	Định nghĩa một bảng
<TR> ... </TR>	Định nghĩa một hàng trong bảng
<TD> ... </TD>	Định nghĩa một ô trong hàng
<TH> ... </TH>	Định nghĩa ô chứa tiêu đề của cột
<CAPTION> ... </CAPTION>	Tiêu đề của bảng

Cú pháp:

<TABLE	
ALIGN	= LEFT / CENTER / RIGHT
BORDER	= n
BORDERCOLOR	= color
BORDERCOLORDARK	= color
BORDERCOLORLIGHT	= color
BACKGROUND	= url
BGCOLOR	= color
CELLSPACING	= spacing
CELLPADDING	= padding
>	
<CAPTION>Tiêu đề của bảng biểu</CAPTION>	
... Định nghĩa các dòng	

```

<TR
  ALIGN = LEFT/CENTER/RIGHT
  VALIGN = TOP/MIDDLE/BOTTOM>
  ... Định nghĩa các ô trong dòng
<TD
  ALIGN                = LEFT / CENTER / RIGHT
  VALIGN              = TOP / MIDDLE / BOTTOM
  BORDERCOLOR        = color
  BORDERCOLORDARK    = color
  BORDERCOLORLIGHT   = color
  BACKGROUND          = url
  BGCOLOR            = color
  COLSPAN            = n
  ROWSPAN            = n
>
... Nội dung của ô
</TD>
...
</TR>
...
</TABLE>

```

Ý nghĩa các tham số:

ALIGN / VALIGN	Căn lề cho bảng và nội dung trong mỗi ô.
BORDER	Kích thước đường kẻ chia ô trong bảng, được đo theo pixel. Giá trị 0 có nghĩa là không xác định lề, giữa các ô trong bảng chỉ có một khoảng trắng nhỏ để phân biệt. Nếu chỉ để border thì ngầm định border=1. Với những bảng có cấu trúc phức tạp, nên đặt lề để người xem có thể phân biệt rõ các dòng và cột.
BORDERCOLOR	Màu đường kẻ
BORDERCOLORDARK	Màu phía tối và phía sáng cho đường kẻ nổi.
BORDERCOLORLIGHT	
BACKGROUND	Địa chỉ tới tệp ảnh dùng làm nền cho bảng
BGCOLOR	Màu nền
CELLSPACING	Khoảng cách giữa các ô trong bảng
CELLPADDING	Khoảng cách giữa nội dung và đường kẻ trong mỗi ô của bảng.
COLSPAN	Chỉ định ô sẽ kéo dài trong bao nhiêu cột
ROWSPAN	Chỉ định ô sẽ kéo dài trong bao nhiêu hàng

2.7. FORM

2.7.1. HTML FORMS

Các HTML Form có thể có các hộp văn bản, hộp danh sách lựa chọn, nút bấm, nút chọn...

2.7.2. TẠO FORM

Để tạo ra một form trong tài liệu HTML, chúng ta sử dụng thẻ FORM với cú pháp như sau:

Cú pháp:

```
<FORM
  ACTION      = url
  METHOD      = GET | POST
  NAME       = name
  TARGET     = frame_name | _blank | _self
>
<!-- Các phần tử điều khiển của form được đặt ở đây -->
<INPUT ...>
<INPUT ...>

</FORM>
```

Trong đó

ACTION	Địa chỉ sẽ gửi dữ liệu tới khi form được submit (có thể là địa chỉ tới một chương trình CGI, một trang ASP...).
METHOD	Phương thức gửi dữ liệu.
NAME	Tên của form.
TARGET	Chỉ định cửa sổ sẽ hiển thị kết quả sau khi gửi dữ liệu từ form đến server.

Đặt các đối tượng điều khiển (như hộp văn bản, ô kiểm tra, nút bấm...) vào trang Web

Cú pháp thẻ **INPUT**:

```
<INPUT
  ALIGN = LEFT | CENTER | RIGHT
  TYPE  = BUTTON | CHECKBOX | FILE | IMAGE | PASSWORD | RADIO
  | RESET | SUBMIT | TEXT
  VALUE = value
>
```

2.7.3. TẠO MỘT DANH SÁCH LỰA CHỌN

Cú pháp:

```
<SELECT NAME="tên danh sách" SIZE="chiều cao">
<OPTION VALUE=value1 SELECTED> Tên mục chọn thứ nhất
<OPTION VALUE=value2> Tên mục chọn thứ hai
<!-- Danh sách các mục chọn -->
</SELECT>
```

2.7.4. TẠO HỘP SOẠN THẢO VĂN BẢN

Cú pháp:

```
<TEXTAREA
  COLS=số cột
  ROWS=số hàng
  NAME=tên
>
  Văn bản ban đầu
</TEXTAREA>
```

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 2

1. Tạo trang web chứa các hyperlink dùng để truy cập nhanh

- Tạo trang web đặt tên là **Links.htm** chứa các hyperlink đã đề cập ở trên. Bổ sung thêm các hyperlink và các phân loại khác mà các anh chị đã biết.
- Upload lên website và kiểm tra lại.

2. Tạo trang HomePage

- Tạo trang web HomePage đặt tên là **Default.htm** giới thiệu về nhóm và các công việc mà nhóm đang triển khai.
- Kết nối hai trang đã tạo vào HomePage.
- Upload lên website và kiểm tra lại.

3. Thiết kế website theo mẫu sau (<http://www.is-edu.hcmuns.edu.vn>)

Welcome to IS-EDU Site - Microsoft Internet Explorer

Address <http://www.is-edu.hcmuns.edu.vn/>

IS-EDU Hệ thống thông tin hỗ trợ Sinh viên

ĐH Khoa Học Tự Nhiên
Khoa Công Nghệ Thông Tin

Trang chủ Eureka Thảo luận Nghiên cứu Thư viện Thư điện tử CLB Tin Học Tìm kiếm

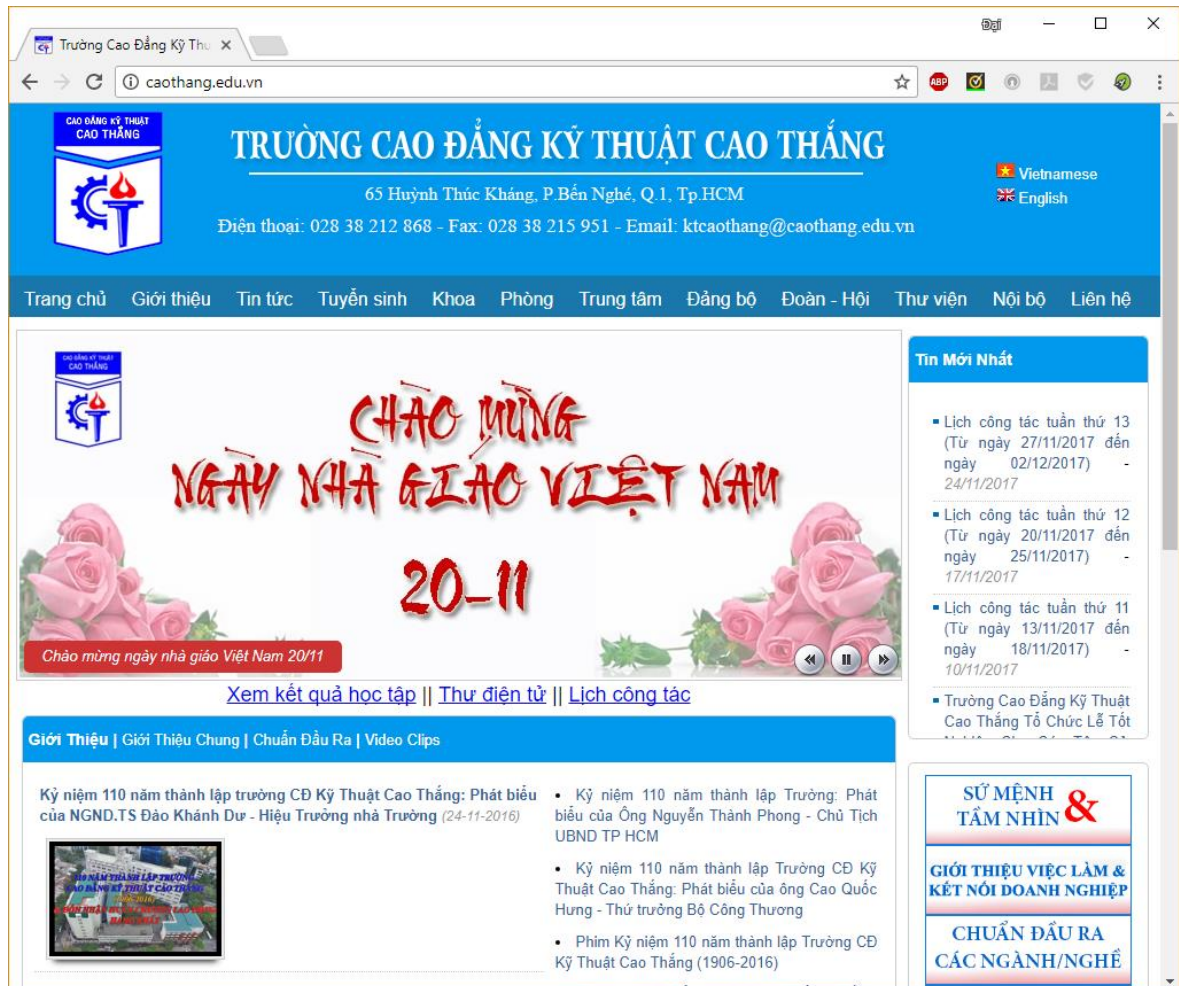
Số lần truy cập: 0136171

Tổng hợp các tin mới - TS tin: 172

- ▶ [Bằng chứng điện tử: vũ khí chống lại tội phạm máy tính](#) 7/21/2001 *New!*
- ▶ [Phát hiện lỗ hổng trong cơ sở dữ liệu Oracle8i](#) 7/21/2001 *New!*
- ▶ [Hacker tấn công nhiều làm thị trường bảo mật dữ liệu sinh lợi](#) 7/21/2001 *New!*
- ▶ [Liệu các công ty có thể tự đối phó với hacker ? \(Kỳ cuối\)](#) 7/21/2001 *New!*
- ▶ [Liệu các công ty có thể tự đối phó với hacker ? \(Kỳ D\)](#) 7/21/2001 *New!*
- ▶ [Thiết bị không dây trước họng súng của hacker](#) 7/21/2001 *New!*
- ▶ [Mạng IRC Efnet bị tấn công theo kiểu DDoS](#) 7/21/2001 *New!*
- ▶ [Ping có thể là dấu hiệu của các hacker nguy hiểm \(Kỳ D\)](#) 7/21/2001 *New!*
- ▶ [Nhóm hacker Poison Box với nạn nhân mới](#) 7/21/2001 *New!*
- ▶ [Hacking 101 \(kỳ cuối\)](#) 7/21/2001 *New!*
- ▶ [679 Website bị tấn công trong vòng một phút](#) 7/21/2001 *New!*
- ▶ [Liên minh hacker chưa phát hành Peekabooby](#) 7/21/2001 *New!*
- ▶ [Phòng văn hacker gửi Viagra cho Bill Gates](#) 7/21/2001 *New!*
- ▶ [Hacking 101 \(Kỳ D\)](#) 7/21/2001 *New!*
- ▶ [Chương trình tấn công khai thác lỗ hổng IIS của Microsoft](#) 7/21/2001 *New!*
- ▶ [Loại worm thứ hai núp dưới dạng bản tin bảo mật của Microsoft](#) 7/21/2001 *New!*
- ▶ [Dễ dàng tiêu diệt worm Code Red](#) 7/21/2001 *New!*

http://www.is-edu.hcmuns.edu.vn/WebNews/ShowContent.asp?NEWS_ID=2751 Local intranet

4. Thiết kế website theo mẫu sau (<http://caothang.edu.vn/>)



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

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

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

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

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

CHƯƠNG 3. SỬ DỤNG DREAMWEAVER THIẾT KẾ WEBSITE

Adobe Dreamweaver là một công cụ xử lý mạnh mẽ dành cho những người thiết kế web, người dùng có thể tự mình lập trình và phát triển ứng dụng web ở nhiều cấp độ. Nếu chỉ dùng ở mức độ hiểu biết chưa nhiều về các ngôn ngữ lập trình web thì *Dreamweaver* vẫn đáp ứng được nhu cầu thiết kế Website chuyên nghiệp.

Người dùng không chuyên chỉ việc cần tìm cho mình một mẫu trang web vừa ý trên Internet, dùng công cụ soạn thảo thông thường để chỉnh sửa hay thêm bớt một số thành phần, rồi tạo ra một *Template* để áp dụng cho toàn bộ Website.

Ngoài những tính năng kéo thả để xây dựng trang web, *Dreamweaver* còn cung cấp một môi trường viết mã với đầy đủ chức năng bao gồm các công cụ viết mã (như tô màu mã, bổ sung thẻ tag, thanh công cụ mã), hỗ trợ các ngôn ngữ lập trình thông dụng *HTML*, *CSS*, *Javascript*, *ASP*, *VBScript*, *PHP* hay *XML*. Nếu muốn xây dựng các ứng dụng web động chạy trên công nghệ máy chủ *ASP.NET*, *ASP*, *JSP* và *PHP* thì *Dreamweaver* hoàn toàn có thể đáp ứng được các dự án lớn này.

Giao diện sử dụng được bố trí trực quan và thân thiện với người sử dụng, khu vực soạn thảo đoạn mã được đánh số thứ tự ở các dòng và có tô màu theo từng thẻ giúp tiện cho việc theo dõi, kiểm tra. Ngoài giao diện mặc định, chương trình còn hỗ trợ nhiều dạng giao diện với cách bố trí vị trí thanh công cụ khác nhau, tại tính năng *Designer* (đối với phiên bản *Dreamweaver CS5*).

Adobe Dreamweaver có thể kết hợp các phần mềm khác của hãng *Adobe* để tạo ra một sản phẩm hoàn hảo, ví dụ như *Adobe Photoshop* giúp chỉnh sửa và thiết kế hình ảnh cho Website.

3.1. Tạo thư mục chứa bộ web

Trước khi bắt tay vào việc tạo ra một trang web thì công việc đầu tiên của người thiết kế là phải tạo ra một thư mục để chứa toàn bộ những gì liên quan đến trang web, ví dụ *webserver*.

Trong thư mục đó, bạn sẽ tạo tiếp các thư mục khác dùng để chứa dữ liệu như:

Thư mục *Image* dùng để chứa toàn bộ hình ảnh của trang web.

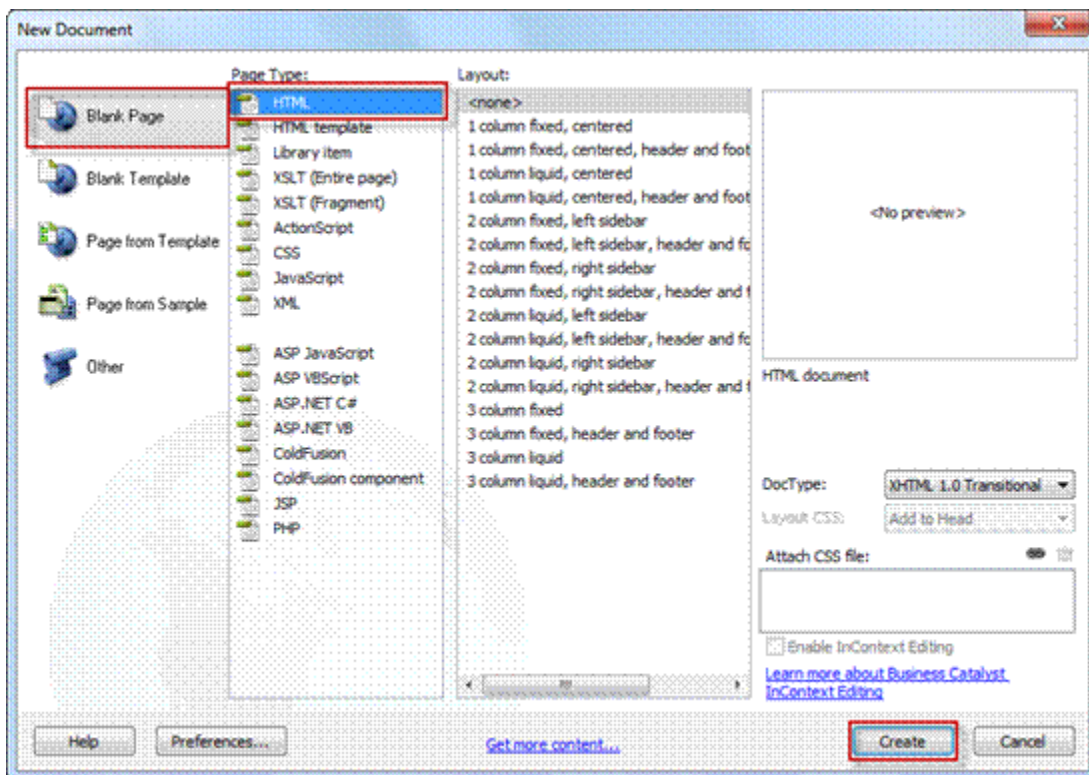
Thư mục *Flash* dùng để chứa những tập tin *flash*, *video clip*, ...

Thư mục *Data* dùng để chứa dữ liệu

Và các thư mục khác tùy theo mục đích của người tạo *web*.

3.2. Tạo mới một trang web

Khởi động *Dreamweaver*, trong cửa sổ đầu tiên của chương trình chọn *HTML* để mở một trang tài liệu mới. Hoặc vào menu *File*, chọn *New*. Trong mục *Blank Page* chọn *HTML* và bấm vào nút *Create* để khởi tạo một trang mới (hình 1).

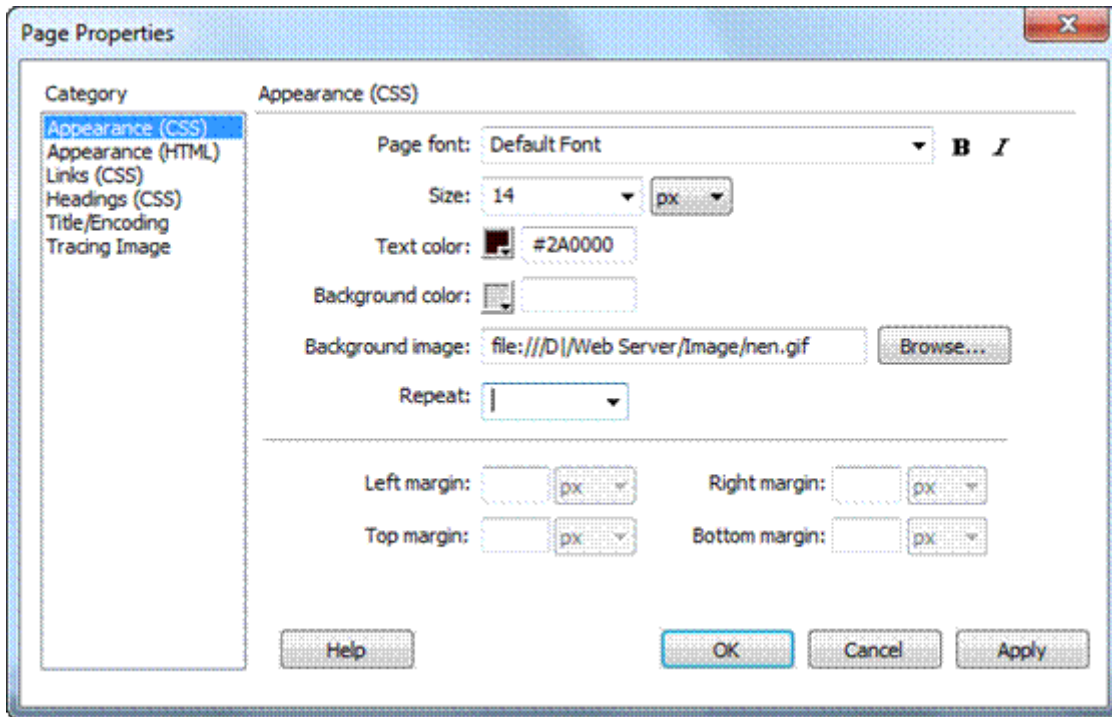


Hình 3.1. Khởi tạo trang web

3.3. Lưu một trang web

Trước khi bắt tay vào thiết kế, bạn nên lưu trang web lại bằng cách vào menu *File*, chọn *Save* hoặc *Save as*. Sau đó lưu trang web vào thư mục chứa bộ web với dạng **.html*. Lưu ý, trang chủ của website bắt buộc phải đặt tên là *index.html* và phải được lưu ngay trong thư mục chứa bộ web.

3.4. Định dạng trang web.



Hình 3.2. Thuộc tính trang web

Sau khi lưu lại trang web thành công, vào menu *Modify* chọn *Page Properties* để cài đặt định dạng cho trang web.

Trong hộp thoại *Page Properties* bạn chỉnh các thông số sau (hình 2):

Mục *Appearance (HTML)*:

+ *Background image*: Chọn *Browse* và duyệt đến tập tin ảnh dùng làm nền cho trang web.

+ *Background*: Bấm vào nút bảng màu và chỉ ra màu nền cho trang web.

+ *Text*: Chỉ định màu chữ mặc định.

+ *Link*: Chỉ định màu chữ mặc định khi bạn tạo liên kết cho nó.

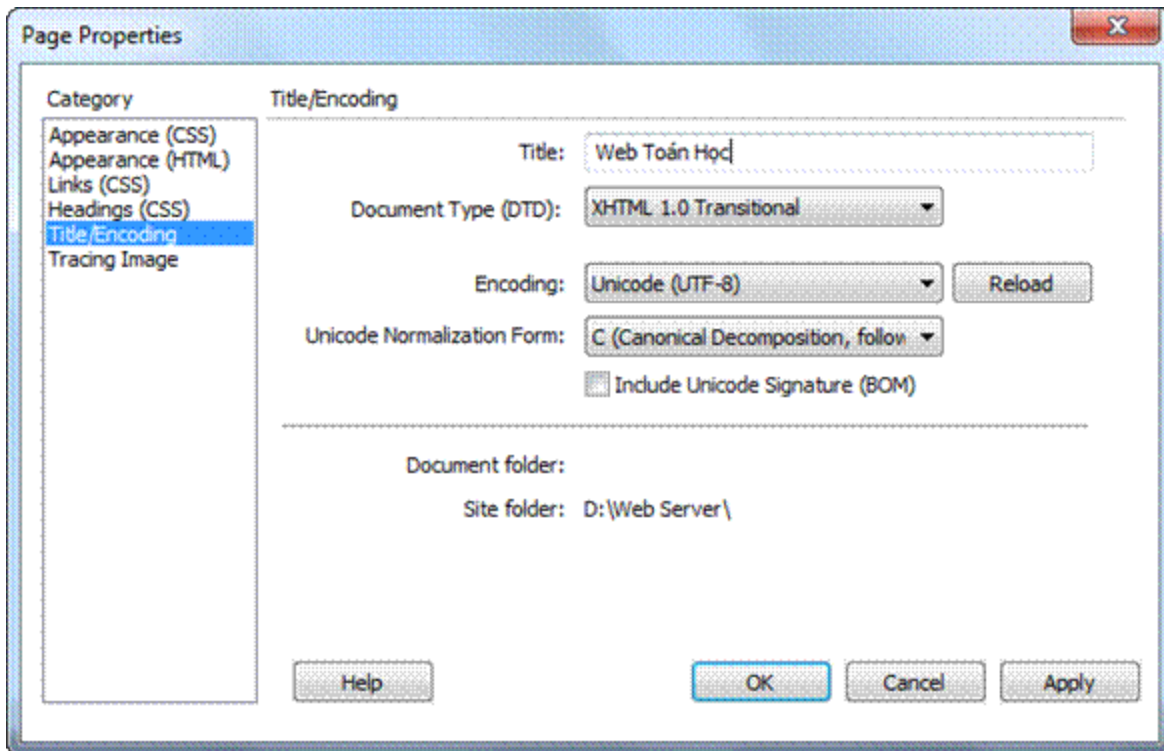
+ *Left margin*: Nhập vào khoảng cách lề trái so với nội dung trang web.

+ *Top margin*: Nhập vào khoảng cách lề trên so với nội dung trang web.

+ *Margin width*: nhập vào độ rộng của trang web.

+ *Margin height*: nhập vào độ cao của trang web.

Mục *Title/Encoding*:



Hình 3.3. Tiêu đề và bảng mã

+ *Title*: Nhập tiêu đề của trang web.

+ Các mục khác cứ để mặc định.

Sau khi tinh chỉnh định dạng trang web xong, chọn OK để thiết lập (hình 3).

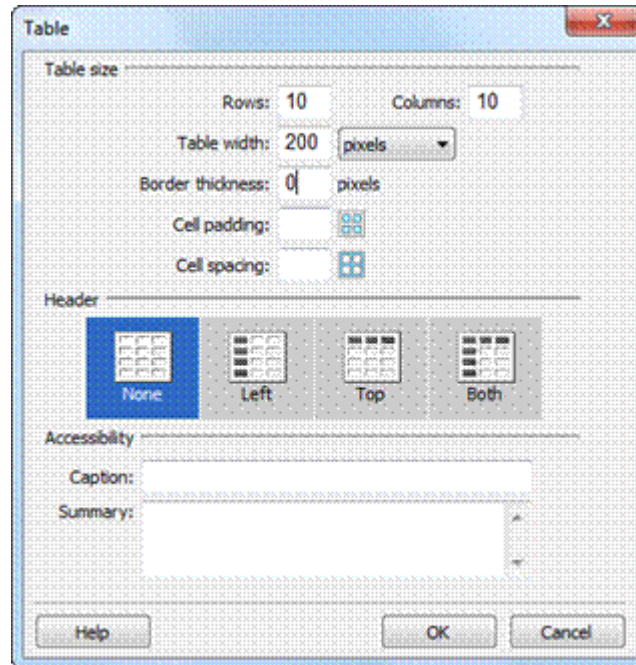
3.5. Xem trang web trên trình duyệt

Trong quá trình làm việc, nếu muốn xem thử trang *web* trên trình duyệt thì bạn bấm *Ctrl+S* để lưu lại, sau đó bấm phím *F12* để xem trên trình duyệt đã cài đặt làm mặc định hoặc bấm chọn vào biểu tượng có hình trái đất ở bên trên và chọn trình duyệt cần xem.

3.6. Tạo bảng trong trang web

- *Chèn bảng*

Nếu muốn tạo một trang web có các nội dung được đặt ở các vị trí phù hợp và dễ tìm ra trên các bộ máy tìm kiếm thì bạn cần tạo ra một bảng trước khi nhập nội dung. Để thực hiện, bạn vào *Menu Insert*, chọn *Table*.

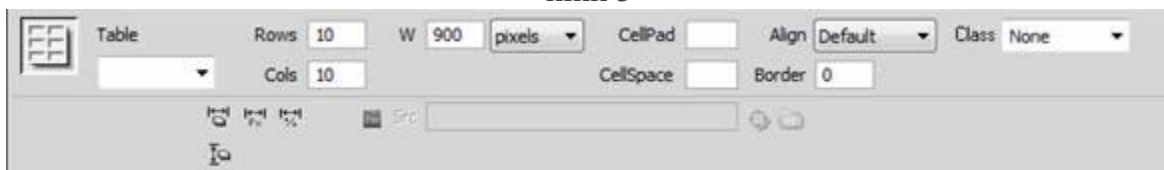


Hình 3.4. Thiết lập thuộc tính bảng

Hộp thoại *Table* hiện ra, nhập số dòng trong mục *Rows*, số cột trong mục *Columns*, độ rộng của *Table* trong mục *Table width* và trong mục *Border thickness* nhập vào giá trị độ lớn của khung trong bảng sau đó bấm OK (hình 4).

- Định dạng bảng

Bấm chuột vào đường khung của bảng, khung thuộc tính của bảng hiện ra bên dưới như hình 5



Hình 3.5. Định dạng bảng

Thiết lập bảng thuộc tính gồm các nội dung sau:

W, H: Chiều rộng và chiều cao của bảng.

Rows: Số hàng.

Cols: Số cột.

CellPad: khoảng cách từ văn bản đến bảng.

CellSpace: khoảng cách giữa các ô của bảng.

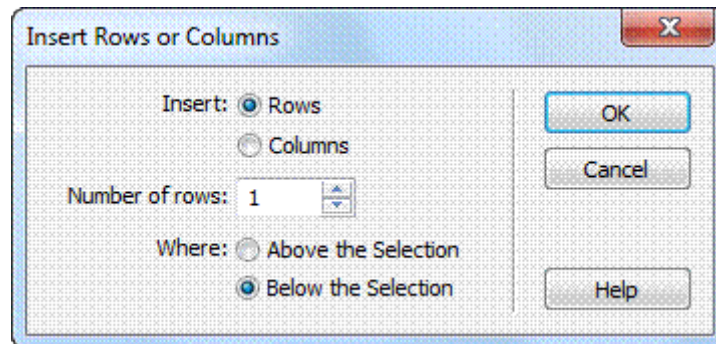
Border: viền của bảng.

Align: chế độ canh lề của bảng.

BgColor: màu nền của bảng hay ô.

Brdr: màu đường viền.

- Thay đổi cấu trúc bảng.



Hình 3.6. Thêm dòng vào bảng

Trong quá trình làm việc, bảng tạo ra có thể chưa phù hợp. Bạn có thể thay đổi cấu trúc của bảng với những nội dung sau:



Thêm dòng: chọn dòng mà bạn muốn chèn thêm một dòng mới phía sau nó. Chọn *Modify > Table > Insert Rows or Columns* .

Trong cửa sổ hiện ra, đánh dấu chọn trước mục *Rows* và *Below the Selection*. Trong mục *Number of rows*, đánh số dòng muốn chèn và bấm OK (hình 6)

Thêm cột: chọn cột mà bạn muốn chèn thêm cột mới phía bên phải nó. Thực hiện lại các thao tác tương tự như thêm dòng nhưng đánh dấu chọn trước chữ *Columns*.

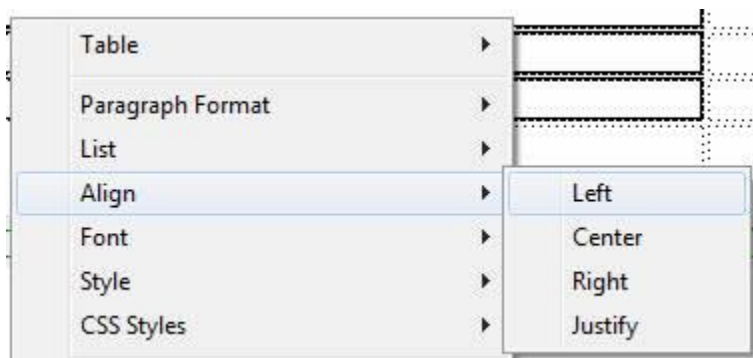
Xóa dòng: chọn dòng muốn xóa, chọn *Modify > Table > Delete row* (hoặc bấm chuột phải vào dòng muốn xóa, chọn *Table > Delete row*).

Xóa cột: chọn cột muốn xóa, vào *Modify > Table > Delete column* (hoặc bấm chuột phải vào cột muốn xóa, chọn *Table > Delete column*).

- Trộn ô: chọn các ô muốn trộn, chọn *Modify > Table > Merge cells* (hoặc bấm vào biểu tượng trộn ô  ở dưới khung thuộc tính)
- Tách ô: chọn vào ô muốn tách, chọn *Modify > Table > Splits cells* (hoặc bấm vào biểu tượng tách ô  ở dưới khung thuộc tính)

- Canh lề văn bản trong ô.

Bạn chọn văn bản cần định dạng trong ô hoặc chọn các ô cần định dạng, rồi bấm chuột phải vào vùng chọn và chọn *Align*. Sau đó chọn một trong bốn kiểu canh lề là *Left* (canh trái), *Right* (canh phải), *Center* (canh giữa) và *Justify* (canh đều) (hình 7).



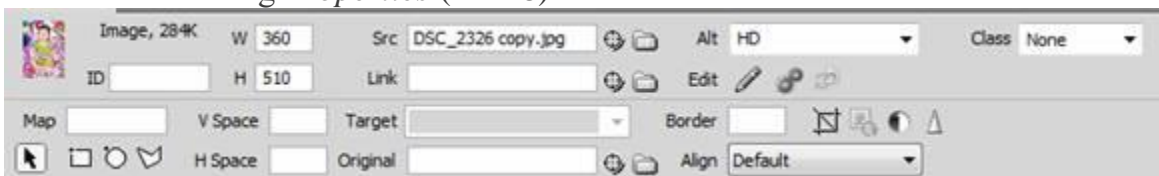
Hình 3.7. Canh lề cho văn bản trong ô

3.7. Chèn hình vào trang web

- Chèn ảnh tĩnh

Trước khi muốn chèn hình ảnh vào trang web, bạn nên chép tất cả các hình ảnh cần chèn vào thư mục *Image* của bộ web. Việc lựa chọn hình ảnh để chèn vào trang web cũng phải chú ý đến số lượng và kích thước của ảnh vì nếu chèn quá nhiều hoặc ảnh quá lớn sẽ làm cho trang web có dung lượng lớn, việc tải về xem cũng chậm hơn. Để thực hiện việc chèn hình ảnh vào trang web, bạn làm như sau: vào *Insert*, chọn *Image* và tìm đến hình ảnh muốn chèn .

Sau khi chèn hình ảnh vào trang web, bạn tiến hành thiết lập các thuộc tính cho hình ảnh vừa mới chèn ở khung *Properties* (hình 8)



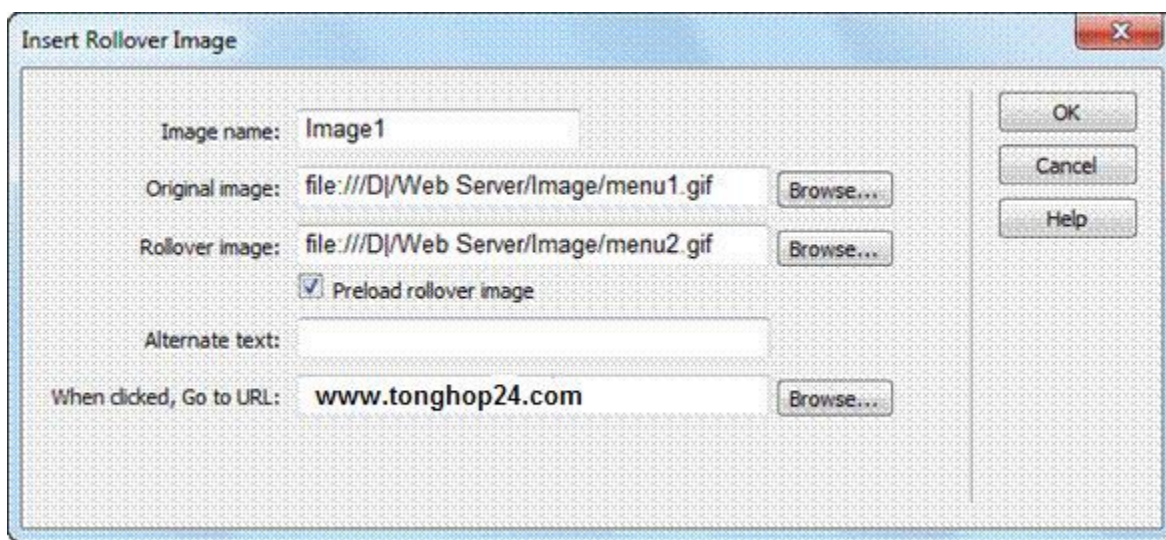
Hình 3.8. Thuộc tính của hình ảnh

- Tạo Rollover Images.

Rollover Images là một hình ảnh sẽ thay đổi khi con trỏ đi ngang qua nó. Khi tạo *Rollover Images* thì hai ảnh phải có kích thước bằng nhau. Nếu hai ảnh có kích thước không bằng nhau thì *Dreamweaver* sẽ tự chỉnh cho chúng bằng nhau.

Trước tiên bạn đặt con trỏ ở vị trí muốn chèn. Sau đó chọn *Insert > Image objects > Rollover Image*.

Hộp thoại *Insert Image Rollover* xuất hiện ta thiết lập các nội dung sau:



Hình 3.9. Tạo hiệu ứng ảnh động khi di chuyển trỏ chuột

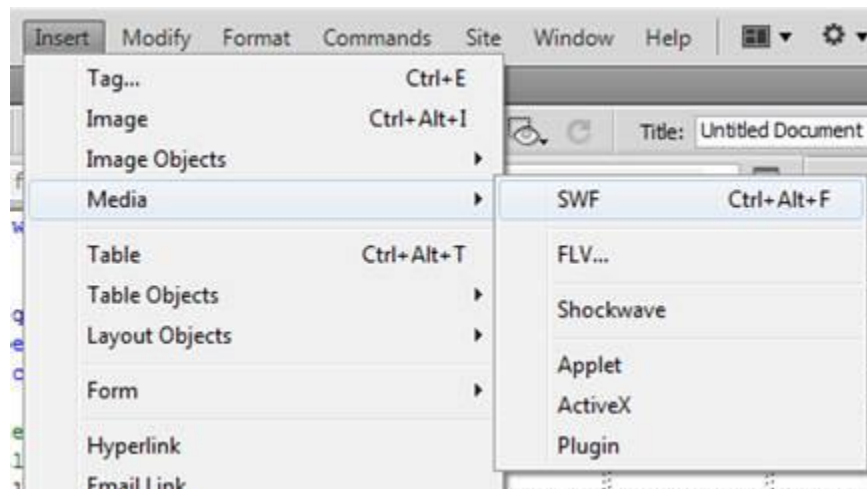
- *Image name*: đặt tên cho *Image*.
- *Original image*: bấm vào nút *Browse* chỉ định hình đầu tiên xuất hiện.
- *Rollover Images*: bấm vào nút *Browse* chỉ định hình sẽ thay đổi khi di chuyển chuột qua nó.

- *When Clicked, Go to URL*: bấm vào nút *Browse* chỉ ra trang web liên kết đến khi bấm vào nó.

Chọn OK để hoàn tất (hình 9).

3.8. Chèn ảnh động flash, video clip vào trang web.

Vào *Insert > media >* chọn *SWF* hoặc *FLV* tùy theo định dạng tập tin cần chèn. Sau đó duyệt đến *File flash* hoặc *File Video* và bấm OK.



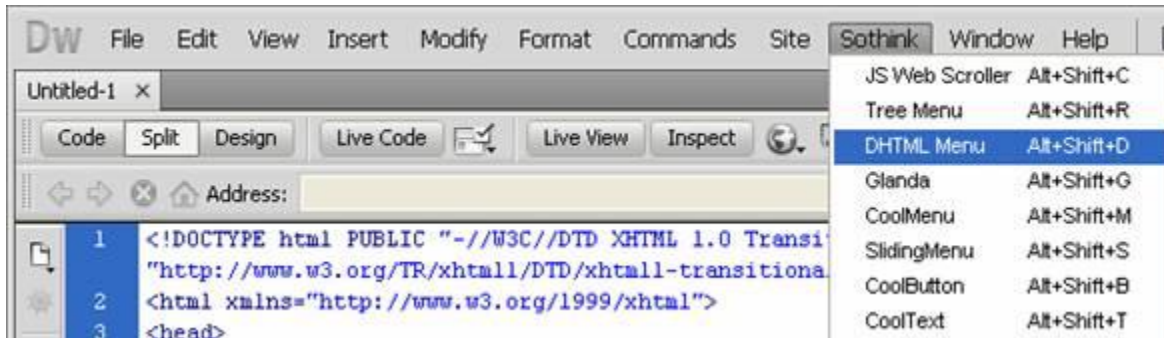
Hình 3.10. Chèn tập tin đa phương tiện

3.9. Tạo menu cho trang web.

Menu là một phần không thể thiếu cho mọi trang web. Nó là nơi liên kết các trang rời rạc của bộ web lại với nhau thành một hệ thống. Việc tạo ra một menu thật đẹp và chuyên nghiệp là ước ao của nhiều người thiết kế web.

Đối với phần mềm thiết kế web *Dreamweaver*, đã có một công cụ tích hợp hỗ trợ tạo menu thật chuyên nghiệp là *SothinkDHTMLMenu*. Bạn có thể dùng phần mềm đó để thiết kế menu cho trang web của mình.

SothinkDHTMLMenu được cung cấp bản dùng thử tại trang chủ <http://www.sothink.com/product/dhtmlmenu>. Sau khi cài đặt thành công phần mềm *SothinkDHTMLMenu*, trên thanh menu của *Adobe Dreamweaver* sẽ có thêm menu *Sothink* và với menu này bạn sẽ tạo ra một menu bóng bẩy cho website của mình (hình 11).



Hình 3.11. Khởi động DHTML Menu

Đầu tiên mở trang web cần tạo menu bằng phần mềm *Adobe Dreamweaver* đặt con trỏ ở vị trí cần chèn menu, chọn menu *Sothink*, trong bảng hiện ra chọn *DHTML Menu* (hoặc bấm tổ hợp phím *Alt+Shift+D*). Lưu ý, bạn phải lưu trang web trước khi tiến hành tạo menu.

Trong cửa sổ mở ra của *SothinkDHTMLMenu*, chọn *Blank Menu* để thiết kế mới, *From Template* nếu muốn chọn một kiểu menu có sẵn. Trong cửa sổ *Menu Tree* (menu đứng ở bên trái), bạn có thể thực hiện các thao tác như cắt, dán,... các chủ đề, chuyên mục cho menu thông qua các nút lệnh trên thanh công cụ.

Toàn bộ quá trình thiết kế trong *SothinkDHTMLMenu* được thực hiện chủ yếu bởi các khung tính năng: *Popup Menu*, *Menu Item*. Ngoài ra, trong quá trình thiết kế, bạn có thể tìm hiểu thêm về các cửa sổ: *Scrolling* và *Global Setting* (hình 12).



Hình 3.12. Các tác vụ thiết kế

- Khung Menu Item:

Cửa sổ này cho phép bạn định dạng cho từng chủ đề, từng mục trong menu (hình 13). Trong cửa sổ này, bạn có thể điều chỉnh các mục như sau:

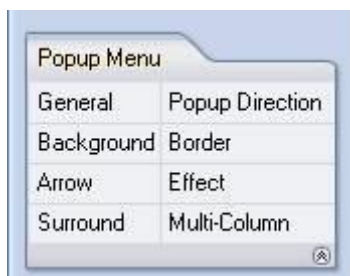


Hình 3.13. Thuộc tính của menu

- + Chọn nhãn *General* để nhập các mục cho menu:
- + Chọn *Text* để nhập tên chủ đề.
- + Chọn mục *HTML* để nhập tên chủ đề được định dạng trong các cặp thẻ HTML.
- + Để chèn ảnh vào *menu*, bạn chọn *Image*, xác định ảnh từ nút lệnh .
- + Căn chỉnh vị trí hiển thị nội dung trên các mục menu với *Align* (*left, right, center*) hoặc *Valign* (*top, middle, bottom*).
- + Đặt liên kết cho các mục menu thông qua mục *Link*.

Ngoài ra trong cửa sổ *Menu Item* bạn còn có thể định dạng font chữ, màu chữ, màu nền, hình nền, khung, kiểu cho menu trong các nhãn tương ứng của cửa sổ *Menu Item* là *Font, Icon, Background, Border, Condition, Style, Advanced*.

- Khung *Popup Menu*

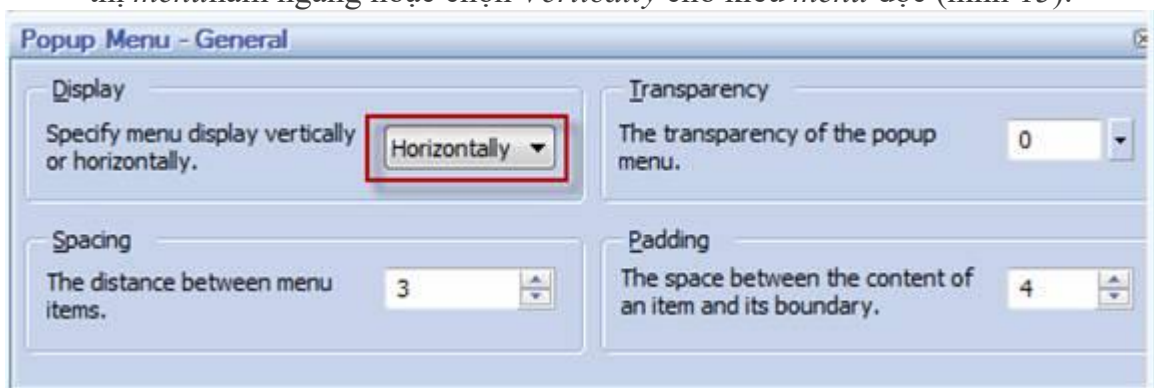


Hình 3.14. Cửa sổ *Popup Menu*

Sau khi tạo xong menu, cửa sổ này sẽ giúp chúng ta thiết kế màu sắc, kiểu hiển thị menu... và điều quan trọng bạn có thể tạo những hiệu ứng đặc biệt cho menu (hình 14).

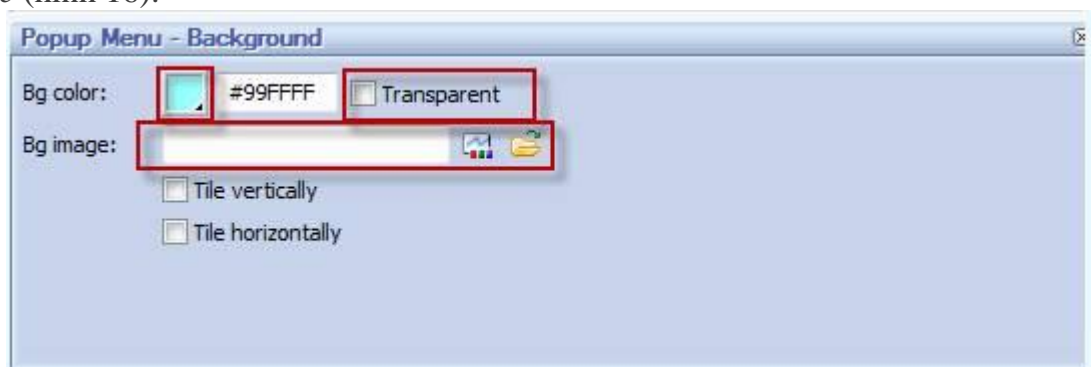
Các bước thực hiện như sau:

Trong nhãn *General* của *Popup Menu*, chọn *Horizontally* trong khung *Display* để hiển thị *menu* nằm ngang hoặc chọn *Vertically* cho kiểu *menu* dọc (hình 15).



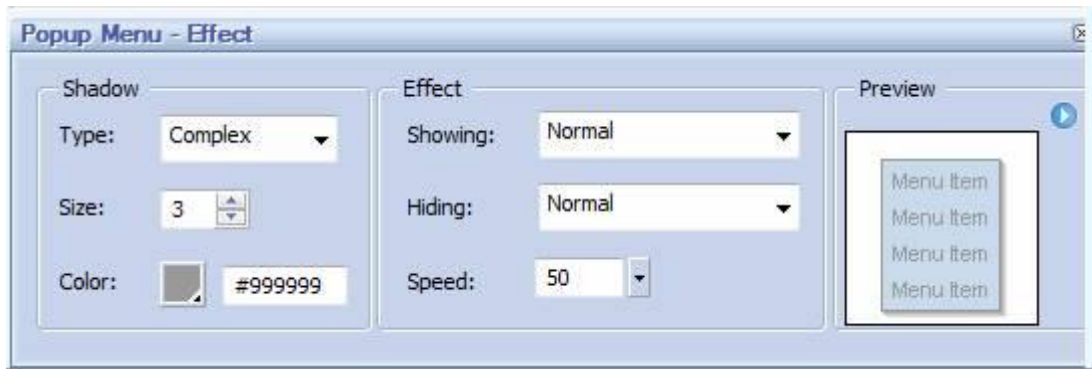
Hình 3.15. Chọn kiểu hiển thị

Tiếp theo chọn nhãn *Background*, trong mục *Bg color*, bấm nút và chọn màu nền cho menu từ bảng màu, còn nếu muốn menu có nền trong suốt thì bấm đánh dấu chọn trước mục *Transparent*. Cũng có thể chọn ảnh nền của menu bằng cách nhấn nút trong mục *Bg image* (hình 16).



Hình 3.16. Chọn màu nền hay hình nền

Để đặt màu nền cho menu, bạn chọn nhãn *Border*, bấm chọn nút mũi tên trên nút để đặt màu cho mỗi cạnh của Menu. Trong nhãn *Effects* từ cửa sổ *Popup Menu*, bạn sẽ đặt hiệu ứng cho menu (hình 16). Các hiệu ứng trong mục như sau:



Hình 3.16. Chọn hiệu ứng

+ Đổ bóng cho menu: Bạn có thể đổ bóng cho menu bằng cách chọn *Simple* hoặc *Complex* từ menu con của *Drop shadow*, chọn đường nét, độ dày cho bóng và màu sắc cho bóng.

+ Hiệu ứng hiển thị menu (*Effects for showing*): bấm chọn menu và chọn một hiệu ứng đặc biệt từ danh sách.

+ Hiệu ứng đặc biệt để ẩn menu (*Effect for Hiding*): bấm chọn mũi tên, chọn hiệu ứng để ẩn menu.

+ *Effect Speed*: Nhập giá trị vào ô text đặt tốc độ hiển thị cho hiệu ứng menu.

- *Khung Scrolling*

Nếu trong menu có quá nhiều menu con, bạn có thể làm cho các menu con cuộn lại bằng cách tùy chỉnh trong cửa sổ *Scrolling* khi trên màn hình không đủ không gian để hiển thị tất cả chúng (hình 17).



Hình 3.17. Tùy chỉnh kiểu dáng thanh cuộn

+ Tùy chỉnh kiểu dáng cho thanh cuộn theo chiều ngang (*Horizontal Popup Menus*) hoặc theo chiều dọc (*Vertical Popup Menus*).

+ Đánh dấu chọn *Enable Scrolling*.

Sau khi đã hoàn tất các bước thiết lập cho menu, bạn chọn lệnh *Save and close* để hoàn tất công việc. *SothinkDHTMLMenu* sẽ tự động đưa menu mà bạn vừa tạo vào trang web của mình mà không cần phải làm việc với những đoạn code nữa. Tuy nhiên, *SothinkDHTMLMenu* lại có một nhược điểm là bị lỗi font tiếng Việt Unicode khi chạy menu trên trình duyệt và các tập tin do chương trình tạo ra là rất nhiều và nằm cùng thư mục với tập tin web vừa tạo nên có phần nào đó làm cho phần thiết kế không có tính khoa học. Nhưng hai lỗi trên vẫn có thể được khắc phục dễ dàng bằng cách làm việc với những đoạn code.

Trước tiên, bạn có thể khắc phục lỗi font bằng cách vào đoạn code do chương trình tạo ra trong *Dreamweaver* tìm đến các từ bị lỗi font và sửa lại cho đúng. Tiếp theo, để khắc phục các tập tin do *SothinkDHTMLMenu* tạo ra nằm “mất trật tự” trong thư mục chứa trang web bạn tạo ra một thư mục mới trong thư mục chứa trang web (thư mục Menu chẳng hạn), tiếp theo sao chép tất cả các tập tin mà *SothinkDHTMLMenu* tạo ra vào thư mục mới này. Sau đó vào cửa sổ soạn thảo code và thêm phần “*menu*” vào trước các dẫn cũ là xong (các đường dẫn cũ là tên của các tập tin do *SothinkDHTMLMenu* tạo ra).

3.10. Tạo liên kết cho trang web.

Một trang web được thiết kế ra không chỉ chứa thông tin của nội bộ trang web mà nó còn phải được liên kết với nhiều trang web khác để nội dung thêm phong phú. Có thể tạo liên kết từ một hình ảnh đến trang web hoặc từ một dòng chữ tùy theo sở thích của người thiết kế. Các bước thực hiện như sau:

- Chèn hình hoặc nhập dòng chữ cần tạo liên kết.
- Bấm chọn hình hoặc chọn dòng chữ cần tạo liên kết.

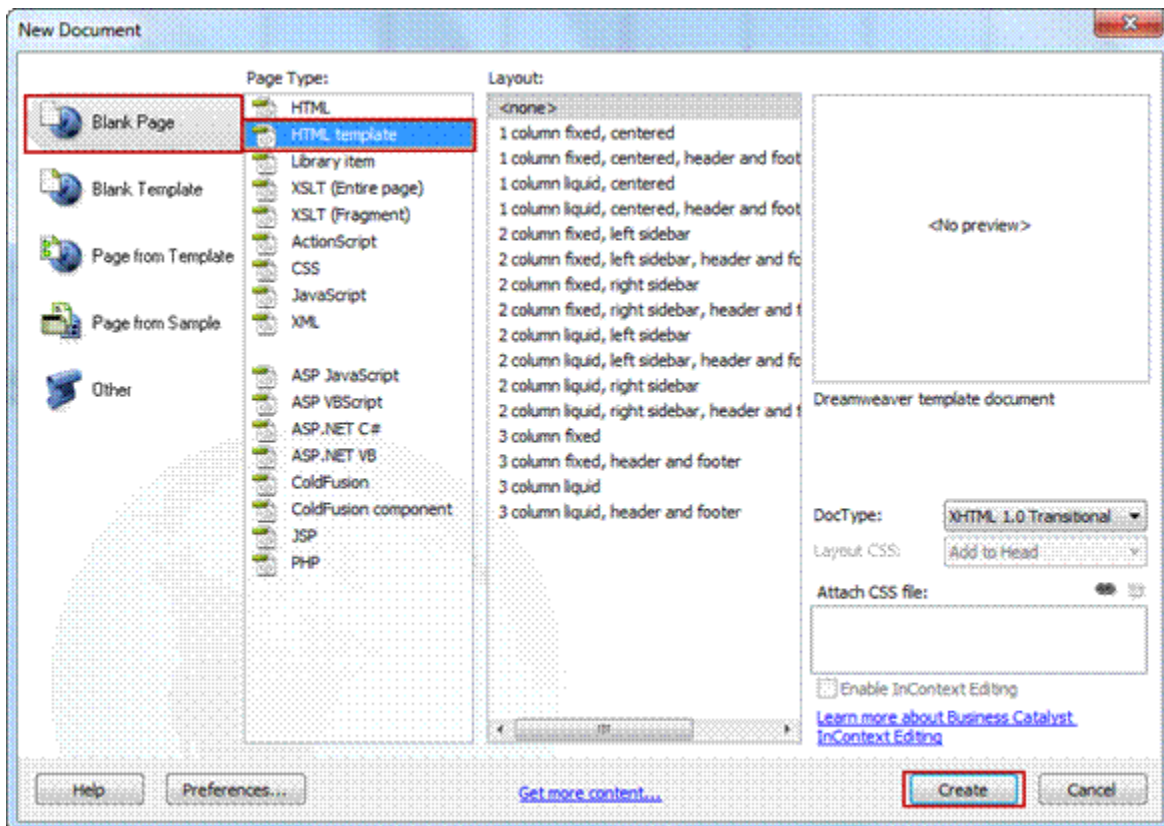
Nhập vào ô *Link* ở khung *Properties* địa chỉ trang web cần liên kết tới.



Hình 3.18. Thuộc tính địa chỉ liên kết

Lưu ý, địa chỉ trang web liên kết nhập vào cần ghi đầy đủ tên của trang web. Ví dụ, liên kết đến trang tìm kiếm *Google* thì cần nhập đúng địa chỉ *http://www.google.com.vn*.

3.11. Cách tạo Template



Hình 3.19. Tạo template cho Website

Trong việc thiết kế web, bạn sẽ thiết kế một trang chủ và hệ thống gồm rất nhiều trang con. Để trang chủ và các trang con đều có cấu trúc nhất quán với nhau, bạn phải tạo một trang gọi là *template*. Khi đó muốn thay đổi nội dung cho toàn bộ Website, bạn chỉ cần đơn giản thay đổi nội dung ở trang *template* và cập nhật lại cho toàn bộ Website với những thao tác thật đơn giản.

Tập tin *template* tạo ra sẽ có phần đuôi là *.dwt* và nó sẽ mặc định đặt trong *Templates folder*. Để tạo một *template*, bạn tiến hành như sau:

Khởi động *Dreamweaver*. Vào menu *File*, chọn *New*. Ở *Page Type* > chọn *HTML template* sau đó bấm vào nút *Create* để mở một trang mới lên (hình 19).

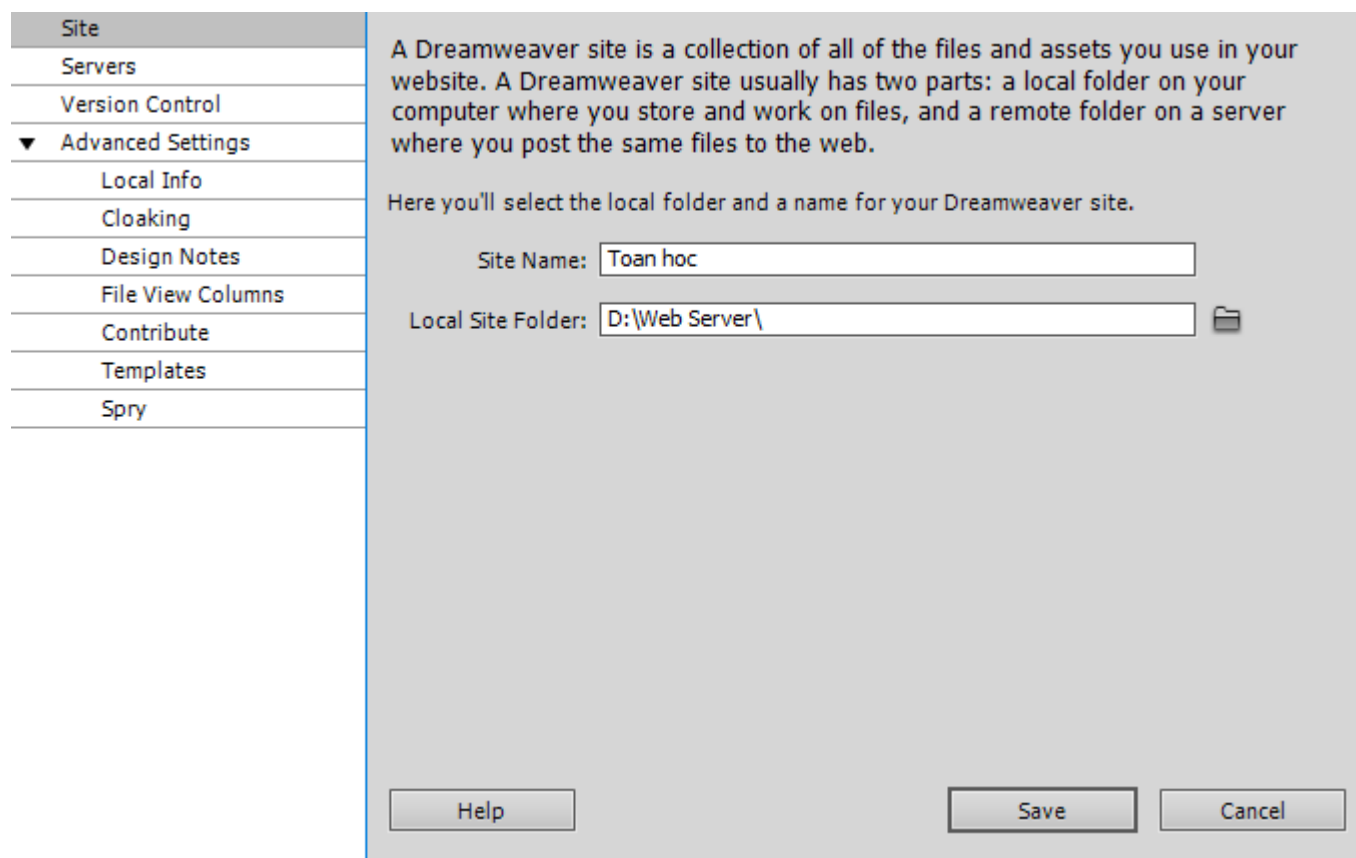
Tiếp theo, bạn khai báo các thông tin về website mới bằng cách vào *Site*, chọn *Manage Sites*, sau đó chọn *New* trong cửa sổ tiếp theo để khai báo.

Tối thiểu cần khai báo một số thông tin sau (hình 20)

- *Site name*: tên Website.

- *Local root folder*: thư mục gốc trên máy.

- *Default images folder*: thư mục mặc định dành cho hình ảnh.



Hình 3.20. Chọn thư mục lưu trữ Website

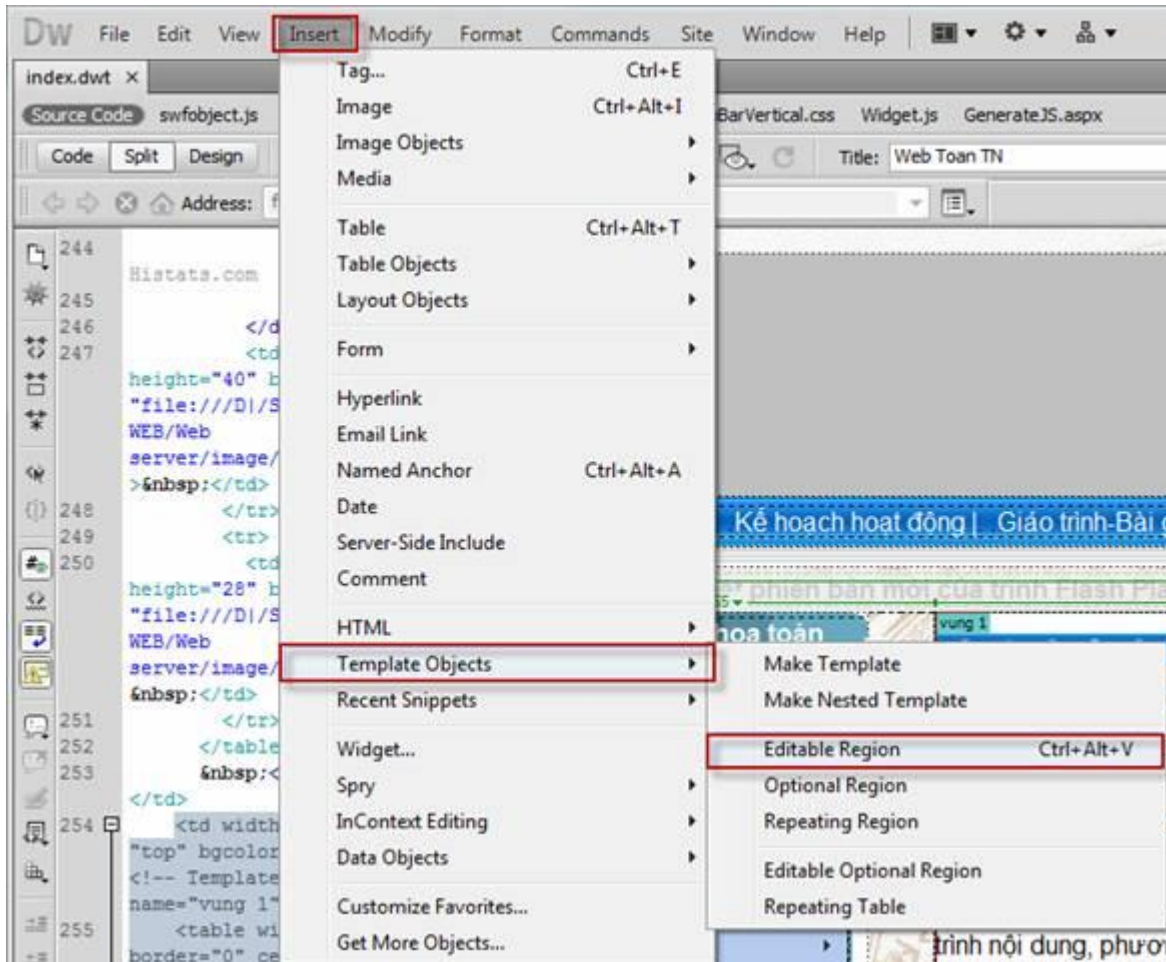
Sau khi khai báo xong chọn *Save* và bắt đầu tạo *template*. Khi xác định thành phần nào trong web là không thay đổi, bạn sẽ tạo trang web với những thành phần đó, sau đó chọn *File > Save as Template*.

Ngoài ra, bạn cũng có thể tạo ra một *template* từ một trang web đã thiết kế trước đó bằng cách mở trang web đó lên bằng *Dreamweaver*, tiến hành khai báo *Site* mới cho trang web đó và chọn *File > Save as template*. Sau đó đặt tên cho *template* vừa tạo.

Tiếp theo, bạn xác định các vùng có thể thay đổi được như sau:

- Chọn vùng có thể thay đổi được bằng cách bấm chuột vào bảng của vùng đó (vùng được chọn sẽ có viền màu đen đậm bao quanh).

- Chọn *Insert > Template > Editable Region* (hình 21).



Hình 3.21. Chọn vùng thay đổi trên template

- Đặt tên cho vùng vừa chọn và bấm OK trong cửa sổ tiếp theo.

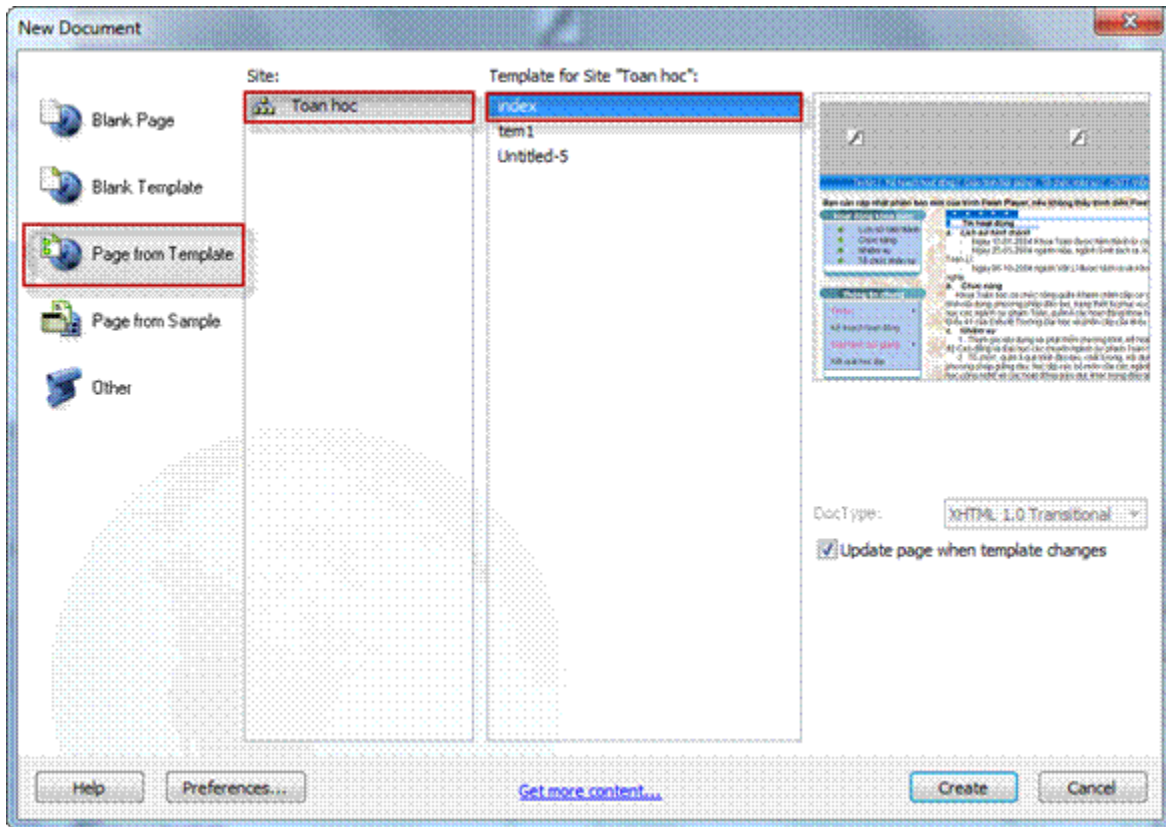
Cuối cùng, chọn *File > Save* để lưu lại trang *template* vừa tạo.

3.12. Tạo trang web mới từ template

Bây giờ, bạn có thể tạo ra trang web đầu tiên từ *template* vừa tạo.

Chọn *File > New*. Trong cửa sổ hiện ra, chọn *Page from Template* và chọn *File Template* vừa tạo sau đó bấm nút *Create* (hình 22).

Lưu ý, các tập tin tạo ra từ template chỉ có thể soạn thảo ở vùng *Editable Region* mà bạn đã chọn lúc tạo *template*. Còn các vùng khác thì sẽ không được phép thay đổi.



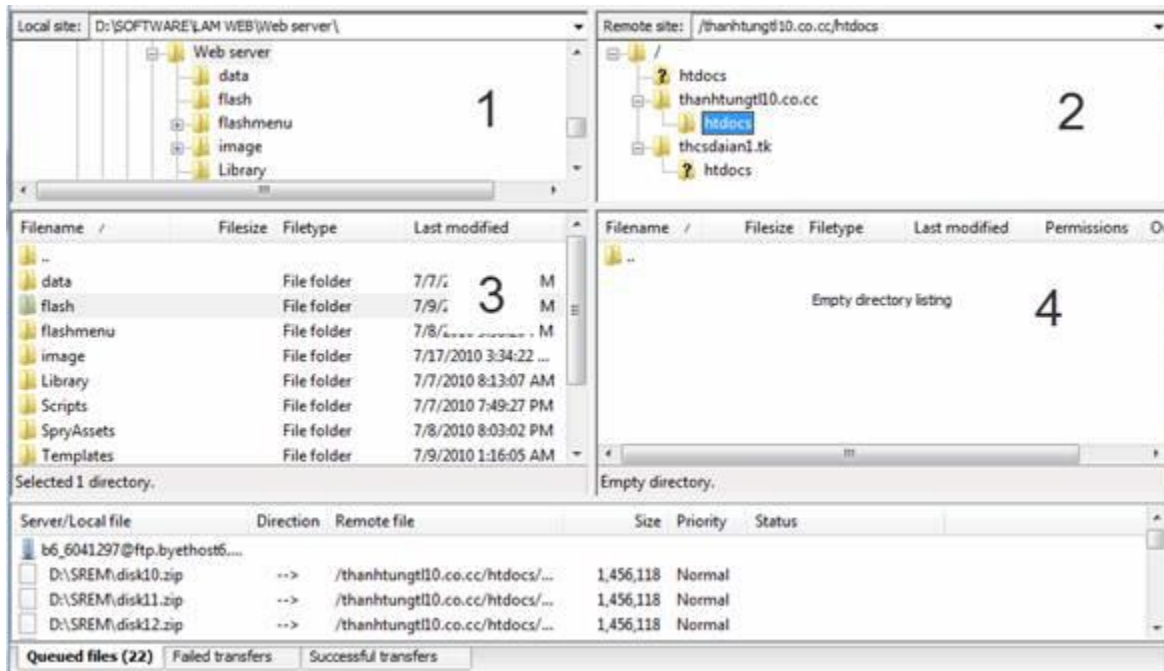
Hình 3.22. Tạo trang web mới từ template

Ngoài ra, khi đã tạo xong template với đầy đủ nội dung và liên kết thì bạn lưu thay đổi trên template này. *Dreamweaver* sẽ xuất hiện hộp thoại nhắc nhở có muốn cập nhật thay đổi trên các trang dùng template này hay không, bấm *Update* để cập nhật.

3.13. Đưa website lên hosting

Sau khi đã hoàn tất việc thiết kế một trang web, bạn tiến hành đưa trang web lên mạng Internet để thử nghiệm và chia sẻ thông tin. Đối với trang web thiết kế bằng *Dreamweaver* trong loạt bài này, để đưa website lên mạng bạn làm như sau.

Dùng phần mềm *FileZilla FTP Client* đưa trang web lên hosting. Khởi động *FileZilla FTP Client* nhập các thông tin cần thiết để kết nối với *hosting*. Trong cửa sổ chương trình có 4 khung chính như hình 23



Hình 3.23. Cửa sổ chương trình FileZilla

- Trong khung số 1, tìm đến đường dẫn chứa bộ *web* (trong hình là thư mục *Web server*).
- Trong khung số 2, tìm và duyệt đến thư mục *htdocs* (đối với *byethost*) hoặc thư mục *public_html* đối với *000webhost*,....
- Trong khung thứ 3, lần lượt bấm chuột phải vào từng tập tin, thư mục (hoặc bấm *Ctrl + A* để chọn tất cả) và chọn *Upload*.

Đợi cho chương trình tải toàn bộ trang web lên hosting. Thời gian nhanh hay chậm tùy theo dung lượng tập tin, số lượng tập tin cần tải lên nhiều hay ít. Lưu ý, đối với *Byethost* thì có nhiều thư mục *htdocs*, bạn phải duyệt đến thư mục *htdocs* nằm trong thư mục tên miền và tập tin *index.html* phải nằm trong thư mục *htdocs*.

Sau khi chương trình tải xong, bạn mở trình duyệt lên và nhập vào địa chỉ web đã đăng ký, sẽ thấy được thành quả lao động của mình.

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 3

1. Sử dụng Dreamwaver Thiết kế trang web sau:



2. Thiết kế Website giới thiệu cửa hàng sách với cấu trúc trang chủ như sau:

<i>Phần banner: hình ảnh, giới thiệu cửa hàng sách, hình ảnh quảng cáo (nếu có)</i>	
	<i>Ngày tháng năm</i>
<i>Tạo form đăng nhập</i>	<i>Các liên kết nhanh: quay lại trang chủ/Sách bán chạy/Sách giảm giá</i>
<i>Tạo form tìm kiếm thông tin</i>	<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <p>Phần trang chính, hiển thị các nội dung khi nhấn chuột vào các liên kết</p> </div> <div style="width: 35%;"> <p><i>Phần phụ giới thiệu:</i></p> <ul style="list-style-type: none"> - các loại sách bán chạy.(tối đa 5 đầu sách) - các loại sách giảm giá.(tối đa 5 đầu sách) </div> </div>
<p><i>Menu chính:</i> (giới thiệu các chủ đề của sách)</p> <p>Kinh tế</p> <p>Phát luật</p> <p>Tin học</p> <p>...</p>	
<p><i>Thông tin về bản quyền Website.</i></p> <p><i>Thông tin liên hệ</i></p>	

Yêu cầu:

1) Sử dụng công cụ thiết kế phần mềm Dreamwaver 8.0.

2) Sử dụng tư liệu và ảnh qua CD

3) Tạo Website lưu trữ tại ổ đĩa: D:/Tên sinh viên

4) Tạo trang chủ.

- theo cấu trúc yêu cầu, hình thức dễ nhìn, có liên kết đầy đủ đến các trang liên kết ở câu 5.

5) Tạo các trang liên kết (*giống phần banner, menu chính và phần phụ, chỉ thay đổi nội dung ở phần trang chính*):

- Trang giới thiệu sách Kinh tế khi nhấn chuột vào chủ đề Kinh tế ở menu chính

- Trang giới thiệu sách Pháp luật khi nhấn chuột vào chủ đề Pháp luật ở menu chính.

- Trang giới thiệu sách Tin học khi nhấn chuột vào chủ đề Tin học ở menu chính .

- Trang giới thiệu chi tiết thông tin về sách bán chạy khi nhấn chuột chọn 1 sách bán chạy tại phần phụ. (*thực hiện 1 trang về 1 loại sách*) .

- Trang giới thiệu chi tiết thông tin về sách giảm giá khi nhấn chuột chọn 1 sách giảm giá tại phần phụ. (*thực hiện 1 trang về 1 loại sách*)

- Trang giới thiệu sách bán chạy trong cửa hàng (tối đa 10 sách) khi nhấn chuột vào liên kết nhanh dưới phần banner.

- Trang giới thiệu sách giảm giá trong cửa hàng (tối đa 10 sách) khi nhấn chuột vào liên kết nhanh dưới phần banner.

CHƯƠNG 4. CASCADING STYLE SHEETS

4.1. CSS là gì?

Cascading Style Sheets (CSS) là ngôn ngữ dùng để trình diễn, định dạng cho các file HTML (màu sắc, kích cỡ, font chữ...). Một số điểm về lịch sử phát triển:

- Năm 1994, Cascading HTML Style Sheets được đưa ra bởi Håkon Wium Lie (CTO (Chief Technology Officer) của công ty phần mềm Opera).
- Tháng 12 năm 1996, CSS level 1 Recommendation ra đời.
- 1998, CSS 2.1 ra đời
- Hiện nay đã sử dụng CSS 3.

4.2. Lợi ích việc sử dụng CSS

Có thể trình diễn cho nhiều trang
Độc lập với file HTML, dễ chỉnh sửa.
Hỗ trợ trên nhiều thiết bị
Tăng tốc độ cho Web

4.3. Sử dụng CSS

- ✚ Sử dụng thuộc tính style trong thẻ HTML (inline CSS)

```
<p style="color:#0FC">hello</p>
```

- ✚ Sử dụng thẻ <style></style>(nằm bên trong thẻ head)

```
<style>

p{

    color:#0FC;

}

</style>
```

Sử dụng file .css

Tạo file style.css như sau:

```
p{  
  
    color:#0FC;  
  
}
```

Trong file html sử dụng thẻ link (trong thẻ head) chỉ đến đường dẫn file css:

```
<link type="text/css" href="style.css" />
```


Thứ tự ưu tiên: Từ trên xuống dưới

4.4. Cú pháp CSS

```
Selector{  
  
    Property:Value;  
  
    ...  
  
}
```

4.4.1 Selector

Thành phần muốn định dạng. Bao gồm:

 Các thẻ HTML

```
body{  
  
    ...  
  
}
```

✚ Nhóm thẻ HTML

```
th,td {  
  
    padding: 0 10px 0 0;  
  
    text-align: left;  
  
}
```

✚ Định dạng theo class:

```
.TenClass{  
  
...  
  
}
```

hoặc

```
TheHTML.TenClass{  
  
...  
  
}
```

hoặc (đối với thành phần có 2 class)

```
TheHTML.TenClass1.TenClass2{  
  
...  
  
}/* voi IE6 selector này tương đương với TheHTML.TenClass2*/
```

Ví dụ:

```
<div class="planet jupiter"> <!--2 class-->
```

```
<h2>Jupiter</h2>
```

```
</div>
```

```
    CSS div.planet { margin: 10px 0;
padding: 20px 20px 20px 200px; border: 1px solid #FFF; background-
position: 20px 20px; background-repeat: no-repeat;
}
```

```
.jupiter {
background-image: url(jupiter.jpg);
}
```

hoặc

```
.planet.jupiter {
background-image: url(jupiter.jpg);
}
```

Lưu ý: IE6 không hỗ trợ 2 class

 Định dạng theo ID

```
#TenID{
```

```
...
```

```
}
```

 Định dạng phổ dụng (universal selector): áp dụng cho tất cả

```
* {
```

```
font-family: Arial, Helvetica, sans-serif;

}
```

+ Định dạng con cháu (Descendant selector)

```
div.planet h2 { font-size: 18px; margin-top: 0;

}
```

```
div.planet > h2 { font-size: 18px; margin-top: 0;

}
```

~~Định dạng cho thẻ h2 nằm trong div có class planet~~

```
div.planet table td { padding: 0 10px 0 0; text-align: left;

}
```

Định dạng cho thẻ td nằm trong thẻ table thuộc thẻ div có class planet

+ Định dạng lớp giả lập(Pseudo Class)

Dynamic Pseudo Class

:link: định dạng khi liên kết chưa được click

:visited: định dạng khi liên kết được click

:hover: định dạng khi lướt qua nội dung thẻ (hover)

(IE6 chỉ hỗ trợ cho thẻ siêu liên kết)

:focus: link được chọn từ bàn phím

:active: đang được người dùng click

4.5. Thứ tự ưu tiên và tính kế thừa

4.5.1 Thứ tự ưu tiên

Các style sheet có thể định dạng từ nhiều nguồn khác nhau:

1. Mặc định của trình duyệt
2. Người dùng
3. Website

Thứ tự ưu tiên (3, 2, 1).

Các selector cũng có thứ tự ưu tiên (ưu tiên hơn cả là ID và selector phổ dụng *). Thứ tự ưu tiên được tính theo công thức sau:

1. 1 nếu style được dùng trong thuộc tính style của thẻ (X)HTML, và ngược lại là 0 (biến a)
2. Tổng số thuộc tính ID trong selector (biến b)
3. Tổng số thuộc tính, pseudo-class, tên class trong selector (biến c)
4. Số tên thẻ trong selector (biến d)
5. Không tính các thành phần pseudo (pseudo-element)

Selector	Loại selector	Thứ tự (specificity) (a, b, c, d)
*	Phổ dụng	0, 0, 0, 0
li	Tên thẻ	0, 0, 0, 1
ul li	Tên thẻ	0, 0, 0, 2
div h1 + p	Tên thẻ	0, 0, 0, 3

input[type='text']	Tên thẻ + Thuộc tính	0, 0, 1, 1
.someclass	Tên lớp	0, 0, 1, 0
div.someclass	Tên thẻ + Tên lớp	0, 0, 1, 1
div.someclass.someother	Tên Thẻ + Tên lớp + Tên lớp	0, 0, 2, 1
#someid	Tên ID	0, 1, 0, 0
div#someid	Tên thẻ + Tên ID	0, 1, 0, 1
style(thuộc tính)	Thuộc tính style	1, 0, 0, 0

Selector nào có số bên trái nhất cao nhất thì có độ ưu tiên cao nhất (nếu cùng thì so tiếp theo thứ tự từ trái sang phải). Nếu hai selector có cùng thứ tự ưu tiên thì áp dụng theo thứ tự xuất hiện trong style sheet (cái sau nhất).

Meo: So sánh như một số có 4 chữ số

4.5.1.1.1 Ví dụ:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html lang="en">

<head>

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

<title>Vi du Do Uu Tien</title>

<style type="text/css">

body {

width: 650px; margin: 0 auto; background: #000; color: #FFF;

font: 12px sans-serif;
```

```
}

h1 {

font-size: 24px;

}

h2 {

font-size: 18px;

margin-top: 0;

}

a {

4color: #FFF;

} a:focus, a:hover {

text-decoration: none;

}

table {

margin-bottom: 10px;

border-spacing: 0;

}

caption {

margin-bottom: 10px;
```

```
font-size: 14px; font-weight: bold; text-align: left;
}

th, td {
padding: 0 10px 0 0;

text-align: left;

}

tr.even {

background: none;

}

tr {

background: #666;

}

div.planet {

background: none;

}

div.planet {

margin: 10px 0;

padding: 20px 20px 20px 200px; border: 1px solid #FFF; background-
image: none; background-position: 20px 20px; background-repeat: no-
repeat;
```

```
}

#jupiter {

background-image: url(jupiter.jpg);

}

</style>

</head>

<body>

<h1>The gas giants</h1>

<div class="planet" id="jupiter">

<h2>Jupiter</h2>

<p>Jupiter is the fifth planet from the Sun and the largest planet
within the

Solar System. The Romans named the planet after the god Jupiter. It
is a gas

giant with a mass two and a half times the mass of all the other
planets in

our Solar System combined.</p>

<table>

<caption>Jupiter Facts</caption>

<tbody>

<tr>
```

```
<th>Distance from the Sun</th>

<td>78,412,020 km</td>

</tr>

<tr class="even">

<th>Equatorial Radius</th>

<td>71,492 km</td>

</tr>

<tr>

<th>Volume</th>

<td>1,425,500,000,000,000 km<sup>3</sup></td>

</tr>

<tr class="even">

<th>Mass</th>

<td>1,898,700,000,000,000,000,000,000 kg</td>

</tr>

</tbody>

</table>

<a
```

```
href="http://solarsystem.jpl.nasa.gov/planets/profile.cfm?Object=Jupiter"
>

More Jupiter facts</a>

</div>

</body>

</html>
```

Nhận xét:

Trường hợp sau

```
tr.even {

background: none;

}

tr {

background: #666;

}
```

sẽ áp dụng theo tr.even (0, 0, 1, 1) thay vì tr (0, 0, 1, 0)

```
div.planet {

background: none;

}
```

```

div.planet {

margin: 10px 0;

padding: 20px 20px 20px 200px; border: 1px solid #FFF; background-
image: none; background-position: 20px 20px; background-repeat: no-
repeat;

}

#jupiter {

background-image: url(jupiter.jpg);

}

```

#jupiter có độ ưu tiên cao hơn (có hình nền)

Nếu đổi vị trí 2 div.planet (cùng thứ tự) thì kết quả sẽ khác (lấy theo cái sau nhất).

4.5.1.1.2 Bài tập cơ bản

1. Làm sao định dạng cho thẻ dựa trên class?
2. Có thể sử dụng bao nhiêu class cho một thẻ?
3. Làm sao định dạng thẻ dựa trên ID?
4. Định dạng đổi font chữ khi hover trên liên kết
5. Xác định độ ưu tiên của các selector sau

```

ul#hmenuul.menu ul#hmenuli li:hover ul#hmenuul.menuul.menu
ul#hmenuli#menu-204ul.menuul.menu
ul#hmenuli#menu-848ul.menuul.menuul.menuul.menu

```

```
ul#hmenu li#menu-990ul.menuul.menu
```

```
ul#hmenu > li.menu.eas + li.menu.easul.menuul.menuul.menu
```

6. Xác định xem liên kết theo style sau đây có màu gì?

```
a.context:link {
```

```
color: blue;
```

```
}
```

```
a.context:visited {
```

```
color: purple;
```

```
} a.context:focus, a.context:hover { color: green;
```

```
}
```

```
a.context:active {
```

```
color: red;
```

```
}
```

7. Xác định xem liên kết theo style sau đây có màu gì?

```
a.context:visited {
```

```
color: purple;
```

```
} a.context:focus, a.context:hover { color: green;
```

```
}
```

```
a.context:active {
```

```
color: red;

}

a.context:link {

color: blue;

}
```

8. Xác định xem liên kết theo style sau đây có màu gì?

```
a.context:link {

color: blue;

}

a.context:visited {

color: purple !important;

} a.context:focus, a.context:hover { color: green;

}

a.context:active {

color: red;

}
```

4.6 Các thuộc tính

4.6.1 Font chữ

font-family: loại font (có chân, không chân,...)

font-style: kiểu chữ nghiêng.

font-variant:viết hoa/thường/...

font-weight:độ đậm của chữ

font-size: cỡ chữ

font: định dạng tất cả các thuộc tính trên

4.6.1.1 font-family

```
font-family: [[ <family-name> | <generic-family> ] [, <family-name>|  
<generic-family>]* ]
```

Ví dụ:

```
<style type="text/css">  
  
body {  
  
font-family: arial, helvetica, sans-serif;  
  
}  
  
h1 {  
  
font-family: "Times New Roman", Georgia, Serif;  
  
}  
  
</style>
```

Lưu ý: font-family phụ thuộc vào sự định nghĩa của hệ điều hành, các font phải được cài trên máy tính thì mới hiển thị được.

Kiểu font(Generic font)	Tên font tương ứng
serif	Times, Times New Roman
sans-serif	Helvetica, Arial
cursive	Zapf-Chancery
fantasy	Western
Monospace	Courier, Courier New

4.6.1.2 font-style

font-style: normal | italic | oblique

4.6.1.3 font-weight

font-weight: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900

Trong thực tế một số font không hỗ trợ(thường chỉ hỗ trợ normal(400) và bold(700))

4.6.1.4 font-variant

font-variant: normal | small-caps

4.6.1.4.1 font-size

font-size: <absolute-size> | <relative-size> | <length> | <percentage>

Absolute-size có thể có các giá trị theo bảng sau

Absolute keyword	XX-SMALL	X-SMALL	SMALL	MEDIUM	LARGE	X-LARGE	XX-LARGE
-------------------------	-----------------	----------------	--------------	---------------	--------------	----------------	-----------------

HTML	n/a	<h6>	<h5>	<h4>	<h3>	<h2>	<h1>
-------------	-----	------	------	------	------	------	------

Relative-size có hai giá trị: smaller và larger (co giãn với tỉ lệ 1.2)

Thang đo độ dài pixel và phần trăm %:

Độ đo chính xác (độ đo tuyệt đối)

Đơn vị đo	Giá trị
in	Inches
cm	Centimeters
mm	Millimeters
pt	Point=1/72 inch
pc	Picas=12point

Độ đo tương đối

Đơn vị đo	Giá trị
em	Độ dài tương ứng với font-size gần nhất
ex	Độ cao của ký tự "x" thường của font liên quan
px	Phụ thuộc vào thiết bị trình diễn (màn hình máy tính)
%	Phụ thuộc vào thuộc tính mà nó đang đc áp dụng

Ví dụ1:

```
body {
```

```
font: 12px sans-serif;
```

```
}

h1 {

font-family: 'Times New Roman', Times, serif;

font-size: 46px;

}

</style>

</head>

<body>

<h1>Gas Giants</h1>

<p>Information about the 4 gas planets in our solar system.</p>

</body>

</html>
```

Ví dụ 2:

```
body {

font: 12px sans-serif;

}

h1 {

font-family: 'Times New Roman', Times, serif;

font-size: 2em;
```

```
}
```

Font-size mặc định của các thẻ là 12px(**font: 12px sans-serif;**), font-size của <h1> là 2em tức là gấp đôi font-size bình thường

Ví dụ về độ đo với số âm:

```
body {  
  
font: 12px sans-serif;  
  
}  
  
h2 {  
  
margin-left: -20px;  
  
}  
  
div {  
  
padding-left: 20px;  
  
}
```

Nhận xét: Các thành phần bên trong thẻ div cách lề trái 20px, tuy nhiên h1 và h2 có margin-left:-20px vì vậy nó sẽ lệch về trái so với các thành phần khác là 20px (tương ứng với margin-left thực sự là 0px)

Ví dụ về %:

```
body {  
  
font-family: arial, helvetica, sans-serif;
```

```

}

h1 {

font-size: 1.5em;

font-weight: normal;

}

p {

font-size: 150%;

}

```

Nhận xét: kích thước <h1> và <p> bằng nhau vì điều gấp 1.5 lần font-size mặc định

4.6.1.5 font

```

font: [ <'font-style'> || <'font-variant'> || <'font-weight'> ]?
<'font-size'> [ / <'line-height'> ]? <'font-family'> ] caption |
icon | menu | message-box | small-caption | status-bar

```

caption, icon, menu, message-box, small-caption, và status-bar phụ thuộc vào font hệ thống

Ví dụ 1:

```

body {

font-weight: bold;

```

```
}  
  
p {  
  
font: 100% arial, helvetica, sans-serif;  
  
}
```

Ví dụ 2:

```
body {  
  
font: 62.5% arial, helvetica, sans-serif;  
  
}  
  
h1 {  
  
font: small-caps 1.6em "Times New Roman", Georgia, Serif;  
  
}  
  
.recipe .intro {  
  
font: italic 1em arial, helvetica, sans-serif;  
  
}
```

4.6.1.6 Bài tập font chữ

1. Tại sao giá trị 100-900 và bolder, lighter lại không được dùng trong thiết kế Website trong thực tế.

2. Có thể rút gọn css sau như thế nào?

```
p {  
  
font-family: Arial, sans-serif;  
  
font-weight: bold; font-size: 24px; color: crimson;  
  
}  
  
p.copy {  
  
font-style: italic; font-weight: bold; line-height: 2em;  
  
}  
  
p#footer {  
  
font-size: 12px;  
  
line-height: 2em;  
  
font-family: Helvetica, Arial, sans-serif;  
  
}
```

3. Style sau có gì sai?

```
p {  
  
font-size: 24;  
  
}
```

4. Việc mô tả **font-size: 75%** sẽ làm cho chữ nhỏ hơn hay lớn hơn?

4.6.2 Định dạng đoạn văn bản

line-height: khoảng cách các dòng
letter-spacing: khoảng cách giữa các ký tự
word-spacing: khoảng cách giữa các từ
text-indent: khoảng cách thụt vào(indent) của văn bản
text-align: canh lề theo chiều ngang
text-decoration: gạch dưới, gạch trên, giữa,...
text-transform: in hoa, chữ thường, ...
white-space

4.6.2.1 line-height

line-height: normal | <number> | <length> | <percentage> | inherit

Thường sử dụng number để định dạng cho thuộc tính này

Ví dụ 1:

```
.intro {  
  
line-height: 3;  
  
}
```

Khoảng cách dòng ví dụ trên gấp 3 lần độ cao text

Ví dụ 2:

```
body {  
  
font: 13px arial, helvetica, clean, sans-serif;  
  
}  
  
.recipe .intro { font-style: italic; line-height: 3;
```

```
}
```

Đoạn văn bản trong lớp intro sẽ có độ cao dòng là $3 \times 13\text{px} = 39\text{px}$

4.6.2.2 letter-spacing

```
letter-spacing: normal | <length> | inherit
```

Ví dụ:

```
.intro {  
  
letter-spacing: 0.2em;  
  
}
```

Khoảng cách các ký tự có thể âm, khi đó các chữ sẽ đè lên nhau

4.6.2.3 word-spacing

```
word-spacing: normal | <length> | inherit
```

Mặc định: normal

Ví dụ:

```
.intro {  
  
word-spacing: 0.2em;  
  
}
```

4.6.2.4 text-indent

```
text-indent: <length> | <percentage> | inherit
```

Mặc định:0

Ví dụ:

```
.intro {  
  
text-indent: 2em;  
  
}
```

Chú ý: khi sử dụng % nó sẽ phụ thuộc vào chiều rộng của thẻ cha, vì vậy người ta ít dùng % để định dạng cho indent của văn bản

4.6.2.5 text-align

```
text-align :left | right | center | justify
```

Mặc định: left

4.6.2.6 text-decoration

```
text-decoration: none | [ underline || overline || line-through ||  
blink ] Mặc định:none
```

4.6.2.7 text-transform

```
text-transform :capitalize | uppercase | lowercase | none
```

Mặc định: none

Ví dụ:

```
h1 {
```

```
text-transform: capitalize;

}
```

4.6.2.8 white-space

```
white-space: normal | pre | nowrap
```

Mặc định là normal

Người ta hay dùng **pre** cho thẻ code

```
<html lang="en">

<head>

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

<title>Figure 6-15</title>

<style type="text/css">

code
white-space: pre;

}

</style>

</head>

<body>

<code>

<script type="text/javascript">
```

```
document.documentElement.className = 'js';  
  
&lt;/script&gt;  
  
</code>  
  
</body>  
  
</html>
```

Chạy thử với normal hay nowrap để xem kết quả

4.6.2.9 Bài tập định dạng đoạn văn bản

1. Làm sao để giảm khoảng cách giữa các ký tự? Cho ví dụ minh họa.
2. Bỏ đi gạch dưới của link (chỉ hiện lên khi hover chuột vào hoặc focus vào)
3. Chỉnh indent cho đoạn văn bản theo đơn vị % sẽ đc tính như thế nào?
4. Chuyển đổi chữ hoa, chữ thường
5. Muốn hỗ trợ xuống dòng(không dùng
) khi trình bày <code> thì làm như thế nào?

4.6.3 Định dạng nền và hình ảnh

background-color: màu nền
background-image:hình nền
background-repeat: lặp lại nền
background-position: vị trí
background-attachment: điều khiển cuộn

background

4.6.3.1 background-color background-color: <color> | transparent

mặc định: transparent

Màu sắc trên Web:

Color keywords: tên màu

Hexadecimal:màu biểu diễn bằng hệ Hexa

Shorthand hexadecimal: màu biểu diễn bằng hệ Hexa gọn

RGB values:hệ RGB theo giá trị (0-255)

RGB percentage: hệ RGB theo%

RGBA (RGB with alpha channel)

Tham khảo phụ lục về **Bảng màu**

```
body {  
  
background-color: #000;  
  
color: #FFF;  
  
}
```

4.6.3.2 background-image background-image:<uri> | none Mặc định:

none

Ví dụ:

```
body {  
  
background-image: url(bg-page.png);  
  
}
```

Cần đặt hình **bg-page.png** cùng thư mục với file html

4.6.3.3 background-repeat

background-repeat: repeat | repeat-x | repeat-y | no-repeat

Ví dụ:

```
body {  
  
background-image: url(bg-page.png);  
  
background-repeat: no-repeat;  
  
}
```

4.6.3.4 background-position

background-position: [[<percentage> | <length> | left | center | right] [<percentage> | <length> | top | center | bottom]?] | [[left | center | right] || [top | center | bottom]] | inherit

Mặc định: 0% 0%

Ví dụ 1:

```
body {  
  
background-image: url(bg-page.png); background-repeat: no-repeat;  
background-position: bottom right;  
  
}
```

Ví dụ 2:

```
body {  
  
background-image: url(bg-page.png); background-repeat: no-repeat;  
background-position: right 100%;  
  
}
```

4.6.3.5 background-attachment

```
background-attachment: scroll | fixed
```

Mặc định: scroll

IE6 chỉ hỗ trợ fixed cho thẻ body

```
body {  
  
background-image: url(bg-page.png);  
  
background-attachment: fixed;  
  
}
```

4.6.3.6 background

```
background: <'background-color'> || <'background-image'> ||  
<'background-  
repeat'> || <'background-attachment'> || <'background-position'>
```

Ví dụ:

```
body {  
  
background: #CCC url(bg-page.png) repeat-x fixed top right;  
  
}
```

4.6.4 Định dạng hộp (box model)

Mô hình hộp

Định dạng Margin

Định dạng Padding

Định dạng border

Chỉnh chiều rộng và chiều cao

Chỉnh chiều rộng, chiều cao tối thiểu

Xử lý tràn vùng

Với các ví dụ bên dưới nên để border cho box để thấy rõ được ý nghĩa các thuộc tính.

4.6.4.1 Margin

```
margin: [ <length> | <percentage> | auto ] {1,4}
```

```
margin-top: <length> | <percentage> | auto margin-right: <length> |  
<percentage> | auto margin-bottom: <length> | <percentage> | auto  
margin-left: <length> | <percentage> | auto
```

Ví dụ margin với 4 đối số:

```
.intro {  
  
margin-top: 70px; margin-left: 100px; margin-bottom: 50px; margin-  
right: 100px;  
  
}
```

Với 4 đối số thì thứ tự sẽ là **Trên Phải Dưới Trái**

Ví dụ margin với 3 đối số

Sửa ví dụ trên margin: 70px 100px 50px

Với 3 đối số thì thứ tự sẽ là **trên phải trái** dưới

Với 2 đối số thì thứ tự sẽ là **phải trái**

Với 1 đối số thì sẽ cho cùng một giá trị với cả 4 phía

Hiện tượng co/giãn margin:

Xảy ra khi margin của hai thẻ **đụng nhau** làm khoảng cách 2 thẻ thay đổi, hiện tượng này chỉ xảy ra với chiều dọc.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
  
<html lang="en">  
  
<head>  
  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
<title>Figure 8-4</title>

<style type="text/css">

.h1 {

margin-bottom: 100px;

}

.intro {

margin-top: 100px;

}

</style>

</head>

<body>

<h1>Recipes for Cheese</h1>

<p class="intro">Cheese is a remarkably versatile food,
available in
literally

hundreds of varieties with different flavors and textures.</p>

</body>

</html>
```

Canh lề theo chiều ngang với margin:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html lang="en">

<head>

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

<title>Figure 8-7</title>

<style type="text/css">

h1 {

width: 200px;

margin-right: auto;

}

.intro {

width: 200px;

margin-right: auto;

margin-left: auto;

}

.recipe { width: 200px; margin-left: auto;

}

</style>
```

</head>

<body>

<h1>Recipes for Cheese</h1>

<p class="intro">Cheese is a remarkably versatile food, available in

literally

hundreds of varieties with different flavors and textures.</p>

<div class="recipe">

<h2>Welsh Rarebit</h2>

<p class="intro">Welsh Rarebit is a savory dish made from melted cheese, often

Cheddar, on toasted bread, and a variety of other ingredients such as mustard,

egg, or bacon. Here is one take on this classic.</p>

Lightly toast the bread

Place on a baking tray, and spread with butter.

Add the grated Cheddar cheese and 2 tablespoons of beer to a saucepan. Place the saucepan over a medium heat, and stir the cheese continuously until it has melted. Add a teaspoon of wholegrain mustard and grind in

a little pepper. Keep stirring.

```
<li>When thick and smooth, pour over each piece of toast spreading  
it to the edges to stop the toast from burning.</li>  
  
<li>Place under the grill for a couple of minutes or until golden  
brown.</li>  
  
</ol>  
  
</div>  
  
</body>  
  
</html>
```

margin:auto sử dụng canh vào giữa(lưu ý cần sử dụng DOCTYPE)

4.6.4.2 Border

Độ rộng border

```
border-top-width: thin | medium | thick | <length> border-right-  
width: thin | medium | thick | <length> border-bottom-width: thin |  
medium | thick | <length> border-left-width: thin | medium | thick |  
<length> border-width: <border-width> {1,4}
```

Mặc định: Medium

Ví dụ:

```
.intro {  
  
border-top-width: 1px; border-right-width: 3px; border-bottom-width:  
5px; border-left-width: 7px; border-style: solid;  
}
```

Kiểu border

`border-style: <border-style> {1,4}`

`border-top-style: none | hidden | dotted | dashed | solid | double | groove`

`| ridge | inset | outset`

`border-right-style: none | hidden | dotted | dashed | solid | double |`

`groove | ridge | inset | outset`

`border-bottom-style: none | hidden | dotted | dashed | solid | double |`

`groove | ridge | inset | outset`

`border-left-style: none | hidden | dotted | dashed | solid | double | groove`

`| ridge | inset | outset`

Mặc định: none

Ví dụ:

```
body {
```

```
border-width: 3px;
```

```
border-top-style: ridge; border-right-style: dashed; border-bottom-  
style: dotted; border-left-style: double
```

```
}
```

Thay bằng **border-style: ridge dashed dotted double; Màu sắc border:**

`border-color: [<color> | transparent] {1,4}`

```
border-top-color: [ <color> | transparent ] border-right-color: [
<color> | transparent ] border-bottom-color: [ <color> | transparent
] border-left-color: [ <color> | transparent ]
```

Mặc định: là màu của thuộc tính color

IE6, IE7 không hỗ trợ transparent cho màu border. Trong IE, transparent là màu đen.

Khi màu border không được định nghĩa nó sẽ hiểu là màu của thuộc tính color

Sử dụng thuộc tính tắt border

```
border-top: <border-width> || <border-style> || <color>
border-right: <border-width> || <border-style> || <color>
border-bottom: <border-width> || <border-style> ||
<color>
border-left: <border-width> || <border-style> ||
<color>
border: <border-width> || <border-style> ||
<color>
```

Ví dụ 1:

```
body {
width: 600px; margin: 1em auto; border: 1px solid #666;
}
```

4.6.4.3 Padding

```
padding: [ <length> | <percentage> ] {1,4} padding-top: [ <length> |
<percentage> ] padding-right: [ <length> | <percentage> ] padding-
bottom: [ <length> | <percentage> ] padding-left: [ <length> |
<percentage> ]
```

Ví dụ:

```
body {  
  
width: 600px; margin: 1em auto; border: 1px solid #666; padding:  
10px;  
}
```

auto không hỗ trợ trong padding

4.6.4.4 Các thuộc tính đo kích thước

width

```
width: <length> | <percentage> | auto
```

Ví dụ:

```
body {  
  
width: 600px;  
  
}
```

height

```
height: <length> | <percentage> | auto
```

Ví dụ:

```
body {  
  
width: 600px;  
  
height: 75px;
```

```
border: 1px solid #666;

}
```

Giá trị **auto** với thuộc tính **width** và **height** ảnh hưởng khác nhau với mỗi thẻ. Với thẻ <div> hay có thể tự động mở rộng để chứa thành phần trong nó (văn bản, hình ảnh, hay một box con). Những thẻ như vậy gọi là block-element.

Độ đo %: là độ dài so với thẻ bao ngoài nó

min-width, max-width, min-height, max-height nên được định nghĩa vì việc hiển thị phụ thuộc vào độ phân giải màn hình và việc sử dụng các giá trị thuộc tính như auto, hay % sẽ làm nội dung bị giãn hay co lại. (IE6 ko hỗ trợ)

min-width và min-height

Sử dụng để ràng buộc kích thước bé nhất của một thẻ, sử dụng để tránh việc co lại để vừa nội dung hay cửa sổ.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html lang="en">

<head>

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

<title>Figure 8-17</title>

<style type="text/css">

body {

min-width: 1000px; min-height: 250px; border: 1px solid #666;

}
```

```
</style>

</head>

<body>

<h1>Recipes for Cheese</h1>

<p class="intro">Cheese is a remarkably versatile food,
available in
literally
hundreds of varieties with different flavors and textures.</p>

</body>

</html>
```

Xuất hiện scroll-bar

max-width và max-height

Sử dụng để định nghĩa chiều rộng tối đa khi vùng nội dung được mở rộng.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html lang="en">

<head>

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

<title>Figure 8-18</title>

<style type="text/css">
body {
```

```
width: 600px;

max-height: 100px;

border: 1px solid #666;

}

.intro {

max-width: 200px;

}

</style>

</head>

<body>

<h1>Recipes for Cheese</h1>

<p class="intro">Cheese is a remarkably versatile food,
available in
literally
hundreds of varieties with different flavors and textures.</p>

</body>

</html>
```

4.6.4.5 Tràn vùng

overflow: visible | hidden | scroll | auto overflow-x: visible |
hidden | scroll | auto overflow-y: visible | hidden | scroll | auto
mặc định: visible

Ví dụ 1:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
  
<html lang="en">  
  
<head>  
  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
  
<title>Figure 8-19</title>  
  
<style type="text/css">  
  
body {  
  
width: 600px;  
  
}  
h1 {  
  
width: 50px;  
  
overflow: hidden;  
  
}  
  
.recipe { height: 200px; overflow: auto;  
  
}  
  
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Recipes for Cheese</h1>
```

```
<p class="intro">Cheese is a remarkably versatile food,  
available in literally
```

```
hundreds of varieties with different flavors and textures.</p>
```

```
<div class="recipe">
```

```
<h2>Welsh Rarebit</h2>
```

```
<p class="intro">Welsh Rarebit is a savory dish made from melted  
cheese,
```

```
often
```

```
Cheddar, on toasted bread, and a variety of other ingredients  
such as mustard,
```

```
egg, or bacon. Here is one take on this classic.</p>
```

```
<ol>
```

```
<li>Lightly toast the bread</li>
```

```
<li>Place on a baking tray, and spread with butter.</li>
```

```
<li>Add the grated Cheddar cheese and 2 tablespoons of beer to a  
saucepan. Place the saucepan over a medium heat, and stir the cheese  
continuously until it has melted. Add a teaspoon of wholegrain  
mustard and grind in
```

```
a little pepper. Keep stirring.</li>
```

```
<li>When thick and smooth, pour over each piece of toast spreading  
it to the edges to stop the toast from burning.</li>  
  
<li>Place under the grill for a couple of minutes or until golden  
brown.</li>  
  
</ol>  
  
</div>  
  
</body>  
  
</html>
```

4.6.5 Trượt(floating) và canh chỉnh theo chiều dọc

Thuộc tính **float** và cách sử dụng

Thuộc tính **clear**

Canh chỉnh theo chiều dọc (đoạn văn bản)

4.6.5.1 Floating

```
float: left | right | none
```

Mặc định: none

Ví dụ:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
  
<html lang="en">  
  
<head>
```

```

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

<title>Figure 9-1</title>

<style type="text/css">

h1 {

float: left;

}

</style>

</head>

<body>

<h1>Recipes for Cheese</h1>

<p class="intro">Cheese is a remarkably versatile food,
available in

literally hundreds of varieties with different flavors and
textures.</p>

</body>

</html>

```

Nhận xét: Khi không để thuộc tính float(tức float:none), phần nội dung trong <h1> nằm trên và <p> nằm dưới. Thử điều chỉnh left->right.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html lang="en">

```

```
<head>

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

<title>Figure 9-2</title>

<style type="text/css">

.recipe h2 {

float: left;

}

.recipe .intro,

.recipe ol { float: right; width: 500px;

}

</style>

</head>

<body>

<h1>Recipes for Cheese</h1>

<p class="intro">Cheese is a remarkably versatile food,
available in
literallyhundreds of varieties with different flavors and
textures.</p>

<div class="recipe">

<h2>Welsh Rarebit</h2>
```

`<p class="intro">Welsh Rarebit is a savory dish made from melted cheese,
often`

`Cheddar, on toasted bread, and a variety of other ingredients
such as mustard,`

`egg, or bacon. Here is one take on this classic.</p>`

``

`Lightly toast the bread`

`Place on a baking tray, and spread with butter.`

`Add the grated Cheddar cheese and 2 tablespoons of beer to a
saucepan. Place the saucepan over a medium heat, and stir the cheese
continuously until it has melted. Add a teaspoon of wholegrain
mustard and grind in`

`a little pepper. Keep stirring.`

`When thick and smooth, pour over each piece of toast spreading
it to the edges to stop the toast from burning.`

`Place under the grill for a couple of minutes or until golden`

`brown.`

``

`</div>`

`</body>`

`</html>`

Lưu ý: kết quả hiển thị phụ thuộc vào kích thước màn hình

Ví dụ: Sử dụng thuộc tính float tạo form như hình sau:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html lang="en">

<head>

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

<title>Example 9-1</title>

<style type="text/css">

body {

width: 600px;

margin: 1em auto;

}

h1 {

float: left;

margin-top: 0;

}

.navigation { float: right; margin: 0;
```

```
list-style: none;
```

```
}
```

```
.navigation li {
```

```
float: left;
```

```
}
```

```
.navigation a {
```

```
display: block;
```

```
margin-left: 0.5em;

padding: 0.5em;

border: 1px solid #CCC;

color: #233;

text-decoration: none;

}

.navigation a:focus,

.navigation a:hover { background: #233; color: #FFF;

}

</style>

</head>

<body>

<h1>Recipes for Cheese</h1>

<ul class="navigation">

<li><a href="#">Home</a></li>

<li><a href="#">Recipes</a></li>

<li><a href="#">Suggestions</a></li>

</ul>
```

```
</body>
```

```
</html>
```

Recipes for Cheese

[Home](#)[Recipes](#)[Suggestions](#)

4.6.5.2 Hủy ảnh hưởng của thuộc tính float

```
clear: none | left | right | both
```

Giá trị mặc định: none

Ví dụ trên với thuộc tính clear:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 9-1</title>
<style type="text/css">
body {
width: 600px;
margin: 1em auto;
```

```
}

h1 {

float: left;

margin-top: 0;

}

.navigation { float: right; margin: 0;

list-style: none;

}

.navigation li { float: left; clear:left;

}

.navigation a { display: block; margin-left: 0.5em; padding: 0.5em;

border: 1px solid #CCC;

color: #233;

text-decoration: none;

}

.navigation a:focus,

.navigation a:hover { background: #233; color: #FFF;

}

</style>
```

```

</head>

<body>

<h1>Recipes for Cheese</h1>

<ul class="navigation">

<li><a href="#">Home</a></li>

<li><a href="#">Recipes</a></li>

<li><a href="#">Suggestions</a></li>

</ul>

</body>

</html>

```

4.6.5.3 Canh lề theo chiều dọc

vertical-align: baseline | sub | super | top | text-top | middle | bottom | text-bottom | <percentage> | <length> Mặc định: baseline

Ví dụ cho sub(chỉ số dưới) và super(chỉ số trên):

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html lang="en">

<head>

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

```

```
<title>Figure 9-11</title>

<style type="text/css">

h1 span {

font-size: 0.5em;

}

.superscript {

    vertical-align: super;

}

.subscript {

vertical-align: sub;

}

</style>

</head>

<body>

<h1><span      class="superscript">Tasty</span>Recipes      for
<span class="subscript">

Tasty</span>Cheese</h1>

</body>

</html>
```

Ví dụ cho thuộc tính top, middle, bottom:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html lang="en">

<head>

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

<title>Figure 9-12</title>

<style type="text/css">

h1 span {

font-size: 0.5em;

}

.top {

vertical-align: top;

}

.middle {

vertical-align: middle;

}

.bottom {

vertical-align: bottom;
```

```
}  
  
</style>  
  
</head>  
  
<body>  
  
<h1><span                class="top">Super</span>  
<span class="middle">Tasty</span>Recipes for  
  
<span class="bottom">Tasty</span>Cheese</h1>  
  
</body>  
  
</html>
```

CÂU HỎI VÀ BÀI TẬP CHƯƠNG 4

Câu 1: Sử dụng ngôn ngữ HTML soạn thảo trang Web có nội dung và định dạng theo mẫu::

NỘI DUNG ÔN TẬP TỐT NGHIỆP LỚP THPT36AQN

1. Môn Cơ sở: **10 tiết**, phòng: **B1.01**
2. Môn Chuyên ngành: **10 tiết**, phòng: **B1.01**
3. Môn Chính trị: **5 tiết**, phòng: **B1.01**

DANH SÁCH SINH VIÊN ĐĂNG KÝ ÔN TẬP:

STT	Họ tên	Môn Cơ sở		Môn chuyên ngành		Môn chính trị	
		Học	Tài liệu	Học	Tài liệu	Học	Tài liệu
1	Hoàng Anh	X		X			X
2	Nguyễn Hà	X	X		X		X
3	Đỗ Kỳ	X		X			X
4	Lê Long	X	X				X
5	Phạm Quang	x		X			X
Tổng cộng		5	2	3	1	0	5

Yêu cầu:

- Chọn màu nền tùy ý cho trang
- Chọn màu nền dòng tiêu đề và dòng cuối cho bảng, màu chữ tùy ý
- Độ rộng Table=800, Khoảng cách giữa các ô =0 , độ dày đường viền 1, màu viền tùy ý
- Canh lề và trộn ô đúng theo mẫu.

Câu 2: Cùng mẫu nội dung và định dạng trên, sử dụng CSS trong định dạng: danh sách, kiểu chữ, font chữ, bảng, dòng, ô.

Câu 3: Sử dụng ngôn ngữ HTML soạn thảo trang Web có nội dung và định dạng theo mẫu:

BẢNG GIÁ ĐĂNG KÝ CƯỚC DỊCH VỤ INTERNET

VNN		FPT	
Cước truy cập Internet		Thẻ kết nối Internet	
Thời gian	Cước truy cập(đ/ph)	Số giờ kết nối	Mệnh giá thẻ
0h-07h	100	14h	100.000đ
07h-19h	210	33h	200.000đ
19h-24h	150	55h	300.000đ
Cước thuê bao trong tháng: 27.273đ/tháng		110h	500.000đ

NHẬP THÔNG TIN MUA THẺ

Họ và tên:

Số CMND:

Đăng ký:

Mua thẻ:

- 100.000đ
- 200.000đ
- 300.000đ
- 500.000đ

Cập nhật

Hủy bỏ

Yêu cầu:

- Chọn màu nền tùy ý cho trang
- Chọn màu nền dòng tiêu đề và dòng cuối cho bảng, màu chữ tùy ý
- Độ rộng Table=800, Khoảng cách giữa các ô =0 , độ dày đường viền 1, màu viền tùy ý
- Canh lề và trộn ô đúng theo mẫu.
- Tạo form nhập thông tin mua thẻ như trong mẫu

Câu 4: Sử dụng CSS trong định dạng: kiểu chữ, font chữ, bảng, dòng, ô của bảng ở mẫu trên.

Câu 5: Sử dụng HTML và sử dụng định dạng mẫu CSS định dạng văn bản mẫu sau:

Mùa mưa đang đến gần, và nhiều khả năng ô tô của bạn sẽ phải lội nước. Nếu bạn nghĩ rằng chỉ việc đem xe đi rửa ngay là xong thì hoàn toàn nhầm lẫn. Biết cách tự kiểm tra và chăm sóc cho “xế yêu” của mình vẫn hơn.



Ảnh minh họa

xe.

Dưới đây là một số gợi ý giúp bạn quan tâm đến xe của mình theo đúng cách. Nếu xe bạn vừa phải lội nước, hãy thử thực hiện một số thao tác đơn giản sau:

- Trước tiên, hãy kiểm tra xem thân trải sàn, bọc ghế và các tấm ốp cửa có bị ẩm không. Nếu có, hãy đem xe tới xưởng yêu cầu thợ làm khô chúng ngay để tránh bị nấm mốc trong

- Kiểm tra dầu động cơ, dầu hộp số, vi sai. Nếu thấy chúng có màu trắng đục, loãng, không còn màu sắc như nguyên bản, nhiều khả năng trong dầu có lẫn nước. Hãy đem xe đến xưởng sửa chữa bảo hành.

CHƯƠNG 5. JAVASCRIPT

5.1. JavaScript là gì

Là ngôn ngữ kịch bản, thông dịch và hướng đối tượng hỗ trợ hiển thị các trang Web sinh động hơn, JavaScript thì hoàn toàn miễn phí.

Là ngôn ngữ ở phía client, sử dụng kiểm tra nhập liệu hay các yếu tố khác (phiên bản trình duyệt)

5.2. Chèn mã Javascript

- ▣ Trực tiếp trong file HTML:

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
document.write("Hello World!");
```

```
</script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

- ▣ Chèn từ file .js:

```
<script src="my_script.js"></script>
```

Vị trí chèn thẻ <script></script> thường nằm trong thẻ head. Ghi chú trong javascript sử dụng // hoặc /* */

5.3. THẺ <NOSCRIPT> VÀ </NOSCRIPT>

Cặp thẻ này dùng để định rõ nội dung thông báo cho người sử dụng biết trình duyệt không hỗ trợ JavaScript. Khi đó trình duyệt sẽ không hiểu thẻ <NOSCRIPT> và nó bị lờ đi, còn đoạn mã nằm trong cặp thẻ này sẽ được Navigator hiển thị. Ngược lại, nếu trình duyệt có hỗ trợ JavaScript thì đoạn mã trong cặp thẻ <NOSCRIPT> sẽ được bỏ qua. Tuy nhiên, điều này cũng có thể xảy ra nếu người sử dụng không sử dụng JavaScript trong trình duyệt của mình bằng cách tắt nó đi trong hộp Preferences/Advanced

<NOSCRIPT>

 Trang này có sử dụng JavaScript. Do đó bạn cần sử dụng trình duyệt Netscape Navigator từ version 2.0 trở đi!

 Hãy kích chuột vào đây để tải về phiên bản Netscape mới hơn

</BR>

Nếu bạn đã sử dụng trình duyệt Netscape từ 2.0 trở đi mà vẫn đọc được dòng chữ này thì hãy bật Preferences/Advanced/JavaScript lên

</NOSCRIPT>

5.4. Các lệnh xuất thông báo cơ bản

- Thường sử dụng xuất các thông báo và là công cụ để kiểm lỗi(debug)

alert(),prompt(),confirm()

```
<html>

<head>

<script type="text/javascript">

alert("Hi there");

confirm("I'm gonna do something, okay?");

prompt("What should I do?");

</script>

</head>

<body>

</body>

</html>
```

5.5. Chèn chuỗi vào nội dung trang HTML

```
<html>

<head>

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

<script type="text/javascript">

document.write("<h1>Hello</h1>");

</script>

</head>
```

```
<body> Xin Chào
```

```
</body>
```

```
</html>
```

□ Có thể tạo một trang Web từ javascript

5.6. Biến-Kiểu dữ liệu

5.6.1 Khai báo biến

```
var TenBien;
```

□ Cách đặt tên biến như trong C++, Không cần xác định kiểu dữ liệu cho biến, KDL được xác định dựa trên giá trị gán cho biến.

5.6.2 Gán

```
TenBien = [Giá Trị];
```

5.6.3 Các kiểu dữ liệu

5.6.3.1 Undefined

```
var foo;
```

```
alert(foo); //chưa xác định vì chưa có giá trị gán cho biến
```

5.6.3.2 Null

```
var foo = null;
```

```
alert(foo);
```

5.6.3.3 Numbers

```
var foo = 5;  
  
alert(foo);
```

5.6.3.4 Strings

```
var foo = "five";  
  
alert( foo );
```

5.6.3.5 Booleans

```
var foo = true;
```

5.6.3.6 Arrays

Khai báo, tạo mảng, mảng đc đánh số từ 0.

```
var foo = [5, "five", "5"]; alert( foo[0] ); // Alerts "5" alert( foo[1] ); // Alerts "five"  
alert( foo[2] ); // Also alerts "5"
```

5.6.4 Các toán tử so sánh và toán học

Tương tự C++

5.6.4.1 Toán tử == và ===

```
alert( "5" == 5 ); // "true", không phân biệt KDL
```

```
alert( "5" === 5 ); // "false", phân biệt KDL
```

```
alert( "5" !== 5 ); // "true", phân biệt KDL
```

5.6.4.2 Chuỗi

Khi được sử dụng với chuỗi, toán tử + được coi là kết hợp hai chuỗi, ví dụ:

"abc" + "xyz" được "abcxyz".

5.6.4.3 Logic

JavaScript hỗ trợ các toán tử logic sau đây:

<code>expr1 && expr2</code>	Là toán tử logic AND, trả lại giá trị đúng nếu cả <code>expr1</code> và <code>expr2</code> cùng đúng.
<code>expr1 expr2</code>	Là toán tử logic OR, trả lại giá trị đúng nếu ít nhất một trong hai <code>expr1</code> và <code>expr2</code> đúng.
<code>! expr</code>	Là toán tử logic NOT phủ định giá trị của <code>expr</code> .

5.7. Các Lệnh

5.7.1 Câu Lệnh Điều Kiện

Câu lệnh điều kiện cho phép chương trình ra quyết định và thực hiện công việc nào đấy dựa trên kết quả của quyết định. Trong JavaScript, câu lệnh điều kiện là `if...else`

Câu lệnh này cho phép bạn kiểm tra điều kiện và thực hiện một nhóm lệnh nào đấy dựa trên kết quả của điều kiện vừa kiểm tra. Nhóm lệnh sau `else` không bắt buộc phải có, nó cho phép chỉ ra nhóm lệnh phải thực hiện nếu điều kiện là sai.

Cú pháp

```
if ( <điều kiện> )
```

```
{
```

```

    }

else

    {

    }

//Các câu lệnh với điều kiện đúng
//Các câu lệnh với điều kiện sai
    Ví dụ:

if (x==10)

{

    document.write("x bằng 10, đặt lại x bằng 0.");

    x = 0;

}

else

{

    document.write("x không bằng 10.");

}

```

5.7.2 Câu Lệnh Lặp

5.7.2.1 Vòng Lặp for

Vòng lặp for thiết lập một biểu thức khởi đầu - `initExpr`, sau đó lặp một đoạn mã cho đến khi biểu thức <điều kiện> được đánh giá là đúng. Sau khi kết thúc mỗi vòng lặp, biểu thức `incrExpr` được đánh giá lại.

Cú pháp:

```
for (initExpr; <điều kiện> ; incrExpr){  
  
    //Các lệnh được thực hiện trong khi lặp  
  
}
```

5.7.2.2 While

Vòng lặp while lặp khối lệnh chừng nào <điều kiện> còn được đánh giá là đúng

Cú pháp:

```
while (<điều kiện>)  
  
{  
  
    //Các câu lệnh thực hiện trong khi lặp  
  
}
```

5.7.2.2.1 BREAK

Câu lệnh break dùng để kết thúc việc thực hiện của vòng lặp for hay while. Chương trình được tiếp tục thực hiện tại câu lệnh ngay sau chỗ kết thúc của vòng lặp.

Cú pháp

```
break;
```

Đoạn mã sau lặp cho đến khi x lớn hơn hoặc bằng 100. Tuy nhiên nếu giá trị x đưa vào vòng lặp nhỏ hơn 50, vòng lặp sẽ kết thúc

5.7.2.2.2 CONTINUE

Lệnh continue giống lệnh break nhưng khác ở chỗ việc lặp được kết thúc và bắt đầu từ đầu vòng lặp. Đối với vòng lặp while, lệnh continue điều khiển quay lại <điều kiện>; với for, lệnh continue điều khiển quay lại incrExpr.

Cú pháp continue;

5.7.2.3 CÁC CÂULỆNH THAO TÁC TRÊN ĐỐI TƯỢNG

5.7.2.3.1 FOR...IN

Câu lệnh này được sử dụng để lặp tất cả các thuộc tính (properties) của một đối tượng. Tên biến có thể là một giá trị bất kỳ, chỉ cần thiết khi bạn sử dụng các thuộc tính trong vòng lặp. Ví dụ sau sẽ minh họa điều này

Cú pháp

```
for (<variable> in <object>)
```

```
{
```

```
    //Các câu lệnh
```

```
}
```

Ví dụ

Ví dụ sau sẽ lấy ra tất cả các thuộc tính của đối tượng Window và in ra tên của mỗi thuộc tính. Kết quả được minh họa trên hình 5.2.

```
<html>
```

```
<head>
```

```
<title>for in example </title>
```

```
<script language= "javascript">
```

```
document.write("the properties of the window object are: <br>");
```

```
for (var x in window)
```

```
    document.write("    " + x + ", ");
```

```
</script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

5.7.2.3.2 NEW

Biến new được thực hiện để tạo ra một thể hiện mới của một đối tượng

Cú pháp

```
objectvar = new object_type ( param1 [,param2]... [,paramN])
```

Ví dụ sau tạo đối tượng person có các thuộc tính firstname, lastname, age, sex. Chú ý rằng từ khoá this được sử dụng để chỉ đối tượng trong hàm person. Sau đó ba thể hiện của đối tượng person được tạo ra bằng lệnh new

```
<html>
```

```
<head>
```

```
<title>new example </title>
```

```
<script language= "JavaScript">
```

```

function person(first_name, last_name, age, sex){ this.first_name=first_name;
    this.last_name=last_name;

    this.age=age;

    this.sex=sex;

}

person1= new person("Thuy", "Dau Bich", "23", "Female"); person2= new person("Chung", "Nguyen
Bao", "24", "Male"); person3= new person("Binh", "Nguyen Nhat", "24", "Male"); person4= new
person("Hoàn", "Đỗ Văn", "24", "Male");

document.write ("1. "+person1.last_name+" " + person1.first_name + "<BR>"
);
document.write("2. "+person2.last_name+" "+ person2.first_name + "<br>");

document.write("3. "+ person3.last_name + " "+ person3.first_name + "<br>"); document.write("4. "+
person4.last_name + " "+ person4.first_name+"<BR>");

</script>

</head>

<body>

</body>

</html>

```

5.7.2.3.3 THIS

Từ khoá `this` được sử dụng để chỉ đối tượng hiện thời. Đối tượng được gọi thường là đối tượng hiện thời trong phương thức hoặc trong hàm.

Cú pháp

this [.property]

Có thể xem ví dụ của lệnh new.

5.7.2.3.4 WITH

Lệnh này được sử dụng để thiết lập đối tượng ngầm định cho một nhóm các lệnh, bạn có thể sử dụng các thuộc tính mà không đề cập đến đối tượng.

Cú pháp

with (object)

```
{  
  
    // statement  
  
}
```

Ví dụ:

Ví dụ sau chỉ ra cách sử dụng lệnh with để thiết lập đối tượng ngầm định là document và có thể sử dụng phương thức write mà không cần đề cập đến đối tượng document

```
<html>
```

```
<head>
```

```
<title>With Example </title>
```

```
<script language= "JavaScript">
```

```
with (document){
```

```
write("This is an exemple of the things that can be done <BR>");
```

```
write("With the <B>with<B> statment. <P>");

write("This can really save some typing");

}

</script>

</head>

<body>

</body>

</html>
```

5.8. Hàm (FUNCTIONS)

JavaScript cũng cho phép sử dụng các hàm. Mặc dù không nhất thiết phải có, song các hàm có thể có một hay nhiều tham số truyền vào và một giá trị trả về. Bởi vì JavaScript là ngôn ngữ có tính định kiểu thấp nên không cần định nghĩa kiểu tham số và giá trị trả về của hàm. Hàm có thể là thuộc tính của một đối tượng, trong trường hợp này nó được xem như là phương thức của đối tượng đó.

Lệnh function được sử dụng để tạo ra hàm trong JavaScript.

Cú pháp

```
function fnName([param1],[param2],...,[paramN])

{

//function statement

}
```

Ví dụ:

Ví dụ sau minh hoạ cách thức tạo ra và sử dụng hàm như là thành viên của một đối tượng. Hàm printStats được tạo ra là phương thức của đối tượng person

```
<html> <head>

<title>function example </title>

<script language= "JavaScript">

function person(first_name, last_name, age, sex)

{

    this.first_name=first_name; this.last_name=last_name; this.age=age;

    this.sex=sex;

this.printStats=printStats;

}

function printStats() {

    with (document) {

write (" Name :" + this.last_name + " " + this.first_name + "<BR>" );

    write("Age :"+this.age+"<BR>");

    write("Sex :"+this.sex+"<BR>");

    }

}

}
```

```

person1= new person("Thuy", "Dau Bich", "23", "Female"); person2= new person("Chung", "Nguyen
Bao", "24", "Male"); person3= new person("Binh", "Nguyen Nhat", "24", "Male"); person4= new
person("Hoan", "Do Van", "23", "Male"); person1printStats();

person2printStats(); person3printStats(); person4printStats();

</script>

</head>

<body> </body>

</html>

```

5.8.1 EVAL

Hàm này được sử dụng để đánh giá các biểu thức hay lệnh. Biểu thức, lệnh hay các đối tượng của thuộc tính đều có thể được đánh giá. Đặc biệt hết sức hữu ích khi đánh giá các biểu thức do người dùng đưa vào (ngược lại có thể đánh giá trực tiếp).

Cú pháp:

```
returnval=eval (bất kỳ biểu thức hay lệnh hợp lệ trong Java)
```

Ví dụ:

```

<html>

<head>

<title>eval example </title>

<script language= "JavaScript">

    var string="10+ Math.sqrt(64)";

document.write(string+ "="+ eval(string));

```

```
</script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

5.8.2 PARSEINT

Hàm này chuyển một chuỗi số thành số nguyên với cơ số là tham số thứ hai (tham số này không bắt buộc). Hàm này thường được sử dụng để chuyển các số nguyên sang cơ số 10 và đảm bảo rằng các dữ liệu được nhập dưới dạng ký tự được chuyển thành số trước khi tính toán. Trong trường hợp dữ liệu vào không hợp lệ, hàm `parseInt` sẽ đọc và chuyển dạng chuỗi đến vị trí nó tìm thấy ký tự không phải là số. Ngoài ra hàm này còn cắt dấu phẩy động.

Cú pháp

`parseInt (string, [, radix])` Ví dụ:

```
<html>
```

```
<head>
```

```
<TITLE> PARSEINT EXAMPLE </TITLE>
```

```
<script language= "JavaScript">
```

```
document.write("Converting 0xC hex to base-10: " + parseInt(0xC,10) + "<br>");
```

```
document.write("Converting 1100 binary to base-10: " + parseInt(1100,2) + "<br>");
```

```
</script>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

5.8.3 PARSEFLOAT

Hàm này giống hàm parseInt nhưng nó chuyển chuỗi thành số biểu diễn dưới dạng dấu phẩy động.

Cú pháp

```
parseFloat (string)
```

5.9. Sự kiện

JavaScript là ngôn ngữ định hướng sự kiện, nghĩa là sẽ phản ứng trước các sự kiện xác định trước như kích chuột hay tải một văn bản. Một sự kiện có thể gây ra việc thực hiện một đoạn mã lệnh (gọi là các chương trình xử lý sự kiện) giúp cho chương trình có thể phản ứng một cách thích hợp.

Chương trình xử lý sự kiện (Event handler): Một đoạn mã hay một hàm được thực hiện để phản ứng trước một sự kiện gọi là chương trình xử lý sự kiện. Chương trình xử lý sự kiện được xác định là một thuộc tính của một thẻ HTML:

```
<tagName eventHandler = "JavaScript Code or Function">
```

Ví dụ sau gọi hàm CheckAge() mỗi khi giá trị của trường văn bản thay đổi:

```
<input type=text name="age" onChange="CheckAge()">
```

Đoạn mã của chương trình xử lý sự kiện không là một hàm; nó là các lệnh của JavaScript cách nhau bằng dấu chấm phẩy. Tuy nhiên cho mục đích viết thành các module nên viết dưới dạng các hàm.

Một số chương trình xử lý sự kiện trong JavaScript:

onBlur	Xảy ra khi input focus bị xoá từ thành phần form
onClick	Xảy ra khi người dùng kích vào các thành phần hay liên kết của form.
onChange	Xảy ra khi giá trị của thành phần được chọn thay đổi
onFocus	Xảy ra khi thành phần của form được focus(làm nổi lên).
onLoad	Xảy ra trang Web được tải.
onMouseOver	Xảy ra khi di chuyển chuột qua kết nối hay anchor.
onSelect	Xảy ra khi người sử dụng lựa chọn một trường nhập dữ liệu trên form.
onSubmit	Xảy ra khi người dùng đưa ra một form.
onUnload	Xảy ra khi người dùng đóng một trang

Sau đây là bảng các chương trình xử lý sự kiện có sẵn của một số đối tượng.
 Các đối tượng này sẽ được trình bày kỹ hơn trong phần sau

Đối tượng	Chương trình xử lý sự kiện có sẵn
Selection list	onBlur, onChange, onFocus
Text	onBlur, onChange, onFocus, onSelect
Textarea	onBlur, onChange, onFocus, onSelect
Button	onClick
Checkbox	onClick
Radio button	onClick
Hypertext link	onClick, onMouseOver, onMouseOut
Clickable Imagemap area	onMouseOver, onMouseOut
Reset button	onClick
Submit button	onClick
Document	onLoad, onUnload, onError

Window	onLoad, onUnload, onBlur, onFocus
Framesets	onBlur, onFocus
Form	onSubmit, onReset
Image	onLoad, onError, onAbort

Ví dụ sau là một đoạn mã script đơn giản của chương trình xử lý sự kiện thẩm định giá trị đưa vào trong trường text. Tuổi của người sử dụng được nhập vào trong trường này và chương trình xử lý sự kiện sẽ thẩm định tính hợp lệ của dữ liệu đưa vào. Nếu không hợp lệ sẽ xuất hiện một thông báo yêu cầu nhập lại. Chương trình xử lý sự kiện được gọi mỗi khi trường **age** bị thay đổi và focus chuyển sang trường khác.

```

<html>
<head>

<title> An Event Handler Exemple </title>

<script language= "JavaScript">

function CheckAge(form) {

if ( (form.age.value<0)|| (form.age.value>120))

{

alert("Tuổi nhập vào không hợp lệ! Mời bạn nhập lại");

form.age.value=0;

}

}

</script>

</head>

```

<body>

<form name="phieu_dieu_tra">

Nhập vào tên của bạn:

tên <input type="text" name="ten" maxlength=10 size=10>
 đệm <input type="text" name="dem" maxlength=15 size=10>
 họ <input type="text" name="ho" maxlength=10 size=10>
 age <input type="text" name="age" maxlength=3 size=2 onchange=" CheckAge (phieu_dieu_tra)">

<P>

Bạn thích mùa nào nhất:

Mùa xuân<input type="radio" name="mua" value="mua xuan"> Mùa hạ<input type="radio" name="mua" value="mua ha">

Mùa thu<input type="radio" name="mua" value="mua thu">

mùa đông<input type="radio" name="mua" value="mua dong">

<P>

Hãy chọn những hoạt động ngoài trời mà bạn yêu thích:

Đi bộ<input type="checkbox" name="hoat_dong" value="Di bo">

Trượt tuyết<input type="checkbox" name="hoat_dong" value="Truot tuyet">

Thể thao dưới nước<input type="checkbox" name="hoat_dong" value="Duoi">

```
nuoc">
```

```
Đạp xe<input type="checkbox name" name="hoat_dong" value="Đạp xe">
```

```
<p>
```

```
<input type=submit>
```

```
<input type=reset>
```

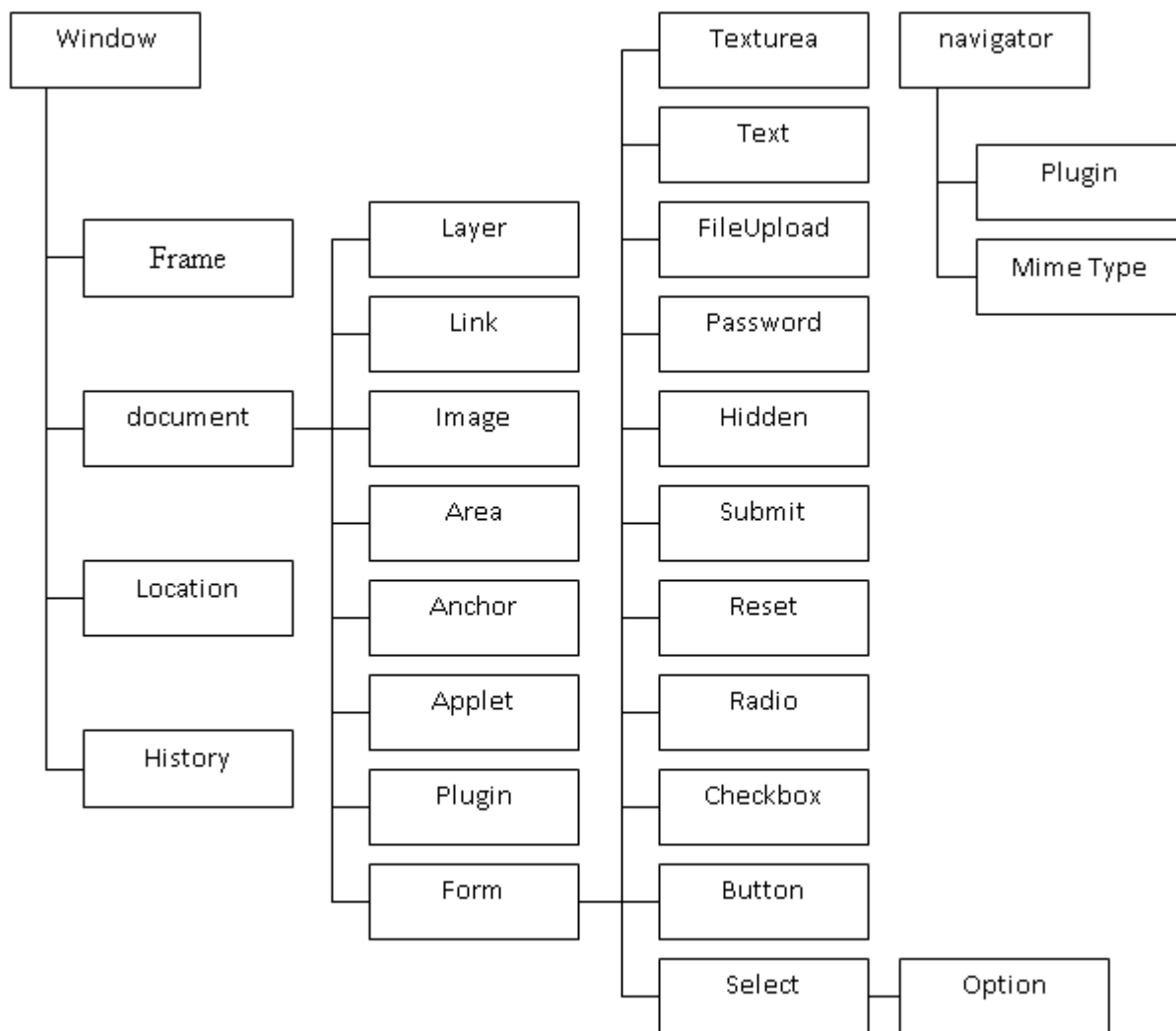
```
</form>
```

```
</body>
```

```
</html>
```

5.10. Các đối tượng trong javascript

Như đã nói JavaScript là ngôn ngữ lập trình dựa trên đối tượng, nhưng không hướng đối tượng bởi vì nó không hỗ trợ các lớp cũng như tính thừa kế.



Trong sơ đồ phân cấp các đối tượng của JavaScript, các đối tượng con thực sự là các thuộc tính của các đối tượng bố mẹ. Trong ví dụ về chương trình xử lý sự kiện trước đây form tên **phieu_dieu_tra** là thuộc tính của đối tượng document và trường text **age** là thuộc tính của form **phieu_dieu_tra**. Để tham chiếu đến giá trị của **age**, bạn phải sử dụng:

`document.phieu_dieu_tra.age.value`

Các đối tượng có thuộc tính (properties), phương thức (methods), và các chương trình xử lý sự kiện (event handlers) gắn với chúng. Ví dụ đối tượng document có thuộc tính title phản ánh nội dung của thẻ <title> của document. Bên cạnh đó bạn thấy phương thức document.write được sử dụng trong nhiều ví dụ để đưa văn bản kết quả ra document.

Đối tượng cũng có thể có các chương trình xử lý sự kiện. Ví dụ đối tượng link có hai chương trình xử lý sự kiện là `onClick` và `onMouseOver`. `onClick` được gọi khi có đối tượng link được kích, `onMouseOver` được gọi khi con trỏ chuột di chuyển qua link.

Khi bạn tải một document xuống Navigator, nó sẽ tạo ra một số đối tượng cùng với những giá trị các thuộc tính của chúng dựa trên file HTML của document đó và một vài thông tin cần thiết khác. Những đối tượng này tồn tại một cách có cấp bậc và phản ánh chính cấu trúc của file HTML đó.

Trong sơ đồ phân cấp này, các đối tượng con chính là các thuộc tính của một đối tượng cha. Ví dụ như một form tên là `form1` chính là một đối tượng con của đối tượng `document` và được gọi tới là `document.form1`

Tất cả các trang đều có các đối tượng sau đây

- **navigator**: có các thuộc tính tên và phiên bản của Navigator đang được sử dụng, dùng cho MIME type được hỗ trợ bởi client và plug-in được cài đặt trên client.
- **window**: là đối tượng ở mức cao nhất, có các thuộc tính thực hiện áp dụng vào toàn bộ cửa sổ.
- **document**: chứa các thuộc tính dựa trên nội dung của document như tên, màu nền, các kết nối và các forms.
- **location**: có các thuộc tính dựa trên địa chỉ URL hiện thời
- **history**: Chứa các thuộc tính về các URL mà client yêu cầu trước đó.

5.10.1 ĐỐI TƯỢNG NAVIGATOR

Đối tượng này được sử dụng để đạt được các thông tin về trình duyệt như số phiên bản. Đối tượng này không có phương thức hay chương trình xử lý sự kiện.

Các thuộc tính

<i>appCodeName</i>	Xác định tên mã nội tại của trình duyệt (Atlas).
--------------------	--

<i>AppName</i>	Xác định tên trình duyệt.
<i>AppVersion</i>	Xác định thông tin về phiên bản của đối tượng navigator.
<i>userAgent</i>	Xác định header của user - agent.

Ví dụ

```

html>

<head>

<title> navigator object exemple </title>

<script language= "JavaScript">

document.write("appCodeName= "+navigator.appCodeName+ "<BR>"); document.write("appName
= "+navigator.appName + "<BR>"); document.write("appVersion= "+navigator.appVersion +
"<BR>"); document.write("userAgent= "+navigator.userAgent + "<BR>");

</script>

</head>

<body>

</body>

</html>

```

5.10.2 ĐỐI TƯỢNG WINDOW

Đối tượng window như đã nói ở trên là đối tượng ở mức cao nhất. Các đối tượng document, frame, vị trí đều là thuộc tính của đối tượng window.

Các thuộc tính

- **defaultStatus** - Thông báo ngầm định hiển thị lên trên thanh trạng thái của cửa sổ
- **Frames** - Mảng xác định tất cả các frame trong cửa sổ.
- **Length** - Số lượng các frame trong cửa sổ cha mẹ.
- **Name** - Tên của cửa sổ hiện thời.
- **Parent** - Đối tượng cửa sổ cha mẹ
- **Self** - Cửa sổ hiện thời.
- **Status** - Được sử dụng cho thông báo tạm thời hiển thị lên trên thanh trạng thái cửa sổ. Được sử dụng để lấy hay đặt lại thông báo trạng thái và ghi đè lên defaultStatus.
- **Top** - Cửa sổ ở trên cùng.
- **Window** - Cửa sổ hiện thời

Các Phương Thức

- **alert("message")** -Hiển thị hộp hội thoại với chuỗi "message" và nút OK.
- **clearTimeout(timeoutID)** -Xóa timeout do setTimeout đặt. setTimeout trả lại timeoutID
- **windowReference.close** -Đóng cửa sổ windowReference.
- **confirm("message")** -Hiển thị hộp hội thoại với chuỗi "message", nút OK và nút Cancel. Trả lại giá trị True cho OK và False cho Cancel.
- **[windowVar =][window]. open("URL", "windowName", ["windowFeatures"])** - Mở cửa sổ mới.

- **prompt ("message" [, "defaultInput"])** - Mở một hộp hội thoại để nhận dữ liệu vào trường text.

- **TimeoutID = setTimeout(expression, msec)** - Đánh giá biểu thức expression sau thời gian msec.

Ví dụ: Sử dụng tên cửa sổ khi gọi tới nó như là đích của một form submit hoặc trong một Hipertext link (thuộc tính TARGET của thẻ FORM và A).

Trong ví dụ tạo ra một tới cửa sổ thứ hai, như nút thứ nhất để mở một cửa sổ rỗng, sau đó một liên kết sẽ tải file doc2.html xuống cửa sổ mới đó rồi một nút khác dùng để đóng cửa sổ thứ hai lại, ví dụ này lưu vào file window.html:

```
<html>

<head>

<title>frame example </title>

</head>

<body>

<form>

<input type="button" value="Open Second Window"
onClick="msgWindow=window.open(,'window2','resizable=no,width=200,height=200')">

<p>

<a href="doc2.html" target="window2"> Load a file into window2 </a>

</p>

<input type="button" value="Close Second Window" onClick="msgWindow.close()">

</form>
```

</body>

</html>

Trình Xử Lý Sự Kiện

- **onLoad** - Xuất hiện khi cửa sổ kết thúc việc tải.
- **onUnload** - Xuất hiện khi cửa sổ được loại bỏ.

5.10.3 ĐỐI TƯỢNG LOCATION

Các thuộc tính của đối tượng location duy trì các thông tin về URL của document hiện thời. Đối tượng này hoàn toàn không có các phương thức và chương trình xử lý sự kiện đi kèm. Ví dụ:

`http:// www.abc.com/ chap1/page2.html#topic3`

Các thuộc tính

- **hash** - Tên anchor của vị trí hiện thời (ví dụ topic3).
- **Host** - Phần hostname:port của URL (ví dụ www.abc.com). Chú ý rằng đây thường là cổng ngầm định và ít khi được chỉ ra.
- **Hostname** - Tên của host và domain (ví dụ www.abc.com).
- **href** - Toàn bộ URL cho document hiện tại.
- **Pathname** - Phần đường dẫn của URL (ví dụ /chap1/page2.html).
- **Port** - Cổng truyền thông được sử dụng cho máy tính host, thường là cổng ngầm định.
- **Protocol** - Giao thức được sử dụng (cùng với dấu hai chấm) (ví dụ http:).
- **Search** - Câu truy vấn tìm kiếm có thể ở cuối URL cho các script CGI.

5.10.4 ĐỐI TƯỢNG FRAME

Một cửa sổ có thể có một vài frame. Các frame có thể cuộn một cách độc lập với nhau và mỗi frame có URL riêng. frame không có các chương trình xử lý sự kiện. Sự kiện onLoad và onUnload là của đối tượng window.

Các Thuộc Tính

- **frames** - Mảng tất cả các frame trong cửa sổ.
- **Name** - Thuộc tính NAME của thẻ <FRAME>
- **Length** - Số lượng các frame con trong một frame.
- **Parent** - Cửa sổ hay frame chứa nhóm frame hiện thời.
- **self** - frame hiện thời.
- **Window** - frame hiện thời

Các Phương Thức

- **clearTimeout (timeoutID)** - Xoá timeout do setTimeout lập. SetTimeout trả lại timeoutID.
- **TimeoutID = setTimeout (expression,msec)** - Đánh giá expression sau khi hết thời gian msec.

Ví dụ:

Tạo một frame

```
<html>
```

```
<head>

<title>Frame Example </title>

<frameset rows="90%,10%">

<frameset cols="30%,70%">

    <frame src=category.htm name="ListFrame">

        <frame src=titles.htm name="contentFrame">

</frameset >

<frame src=navigator.htm name="navigateFrame">

</frameset >

</head>

<body> </body>

</html>
```

5.10.5 ĐỐI TƯỢNG DOCUMENT

Các Thuộc Tính

- **alinkColor** - Giống như thuộc tính ALINK.
- **anchor** - Mảng tất cả các anchor trong document.
- **bgColor** - Giống thuộc tính BGCOLOR.
- **cookie** - Sử dụng để xác định cookie.

- **fgColor** - Giống thuộc tính TEXT.
- **forms** - Mảng tất cả các form trong document.
- **lastModified** - Ngày cuối cùng văn bản được sửa.
- **linkColor** - Giống thuộc tính LINK.
- **links** - Mảng tất cả các link trong document.
- **location** - URL đầy đủ của văn bản.
- **referrer** - URL của văn bản gọi nó.
- **title** - Nội dung của thẻ <TITLE>.
- **vlinkColor** - Giống thuộc tính VLINK

Các Phương Thức

- **document.clear** - Xoá document hiện thời.
- **document.close** - Đóng dòng dữ liệu vào và đưa toàn bộ dữ liệu trong bộ đệm ra màn hình.
- **document.open ("mineType")** - Mở một stream để thu thập dữ liệu vào của các phương thức write và writeln.
- **document.write(expression1 [,expression2]...[,expressionN])** - Viết biểu thức HTML lên văn bản trong một cửa sổ xác định.
- **document.writeln (expression1 [,expression2] ... [,expressionN])** - Giống phương thức trên nhưng khi hết mỗi biểu thức lại xuống dòng.

5.10.6 ĐỐI TƯỢNG ANCHORS

anchor là một đoạn văn bản trong document có thể dùng làm đích cho một siêu liên kết. Nó được xác định bằng cặp thẻ <A> và . Đối tượng anchor không có thuộc tính, phương thức cũng như chương trình xử lý sự kiện. Mảng anchor tham chiếu đến mỗi anchor có tên trong document. Mỗi anchor được tham chiếu bằng cách:

```
document.anchors [index]
```

Mảng anchor có một thuộc tính duy nhất là length xác định số lượng các anchor trong document, nó có thể được xác định như sau:

```
document.anchors.length.
```

5.10.7 ĐỐI TƯỢNG FORMS

Các form được tạo ra nhờ cặp thẻ <FORM> và </FORM>. Phần lớn các thuộc tính của đối tượng form phản ánh các thuộc tính của thẻ <FORM>. Có một vài phần tử (elements) là thuộc tính của đối tượng forms:

Button, checkbox, hidden, password, radio ,reset.select, Submit,text, textarea. Các phần tử này sẽ được trình bày sau.

Nếu document chứa một vài form, chúng có thể được tham chiếu qua mảng forms. Số lượng các form có thể được xác định như sau:

```
document.forms.length.
```

Mỗi một form có thể được tham chiếu như sau:

```
document.forms[index]
```

Các Thuộc Tính

action-thuộc tính ACTION của thẻ FORM.

elements-Mảng chứa tất cả các thành phần trong một form (như checkbox, trường text, danh sách lựa chọn)

encoding-Xâu chứa kiểu MIME được sử dụng để mã hoá nội dung của form gửi cho server.

length-Số lượng các thành phần trong một form.

Method-Thuộc tính METHOD.

target-Xâu chứa tên của cửa sổ đích khi submit form

CácPhươngThức

formName.submit () - Xuất dữ liệu của một form tên formName tới trang xử lý. Phương thức này mô phỏng một click vào nút submit trên form.

CácChươngTrìnhXửLý SựKiện

onSubmit - Chương trình xử lý sự kiện này được gọi khi người sử dụng chuyển dữ liệu từ form đi.

5.10.8 ĐỐI TƯỢNG HISTORY

Đối tượng này được sử dụng để lưu giữ các thông tin về các URL trước được người sử dụng sử dụng. Danh sách các URL được lưu trữ theo thứ tự thời gian. Đối tượng này không có chương trình xử lý sự kiện.

Các Thuộc Tính

length - Số lượng các URL trong đối tượng.

CácPhươngThức

- **history.back()** - Được sử dụng để tham chiếu tới URL mới được thăm trước đây.

- **history.forward()** - Được sử dụng để tham chiếu tới URL kế tiếp trong danh sách. Nó sẽ không gây hiệu ứng gì nếu đã đến cuối của danh sách.

- **history.go (delta | "location")** - Được sử dụng để chuyển lên hay chuyển xuống delta bậc hay di chuyển đến URL xác định bởi location trong danh sách. Nếu delta được sử dụng thì việc dịch chuyển lên phía trên khi delta dương và xuống phía dưới khi delta âm. nếu sử dụng location, URL gần nhất có chứa location là chuỗi con sẽ được tham chiếu.

5.10.9 ĐỐI TƯỢNG LINKS

Đối tượng link là một đoạn văn bản hay một ảnh được xem là một siêu liên kết. Các thuộc tính của đối tượng link chủ yếu xử lý về URL của các siêu liên kết. Đối tượng link cũng không có phương thức nào.

Mảng link chứa danh sách tất cả các liên kết trong document. Có thể xác định số lượng các link qua

```
document.links.length()
```

Có thể tham chiếu tới một liên kết qua

```
document.links [index]
```

Để xác định các thuộc tính của đối tượng link, có thể sử dụng URL tương tự:

```
http://www.abc.com/chap1/page2.html#topic3
```

Các Thuộc Tính

- **hash** - Tên anchor của vị trí hiện thời (ví dụ topic3).
- **Host** - Phần hostname:port của URL (ví dụ www.abc.com). Chú ý rằng đây thường là công ngầm định và ít khi được chỉ ra.
- **Hostname** - Tên của host và domain (ví dụ ww.abc.com).
- **href** - Toàn bộ URL cho document hiện tại.
- **Pathname** - Phần đường dẫn của URL (ví dụ /chap1/page2.html).

- **port** - Cổng truyền thông được sử dụng cho máy tính host, thường là cổng ngầm định.
- **Protocol** - Giao thức được sử dụng (cùng với dấu hai chấm) (ví dụ http:).
- **Search** - Câu truy vấn tìm kiếm có thể ở cuối URL cho các script CGI.
- **Target** - Giống thuộc tính TARGET của <LINK>.

CácChươngTrìnhXửLý SựKiện

- **onClick** - Xảy ra khi người sử dụng nhấn vào link.
- **onMouseOver** - Xảy ra khi con chuột di chuyển qua link.

5.10.10 ĐỐI TƯỢNG MATH

Đối tượng Math là đối tượng nội tại trong JavaScript. Các thuộc tính của đối tượng này chứa nhiều hằng số toán học, các hàm toán học, lượng giác phổ biến. Đối tượng Math không có chương trình xử lý sự kiện.

Việc tham chiếu tới number trong các phương thức có thể là số hay các biểu thức được đánh giá là số hợp lệ.

Các Thuộc Tính

- **E** - Hằng số Euler, khoảng 2,718.
- **LN2** - logarit tự nhiên của 2, khoảng 0,693.
- **LN10** - logarit tự nhiên của 10, khoảng 2,302.
- **LOG2E** - logarit cơ số 2 của e, khoảng 1,442.
- **PI** - Giá trị củ $a \cdot \ln b$ - khoảng 14159.

- SQRT1_2 - Căn bậc 2 của 0,5, khoảng 0,707.
- SQRT2 - Căn bậc 2 của 2, khoảng 1,414.

Các Phương Thức

- Math.abs (number) - Trả lại giá trị tuyệt đối của number.
- Math.acos (number) - Trả lại giá trị arc cosine (theo radian) của number. Giá trị của number phải nằm giữa -1 và 1.
- Math.asin (number) - Trả lại giá trị arc sine (theo radian) của number. Giá trị của number phải nằm giữa -1 và 1.
- Math.atan (number) - Trả lại giá trị arc tan (theo radian) của number.
- Math.ceil (number) - Trả lại số nguyên nhỏ nhất lớn hơn hoặc bằng number.
- Math.cos (number) - Trả lại giá trị cosine của number.
- Math.exp (number) - Trả lại giá trị e^{number} , với e là hằng số Euler.
- Math.floor (number) - Trả lại số nguyên lớn nhất nhỏ hơn hoặc bằng number.
- Math.log (number) - Trả lại logarit tự nhiên của number.
- Math.max (num1,num2) - Trả lại giá trị lớn nhất giữa num1 và num2
- Math.min (num1,num2) - Trả lại giá trị nhỏ nhất giữa num1 và num2.
- Math.pos (base,exponent) - Trả lại giá trị base lũy thừa exponent.
- Math.random (r) - Trả lại một số ngẫu nhiên giữa 0 và 1. Phwong thức này chỉ thực hiện được trên nền tảng UNIX.

- Math.round (number) - Trả lại giá trị của number làm tròn tới số nguyên gần nhất.
- Math.sin (number) - Trả lại sin của number.
- Math.sqrt (number) - Trả lại căn bậc 2 của number.
- Math.tan (number) - Trả lại tang của number.

5.10.11 ĐỐI TƯỢNG DATE

Đối tượng Date là đối tượng có sẵn trong JavaScript. Nó cung cấp nhiều phương thức có ích để xử lý về thời gian và ngày tháng. Đối tượng Date không có thuộc tính và chương trình xử lý sự kiện.

Phần lớn các phương thức date đều có một đối tượng Date đi cùng. Các phương thức giới thiệu trong phần này sử dụng đối tượng Date dateVar, ví dụ:

```
dateVar = new Date ('August 16, 1996 20:45:04');
```

Các Phương Thức

- dateVar.getDate() - Trả lại ngày trong tháng (1-31) cho dateVar.
- dateVar.getDay() - Trả lại ngày trong tuần (0=chủ nhật,...6=thứ bảy) cho dateVar.
- dateVar.getHours() - Trả lại giờ (0-23) cho dateVar.
- dateVar.getMinutes() - Trả lại phút (0-59) cho dateVar.
- dateVar.getSeconds() - Trả lại giây (0-59) cho dateVar.
- dateVar.getTime() - Trả lại số lượng các mili giây từ 00:00:00 ngày 1/1/1970.
- dateVar.getTimeZoneOffset() - Trả lại độ dịch chuyển bằng phút của giờ địa phương hiện tại so với giờ quốc tế GMT.

- `dateVar.getYear()`-Trả lại năm cho `dateVar`.
- `Date.parse (dateStr)` - Phân tích chuỗi `dateStr` và trả lại số lượng các mili giây tính từ 00:00:00 ngày 01/01/1970.
- `dateVar.setDay(day)` - Đặt ngày trong tháng là `day` cho `dateVar`.
- `dateVar.setHours(hours)` - Đặt giờ là `hours` cho `dateVar`.
- `dateVar.setMinutes(minutes)` - Đặt phút là `minutes` cho `dateVar`.
- `dateVar.setMonths(months)` - Đặt tháng là `months` cho `dateVar`.
- `dateVar.setSeconds(seconds)` - Đặt giây là `seconds` cho `dateVar`.
- `dateVar.setTime(value)` - Đặt thời gian là `value`, trong đó `value` biểu diễn số lượng mili giây từ 00:00:00 ngày 01/01/1970.
- `dateVar.setYear(years)` - Đặt năm là `years` cho `dateVar`.
- `dateVar.toGMTString()` - Trả lại chuỗi biểu diễn `dateVar` dưới dạng GMT.
- `dateVar.toLocaleString()`-Trả lại chuỗi biểu diễn `dateVar` theo khu vực thời gian hiện thời.
- `Date.UTC (year, month, day [,hours] [,minutes] [,seconds])` - Trả lại số lượng mili giây từ 00:0

5.10.12 ĐỐI TƯỢNG STRING

Đối tượng `String` là đối tượng được xây dựng nội tại trong JavaScript cung cấp nhiều phương thức thao tác trên chuỗi. Đối tượng này có thuộc tính duy nhất là độ dài (`length`) và không có chương trình xử lý sự kiện.

CácPhươngThức

- `str.anchor(name)` - Được sử dụng để tạo ra thẻ `<A>` (một cách động). Tham số `name` là thuộc tính `NAME` của thẻ `<A>`.
- `str.big()` - Kết quả giống như thẻ `<BIG>` trên chuỗi `str`.
- `str.blink()` - Kết quả giống như thẻ `<BLINK>` trên chuỗi `str`.
- `str.bold()` - Kết quả giống như thẻ `<BOLD>` trên chuỗi `str`.
- `str.charAt(a)` - Trả lại ký tự thứ `a` trong chuỗi `str`.
- `str.fixed()` - Kết quả giống như thẻ `<TT>` trên chuỗi `str`.
- `str.fontcolor()` - Kết quả giống như thẻ `<FONTCOLOR = color>`.
- `str.fontSize(size)` - Kết quả giống như thẻ `<FONTSIZE = size>`.
- `str.indexOf(srchStr [,index])` - Trả lại vị trí trong chuỗi `str` vị trí xuất hiện đầu tiên của chuỗi `srchStr`. Chuỗi `str` được tìm từ trái sang phải. Tham số `index` có thể được sử dụng để xác định vị trí bắt đầu tìm kiếm trong chuỗi.
- `str italics()` - Kết quả giống như thẻ `<I>` trên chuỗi `str`.
- `str.lastIndexOf(srchStr [,index])` - Trả lại vị trí trong chuỗi `str` vị trí xuất hiện cuối cùng của chuỗi `srchStr`. Chuỗi `str` được tìm từ phải sang trái. Tham số `index` có thể được sử dụng để xác định vị trí bắt đầu tìm kiếm trong chuỗi.
- `str.link(href)` - Được sử dụng để tạo ra một kết nối HTML động cho chuỗi `str`. Tham số `href` là URL đích của liên kết.
- `str.small()` - Kết quả giống như thẻ `<SMALL>` trên chuỗi `str`.
- `str.strike()` - Kết quả giống như thẻ `<STRIKE>` trên chuỗi `str`.
- `str.sub()` - Tạo ra một subscript cho chuỗi `str`, giống thẻ `<SUB>`.

- `str.substring(a,b)` - Trả lại chuỗi con của `str` là các ký tự từ vị trí thứ `a` tới vị trí thứ `b`. Các ký tự được đếm từ trái sang phải bắt đầu từ 0.
- `str.sup()` - Tạo ra superscript cho chuỗi `str`, giống thẻ `<SUP>`.
- `str.toLowerCase()` - Đổi chuỗi `str` thành chữ thường.
- `str.toUpperCase()` - Đổi chuỗi `str` thành chữ hoa.0:00 01/01/1970 GMT.

5.11. Các phần tử của đối tượng form

Form được tạo bởi các phần tử cho phép người sử dụng đưa thông tin vào. Khi đó, nội dung (hoặc giá trị) của các phần tử sẽ được chuyển đến một chương trình trên server qua một giao diện được gọi là Common Gateway Interface(Giao tiếp qua một cổng chung) gọi tắt là CGI

Sử dụng JavaScript bạn có thể viết những đoạn scripts chèn vào HTML của bạn để làm việc với các phần tử của form và các giá trị của chúng.

Phần tử	Mô tả
button	Là một nút bấm hơn là nút submit hay nút reset
checkbox	Một checkbox (<code><input type="checkbox"></code>)
FileUpload	Là một phần tử tải file lên cho phép người sử dụng gửi lên một file (<code><input type="file"></code>)
hidden	Một trường ẩn (<code><input type="hidden"></code>)
password	Một trường text để nhập mật khẩu mà tất cả các ký tự nhập vào đều hiển thị là dấu (*) (<code><input type="password"></code>)
radio	Một nút bấm (<code><input type="radio"></code>)

reset	Một nút reset(<input type="reset">)
select	Một danh sách lựa chọn (<select><option>option1 </option>)
submit	Một nút submit (<input type="submit">)
text	Một trường text (<input type="text">)
textArea	Một trường text cho phép nhập vấp nhiều dòng

5.11.1 THUỘC TÍNH TYPE

Trong mỗi phần tử của form đều có thuộc tính type, đó là một xâu chỉ định rõ kiểu của phần tử được đưa vào như nút bấm, một trường text hay một checkbox...

Xâu đó có thể là một trong các giá trị sau: Text field: "text"

Radio button: "radio"

Checkbox: "checkbox" Hidden field: "hidden" Submit button: "submit" Reset button: "reset"

Password field: "password" Button: "button"

Select list: "select-one"

Multiple select lists: "select-multiple" Textarea field: "textarea"

5.11.2 PHẦN TỬ BUTTON

Trong một form HTML chuẩn, chỉ có hai nút bấm có sẵn là submit và reset bởi vì dữ liệu trong form phải được gửi tới một vài địa chỉ URL (thường là CGI-BIN script) để xử lý và lưu trữ.

Một phần tử button được chỉ định rõ khi sử dụng thẻ INPUT:

```
<input type="button" name="name" value="buttonname">
```

Trong thẻ INPUT, name là tên của button, thuộc tính VALUE có chứa nhãn của button sẽ được hiển thị trên Navigator của browser.

Chỉ có một thẻ sự kiện duy nhất đối với button là onClick. Kết hợp với nó là cách thức duy nhất click. Phần tử button có khả năng mở rộng cho phép người lập trình JavaScript có thể viết được một đoạn mã lệnh JavaScript để thực thi việc thêm vào một nút bấm trong một script.

Trong ví dụ sau, thay vì sử dụng onChange, bạn có thể chỉnh sửa script để định giá biểu thức khi button được bấm.

Ví dụ: Định giá một form sử dụng phần tử button.

```
<html>

<head>

<title>button example</title>

<script language="JavaScript">

<!-- HIDE FROM OTHER BROWSERS

function calculate(form) {

form.results.value = eval(form.entry.value);

}

// STOP HIDING FROM OTHER BROWSERS -->

</script>
```

```
</head>
```

```
<body>
```

```
<form method=post>
```

Enter a JavaScript mathematical expression:

```
<input type="text" name="entry" value="">
```

```
<br>
```

the result of this expression is:

```
<input type=text name="results" onFocus="this.blur();">
```

```
<br>
```

```
<input type="button" value="calculate" onClick="calculate(this.form);">
```

```
</form>
```

```
</body>
```

```
</html>
```

5.11.3 PHẦN TỬ CHECKBOX

Các phần tử checkbox có khả năng bật tắt dùng để chọn hoặc không chọn một thông tin. Các checkbox có nhiều thuộc tính và cách thức hơn button. Bảng dưới đây là danh sách các thuộc tính và các cách thức của phần tử checkbox.

Cách thức và thuộc tính	Mô tả
checked	Cho biết trạng thái hiện thời của checkbox (thuộc tính)

defaultChecked	Cho biết trạng thái mặc định của phần tử (thuộc tính)
name	Cho biết tên của phần tử được chỉ định trong thẻ INPUT (thuộc tính)
value	Cho biết giá trị hiện thời của phần tử được chỉ định trong thẻ INPUT (thuộc tính)
click()	Mô tả một click vào checkbox (Cách thức)

Phần tử checkbox chỉ có một thẻ sự kiện là onClick

Ví dụ: Tạo hộp checkbox để nhập vào một số rồi lựa chọn tính nhân đôi và bình phương:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>checkbox Example</TITLE>
```

```
<SCRIPT>
```

```
<!-- HIDE FROM OTHER BROWSERS
```

```
function calculate(form,callingField) {
```

```
if (callingField == "result") { // if(1)
```

```
if (form.square.checked) { // if(2)
```

```
    form.entry.value = Math.sqrt(form.result.value);
```

```
    }
```

```
else {
```

```
    form.entry.value = form.result.value / 2;
```

```
        } //end if(2)

    }

else{

    if (form.square.checked) {    // if(3)

        form.result.value=form.entry.value*form.entry.value;

    }

    else {

        form.result.value = form.entry.value * 2;

    } //enfzd if(3)

} //end if(1)

} //end function

// STOP HIDING FROM OTHER BROWSERS -->

</SCRIPT>

</HEAD>

<BODY>

<FORM METHOD=POST>

Value: <INPUT TYPE="text" NAME="entry" VALUE=0
onChange="calculate(this.form,this.name);">

<BR>
```

```
Action (default double): <INPUT TYPE=checkbox NAME=square
    onClick="calculate(this.form,this.name);">
```

Square

```
<BR>
```

```
Result: <INPUT TYPE="text" NAME="result" VALUE=0
    onChange="calculate(this.form,this.name);">
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

Trong script này, bạn đã thấy cách sử dụng thẻ sự kiện `onClick` cũng như thuộc tính `checked` là một giá trị kiểu Boolean có thể dùng làm điều kiện trong câu lệnh `if...else`

Bạn có thể thêm một checkbox tên là `square` vào form. Nếu hộp này được check, chương trình sẽ lấy giá trị của nó, nếu không, một thực thi được mặc định sẽ nhân đôi giá trị của nó. Thẻ sự kiện `onClick` trong checkbox được định nghĩa:

```
(<INPUT TYPE=checkbox NAME=square onClick= "calculate( this.form, this.name);">)
```

Khi đó nếu người dùng thay đổi một câu lệnh khác, form sẽ được tính toán lại.

Để tạo ra sự mở rộng cho checkbox, bạn có thể thay đổi hàm `calculate()` như sau:

```
function calculate(form,callingField) {
    if (callingField == "result") { // if (1)
        if (form.square.checked) { // if (2)
            form.entry.value = Math.sqrt(form.result.value);
        }
    }
}
```

```
else {  
  
form.entry.value = form.result.value / 2;  
  
} //end if(2)  
  
}  
  
else {  
  
if (form.square.checked) { // if (3)  
  
form.result.value=form.entry.value*form.entry.value;  
}  
  
else {  
  
form.result.value = form.entry.value * 2;  
  
} // end if (3)  
  
} // end if (1)  
  
}
```

5.11.4 PHẦN TỬ FILE UPLOAD

Phần tử này cung cấp cho form một cách để người sử dụng có thể chỉ rõ một file đưa vào form xử lý. Phần tử file Upload được chỉ định rõ trong JavaScript bằng đối tượng FileUpload.

Đối tượng chỉ có hai thuộc tính là name và value, cả hai đều là giá trị xâu như các đối tượng khác. Không có cách thức hay thẻ file cho đối tượng này.

5.11.5 PHẦN TỬ HIDDEN

Phần tử hidden là phần tử duy nhất trong số tất cả các phần tử của form không được hiển thị trên Web browser. Trường hidden có thể sử dụng để lưu các giá trị cần thiết để gửi tới

server song song với sự xuất ra từ form (form submission) nhưng nó không được hiển thị trên trang. Mọi người có thể sử dụng trong JavaScript để lưu các giá trị trong suốt một script và để tính toán không cần form.

Đối tượng hidden chỉ có hai thuộc tính là name và value, đó cũng là những giá trị xâu giống các đối tượng khác. Không có cách thức hay thẻ sự kiện nào cho đối tượng này.

5.11.6 PHẦN TỬ PASSWORD

Đối tượng Password là đối tượng duy nhất trong các đối tượng của form mà khi gõ bất kỳ ký tự nào vào cũng đều hiển thị dấu sao(*). Nó cho phép đưa vào những thông tin bí mật như đăng ký mật khẩu...

Đối tượng Password có 3 thuộc tính giống trường text là: defaultValue, name và value. Không giống với hai phần tử ở trên, trường Password có nhiều cách thức hơn(focus(), blur(), and select()) và tương ứng với các thẻ sự kiện: onFocus, onBlur, and onSelect.

Phần này sẽ được nói kỹ hơn trong đối tượng text.

5.11.7 PHẦN TỬ RADIO

Đối tượng radio gần giống sự bật tắt checkbox khi có hai nút radio kết hợp thành một nhóm. Khi nhiều radio được kết hợp thành một nhóm, chỉ có một nút được chọn trong bất kỳ một thời điểm nào. Ví dụ dòng lệnh sau tạo ra một nhóm radio có ba

nút tên là test:

```
<INPUT TYPE="radio" NAME="test" VALUE="1" checked>1<BR>
<INPUT TYPE="radio" NAME="test" VALUE="2">2<BR>
<INPUT TYPE="radio" NAME="test" VALUE="3">3<BR>
```

Nhóm các nút radio lại bằng cách đặt cho chúng có cùng một tên trong các thẻ INPUT.

Có một vài thuộc tính để kiểm tra trạng thái hiện thời của một nhóm nút radio.

Bảng sau hiển thị các thuộc tính và cách thức của đối tượng radio.

Thuộc tính và cách thức	Mô tả
checked	Mô tả trạng thái hiện thời của phần tử radio (thuộc tính)
defaultChecked	Mô tả trạng thái mặc định của phần tử (thuộc tính)
index	Mô tả thứ tự của nút radio được chọn hiện thời trong một nhóm
length	Mô tả tổng số nút radio trong một nhóm
name	Mô tả tên của phần tử được chỉ định trong thẻ INPUT (thuộc tính)
value	Mô tả giá trị hiện thời của phần tử được định ra trong thẻ INPUT (thuộc tính)
click()	Mô phỏng một click trên nút radio (cách thức)

Cũng như checkbox, radio chỉ có một thẻ sự kiện là onClick.

Không có bất kỳ một đối tượng form nào có thuộc tính index và length. Do một nhóm radio gồm nhiều phần tử radio, nên chúng được đặt trong một mảng các nút radio và được đánh số từ 0. Trong ví dụ nhóm radio có tên test ở trên, nếu nhóm đó nằm trong một form có tên là "testform", bạn có thể gọi tới nút radio thứ hai bằng tên "testform.test[1]" và có thể kiểm tra giá trị của nó bằng "testform.test[1].checked"

Để minh họa rõ cách dùng đối tượng radio, ta xem ví dụ sau: Ví dụ:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>radio button Example</TITLE>
```

```
<SCRIPT>
```

```
<!-- HIDE FROM OTHER BROWSERS
```

```
function calculate(form,callingField) {
```

```
    if (callingField == "result") {
```

```
        if (form.action[1].checked) {  
form.entry.value = Math.sqrt(form.result.value);  
  
        } else {  
        form.entry.value = form.result.value / 2;  
        }  
    } else {  
        if (form.action[1].checked) {  
        form.result.value=form.entry.value*form.entry.value;  
        } else {  
        form.result.value = form.entry.value * 2;  
        }  
    }  
}
```

```
// STOP HIDING FROM OTHER BROWSERS -->
```

```
</SCRIPT>
```

```
</HEAD>
```

```
<BODY>
```

```
<FORM METHOD=POST>
```

```
Value: <INPUT TYPE="text" NAME="entry" VALUE=0
      onChange="calculate(this.form,this.name);" > <BR>
```

```
Action:<BR>
```

```
<INPUT TYPE="radio" NAME="action" VALUE="twice"
      onClick="calculate(this.form,this.name);" > Double<BR>
```

```
<INPUT TYPE="radio" NAME="action" VALUE="square"
      onClick="calculate(this.form,this.name);" > Square <BR> Result: <INPUT TYPE=text
NAME="result" VALUE=0
      onChange="calculate(this.form,this.name);" >
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

Trong ví dụ này, sự thay đổi từ checkbox ở trên là rất khó nhận biết. Thay cho một checkbox trong ví dụ trước, ở đây ta sử dụng hai nút radio với hai giá trị khác nhau: double và square

Như ta đã biết có thể truy nhập đến các nút radio qua một mảng, do đó hai nút này có thể truy nhập bằng `action[0]` và `action[1]`. Bằng cách này, bạn chỉ cần thay

đổi tham chiếu đến hàm calculate() từ form.square.checked thành form.action[1].checked.

5.11.8 PHẦN TỬ RESET

Sử dụng đối tượng reset, bạn có thể tác động ngược lại để click vào nút Reset. Cũng giống đối tượng button, đối tượng reset có hai thuộc tính là name và value, và một cách thức click(), một thẻ sự kiện onClick.

Hầu hết những người lập trình không sử dụng thẻ sự kiện onClick của nút reset để kiểm tra giá trị của nút này, đối tượng reset thường dùng để xoá form.

Ví dụ sau minh hoạ cách sử dụng nút reset để xoá các giá trị của form. Ví dụ:

```
<HTML>

<HEAD>

<TITLE>reset Example</TITLE>

<SCRIPT LANGUAGE="JavaScript">

<!-- HIDE FROM OTHER BROWSERS function clearForm(form) { form.value1.value = "Form";
    form.value2.value = "Cleared";
}

// STOP HIDING FROM OTHER BROWSERS -->

</SCRIPT>

</HEAD>
```

```
<BODY>

<FORM METHOD=POST>

<INPUT TYPE="text" NAME="value1"><BR>

<INPUT TYPE="text" NAME="value2"><BR>

<INPUT TYPE="reset" VALUE="Clear Form" onClick="clearForm(this.form);">

</FORM>

</BODY>

</HTML>
```

5.11.9 PHẦN TỬ SELECT

Danh sách lựa chọn trong các form HTML xuất hiện menu drop-down hoặc danh sách cuộn được của các đối tượng có thể được lựa chọn. Các danh sách được xây dựng bằng cách sử dụng hai thẻ SELECT và OPTION. Ví dụ:

```
<SELECT NAME="test">

    <OPTION SELECTED>1

    <OPTION>2

    <OPTION>3

</SELECT>
```

tạo ra ba thành phần của menu thả drop-down với ba lựa chọn 1,2 và 3. Sử dụng thuộc tính SIZE bạn có thể tạo ra một danh sách cuộn với số phần tử hiển thị ở lần thứ nhất. Để bật menu drop-down trong một menu cuộn với hai thành phần hiển thị, bạn có thể sử dụng như sau:

```
<SELECT NAME="test" SIZE=2>
```

```
<OPTION SELECTED>1
```

```
<OPTION>2
```

```
<OPTION>3
```

```
</SELECT>
```

Trong cả hai ví dụ trên, người sử dụng chỉ có thể có một lựa chọn. Nếu sử dụng thuộc tính MULTIPLE, bạn có thể cho phép người sử dụng lựa chọn nhiều hơn một giá trị trong danh sách lựa chọn:

```
<SELECT NAME="test" SIZE=2 MULTIPLE>
```

```
<OPTION SELECTED>1
```

```
<OPTION>2
```

```
<OPTION>3
```

```
</SELECT>
```

Danh sách lựa chọn trong JavaScript là đối tượng select. Đối tượng này tạo ra một vài thành phần tương tự các button và radio.

Với các thành phần lựa chọn, danh sách các lựa chọn được chứa trong một mảng được đánh số từ 0. Trong trường hợp này, mảng là một thuộc tính của đối tượng select gọi là options.

Cả việc lựa chọn các option và từng phần tử option riêng biệt đều có những thuộc tính. Bổ sung thêm vào mảng option, phần tử select có thuộc tính selectedIndex, có chứa số thứ tự của option được lựa chọn hiện thời.

Mỗi option trong danh sách lựa chọn đều có một vài thuộc tính:

- **DEFAULTSELECTED**: cho biết option có được mặc định là lựa chọn trong thẻ **OPTION** hay không.
- **INDEX**: chứa giá trị số thứ tự của option hiện thời trong mảng option.
- **SELECTED**: cho biết trạng thái hiện thời của option
- **TEXT**: có chứa giá trị của dòng text hiển thị trên menu cho mỗi option, và thuộc tính value mọi giá trị chỉ ra trong thẻ **OPTION**.

Đối tượng select không có các cách thức được định nghĩa sẵn. Tuy nhiên, đối tượng select có ba thẻ sự kiện, đó là onBlue, onFocus, onChange, chúng đều là những đối tượng text.

Ví dụ bạn có danh sách lựa chọn sau:

```
<SELECT NAME="example" onFocus="react();">
```

```
  <OPTION SELECTED VALUE="Number One">1
```

```
  <OPTION VALUE="The Second">2
```

```
  <OPTION VALUE="Three is It">3
```

```
</SELECT>
```

Khi lần đầu tiên hiển thị bạn có thể truy nhập tới các thông tin sau:

```
example.options[1].value = "The Second" example.options[2].text = "3" example.selectedIndex = 0
example.options[0].defaultSelected = true example.options[1].selected = false
```

Nếu người sử dụng kích vào menu và lựa chọn option thứ hai, thì thẻ onFocus sẽ thực hiện, và khi đó giá trị của thuộc tính sẽ là:

```
example.options[1].value = "The Second" example.options[2].text = "3" example.selectedIndex
= 1 example.options[0].defaultSelected = true example.options[1].selected = true
```

Sửa các danh sách lựa chọn

Navigator 3.0 cho phép thay đổi nội dung của danh sách lựa chọn từ JavaScript bằng cách liên kết các giá trị mới cho thuộc tính text của các thực thể trong danh sách.

Ví dụ, trong ví dụ trước, bạn đã tạo ra một danh sách lựa chọn như sau:

```
<SELECT NAME="example" onFocus="react();">
  <OPTION SELECTED VALUE="Number One">1
  <OPTION VALUE="The Second">2
  <OPTION VALUE="Three is It">3
</SELECT>
```

Có thể thay đổi được dòng text hiển thị trên nút thứ hai thành "two" bằng:

```
example.options[1].text = "two";
```

Có thể thêm các lựa chọn mới vào danh sách bằng cách sử dụng đối tượng xây dựng Option() theo cú pháp:

```
newOptionName = new Option(optionText, optionValue, defaultSelected, selected);
```

```
selectListName.options[index] = newOptionName;
```

Việc tạo đối tượng option() này với dòng text được chỉ trước, defaultSelected và selected như trên đã định ra những giá trị kiểu Boolean. Đối tượng này được liên kết vào danh sách lựa chọn được thực hiện bằng index.

Các lựa chọn có thể bị xoá trong danh sách lựa chọn bằng cách gán giá trị null cho đối tượng muốn xoá

```
selectListName.options[index]= null;
```

5.11.10 Phần tử submit

Nút Submit là một trường hợp đặc biệt của button, cũng như nút Reset. Nút này đưa thông tin hiện tại từ các trường của form tới địa chỉ URL được chỉ ra trong thuộc tính ACTION của thẻ form sử dụng cách thức METHOD chỉ ra trong thẻ FORM.

Giống như đối tượng button và reset, đối tượng submit có sẵn thuộc tính name và value, cách thức click() và thẻ sự kiện onClick.

5.11.11 Phần tử Text

Phần tử này nằm trong những phần tử hay được sử dụng nhất trong các form HTML. Tương tự như trường Password, trường text cho phép nhập vào một dòng đơn, nhưng các ký tự của nó hiện ra bình thường.

Đối tượng text có ba thuộc tính: defaultValue, name và value. Ba cách thức mô phỏng sự kiện của người sử dụng: focus(), blur() và select(). Có 4 thẻ sự kiện là: onBlur, onFocus, onChange, onSelect. Chú ý các sự kiện này chỉ thực hiện khi con trỏ đã được kích ra ngoài trường text.

Bảng sau mô tả các thuộc tính và cách thức của đối tượng text.

Cách thức và thuộc tính	Mô tả
defaultValue	Chỉ ra giá trị mặc định của phần tử được chỉ ra trong thẻ INPUT (thuộc tính)
name	Tên của đối tượng được chỉ ra trong thẻ INPUT (thuộc tính)
value	Giá trị hiện thời của phần tử (thuộc tính)
focus()	Mô tả việc con trỏ tới trường text (cách thức)
blur()	Mô tả việc con trỏ rời trường text (cách thức)
select()	Mô tả việc lựa chọn dòng text trong trường text (cách thức)

Một chú ý quan trọng là có thể gán giá trị cho trường text bằng cách liên kết các giá trị với thuộc tính value. Trong ví dụ sau đây, dòng text được đưa vào trường đầu tiên được lặp lại trong trường text thứ hai, và mọi dòng text được đưa vào trường text thứ hai lại được lặp lại trong trường text thứ nhất. Khả năng này của nó có thể áp dụng để tự động cập nhật hoặc thay đổi dữ liệu.

Ví dụ. Tự động cập nhật các trường text .

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>text Example</TITLE>
```

```
<SCRIPT LANGUAGE="JavaScript">
```

```
<!-- HIDE FROM OTHER BROWSERS
```

```
function echo(form,currentField) {
```

```

        if (currentField == "first")

form.second.value = form.first.value;

        else

        form.first.value = form.second.value;

    }

// STOP HIDING FROM OTHER BROWSERS -->

</SCRIPT>

</HEAD>

<BODY>

<FORM>

<INPUT TYPE=text NAME="first" onChange="echo(this.form,this.name);">

<INPUT TYPE=text NAME="second" onChange="echo(this.form,this.name);">

</FORM>

</BODY>

</HTML>

```

5.11.12 Phần tử Textarea

Thẻ TEXTAREA cung cấp một hộp cho phép nhập số dòng text do người thiết kế định trước. Ví dụ:

```
<TEXTAREA NAME="fieldName" ROWS=10 COLS=25> Default Text Here
```

```
</TEXTAREA>
```

ví dụ này tạo ra một trường text cho phép đưa vào 10 hàng ,mỗi hàng 25 ký tự. Dòng "Default Text Here" sẽ xuất hiện trong trường này vào lần hiển thị đầu tiên.

Cũng như phần tử text , JavaScript cung cấp cho bạn các thuộc tính defaultValue, name, và value, các cách thức focus(), select(), và blur(), các thẻ sự kiện onBlur, onFocus, onChange, onSelect.

5.11.13 Mảng elements[]

Các đối tượng của form có thể được gọi tới bằng mảng elements[]. Ví dụ bạn tạo ra một form sau:

```
<FORM METHOD=POST NAME=testform>
```

```
  <INPUT TYPE="text" NAME="one">
```

```
  <INPUT TYPE="text" NAME="two">
```

```
  <INPUT TYPE="text" NAME="three">
```

```
</FORM>
```

bạn có thể gọi tới ba thành phần này như sau: document.elements[0], document.elements[1], document.elements[2], hơn nữa còn có thể gọi document.testform.one, document.testform.two,

document.testform.three.

Thuộc tính này thường được sử dụng trong các mối quan hệ tuần tự của các phần tử hơn là dùng tên của chúng.

5.11.14 Mảng form[]

Các thẻ sự kiện được thiết kế để làm việc với các form riêng biệt hoặc các trường ở một thời điểm, nó rất hữu dụng để cho phép gọi tới các form có liên quan trong cùng một trang.

Mảng form[] đề cập đến ở đây có thể có nhiều xác định các nhân của form trên cùng một trang và have information in a single field match in all three forms. Có thể gọi bằng document.forms[] thay vì gọi bằng tên form. Trong script này, bạn có hai trường text để nhập và nằm trên hai form độc lập với nhau. Sử dụng mảng form bạn có thể tương tác trên các giá trị của các trường trong hai form cùng một lúc khi người sử dụng thay đổi giá trị trên một form.

```
<HTML>

<HEAD>

<TITLE>forms[] Example</TITLE>

</HEAD>

<BODY>

<FORM METHOD=POST>

  <INPUT TYPE=text onChange="document.forms[1].elements[0].value =
this.value;">

</FORM>

<FORM METHOD=POST>

  <INPUT TYPE=text onChange="document.forms[0].elements[0].value =
this.value;">

</FORM>
```

</BODY>

</HTML>

Mặt khác, cũng có thể truy nhập đến form bằng tên form được đặt trong thẻ FORM:

```
<FORM METHOD=POST NAME="name">
```

Khi đó bạn có thể gọi là `document.forms["name"]` hoặc `document.name`

5.12. JQuery cơ bản

5.12.1 JQuery là gì?

- JQuery là một thư viện các hàm JavaScript.
- JQuery là một thư viện JavaScript “write less, do more” (“viết ít hơn, làm nhiều hơn”).

Thư viện JQuery có chứa các tính năng sau đây:

- Lựa chọn các thẻ HTML.
- Thao tác trên thẻ HTML.
- Thao tác trên CSS.
- Các hàm sự kiện HTML.
- Hiệu ứng và chuyển động JavaScript.
- Duyệt và hiệu chỉnh dữ liệu HTML.
- AJAX.
- Các tiện ích khác.

5.12.2 Chèn thư viện JQuery vào trang web

Các thư viện jQuery được lưu trữ trong một tập tin duy nhất, chứa tất cả các phương thức của jQuery.

Nó có thể thêm vào trang web với dòng đánh dấu sau:

```
<head>
```

```
<script type="text/javascript" src=[DuongDan file jQuery]></script>
```

```
</head>
```

Xin lưu ý rằng thẻ `<script>` nên đặt trong phần `<head>` của trang. Có thể download tập tin jQuery từ trang: <http://jquery.com/>

5.12.3 Cú pháp sử dụng jQuery

```
$(selector).action()
```

Kí hiệu \$ (đồng đô la) để xác định jQuery.

selector để “truy vấn (hoặc tìm)” các thẻ HTML.

action là một hàm dùng để thực hiện hành động trên các thẻ. Ví dụ:

```
$(this).hide() - ẩn thẻ hiện tại
```

```
$("p").hide() - ẩn tất cả đoạn văn
```

```
$("p.test").hide() - ẩn tất cả đoạn văn có class="test"
```

```
$("#test").hide() - ẩn thẻ có id="test"
```

5.12.4 Hàm `document.ready()`

Các phương thức của jQuery nên được đặt trong hàm `document.ready()`.

```
$(document).ready(function(){  
  
    // các hàm jQuery đặt tại đây...  
  
});
```

Điều này ngăn cản các lệnh jQuery chạy trước khi trang được tải xong (đã sẵn sàng).

Dưới đây là một số ví dụ về các hành động có thể thất bại nếu các chức năng được chạy trước khi tài liệu được nạp đầy đủ:

- Ấn một thẻ không tồn tại.
- Lấy kích thước của một hình mà chưa được tải xong.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="jquery.min.js">
```

```
</script>
```

```
<script>
```

```
$(document).ready(function(){
```

```
    $("button").click(function(){
```

```
        $("p").hide();
```

```
});  
  
});  
  
</script>  
  
</head>  
  
<body>  
  
<h2>This is a heading</h2>  
  
<p>This is a paragraph.</p>  
  
<p>This is another paragraph.</p>  
  
<button>Click me</button>  
  
</body>  
  
</html>
```

5.12.5 Các jQuery Selector

jQuery selector cho phép chọn và thao tác trên một hay một nhóm các thẻ HTML.

jQuery selector cho phép chọn các thẻ HTML (hoặc một nhóm các thẻ) theo tên thẻ, tên thuộc tính, hoặc theo nội dung.

5.12.5.1 jQuery Element Selector

jQuery sử dụng CSS selector để chọn các thẻ HTML. Ví dụ:

`$("p")` chọn tất cả thẻ `<p>`

`$("p.intro")` chọn tất cả thẻ `<p>` có `class="intro"`.

`$("#demo")` chọn tất cả thẻ `<p>` có `id="demo"`.

5.12.5.2 jQuery Attribute Selector

jQuery sử dụng biểu thức XPath để chọn thẻ với các thuộc tính nhất định.

`$("[href]")` chọn tất cả thẻ có thuộc tính `href`.

`$("[href='#']")` chọn tất cả thẻ có thuộc tính `href` là `"#"`.

`$("[href!='#']")` chọn tất cả thẻ có thuộc tính `href` khác `"#"`.

`$("[href$='.jpg']")` chọn tất cả thẻ có thuộc tính `href` kết thúc bởi `".jpg"`.

5.12.5.3 jQuery CSS Selector

jQuery CSS Selector có thể được sử dụng để thay đổi thuộc tính CSS cho thẻ HTML.

Ví dụ sau đây đổi `background-color` của tất cả các thẻ `<p>` sang màu vàng:

```
$("p").css("background-color","yellow");
```

5.12.6 Một số ví dụ

Cú pháp	Mô tả
---------	-------

<code>\$(this)</code>	Thẻ HTML. hiện tại
<code>\$("n")</code>	Tất cả thẻ <code><n></code>
<code>\$("n intro")</code>	Tất cả thẻ <code><n></code> có <code>class="intro"</code>
<code>\$("n#intro")</code>	Tất cả thẻ <code><n></code> có <code>id="intro"</code>
<code>\$("n#intro:first")</code>	Thẻ <code><n></code> đầu tiên có <code>id="intro"</code>
<code>\$(" intro")</code>	Tất cả thẻ có <code>class="intro"</code>
<code>\$("#intro")</code>	Thẻ đầu tiên có <code>id="intro"</code>
<code>\$("ul li:first")</code>	Thẻ <code></code> đầu tiên của mỗi thẻ <code></code>
<code>\$("[href\$=' ino'l"])</code>	Tất cả thẻ có thuộc tính <code>href</code> kết thúc bởi <code>" ino"</code>
<code>\$("div#intro .head")</code>	Tất cả thẻ có <code>class="head"</code> trong một thẻ <code><div></code> có <code>id="intro"</code>

5.12.7 Bảng tham chiếu jQuery Selector

Selector	Example	Selects
<i>*</i>	<code>\$("*")</code>	Tất cả thẻ
<i>#id</i>	<code>\$("#lastname")</code>	Thẻ với <code>id=lastname</code>
<i>.class</i>	<code>\$(".intro")</code>	Tất cả thẻ với <code>class="intro"</code>
<i>element</i>	<code>\$("n")</code>	Tất cả thẻ <code>n</code>
<i>.class.class</i>	<code>\$(".intro.demo")</code>	Tất cả thẻ với <code>class</code> là <code>"intro"</code> và <code>"demo"</code>
<i>:first</i>	<code>\$("n:first")</code>	Thẻ <code>n</code> đầu tiên
<i>:last</i>	<code>\$("n:last")</code>	Thẻ <code>n</code> cuối cùng
<i>:even</i>	<code>\$("tr:even")</code>	Tất cả thẻ <code>tr</code> ở vị trí chẵn
<i>:odd</i>	<code>\$("tr:odd")</code>	Tất cả thẻ <code>tr</code> ở vị trí lẻ
<i>:ea(index)</i>	<code>\$("ul li:ea(3)")</code>	Thẻ thứ 4 trong một danh sách (đánh số từ 0)

<code>:gt(<i>no</i>)</code>	<code>\$(<i>"ul li:gt(3)"</i>)</code>	Danh sách thẻ với chỉ số lớn hơn 3
<code>:lt(<i>no</i>)</code>	<code>\$(<i>"ul li:lt(3)"</i>)</code>	Danh sách thẻ với chỉ số nhỏ hơn 3
<code>:not(<i>selector</i>)</code>	<code>\$(<i>"input:not(:emtv)"</i>)</code>	Tất cả thẻ input khác rỗng
<code>.header</code>	<code>\$(<i>"header"</i>)</code>	Tất cả thẻ header h1_h2

<code>:animated</code>	<code>\$(<i>"animated"</i>)</code>	Tất cả thẻ chuyển động
<code>:contains(<i>text</i>)</code>	<code>\$(<i>"contains('W3Schools')"</i>)</code>	Tất cả thẻ chứa chuỗi
<code>:emtv</code>	<code>\$(<i>"emtv"</i>)</code>	Tất cả thẻ không có thẻ con
<code>:hidden</code>	<code>\$(<i>"n:hidden"</i>)</code>	Tất cả thẻ ẩn
<code>:visible</code>	<code>\$(<i>"table:visible"</i>)</code>	Tất cả thẻ table hiển thị
<code>s1.s2.s3</code>	<code>\$(<i>"th.td.intro"</i>)</code>	Tất cả thẻ khớp với selector
<code>[<i>attribute</i>]</code>	<code>\$(<i>"href"</i>)</code>	Tất cả thẻ có thuộc tính href
<code>[<i>attribute=</i>value]</code>	<code>\$(<i>"href='default.htm'"</i>)</code>	Tất cả thẻ có thuộc tính href bằng "default.htm"
<code>[<i>attribute!=</i>value]</code>	<code>\$(<i>"href!='default.htm'"</i>)</code>	Tất cả thẻ có thuộc tính href khác "default.htm"
<code>[<i>attribute\$=</i>value]</code>	<code>\$(<i>"href\$='.in'"</i>)</code>	Tất cả thẻ có thuộc tính href kết thúc bằng ".in"
<code>:input</code>	<code>\$(<i>"input"</i>)</code>	Tất cả thẻ input
<code>:text</code>	<code>\$(<i>"text"</i>)</code>	Tất cả thẻ input có tvpe="text"
<code>:password</code>	<code>\$(<i>"password"</i>)</code>	Tất cả thẻ input có tvpe="password"
<code>:radio</code>	<code>\$(<i>"radio"</i>)</code>	Tất cả thẻ input có tvpe="radio"
<code>:checkbox</code>	<code>\$(<i>"checkbox"</i>)</code>	Tất cả thẻ input có tvpe="checkbox"
<code>:submit</code>	<code>\$(<i>"submit"</i>)</code>	Tất cả thẻ input có tvpe="submit"
<code>:reset</code>	<code>\$(<i>"reset"</i>)</code>	Tất cả thẻ input có tvpe="reset"

<code>:button</code>	<code>\$(":button")</code>	Tất cả thẻ input có <code>type="button"</code>
<code>:image</code>	<code>\$(":image")</code>	Tất cả thẻ input có <code>type="image"</code>
<code>:file</code>	<code>\$(":file")</code>	Tất cả thẻ input có <code>type="file"</code>
<code>:enabled</code>	<code>\$(":enabled")</code>	Tất cả thẻ input được kích hoạt
<code>:disabled</code>	<code>\$(":disabled")</code>	Tất cả thẻ input được hủy kích hoạt

<code>:selected</code>	<code>\$(":selected")</code>	Tất cả thẻ input lựa chọn
<code>:checked</code>	<code>\$(":checked")</code>	Tất cả thẻ input kiểm tra

5.12.8 Các jQuery Event

5.12.8.1 Các hàm sự kiện

Các phương thức bắt sự kiện là chức năng cốt lõi trong jQuery.

Các phương thức bắt sự kiện được gọi khi “một cái gì đó xảy ra” trong HTML. Thuật ngữ “trigger” (hoặc “fired”) thường được sử dụng cho một sự kiện khi nó xảy ra.

Thông thường ta thường đặt phương thức bắt sự kiện trong phần `<head>`:

```
<html>
```

```
<head>
```

```
<script type="text/javascript" src="jquery.js"></script>
```

```
<script type="text/javascript">
```

```
$(document).ready(function(){
```

```
    $("button").click(function(){  
  
        $("p").hide();  
  
        });  
  
});  
  
</script>  
  
</head>  
  
<body>  
  
<h2>This is a heading</h2>  
  
<p>This is a paragraph.</p>  
  
<p>This is another paragraph.</p>  
  
<button>Click me</button>  
  
</body>  
  
</html>
```

Trong ví dụ trên, một hàm được gọi khi sự kiện click của một nút được kích hoạt:

```
$("button").click(function() { ...some code... } )
```

Phương thức ẩn tất cả thẻ <p>:

```
$("#p").hide();
```

5.12.8.2 Các chức năng trong một tập tin riêng biệt

Nếu trang web có chứa nhiều trang, và để dễ quản lý các hàm jQuery, thì ta nên đặt các hàm jQuery trong các tập tin .js riêng biệt.

```
<head>
```

```
<script type="text/javascript" src="jquery.js"></script>
```

```
<script type="text/javascript" src="my_jquery_functions.js"></script>
```

```
</head>
```

5.12.8.3 Xung đột tên

jQuery sử dụng ký hiệu \$ như một phím tắt cho jQuery.

Một số thư viện JavaScript khác cũng sử dụng ký hiệu đô la cho các chức năng của họ.

Phương thức noConflict() xác định một tên tùy chỉnh (như jq), thay vì sử dụng ký hiệu đô la.

5.12.8.4 Các sự kiện

Phương thức sự kiện	Mô tả
<code>\$(document).ready(function)</code>	Liên kết một hàm cho sự kiện ready của trang (khi trang được tải xong).
<code>\$(selector).click(function)</code>	Kích hoạt, hoặc liên kết một hàm cho sự kiện click cho thẻ được chọn.

<code>\$(selector).dblclick(function)</code>	Kích hoạt, hoặc liên kết một hàm cho sự kiện <code>double click</code> cho thẻ được chọn.
<code>\$(selector).focus(function)</code>	Kích hoạt, hoặc liên kết một hàm cho sự kiện <code>focus</code> cho thẻ được chọn.
<code>\$(selector).mouseover(function)</code>	Kích hoạt, hoặc liên kết một hàm cho sự kiện <code>mouse over</code> cho thẻ được chọn.

5.12.8.5 Bảng sự kiện

Phương thức	Mô tả
<u>bind()</u>	Thêm một hoặc nhiều trình bắt sự kiện cho các thẻ phù hợp
<u>blur()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện <code>blur</code> của thẻ
<u>change()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện <code>change</code> của thẻ chọn
<u>click()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện <code>click</code> của thẻ
<u>dblclick()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện <code>dblclick</code> của thẻ chọn
<u>delegate()</u>	Thêm một hoặc nhiều trình bắt sự kiện cho thẻ hiện tại, hoặc một thẻ con cụ thể của thẻ phù hợp
<u>die()</u>	Hủy bỏ tất cả trình bắt sự kiện đã thêm bằng hàm <code>live()</code>
<u>error()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện <code>error</code> của thẻ
<code>event.currentTarget</code>	Thẻ DOM hiện tại trong giai đoạn gây nên sự kiện
<code>event.data</code>	Chứa dữ liệu tùy chọn thông qua <code>jQuery.fn.bind</code> khi trình xử lý đang thực thi bị ràng buộc
<u>event.isDefaultPrevented()</u>	Trả về giá trị xác định <code>event.preventDefault()</code> đã được gọi
<code>event.isImmediatePropagationStopped()</code>	Trả về giá trị xác định <code>event.stopImmediatePropagation()</code>

<code>event.isPropagationStopped()</code>	Trả về giá trị xác định <code>event.stopPropagation()</code> đã được gọi cho đối tượng sự kiện
<code>event.pageX</code>	Vị trí chuột so với canh trái của trang
<code>event.pageY</code>	Vị trí chuột so với canh trên của trang
<code>event.preventDefault()</code>	Ngăn chặn hành động mặc định của sự kiện
<code>event.relatedTarget</code>	Thẻ DOM khác liên quan đến sự kiện, nếu tồn tại
<code>event.result</code>	Thuộc tính này chứa giá trị cuối được trả về bởi một hàm xử lý sự kiện được kích hoạt bởi sự kiện này, trừ khi giá trị không xác định
<code>event.stonImmediatePronagation()</code>	Ngăn chặn các hàm xử lý sự kiện khác được gọi
<code>event.stopPropagation()</code>	Ngăn chặn sự kiện gây nên cây DOM, ngăn chặn bất cứ
<code>event.target</code>	Thẻ DOM bắt đầu sự kiện
<code>event.timeStamp</code>	Thuộc tính này trả về số phần nghìn giây từ 01/01/1970 khi sự kiện được kích hoạt
<code>event.type</code>	Mô tả bản chất của sự kiện
<code>event.which</code>	Phím hoặc nút nào được ấn cho sự kiện phím hoặc nút
<code>focus()</code>	Kích hoạt hoặc liên kết một hàm cho sự kiện focus của thẻ
<code>focusin()</code>	Liên kết một hàm cho sự kiện focusin của thẻ chọn
<code>focusout()</code>	Liên kết một hàm cho sự kiện focusout của thẻ chọn
<code>hover()</code>	Liên kết một hoặc hai hàm cho sự kiện hover của sự kiện chọn
<code>keydown()</code>	Kích hoạt hoặc liên kết một hàm cho sự kiện keydown của thẻ chọn
<code>keypress()</code>	Kích hoạt hoặc liên kết một hàm cho sự kiện keypress của thẻ chọn
<code>keyup()</code>	Kích hoạt hoặc liên kết một hàm cho sự kiện keyup của thẻ chọn
<code>live()</code>	Thêm một hoặc nhiều hàm xử lý sự kiện cho thẻ hiện tại, hoặc thẻ phù hợp

<u>load()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện load của thẻ
<u>mousedown()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện mousedown của thẻ chọn
<u>mouseenter()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện mouseenter của thẻ chọn
<u>mouseleave()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện mouseleave của thẻ chọn
<u>mousemove()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện mousemove của thẻ chọn
<u>mouseout()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện mouseout của thẻ chọn
<u>mouseover()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện mouseover của thẻ chọn
<u>mouseup()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện mouseup của thẻ chọn
<u>one()</u>	Thêm một hoặc nhiều hàm xử lý sự kiện cho thẻ phù hợp. Hàm xử lý này chỉ có thể kích hoạt một lần trên một thẻ
<u>ready()</u>	Liên kết một hàm cho sự kiện ready của tài liệu (khi một nội dung HTML sẵn sàng được sử dụng)
<u>resize()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện resize của thẻ chọn
<u>scroll()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện scroll của thẻ
<u>select()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện select của thẻ chọn
<u>submit()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện submit của thẻ chọn
<u>toggle()</u>	Liên kết một hoặc hai hàm chuyển đổi cho sự kiện click của thẻ chọn
<u>trigger()</u>	Kích hoạt tất cả sự kiện ràng buộc với thẻ chọn
<u>triggerHandler()</u>	Kích hoạt tất cả hàm ràng buộc cho một sự kiện cụ thể
<u>unbind()</u>	Hủy bỏ một hàm xử lý sự kiện đã thêm vào thẻ chọn

<u>undelegate()</u>	Hủy bỏ một hàm xử lý sự kiện của thẻ chọn hiện tại và
<u>unload()</u>	Kích hoạt hoặc liên kết một hàm cho sự kiện unload của thẻ chọn

5.12.9 Hiệu ứng

5.12.9.1 Hiệu ứng ẩn hiện

Với jQuery, ta có thể ẩn và hiển thị thẻ HTML bằng phương thức `hide()` và `show()`.

Ví dụ:

```
$("#hide").click(function(){
```

```
    $("#p").hide();
```

```
});
```

```
$("#show").click(function(){
```

```
    $("#p").show();
```

```
});
```

Cả `hide()` và `show()` có thể nhận 2 tham số: `speed` và `callback`. Cú pháp:

```
$(selector).hide(speed,callback)
```

```
$(selector).show(speed,callback)
```

Biến `speed` quy định tốc độ ẩn/hiện, và có thể nhận các giá trị sau: “slow”, “fast”, “normal”, hoặc phần nghìn giây.

Biến callback là tên của hàm sẽ được thực hiện sau khi hàm ẩn (hoặc hiện) hoàn tất.

5.12.9.2 Hiện ứng chuyển đổi ẩn/hiện

Phương thức toggle() chuyển đổi sự hiển thị của thẻ HTML sử dụng phương thức show() hoặc hide().

Các thẻ hiển thị được ẩn và các thẻ ẩn được hiển thị. Cú pháp:

```
$(selector).toggle(speed,callback)
```

Biến speed có thể nhận các giá trị sau: “slow”, “fast”, “normal”, hoặc phần nghìn giây.

Biến callback là tên của hàm sẽ được thực hiện sau khi hàm ẩn (hoặc hiện) hoàn tất.

Ví dụ:

```
$("#button").click(function(){  
  
    $("#p").toggle();  
  
});
```

5.12.9.3 Hiệu ứng trượt

Các phương thức trượt dần dần thay đổi chiều cao của các phần tử đã chọn.

jQuery có các phương thức trượt sau đây

```
$(selector).slideDown(speed,callback)
```

```
$(selector).slideUp(speed,callback)
```

```
$(selector).slideToggle(speed,callback)
```

Biến speed sẽ có nhận các giá trị sau: “slow”, “fast”, “normal”, hoặc phần nghìn giây.

Biến callback là tên của hàm sẽ được thực hiện sau khi hàm hoàn tất. Ví dụ slideDown():

```
$("#flip").click(function(){  
  
    $(".panel").slideDown();  
  
});
```

Ví dụ slideUp():

```
$("#flip").click(function(){  
  
    $(".panel").slideUp()  
  
});
```

Ví dụ slideToggle():

```
$("#flip").click(function(){  
  
    $(".panel").slideToggle();  
  
});
```

5.12.9.4 Hiệu ứng mờ

Các phương thức mờ dần dần thay đổi độ mờ của phần tử chọn.

jQuery có các phương thức mờ sau đây:

```
$(selector).fadeIn(speed,callback)
```

```
$(selector).fadeOut(speed,callback)
```

```
$(selector).fadeTo(speed,opacity,callback)
```

Biến `speed` có thể nhận các giá trị sau: “slow”, “fast”, “normal”, hoặc phần nghìn giây.

Biến `opacity` trong phương thức `fadeTo()` cho phép làm mờ đến độ mờ cụ thể. Biến `callback` là tên của hàm sẽ được thực hiện sau khi hàm hoàn tất.

Ví dụ `fadeTo()`:

```
$("#button").click(function(){  
  
    $("#div").fadeTo("slow",0.25);  
  
});
```

Ví dụ `fadeOut()`:

```
$("#button").click(function(){  
  
    $("#div").fadeOut(4000);  
  
});
```

5.12.9.5 Tùy chỉnh chuyển động

Cú pháp của phương thức jQuery dùng để tùy chỉnh chuyển động là:

```
$(selector).animate({params},[duration],[easing],[callback])
```

Biến chính là `params`. Nó định nghĩa các thuộc tính CSS sẽ được hoạt hình. Nhiều thuộc tính có thể được hoạt hình cùng lúc:

```
animate({ width:"70%", opacity:0.4, marginLeft:"0.6in", fontSize:"3em" });
```

Biến thứ hai là duration. Nó quy định tốc độ của chuyển động. Các giá trị có thể nhận là “fast”, “slow”, “normal”, hoặc phần nghìn giây.

Ví dụ 1:

```
<script type="text/javascript">  
  
$(document).ready(function(){  
  
    $("button").click(function(){  
  
        $("div").animate({height:300},"slow");  
  
        $("div").animate({width:300},"slow");  
  
        $("div").animate({height:100},"slow");  
  
        $("div").animate({width:100},"slow");  
  
    });  
  
});  
  
</script>
```

Ví dụ 2:

```
<script type="text/javascript">  
  
$(document).ready(function(){  
  
    $("button").click(function(){
```

```

$("div").animate({left:"100px"},"slow");

$("div").animate({fontSize:"3em"},"slow");

});

});

</script>

```

5.12.9.6 Tổng kết các hiệu ứng

Hàm	Mô tả
<code>\$(selector).hide()</code>	Ẩn thẻ được chọn
<code>\$(selector).show()</code>	Hiện thẻ được chọn
<code>\$(selector).toggle()</code>	Chuyển đổi (giữa ẩn và hiện) thẻ được chọn
<code>\$(selector).slideDown()</code>	Slide-down (hiển thị) thẻ được chọn
<code>\$(selector).slideUp()</code>	Slide-up (ẩn) thẻ được chọn
<code>\$(selector).slideToggle()</code>	Chuyển đổi slide-up và slide-down của thẻ được chọn.
<code>\$(selector).fadeIn()</code>	Fade in thẻ được chọn
<code>\$(selector).fadeOut()</code>	Fade out thẻ được chọn
<code>\$(selector).fadeTo()</code>	Fade out thẻ được chọn đến độ mờ được xác định
<code>\$(selector).animate()</code>	Chạy một chuyển động tùy chỉnh trên thẻ được chọn

5.12.10 Hàm gọi lại (Callback Function)

5.12.10.1 Các hàm gọi lại

Một hàm gọi lại được thực thi khi chuyển động hoàn toàn hoàn thành.

Các biểu thức JavaScript được thực hiện từng dòng. Tuy nhiên với sự chuyển động, các dòng lệnh kế có thể chạy ngay cả khi chuyển động chưa hoàn tất. Điều này có thể tạo ra các lỗi.

Để ngăn chặn điều này, ta có thể tạo hàm gọi lại.

Một hàm gọi được thực thi sau khi chuyển động hiện tại (hiệu ứng) kết thúc.

5.12.10.2 Ví dụ

Điền hình cú pháp: `$(selector).hide(speed,callback)`

Biến callback là một hàm được thực thi sau khi hiệu ứng hide hoàn thành. Ví dụ:

```
$("#p").hide(1000,function(){  
  
    alert("The paragraph is now hidden");  
  
});
```

Nếu không có biến callback, cửa sổ thông báo sẽ được hiển thị trước khi hiệu ứng ẩn hoàn thành.

Ví dụ không có hàm gọi lại:

```
$("#p").hide(1000);  
  
alert("The paragraph is now hidden");
```

5.12.11 Thao tác trên HTML Tag

jQuery chức các phương thức mạnh mẽ (các hàm) để thay đổi và thao tác các thẻ và thuộc tính HTML.

5.12.11.1 Thay đổi nội dung HTML

`$(selector).html(content)`

Phương thức `html()` thay đổi nội dung (innerHTML) cho các thẻ phù hợp. Ví dụ:

```
$("#p").html("W3Schools");
```

5.12.11.2 Thêm nội dung HTML

`$(selector).append(content)`

Phương thức `append()` nối thêm nội dung của các thẻ phù hợp.

`$(selector).prepend(content)`

Phương thức `prepend()` thêm đằng trước nội dung của các thẻ phù hợp. Ví dụ:

```
$("#p").append(" W3Schools");
```

`$(selector).after(content)`

Phương thức `after()` chèn nội dung HTML sau các thẻ phù hợp.

`$(selector).before(content)`

Phương thức `before()` chèn nội dung HTML trước các thẻ phù hợp. Ví dụ:

```
$("#p").after(" W3Schools.");
```

5.12.11.3 Bảng các hàm thao tác HTML Tag

Phương thức	Mô tả
<code>addClass()</code>	Thêm một hoặc nhiều lớp (đối với CSS) cho thẻ chọn.

<code>after()</code>	Chèn nội dung sau thẻ chọn
<code>append()</code>	Chèn nội dung tại cuối của thẻ chọn (nhưng vẫn giữ nội dung ở trong)
<code>appendTo()</code>	Chèn nội dung tại cuối của thẻ chọn (nhưng vẫn giữ nội dung ở trong)
<code>attr()</code>	Đặt hoặc trả về một thuộc tính và giá trị của thẻ chọn
<code>before()</code>	Chèn nội dung trước thẻ chọn
<code>clone()</code>	Tạo một bản sao của thẻ chọn
<code>detach()</code>	Xóa (nhưng giữ một bản sao) thẻ chọn
<code>empty()</code>	Xóa tất cả thẻ con và nội dung của thẻ chọn
<code>hasClass()</code>	Kiểm tra thẻ chọn có chứa một lớp hay không
<code>html()</code>	Đặt hoặc trả về nội dung thẻ chọn
<code>insertAfter()</code>	Chèn nội dung HTML hoặc các thẻ sau thẻ chọn
<code>insertBefore()</code>	Chèn nội dung HTML hoặc các thẻ trước thẻ chọn
<code>prepend()</code>	Chèn nội dung tại vị trí đầu của thẻ chọn (vẫn giữ nội dung ở trong)
<code>prependTo()</code>	Chèn nội dung tại vị trí đầu của thẻ chọn (vẫn giữ nội dung ở trong)
<code>remove()</code>	Xóa thẻ chọn
<code>removeAttr()</code>	Xóa một thuộc tính của thẻ chọn
<code>removeClass()</code>	Xóa một hoặc nhiều lớp của thẻ chọn
<code>replaceAll()</code>	Thay thế thẻ chọn với nội dung mới
<code>replaceWith()</code>	Thay thế thẻ chọn với nội dung mới
<code>text()</code>	Đặt hoặc trả về nội dung văn bản của thẻ chọn
<code>toggleClass()</code>	Chuyển đổi giữa thêm/xóa một hoặc nhiều lớp của thẻ chọn
<code>unwrap()</code>	Xóa thẻ cha của thẻ chọn
<code>val()</code>	Đặt hoặc trả về giá trị thuộc tính của thẻ chọn
<code>wrap()</code>	Hoán đổi nội dung HTML các thẻ xung quanh mỗi thẻ chọn
<code>wrapAll()</code>	Hoán đổi nội dung HTML các thẻ xung quanh thẻ chọn
<code>wrapInner()</code>	Hoán đổi nội dung HTML các thẻ xung quanh nội dung của mỗi thẻ chọn

5.12.12 Thao tác CSS

5.12.12.1 Phương thức css()

jQuery chứa một phương thức quan trọng cho việc thao tác CSS: `css()`

Phương thức `css()` có 3 cú pháp khác nhau để thực hiện các nhiệm vụ khác nhau:

- `css(name)` – Trả về giá trị của thuộc tính CSS.
- `css(name,value)` – Đặt thuộc tính CSS và giá trị.
- `css({properties})` – Đặt nhiều thuộc tính CSS và giá trị.

5.12.12.2 Trả về thuộc tính CSS

Sử dụng `css(name)` để trả về giá trị thuộc tính CSS nêu ra của thẻ phù hợp đầu tiên.

Ví dụ:

```
$(this).css("background-color");
```

5.12.12.3 Đặt thuộc tính CSS và giá trị

Sử dụng `css(name,value)` để thiết lập thuộc tính CSS và giá trị cho tất cả thẻ phù hợp.

Ví dụ:

```
$("p").css("background-color","yellow");
```

5.12.12.4 Đặt nhiều cặp thuộc tính CSS và giá trị

Sử dụng `css({properties})` để thiết lập một hoặc nhiều cặp thuộc tính/giá trị CSS cho các thẻ được chọn.

Ví dụ:

```
$("#p").css({"background-color":"yellow","font-size":"200%"});
```

5.12.12.5 Phương thức `height()` và `width()`

jQuery có 2 phương thức quan trọng cho việc thao tác kích thước.

- `height()`
- `width()`

Ví dụ thao tác kích thước

Phương thức `height()` thiết lập chiều cao cho tất cả thẻ phù hợp.

Ví dụ:

```
$("#div1").height("200px");
```

Phương thức `width()` thiết lập chiều rộng cho tất cả thẻ phù hợp. Ví dụ:

```
$("#div2").width("300px");
```

5.12.12.6 Bảng các hàm CSS

Phương thức	Mô tả
<code>addClass()</code>	Thêm một hoặc nhiều lớp vào thẻ chọn
<code>css()</code>	Đặt hoặc trả về một hoặc nhiều thuộc tính kiểu của thẻ chọn
<code>hasClass()</code>	Kiểm tra xem thẻ chọn có chứa một lớp chỉ ra hay không

<code>height()</code>	Đặt hoặc trả về chiều cao của thẻ chọn
<code>offset()</code>	Đặt hoặc trả về vị trí (tương đối so với trang) của thẻ chọn
<code>offsetParent()</code>	Trả về vị trí thẻ cha đầu.
<code>position()</code>	Trả về vị trí (tương đối so với thẻ cha) của thẻ chọn đầu tiên
<code>removeClass()</code>	Xóa một hoặc nhiều lớp của thẻ chọn
<code>scrollLeft()</code>	Đặt hoặc trả về vị trí ngang của thanh cuộn đối với thẻ chọn
<code>scrollTop()</code>	Đặt hoặc trả về vị trí dọc của thanh cuộn đối với thẻ chọn
<code>toggleClass()</code>	Chuyển đổi giữa thêm/xóa một hoặc nhiều lớp từ thẻ chọn
<code>width()</code>	Đặt hoặc trả về chiều rộng của thẻ chọn

5.12.13 AJAX

jQuery có một thư viện phong phú các phương thức (các hàm) để phát triển AJAX.

5.12.13.1 AJAX là gì?

AJAX = Asynchronous JavaScript and XML. AJAX là một kỹ thuật để tạo ra các trang web nhanh chóng và linh động. AJAX cho phép trang web cập nhật không đồng bộ bằng các trao đổi một lượng nhỏ dữ liệu với máy phục vụ. Điều này có nghĩa là nó có thể cập nhật một phần của một trang web, mà không cần tải lại toàn bộ trang.

5.12.13.2 AJAX và jQuery

jQuery cung cấp một tập phong phú các phương thức cho việc phát triển web AJAX.

Với jQuery AJAX, ta có thể yêu cầu TXT, HTML, XML hoặc dữ liệu JSON từ máy phục vụ sử dụng cả hai phương thức GET và POST.

Và ta có thể tải trực tiếp dữ liệu từ xa vào các thẻ HTML của trang web.

5.12.13.3 Viết ít hơn, làm nhiều hơn

Phương thức load() là một hàm AJAX đơn giản (nhưng rất mạnh mẽ).

`$(selector).load(url,data,callback)`

Sử dụng selector để xác định các thẻ HTML cần thay đổi, và biến url để chỉ định địa chỉ cho dữ liệu.

Chỉ khi ta muốn gửi dữ liệu đến máy phục vụ, ta mới cần sử dụng biến data.

5.12.13.4 AJAX cấp thấp

`$.ajax(options)` là cú pháp của hàm AJAX cấp thấp.

`$.ajax` cung cấp nhiều chức năng hơn so với các hàm cấp cao hơn như load, get, và post, nhưng nó cũng khó khăn hơn khi sử dụng.

Các tham số tùy chọn chứa các cặp tên|giá trị xác định dữ liệu địa chỉ, mật khẩu, kiểu dữ liệu, các dữ liệu lọc, tập các ký tự, thời hạn và hàm lỗi.

5.12.13.5 Bảng các phương thức AJAX

Phương thức	Mô tả
<code>\$.ajax()</code>	Thực hiện một AJAX request
<code>ajaxComplete()</code>	Xác định một hàm sẽ chạy khi AJAX request hoàn thành
<code>ajaxError()</code>	Xác định một hàm sẽ chạy khi AJAX request hoàn thành có lỗi
<code>ajaxSend()</code>	Xác định một hàm sẽ chạy trước khi AJAX request được gửi
<code>\$.ajaxSetup()</code>	Đặt giá trị mặc định cho AJAX request
<code>ajaxStart()</code>	Xác định một hàm sẽ chạy khi AJAX request bắt đầu
<code>ajaxStop()</code>	Xác định một hàm sẽ chạy khi tất cả AJAX request hoàn thành
<code>ajaxSuccess()</code>	Specifies a function to run an AJAX request completes successfully

\$.get()	Tải tất cả dữ liệu từ server sử dụng AJAX HTTP GET request
\$.getJSON()	Tải dữ liệu mã hóa JSON từ server sử dụng HTTP GET request
\$.getScript()	Tải (và thực thi) một JavaScript từ server sử dụng một AJAX HTTP GET request
load()	Tải dữ liệu từ server và gán nội dung HTML trả về vào thẻ chọn
\$.param()	Tạo một dãy nối tiếp biểu diễn cho một mảng hoặc đối tượng (có thể sử dụng như ~ ~ ~)
\$.post()	Tải dữ liệu từ server sử dụng AJAX HTTP POST request
serialize()	Mã hóa một tập các thẻ trong biểu mẫu thành một chuỗi để gửi
serializeArray()	Mã hóa một tập các thẻ trong biểu mẫu thành một mảng của tên và giá trị

BÀI TẬP CHƯƠNG 5

1. Bài tập 1:

Tạo giao diện như sau:

Radio Button Alert

Click on one of the radio buttons to see a message

1: 2: 3:

Yêu cầu :

Khi Click chuột vào Radio Button thì thông điệp (Message) tương ứng:

<FORM>

<p>1:

<INPUT TYPE="radio" NAME="radio" value="Bạn chọn số 1" onClick="alert(value)">2:

<INPUT TYPE="radio" NAME="radio" value=" Bạn chọn số 2 " onClick="alert(value)">3:

<INPUT TYPE="radio" NAME="radio" value=" Bạn chọn số 3" onClick="alert(value)" >

</form>

2. Bài tập 2

Button Alert

Push this button to make a message appear.

message

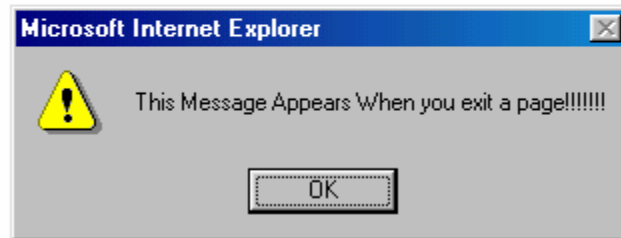
Khi Click chuột vào nút Message thì hiện lên câu chào:

```
<form>
<input type="button" value="message" onClick="alert('How are you'); return true">
</form>
```

3. Bài tập 3

BACK

Click the back to see the Example!



Chào tạm biệt

Khi đóng cửa sổ trình duyệt thì xuất hiện lời chào tạm biệt.

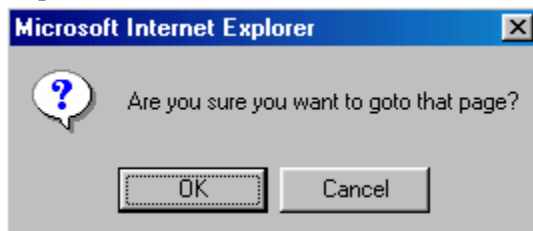
```
<html>
<head>
<title>Click the back to see the Example</title>
<meta name="GENERATOR" content="Copyright Next Step Software 1998 (c)">
</head>
<body onUnload="window.alert('This Message Appears When you exit a page!!!!!!')">
<form>
<p><input TYPE="button" VALUE=" BACK " onClick="history.go(-1)"> </p>
</form>
<p>Click the back to see the Example!</p>
</body>
</html>
```

4. Bài tập 4:

Yêu cầu:

Khi nhập vào liên kết thì Windows hỏi, nếu OK thì chuyển đến trang liên kết, nếu Cancel thì không thực hiện gì cả.

<http://www.cidnet.vn>



```
<script>
```

```
function rusure(){
question = confirm("YOUR CONFIRM MESSAGE")
if (question !="0"){
top.location = "YOUR LINK GOES HERE"
}
}
</script>
```

Now put this anywhere in your page and change YOUR LINK DESCRIPTION
YOUR LINK DESCRIPTION

5.Bài tập 5:

Hãy tạo một chương trình máy tính như sau:



```
<html>
<head><script LANGUAGE="JavaScript">
function a_plus_b(form) {
a=eval(form.a.value)
b=eval(form.b.value)
c=a+b
form.ans.value = c
}
function a_minus_b(form) {
a=eval(form.a.value)
b=eval(form.b.value)
c=a-b
form.ans.value=c
}
function a_times_b(form) {
a=eval(form.a.value)
b=eval(form.b.value)
c=a*b
form.ans.value=c
}
function a_div_b(form) {
a=eval(form.a.value)
b=eval(form.b.value)
c=a/b
form.ans.value = c
}
function a_pow_b(form) {
```

```

a=eval(form.a.value)
b=eval(form.b.value)
c=Math.pow(a, b)
form.ans.value = c
}
</script>
<title>E:\button\windowsizer_.htm</title>
</head>
<body>
<form name="formx">
<p><input type="text" size="4" value="12" name="a"> <input type="button" value=" + "
onClick="a_plus_b(this.form)"> <input type="button" value=" - "
onClick="a_minus_b(this.form)"> <input type="button" value=" x "
onClick="a_times_b(this.form)"> <input type="button" value=" / "
onClick="a_div_b(this.form)"> <input type="button" value=" ^ "
onClick="a_pow_b(this.form)"> <input type="number" size="4" value="3" name="b"> =
<input
type "number" value="0" name="ans" size="9"> </p>
</form>
</body>
</html>

```

6. Bài tập 6:

Tạo một chương trình lịch để bàn như sau:

November 2002						
Su	M	Tu	W	Th	F	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

```

<html>
<head>
<title> Lịch nam </title>
</head>
<body>
<script LANGUAGE="JavaScript">
<!-- Begin

```

```

monthnames = new Array("January", "February", "March", "April", "May", "June",
"July", "August", "September", "October", "November", "Decemeber");
var linkcount=0;
function addlink(month, day, href) {
var entry = new Array(3);
entry[0] = month;
entry[1] = day;
entry[2] = href;
this[linkcount++] = entry;
}
Array.prototype.addlink = addlink;
linkdays = new Array();
monthdays = new Array(12);
monthdays[0]=31;
monthdays[1]=28;
monthdays[2]=31;
monthdays[3]=30;
monthdays[4]=31;
monthdays[5]=30;
monthdays[6]=31;
monthdays[7]=31;
monthdays[8]=30;
monthdays[9]=31;
monthdays[10]=30;
monthdays[11]=31;
todayDate=new Date();
thisday=todayDate.getDay();
thismonth=todayDate.getMonth();
thisdate=todayDate.getDate();
thisyear=todayDate.getYear();
thisyear = thisyear % 100;
thisyear = ((thisyear < 50) ? (2000 + thisyear) : (1900 + thisyear));
if (((thisyear % 4 == 0) && !(thisyear % 100 == 0)) || (thisyear % 400 == 0)) monthdays[1]++;
startspaces=thisdate;
while (startspaces > 7) startspaces-=7;
startspaces = thisday - startspaces + 1;
if (startspaces < 0) startspaces+=7;
document.write("<table border=2 bgcolor=white ");
document.write("bordercolor=black><font color=black>");

```

```

document.write("<tr><td colspan=7><center><strong>" + monthnames[thismonth] + " " + thisyear+
"</strong></center></font></td></tr>");
document.write("<tr>");
document.write("<td align=center>Su</td>");
document.write("<td align=center>M</td>");
document.write("<td align=center>Tu</td>");
document.write("<td align=center>W</td>");
document.write("<td align=center>Th</td>");
document.write("<td align=center>F</td>");
document.write("<td align=center>Sa</td>");
document.write("</tr>");
document.write("<tr>");
for (s=0;s<startspaces;s++) {
document.write("<td> </td>");
}
count=1;
while (count <= monthdays[thismonth]) {
for (b = startspaces;b<7;b++) {
linktrue=false;
document.write("<td>");
for (c=0;c<linkdays.length;c++) {
if (linkdays[c] != null) {
if ((linkdays[c][0]==thismonth + 1) && (linkdays[c][1]==count)) {
document.write("<a href=\"\" + linkdays[c][2] + \"\">");
linktrue=true;
}
}
}
if (count==thisdate) {
document.write("<font color='FF0000'><strong>");
}
if (count <= monthdays[thismonth]) {
document.write(count);
}
else {
document.write(" ");
}
if (count==thisdate) {
document.write("</strong></font>");
}if (

```

```

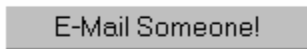
linktrue)
document.write("</a>");
document.write("</td>");
count++;
}
document.write("</tr>");
document.write("<tr>");
startspaces=0;
}
document.write("</table></p>");
// End -->
</script>
</body>
</html>

```

7. Bài tập 7:

Khi click vào link và button thì cho phép nhập vào địa chỉ người nhận và tiêu đề thư.

E-Mail Someone!



```

<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!-- Begin
function mailsome1(){
who=prompt("Enter recipient's email address: ","antispammer@earthling.net");
what=prompt("Enter the subject: ","none");
if (confirm("Are you sure you want to mail "+who+" with the subject of "+what+"?")==true){
parent.location.href='mailto:'+who+'?subject='+what+'';
}
}
// End -->
</SCRIPT>
<BODY>
<CENTER>
<a href='javascript:mailsome1()'>E-Mail Someone!</a>
<FORM>
<input type=button value="E-Mail Someone!" onClick="mailsome1()">
</FORM>
</CENTER>

```

8. Bài tập 8

Viết chương trình link đến một trang Web khác trong đó cho phép tùy chọn các đối tượng Windows

http://www.cidnet.vn : URL

: Toolbar

: Location

: Directories

: Status

: Menubar

: Scrollbars

: Resizable

: Width

: Height

=ENTER=

=RESET=

```
<HEAD>
```

```
<SCRIPT LANGUAGE="JavaScript">
```

```
<!-- Begin
```

```
function customize(form) {
```

```
var address = document.form1.url.value;
```

```
var op_tool = (document.form1.tool.checked == true) ? 1 : 0;
```

```
var op_loc_box = (document.form1.loc_box.checked == true) ? 1 : 0;
```

```
var op_dir = (document.form1.dir.checked == true) ? 1 : 0;
```

```
var op_stat = (document.form1.stat.checked == true) ? 1 : 0;
```

```
var op_menu = (document.form1.menu.checked == true) ? 1 : 0;
```

```
var op_scroll = (document.form1.scroll.checked == true) ? 1 : 0;
```

```
var op_resize = (document.form1.resize.checked == true) ? 1 : 0;
```

```
var op_wid = document.form1.wid.value;
```

```
var op_heigh = document.form1.heigh.value;
```

```
var option = "toolbar="+ op_tool +",location="+ op_loc_box +",directories="
```

```
+ op_dir +",status="+ op_stat +",menubar="+ op_menu +",scrollbars="
```

```
+ op_scroll +",resizable="+ op_resize +",width="+ op_wid +",height="+ op_heigh;
```

```
var win3 = window.open("", "what_I_want", option);
```

```
var win4 = window.open(address, "what_I_want");
```

```
}f
```

```
unction clear(form) {
```

```
document.form1.wid.value="";
```

```

document.form1.heigh.value="";
}
// End -->
</SCRIPT>
<BODY>
<CENTER>
<h4>Please choose from the following selections to customize your window</h4>
<br>
<TABLE cellpadding=5 border><TR><TD><PRE>
<FORM name=form1 ACTION="javascript:" METHOD="POST">
<INPUT TYPE="text" NAME="url" value="http://www.geocities.com" >: URL
<INPUT TYPE="checkbox" NAME="tool">: Toolbar
<INPUT TYPE="checkbox" NAME="loc_box">: Location
<INPUT TYPE="checkbox" NAME="dir">: Directories
<INPUT TYPE="checkbox" NAME="stat">: Status
<INPUT TYPE="checkbox" NAME="menu">: Menubar
<INPUT TYPE="checkbox" NAME="scroll">: Scrollbars
<INPUT TYPE="checkbox" NAME="resize">: Resizable
<INPUT TYPE="text" NAME="wid" value= >: Width
<INPUT TYPE="text" NAME="heigh" value= >: Height
<BR><CENTER>
<INPUT TYPE="button" VALUE="=ENTER=" OnClick="customize(this.form)">
<INPUT TYPE="reset" VALUE="=RESET=" onClick="clear(this.form)">
</PRE></TD></TR></TABLE>
</FORM>
</CENTER>

```

10. Bài tập 10:

Kiểm tra tính hợp lệ của thông tin nhập vào:

Name: Age:

```

<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!-- Begin
function validate(){
var digits="0123456789"
var temp
if (document.testform.Name.value=="") {
alert("No Name !")
return false
}
}

```

```
if (document.testform.age.value=="") {
alert("Invalid Age !")
return false
}
for (var i=0;i<document.testform.age.value.length;i++){
temp=document.testform.age.value.substring(i,i+1)
if (digits.indexOf(temp)==-1){
alert("Invalid Age !")
return false
}
}
return true
} // End -->
</SCRIPT>
<BODY>
<FORM name="testform" onSubmit="return validate()">
Name:<input type="text" size=30 name="Name">
Age:<input type="text" size=3 name="age">
<input type="submit" value="Submit">
</FORM>
```

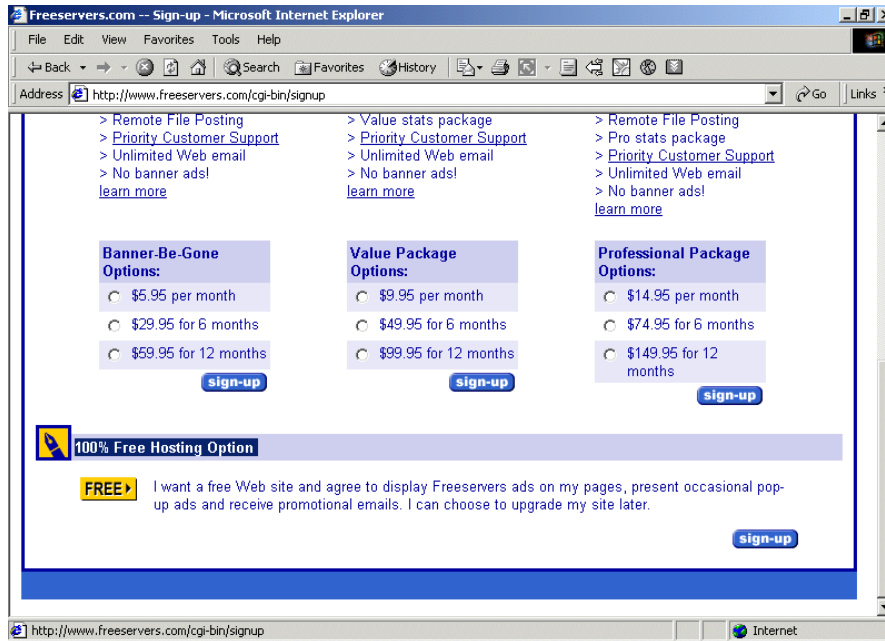
CHƯƠNG 6. QUẢN TRỊ WEBSITE

6.1. Đăng kí website miễn phí trên internet

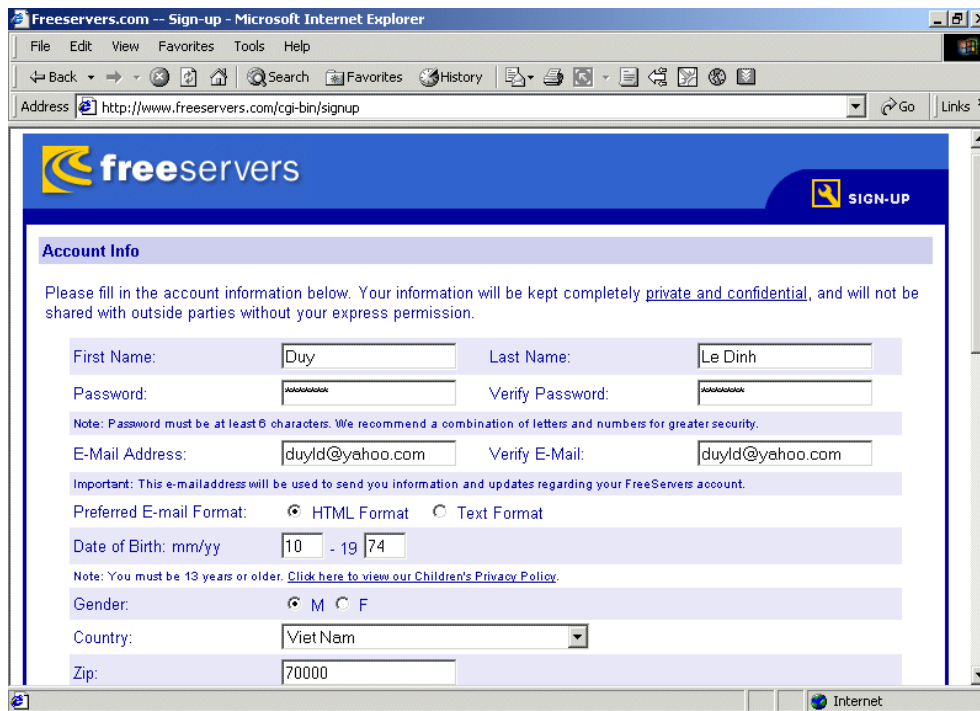
1. Vào địa chỉ <http://www.freesevers.com> để đăng kí website miễn phí với địa chỉ truy cập có dạng <http://<yourname>.freesevers.com>



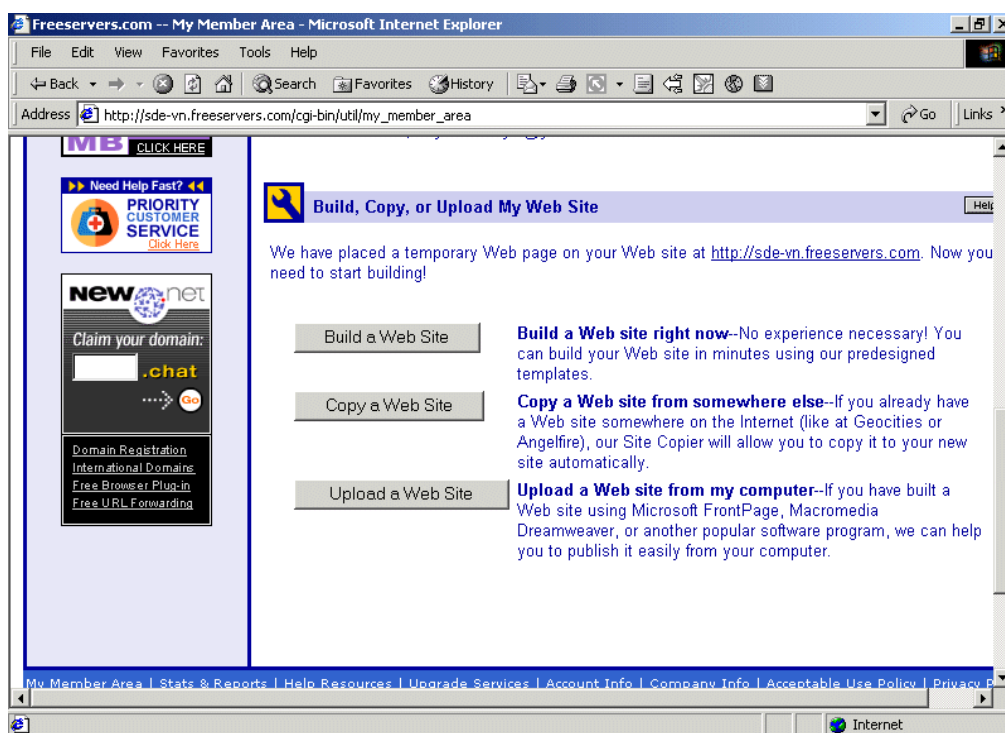
2. Gõ vào địa chỉ mà bạn dự định đăng kí tại ô **yourname**. Sau đó click **Go**. Màn hình tiếp theo sẽ hiện ra, bạn chọn mục cuối cùng **100% Free Hosting Option** và click vào **SignUp**



3. Sau đó bạn hãy điền vào các thông tin liên quan cần thiết như Họ tên, mật khẩu, email, ... Cuối cùng gõ vào ARTK để chắc rằng bạn đã đồng ý với các qui định của FreeServers khi cung cấp dịch vụ miễn phí này.



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



6.2. WordPress là gì?

WordPress được bắt đầu từ năm 2003 khi nhà sáng lập Matt Mullenweg phát triển một dự án nhỏ mang tên b2 / Cafelog. Với sự trợ giúp của Mike Little, họ đã sử dụng bộ mã nguồn đó để tạo nên phiên bản đầu tiên của WordPress vào ngày 27/5/2003. Và tôi tin chắc rằng, trong khoảng thời gian đó họ không hề nghĩ rằng thứ họ vừa tạo ra lại ảnh hưởng đến thế giới ngày hôm nay.

Từ lúc đó đến nay, WordPress được phát triển từ một công cụ cho việc viết blog trở thành một website quản lý nội dung (CMS) – mà từ đó bạn có thể tạo ra bất kỳ loại website nào mà bạn nghĩ ra, từ blog cá nhân, đến trang giới thiệu sản phẩm, triển lãm ảnh, đấu giá, học trực tuyến hay một trang thương mại điện tử...

Có hàng nghìn giao diện để thay đổi bộ mặt của một website và còn nhiều hơn nữa là các plugin cho phép mở rộng tính năng. Chưa kể đến là chúng hoàn toàn miễn phí cho bất kỳ ai đang sử dụng WordPress.

Trong phần hướng dẫn sử dụng WordPress cho người mới bắt đầu này. Chúng tôi sẽ không hướng dẫn bạn cách cài đặt, hay chọn [hosting như thế nào cho phù hợp với bạn](#). Thứ mà chúng tôi muốn hướng đến là một bài hướng dẫn sử dụng WordPress đầy đủ và thật chi tiết nhất có thể.

6.3. WordPress.com và WordPress.org khác nhau như thế nào?

Nhiều người có thể nhầm lẫn giữa WordPress.com và WordPress.org rằng chúng giống nhau. Nhưng thực tế chúng là hai khái niệm hoàn toàn khác nhau.

Tôi sẽ giải thích dễ hiểu nhất cho bạn.

Với WordPress.com là một dịch vụ cho phép bạn tạo một website miễn phí trên website của họ. Họ có có giao diện có sẵn để bạn tùy chỉnh với chức năng kéo thả dễ dàng. Bạn được phép sử dụng một dung lượng họ cho phép (**3GB nếu vượt quá sẽ phải trả thêm tiền**).

Nói chung là cài đặt rất dễ dàng, ko cần thuê hosting. Để sử dụng tên miền riêng bạn cần chi trả 25\$/năm. Rất tiện đối với người mới sử dụng WordPress

Nhưng nó cũng có những nhược điểm đó là:

- Bạn không thể cài đặt plugin (tính năng làm nên tên tuổi của wordpress). Nếu muốn sử dụng plugin bạn phải trả phí (3750\$/tháng, quá chất)
- Không thể sử dụng theme bên ngoài những theme của họ
- Bạn không có quyền kiểm soát 100% với website của mình. Họ có thể xóa website của bạn bất kỳ lúc nào..

Với WordPress.org là một website cung cấp bộ mã nguồn mở của nền tảng WordPress được cập nhật liên tục và hoàn toàn MIỄN PHÍ.

Kèm theo đó là thư viện plugin miễn phí, diễn đàn hỗ trợ về wordprss. Điều bạn cần làm là:

- Tải bộ mã nguồn tại wordpress.org
- Mua một hosting và tên miền
- Tiên hành cài đặt WordPress

Ưu điểm của WordPress.org các bạn có thể thấy bạn có thể dễ dàng thay đổi giao diện, sử dụng plugin, toàn quyền kiểm soát 100% với website. Nếu bạn đang có ý định xây dựng website để kinh doanh thì hãy lựa WordPress.org ngay nhé.

Mình đã có bài viết so sánh về WordPress.com và WordPress.org bấm vào [link này](#) để xem.

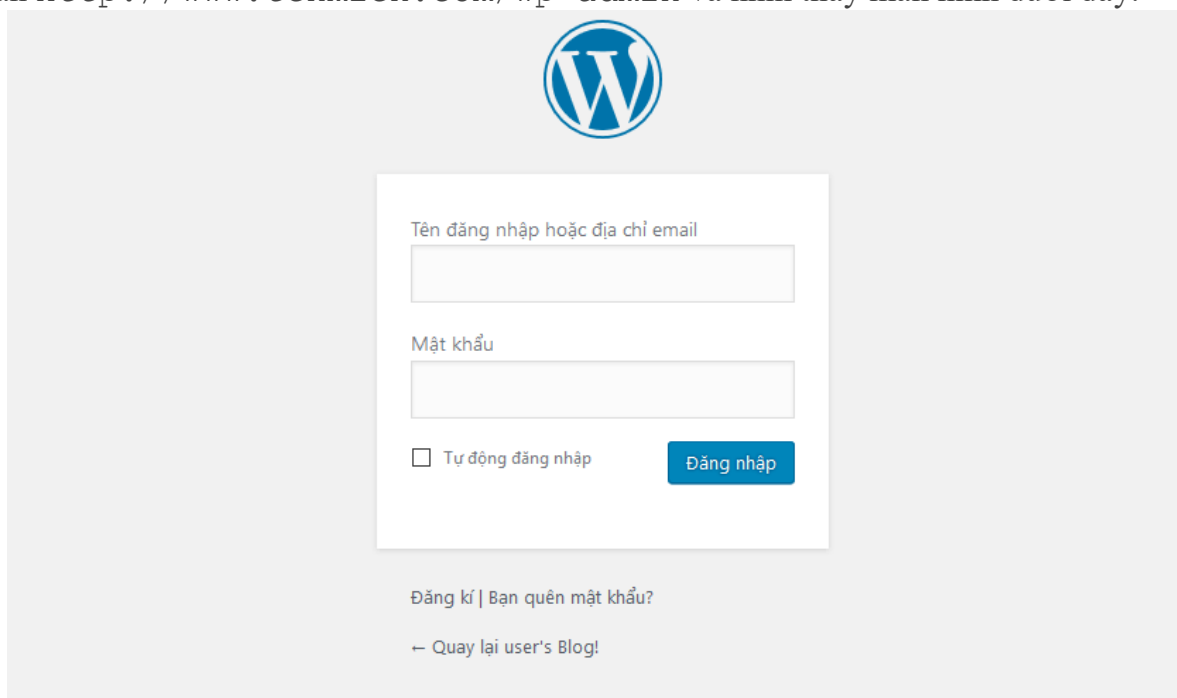
6.4. Cài Đặt WordPress

Để sử dụng WordPress bạn cần có một tên miền và một hosting (thường thì khi mới sử dụng WordPress bạn chỉ cần [Shared Hosting](#) là đủ).

Tiếp theo là mua một tên miền. Có một số nhà cung cấp tên miền rất nổi tiếng như: [Godaddy](#), [NameCheap](#), [BigRock](#) và [1and1](#)

Sau khi có tên miền và hosting là bạn có thể tiến hành cài đặt WordPress. Những bước tiến hành cài đặt khá đơn giản, bạn không cần phải là người am hiểu về kỹ thuật cũng có thể làm được

Để bắt đầu sử dụng website, bạn cần đăng nhập vào hệ thống bằng tên đăng nhập và mật khẩu được điền trong quá trình cài đặt. Bạn có thể đăng nhập qua đường dẫn <http://www.tenmien.com/wp-admin> và nhìn thấy màn hình dưới đây:



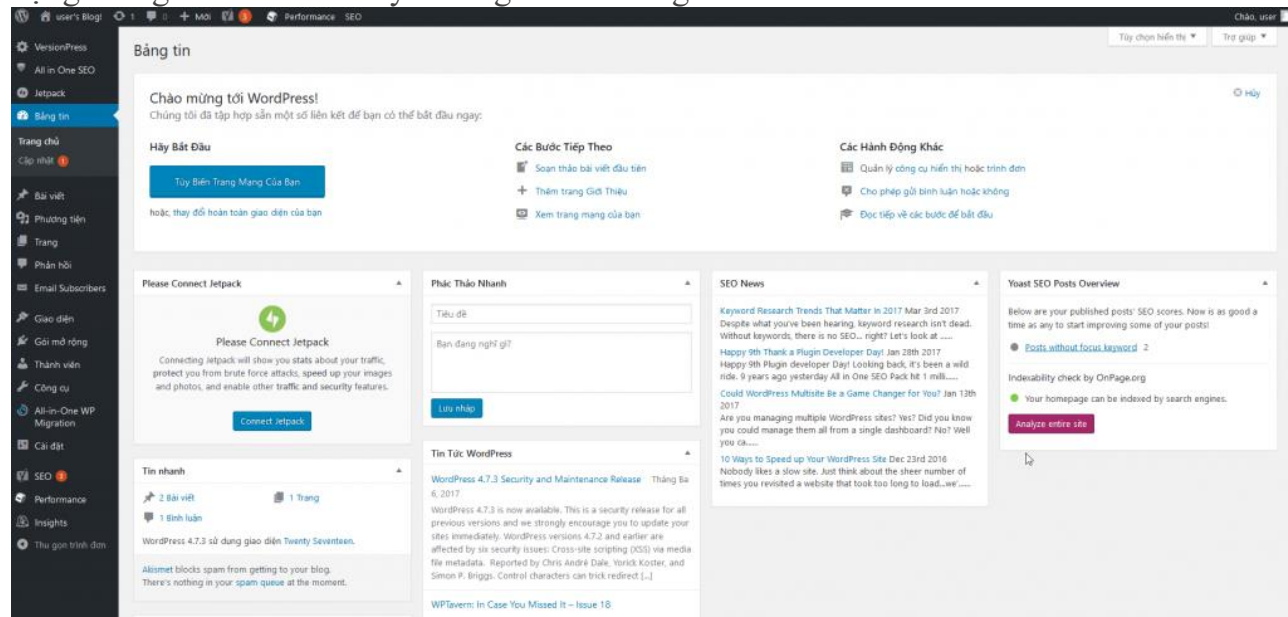
giao diện đăng nhập WordPress

Khi đã đăng nhập thành công, trang đầu tiên nhìn thấy sẽ là WordPress dashboard (bảng tin).

Đây là khu vực chính của quản trị viên WordPress nơi bạn cài đặt thông số kỹ thuật cho website, tạo bài viết và trang, cài đặt giao diện và plugin và nhiều tính năng khác nữa.

Trong lần cài đặt đầu tiên của WordPress, bạn sẽ nhìn thấy khu vực “Chào mừng tới WordPress” ngay trên đầu màn hình. Khu vực này chứa một số đường dẫn giúp bạn hiểu rõ hơn về [WordPress](#) cũng như cách sử dụng chúng.

Ở trong bài viết này mình sử dụng [WordPress Tiếng Việt](#), nếu các bạn không quen sử dụng Tiếng Anh có thể chuyển sang cho dễ dùng.



bảng tin WordPress

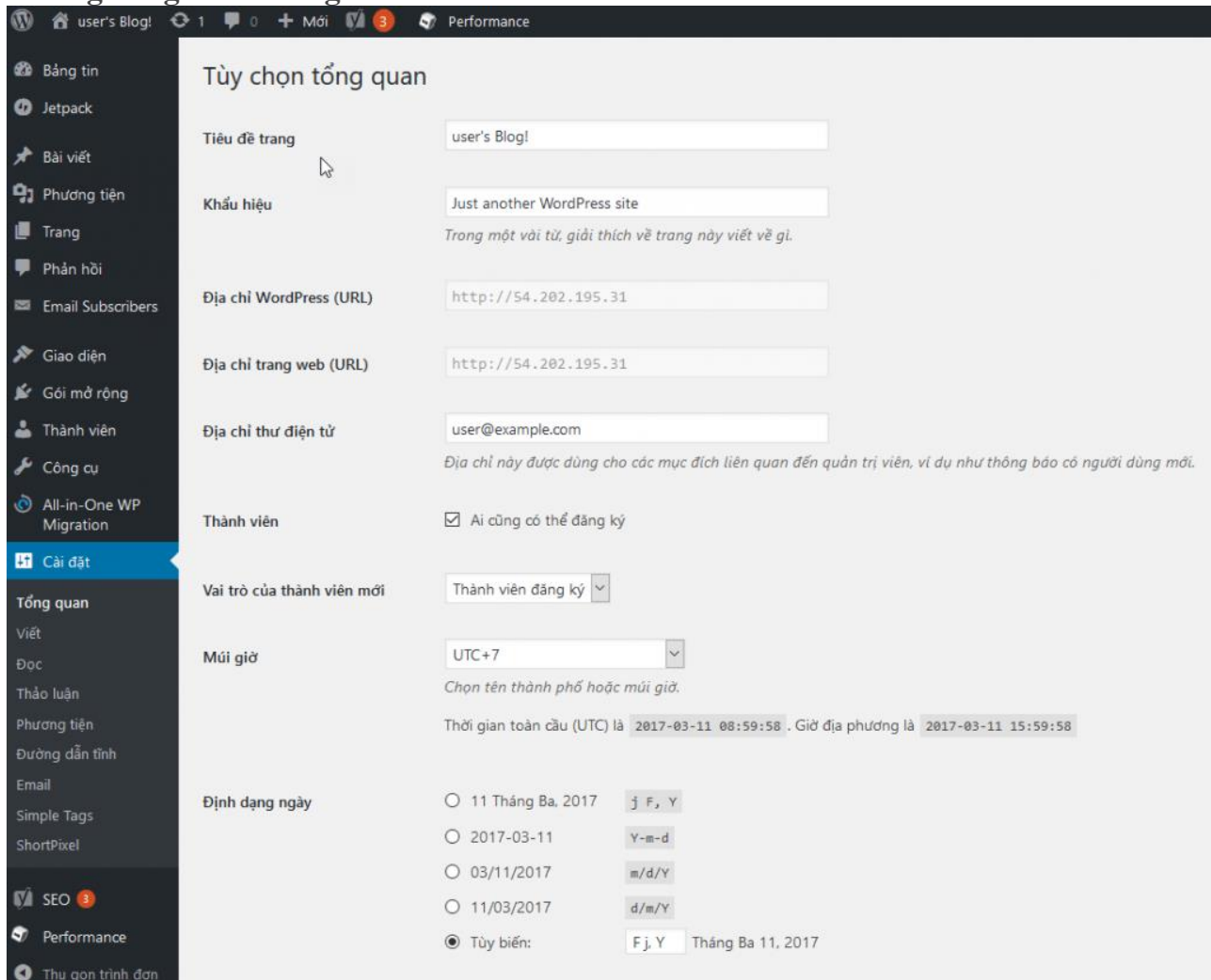
Bạn có thể quan sát, chiêm ngưỡng giao diện mặc định của WordPress được cài đặt ở ngoài trang chủ. Trong phần cuối của bài viết này, chúng tôi sẽ hướng dẫn bạn cách thay đổi giao diện của website bằng cách cài đặt một giao diện mới.

6.4.1. Cài đặt

Có một vài thông số cài đặt mà bạn cần phải quan tâm trong lần đầu tiên sử dụng WordPress. Đây là cơ hội tốt để bạn làm quen với hệ thống. Để bắt đầu cài đặt căn bản, trong danh mục bên trái, chọn **Cài đặt** > **Tổng quan** và bạn sẽ nhìn thấy một vài tùy chọn cho website:

- **Tiêu đề trang** (tên của website)
- **Khẩu hiệu** (miêu tả trang web)
- **Địa chỉ WordPress** (vị trí của website)
- **Địa chỉ trang web** (trang chủ của website nếu bạn cài đặt WordPress trong thư mục khác)
- **Địa chỉ thử điện tử** (email của quản trị viên)
- **Thành viên** (chọn nếu bạn muốn cho người khác đăng ký là thành viên của website)
- **Vai trò thành viên mới** (vai trò của người mới đăng ký trong website)
- **Múi giờ**
- **Định dạng ngày**

- Định dạng thời gian
- Tuần bắt đầu vào
- Ngôn ngữ của trang



cài đặt tổng quan WordPress

Bạn không nhất thiết phải thay đổi mọi thứ ở đây, nhưng hãy nhớ rằng có thể bạn sẽ cần đến chúng trong tương lai khi bạn [thay đổi tên miền](#), [thay đổi hosting](#) hay đổi tiêu đề.

6.4.2. Đường dẫn tĩnh

Việc tiếp theo cần phải làm là cài đặt “đường dẫn tĩnh” – cấu trúc URL mà website sẽ sử dụng. Mặc định của WordPress, cấu trúc này sẽ có dạng:

`http://www.yourdomain.com/?p=123`

Đây không phải là một cấu trúc tuyệt vời và không thân thiện với các công cụ tìm kiếm. Tuy nhiên, WordPress cung cấp cho người dùng rất nhiều lựa chọn có sẵn để thay đổi cấu trúc mặc định như:

<http://www.yourdomain.com/2015/08/19/sample-post/>

Trông đã tốt hơn nhiều rồi đúng không nào. Tuy nhiên, chúng tôi khuyên bạn nên sử dụng lựa chọn **Tên bài** và sử dụng tên bài viết để có đường dẫn ngắn nhất và đẹp nhất trong mắt người dùng:

<http://www.yourdomain.com/sample-post/>

Và đừng quên chọn nút “**Lưu thay đổi**”.

Tùy chọn đường dẫn tĩnh

WordPress mang đến cho bạn khả năng tùy chỉnh cấu trúc URL cho các đường dẫn và trang lưu trữ. Cấu trúc URL tùy chỉnh có thể

Cài Đặt Thông Dụng

- Bản thô `http://54.202.195.31/?p=123`
- Ngày và tiêu đề `http://54.202.195.31/2017/03/11/bai-mau/`
- Tháng và tiêu `http://54.202.195.31/2017/03/bai-mau/`
- Dạng số `http://54.202.195.31/luu-tru/123`
- Tên bài `http://54.202.195.31/bai-mau/`
- Tùy biến `http://54.202.195.31 /%year%/%monthnum%/%day%/%postname%/`

Tùy chọn thêm

Nếu bạn muốn, bạn có thể nhập các cấu trúc tùy biến cho các danh mục và thẻ URL tại đây. Ví dụ, sử dụng `topics` như cơ sở d được sử dụng.

Cơ sở cho Chuyên mục

Cơ sở của Thẻ

Lưu thay đổi

tùy chỉnh permalinks

6.4.3. Thảo luận

Bước tiếp theo với những trang WordPress mới là cài đặt thảo luận – tính năng cho phép người dùng bình luận trong website của bạn.

Đây là một tính năng tuyệt vời của WordPress nhưng cần phải được quản lý để tránh các spammer.

Để làm việc này, truy cập vào “Cài đặt” > “Thảo luận”. Chúng tôi khuyên bạn nên bỏ chọn “Cho phép liên kết các thông báo từ các blog khác (pingbacks và trackbacks) trên các bài viết mới” vì tính năng này sẽ gây nhiều phiền phức cho website sau này.

Để hiểu rõ hơn vấn đề này, chúng tôi đã miêu tả rất chi tiết trong bài viết [làm thế nào để vô hiệu hóa self pingbacks](#).

Những lựa chọn còn lại hãy cứ để mặc định nếu như bạn không biết chắc mình đang làm gì. Bạn có thể thay đổi chúng sau nếu như website bắt đầu có nhiều bình luận spam.

Tùy chọn thảo luận Trợ giúp

Tùy chọn mặc định cho bài viết/trang

- Gửi thông báo tới bất kỳ trang nhật ký trực tuyến nào được bài viết này liên kết tới
- Cho phép liên kết các thông báo từ các blog khác (pingbacks và trackbacks) trên các bài viết mới
- Cho phép mọi người gửi phản hồi cho bài viết mới
(Những tùy chọn này có thể được chỉnh riêng cho từng bài viết/trang.)

Các tùy chọn khác về phản hồi

- Tác giả của phản hồi phải cung cấp tên và thư điện tử.
- Thành viên phải đăng ký và đăng nhập để phản hồi.
- Tự động tắt chức năng gửi phản hồi nếu bài viết cũ hơn 14 ngày.
- Dùng chức năng chia lớp phản hồi với 5 lớp trả lời.
- Chia phản hồi thành nhiều trang với 50 phản hồi cấp cao nhất trong mỗi trang và trang cuối cùng được dùng làm trang đầu tiên.
Phản hồi được hiển thị với những phản hồi cũ hơn ở đầu mỗi trang.

Gửi thư điện tử cho tôi khi

- Có ai đó gửi phản hồi
- Có phản hồi chờ xét duyệt

Trước khi phản hồi được đăng

- Nhận xét phải chờ được kiểm duyệt
- Người gửi phản hồi đã từng có phản hồi được chấp nhận.

Xét duyệt phản hồi

Giữ một phản hồi trong hàng đợi nếu nó chứa 2 liên kết hoặc nhiều hơn. (Tính chất chung của các phản hồi spam là có nhiều liên kết).

Nếu chứa những từ này trong nội dung, tên, liên kết, thư điện tử hay IP, phản hồi sẽ bị đưa vào [danh sách chờ xét duyệt](#). Mỗi dòng một từ hoặc IP. Chuỗi kí tự nằm trong từ cũng được tính, ví dụ "press" cũng được tính là xuất hiện trong "WordPress".

Tùy chọn thảo luận

Ở phía cuối của trang web, có một vài lựa chọn cho ảnh đại diện. Nơi bạn nhìn thấy hình người nhỏ bên trái tên người. Bạn có thể chọn ảnh mặc định người cho các thành viên trong trường hợp họ không có bất kỳ ảnh đại diện nào hoặc có thể tắt chức năng đó nếu bạn muốn.

Ảnh đại diện

Hình ảnh đại diện sẽ theo bạn từ nhật ký trực tuyến này đến nhật ký khác, hiển thị bên cạnh tên của bạn khi bạn bình luận tại các trang mạng hỗ trợ hiển thị ảnh đại diện. Tại đây bạn có thể cho phép hiển thị ảnh đại diện của những người bình luận trên trang mạng của bạn.

Hiển thị ảnh đại diện Hiện ảnh đại diện

Đánh giá cao nhất

- G — Phù hợp cho mọi đối tượng
- PG — Cho người trên 13 tuổi trở lên
- PR — Cho người trên 18 tuổi
- X — Cho người trưởng thành

Ảnh đại diện mặc định

Với các thành viên không có ảnh đại diện, bạn có thể chọn hiển thị ảnh đại diện thay thế hoặc một ảnh đại diện được tạo ra tự động dựa trên địa chỉ thư điện tử của thành viên đó.

-  Hình người vô danh
- Trống
-  Logo của Gravatar
-  Identicon (tự động tạo)
-  Ảnh sóng (tự động tạo)
-  MonsterID (tự động tạo)
-  Quả Khứ (Mô phỏng)

ảnh đại diện bình luận

6.4.4. Hồ sơ của bạn

Một phần bạn không thể bỏ qua đó là trang hồ sơ cá nhân của bạn. Phần này nằm trong “Thành viên” > “Hồ sơ của bạn”, cho phép bạn cài đặt một vài tùy chọn trong quá trình sử dụng website và thông tin cá nhân của bạn.

Phần trên cùng của trang, có một lựa chọn cho phép bạn có thể tắt tính năng “Hiển thị”. Chúng tôi khuyên bạn không nên làm như vậy vì nó sẽ gây khó khăn trong quá trình soạn thảo bài viết.

Bạn có thể tùy chọn màu sắc hiển thị cho khu vực admin nếu thấy màu sắc mặc định quá nhàm chán. Hay ấn “Thanh công cụ” trong quá trình sử dụng website. Thanh công cụ được hiển thị trên cùng của trang web với một vài tính năng cho phép bạn chỉnh sửa bài viết, website một cách nhanh và thuận tiện nhất.

Cá nhân tôi khuyên bạn cứ giữ nguyên như vậy bởi tôi thích sự thuận tiện trong quá trình vận hành website của mình.

Lý lịch Trợ giúp ▾

Tùy chọn cá nhân

Hiển thị Không sử dụng khung soạn thảo nâng cao khi viết bài

Màu của trang quản lý

Mặc định Sáng Xanh Cà phê

Kính dị Đêm khuya Đại dương Bình minh

Phím tắt Sử dụng phím tắt cho việc xét duyệt phản hồi. [Thêm thông tin](#)

Thanh công cụ Hiển thị Thanh Công Cụ khi xem trang mạng

Ngôn ngữ

Tên

Tên đăng nhập Tên đăng nhập không thể đổi được.

Tên

Họ

Tên mạng (bắt buộc)

Tên hiển thị:

hồ sơ cá nhân của bạn

Ngay bên dưới những tùy chọn này, bạn có thể nhập tên, thông tin liên hệ, website cá nhân và đoạn miêu tả ngắn về bạn.

6.5. Các Chức Năng Chính Trong WordPress

Vậy là bạn đã khá quen thuộc với cách cài đặt wordpress, cấu hình chung của WordPress. Bây giờ là thời điểm sử dụng chúng vào những mục đích hữu ích hơn và xây dựng nội dung cho website của bạn. Trong phần này, tôi sẽ giới thiệu cho bạn cách đăng bài viết và trang với một vài tùy chọn mà bạn có thể sử dụng sau này.

Vậy bài viết là gì? và trang là gì? Chúng tôi đã chỉ ra [sự khác biệt giữa Post và Page trong WordPress](#) rất kỹ lưỡng trong bài viết trên.

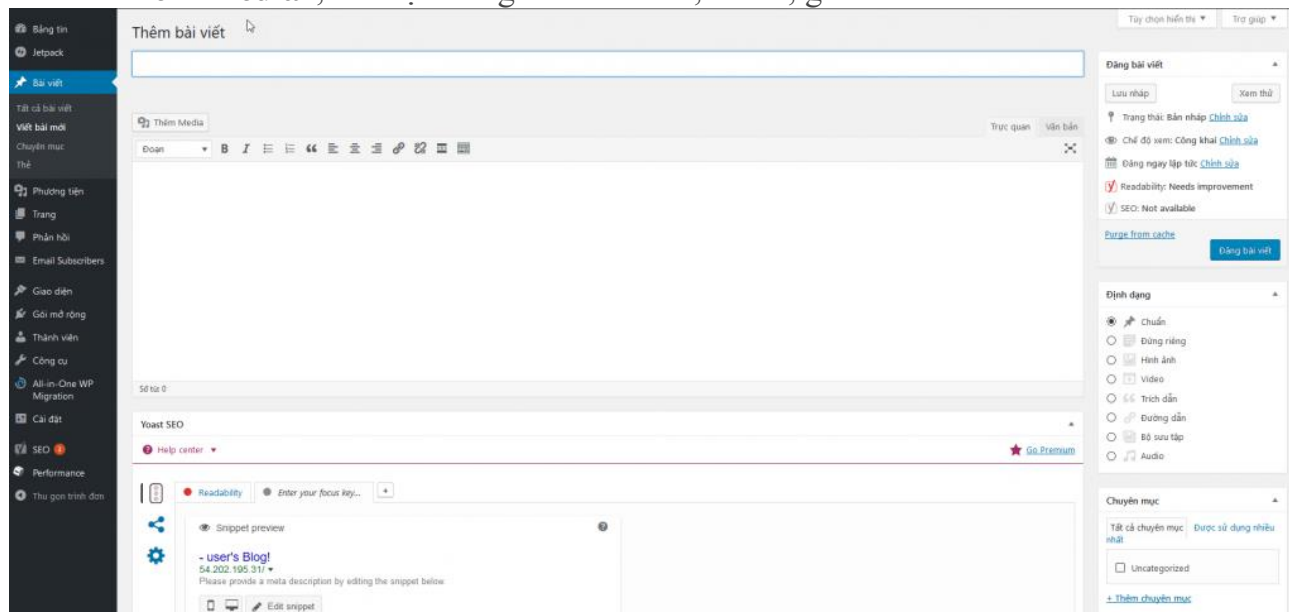
6.5.1. Soạn thảo bài viết

Chúng ta cùng đi đến trang soạn thảo bài viết, nơi bạn sẽ dành gần như toàn bộ thời gian để tạo ra những nội dung giá trị và công khai chúng trên trang web của mình.

Thậm chí, bạn cũng có thể đặt lịch xuất bản cho chúng vào thời điểm mà bạn muốn. Để làm được việc đó, hãy truy cập vào “**Bài viết**” > “**Viết bài mới**”.

Trang soạn thảo bài viết có chứa một ô trống cho phép bạn nhập tiêu đề bài viết và một ô to hơn nằm ngay bên dưới để nhập nội dung của bài viết.

Ngay phía trên, bạn sẽ thấy “trình soạn thảo nâng cao”, nơi cho phép bạn chỉnh sửa định dạng bài viết như bôi đậm, in nghiêng, thêm đường dẫn... Và phía trên công cụ soạn thảo là nút “**Thêm Media**“, nơi bạn dùng để thêm ảnh, video, gif...v.. cho bài viết.



thêm bài viết mới

Hãy trải nghiệm ngay bây giờ bằng cách nhập tiêu đề bài viết và một vài nội dung tùy thích, sau đó bấm “Lưu nháp”. Với chức năng lưu nháp bài viết đó sẽ được lưu tạm như một bản nháp để bạn có thể tiếp tục khi bài viết còn chưa hoàn thiện

Nếu muốn thêm đường dẫn, hãy chọn đoạn chữ mà bạn muốn gắn link cho chúng và chọn biểu tượng đường dẫn, rồi nhập đường dẫn bất kỳ bạn muốn. Bạn có thể chọn nút “Xem thử” để xem thành quả của mình trên trang web. Bạn yên tâm, bài viết chưa được công khai trong chế độ này.

Tại thời điểm này, phần lớn (hoặc tất cả) các công việc khó khăn sẽ được hoàn các tài liệu, và các giấy phép của mình tại cơ sở trung tâm kiểm soát, dựa cơ các lịch trình cập nhật thường xuyên và một hệ thống hỗ trợ khách hàng tại chỗ. Giờ là lúc để bạn nghĩ về marketing rồi đấy.

Mặc định, bạn sẽ có nhiều lựa chọn để định dạng bài viết trên thanh công cụ soạn thảo.

Khi bấm vào biểu tượng ở ngoài cùng bên tay phải trên thanh công cụ với tên gọi “Hiện hoặc ẩn thanh công cụ”, sẽ xuất hiện dòng thứ hai ngay bên dưới với nhiều lựa chọn hơn

cho định dạng bài viết.

Tại thời điểm này, phần lớn (hoặc tất cả) các công việc khó khăn sẽ được hoàn thành ở bên phát triển. Bạn có các sản phẩm demo, các tài liệu, và các giấy phép của mình tất cả đã trong tầm kiểm soát, bạn có các lịch trình cập nhật thường xuyên và một hệ thống hỗ trợ khách hàng tại chỗ. Giờ là lúc để bạn nghĩ về marketing rồi đây.

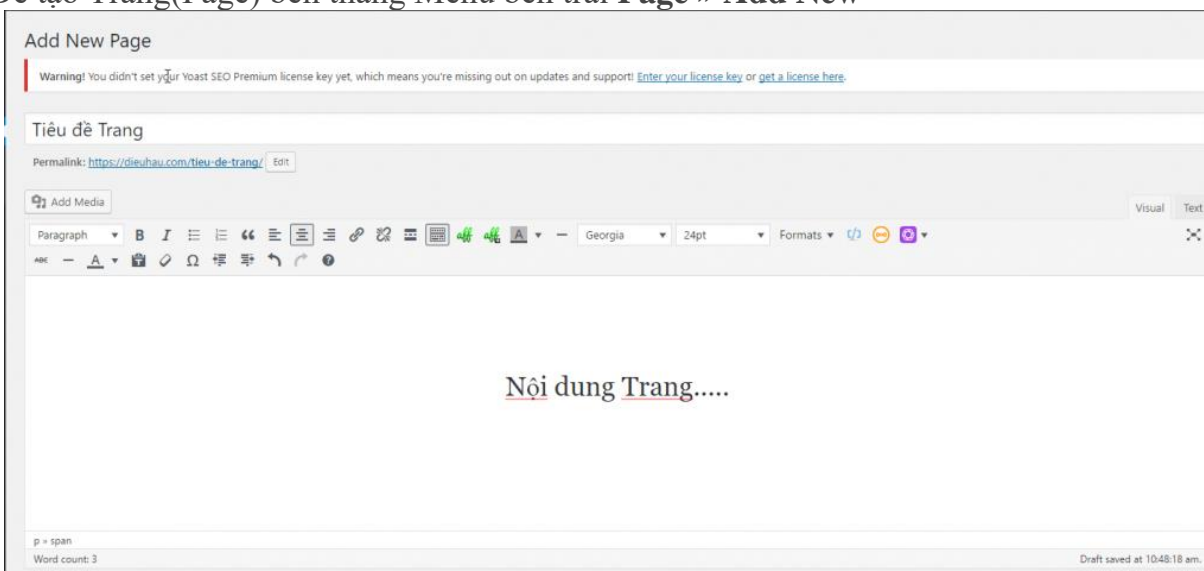
Một điều quan trọng nữa là đưa bài viết đến được nhiều người đọc. Một bài viết có hay đến đâu mà không ai đọc thì cũng thật vô ích phải không.

Vì vậy cho nên chúng ta phải viết bài chuẩn SEO. Điều đó giúp bài viết của bạn có thứ hạng cao hơn trong công cụ tìm kiếm, thứ hạng cao hơn đồng nghĩa việc có nhiều traffic hơn và được nhiều người biết đến hơn.

6.5.2. Hướng dẫn tạo trang (Page)

Như ở trên mình đã có bài giải thích sự khác biệt giữa Post và Page. Nói tóm lại Post mang tính cập nhật thay đổi liên tục, còn Page rất ít thay đổi, và có tính độc lập thường chỉ tạo một lần sử dụng mãi mãi như trang Liên hệ của website.

Đề tạo Trang(Page) bên thang Menu bên trái **Page » Add New**



Giống như tạo một bài viết mới sẽ có một khung soạn thảo văn bản hiện ra cho bạn. Nhập nội dung mà bạn muốn ở dưới khung

Ở bên tay phải sẽ có thêm một ô đó là Page Attributes (thuộc tính trang).

Thuộc tính trang ▲

cha mẹ

Giao diện

Thứ tự

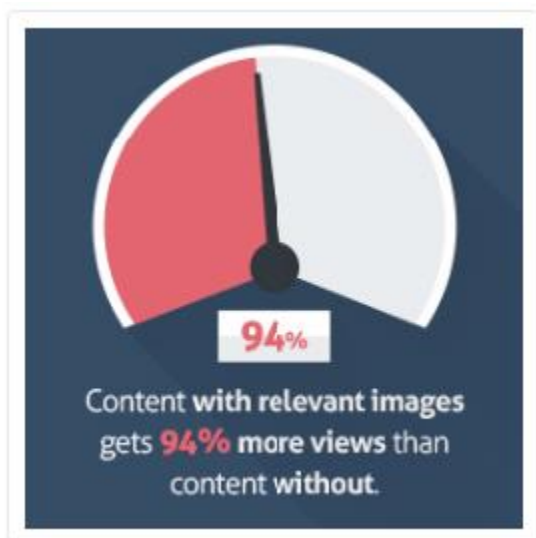
Cần giúp đỡ? Sử dụng thẻ Trợ giúp ở trên tiêu đề màn hình

Bạn có thể thêm trang mẹ con cho trang mới tạo: giống như việc tạo một thư mục con trên máy tính. Trang con sẽ có đường dẫn: tenmiencuaban/trangme/trangcon/

Ngoài ra bạn có thể chọn giao diện riêng cho trang.

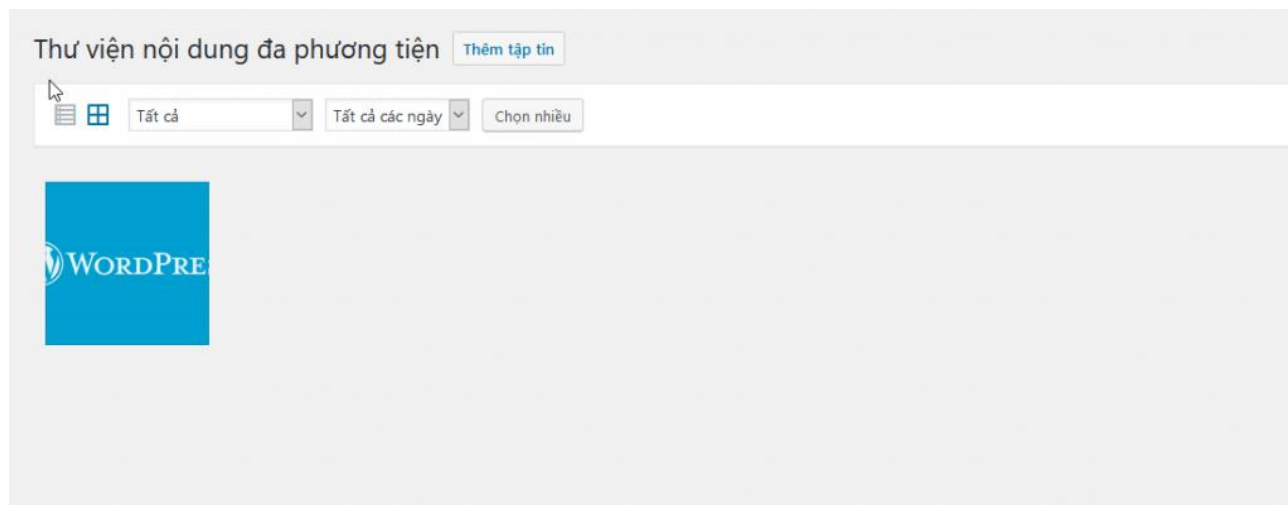
6.5.3. Quản lý thư viện Media

Phần quan trọng của bất kỳ blog chuyên nghiệp nào chính là hình ảnh. Đã có nhiều chứng minh rằng bài viết có hình ảnh có nhiều lượt view hơn những bài viết không có.



Tất cả hình ảnh, video và âm thanh trong WordPress đều được quản lý bởi “Thư viện Media”. Khi soạn thảo bài viết, bạn có thể truy cập vào thư viện để tải ảnh hoặc sử dụng hình ảnh có sẵn từ trước.

Trong trang soạn thảo bài viết, chọn nút “Thêm Media” phía trên thanh công cụ. Một popup xuất hiện và hiển thị toàn bộ hình ảnh mà bạn đã sử dụng trong website hoặc tải lên trước đó.

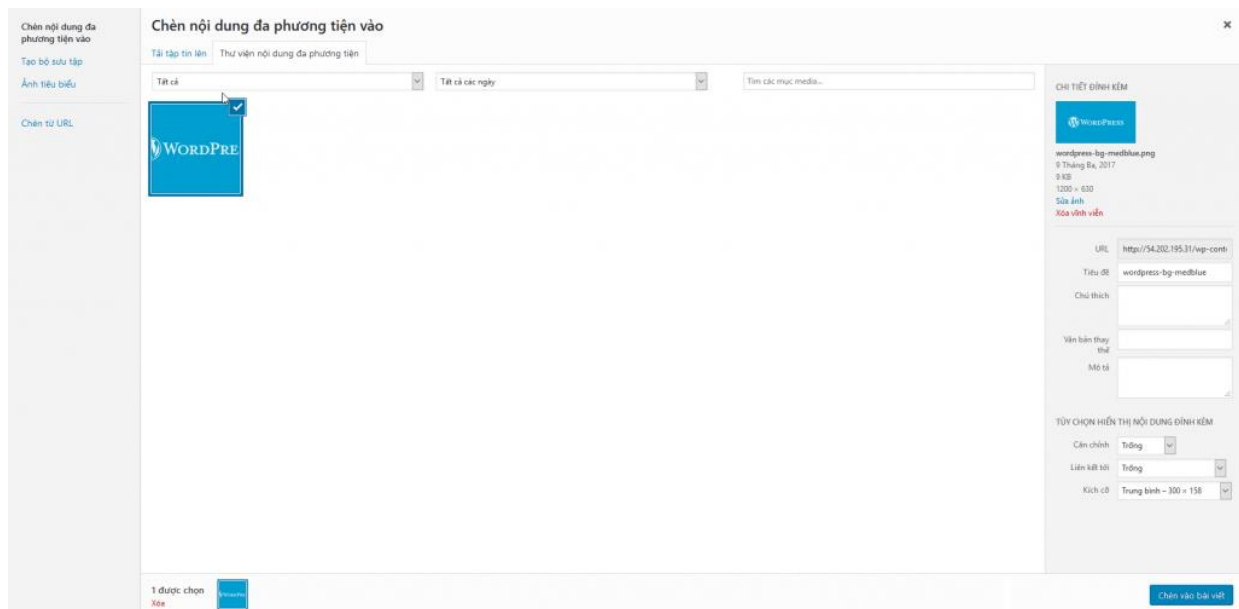


thư viện media

Để thêm ảnh mới, chọn tab “Tải tập tin lên”, sẽ mở ra chức năng tải file của WordPress.

Tại đây, bạn chỉ cần kéo thả một hoặc nhiều hình ảnh từ máy tính vào trong khu vực này hoặc bấm nút “Chọn tập tin”. Một khi ảnh đã được tải lên thành công, bạn sẽ nhìn thấy một vài tùy chọn bên tay phải màn hình.

Các tính năng này cho phép bạn đặt tiêu đề hay miêu tả cho bức ảnh nếu cần thiết, thẻ Alt và một vài tùy chọn hiển thị trong bài viết. Tham khảo bài viết để hiểu rõ hơn về sự [khác nhau giữa Alt Text Và tiêu đề hình ảnh](#).



chèn ảnh vào bài viết

Nhấn “Chèn vào bài viết” và hình ảnh đó sẽ được thêm vào bài viết. Nếu cần thay đổi bất cứ tùy chọn gì, chỉ việc chọn hình ảnh đó và thanh công cụ sẽ xuất hiện để bạn thực hiện việc đó.

Để học cách quản lý thư viện Media một cách chuyên nghiệp hãy đọc bài viết: [Hướng dẫn hoàn thiện về cách quản lý hình ảnh trong WordPress.](#)

Ngoài ra bạn cũng có thể [tạo và chèn Gallery hình ảnh trong trang WordPress](#), nếu bạn là một nhiếp ảnh gia thì tạo một Gallery ảnh sẽ giúp website trông rất chuyên nghiệp và đẹp mắt

6.5.4. Đăng bài viết

Vậy là bạn đã hoàn thành việc soạn thảo bài viết đầu tiên cho website, bao gồm cả việc thêm đường dẫn và hình ảnh trong đó, nên tôi tin chắc rằng bạn đã sẵn sàng công khai nội dung đó với toàn bộ thế giới. Nhưng trước khi đăng bài viết, còn một vài việc bạn phải hoàn thành trước đã.

Bên tay phải của màn hình soạn thảo, có một vài lựa chọn mà bạn nên hoàn tất. Bỏ qua mục đầu tiên “Đăng bài viết”, chúng ta sẽ quay lại đây cuối cùng.

Hãy lựa chọn chuyên mục cho bài viết từ danh sách đã được tạo trước đó. Nếu muốn, bạn cũng có thể tạo thêm chuyên mục mới ngay tại đây.






Điền một vài tag và bấm “Thêm” ngay sau đó. Và cuối cùng là ô hình ảnh đại diện. Tại đây, bạn sẽ tải lên hình ảnh liên kết với bài viết này. Mỗi một giao diện sẽ hiển thị hình

ảnh này một cách khác nhau, nhưng về cơ bản thì sẽ hiển thị một ảnh to phía trên hoặc bên cạnh tiêu đề bài viết.

Đăng bài viết

Lưu nháp









Xem thử

-  Trạng thái: Bản nháp [Chỉnh sửa](#)
-  Chế độ xem: Công khai [Chỉnh sửa](#)
-  Đăng ngay lập tức [Chỉnh sửa](#)
-  Readability: Needs improvement
-  SEO: Not available

[Purge from cache](#)

Đăng bài viết

Định dạng

-  Chuẩn
-  Đứng riêng
-  Hình ảnh
-  Video
-  Trích dẫn
-  Đường dẫn
-  Bộ sưu tập
-  Audio

Chuyên mục

Tất cả chuyên mục [Được sử dụng nhiều nhất](#)

Uncategorized

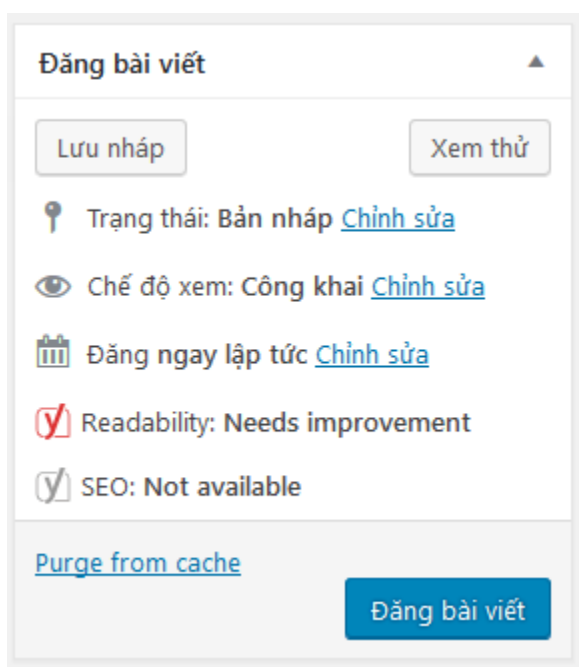
[+ Thêm chuyên mục](#)

Một khi bạn đã cảm thấy thoả mãn với những lựa chọn của mình, hãy quay lại khu vực trên cùng bên phải, nơi chúng ta đã bỏ qua lúc này “Đăng bài viết”.

Để công khai bài viết của bạn trên trang web, đơn giản là bấm nút “Đăng bài viết” và thế là xong. Trước khi làm việc đó, nếu bạn muốn trải nghiệm một vài [thu thuật wordpress](#) hay tính năng khác của WordPress như lập lịch bài đăng, thì bạn chỉ việc chọn “Chỉnh sửa” ngay bên cạnh đoạn “Đăng ngay lập tức”, bạn sẽ chọn được chính xác ngày và giờ mà bài viết sẽ được công khai.

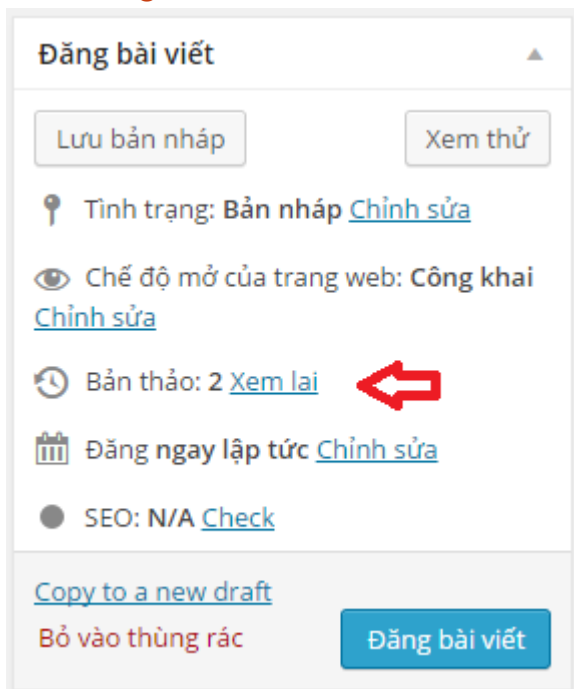
Nếu làm như vậy, nút “Đăng bài viết” sẽ tự động chuyển thành “Lên lịch” và khi đó bài viết của bạn sẽ được đăng vào đúng thời điểm mà bạn đã chọn trước đó. Đây là một tính năng tuyệt vời nếu bạn muốn thêm nhiều bài viết cùng lúc và phân chia thời gian đăng bài trong một khoảng thời gian khác nhau.

Trước khi đăng bài viết nhớ bấm “Xem thử” để xem mình còn sai sót gì trước khi công khai bài viết hay không



Giờ thì bạn đã biết làm thế nào để đăng bài viết trong WordPress website. Và trang cũng được thêm vào một cách tương tự như vậy. Truy cập “**Trang**” > “**Tạo mới**” và một màn hình soạn thảo tương tự hiện ra, tuy nhiên sẽ không có lựa chọn chuyên mục và tag. Trang cũng có tính năng lên lịch như bài viết.

Chức năng Revision



Revision (quản lý nhật ký soạn thảo) chức năng này cho phép bạn lưu lại những bản chỉnh sửa gần nhất của một bài viết khi lưu bản nháp.

Giúp bạn có thể dễ dàng quay lại bản chỉnh sửa cũ nếu muốn.

Bạn sẽ cần ít nhất 2 bản nháp để sử dụng tính năng này. Bấm vào Revision(Xem lại) để chọn phiên bản muốn khôi phục



6.5.5. Cách chèn video vào bài viết

Để chèn video vào bài viết rất đơn giản, bạn chỉ cần copy đường dẫn của video đó và dán vào bài viết là được.

Hoặc bạn có thêm bằng cách vào Thêm Media » Chèn từ URL. Nhập đường dẫn video (tốt nhất là từ youtube), rồi chọn chèn vào bài viết là được

Trong WordPress có tính năng Embed cho phép tự động nhận diện những liên kết từ video và chuyển nó thành dạng nhúng video. Vì vậy mà bạn chỉ cần copy paste đường dẫn video.

TÀI LIỆU THAM KHẢO

- [1] Chu Văn Hoàn, *Giáo trình thiết kế Web*.
- [2] *Giáo trình thiết kế web*, NXB Giáo dục, 2007.
- [3] Hoàng Mạnh Hùng, *Lập trình Web*, Đại học Đà Lạt.
- [4] Lê Minh Hoàng, *Tự học thiết kế Web tập 1 và tập 2*, NXB Lao động, 2007.
- [5] Trung tâm CNTT – Bộ giáo dục đào tạo, Tài liệu giảng dạy: *Thiết kế WEB với Front Page*, NXB Giáo dục, 2003.