

UBND TỈNH LÂM ĐỒNG
TRƯỜNG CAO ĐẲNG ĐÀ LẠT

GIÁO TRÌNH

MÔ ĐUN: THIẾT KẾ VÀ QUẢN TRỊ WEBSITE

NGÀNH/NGHỀ: CÔNG NGHỆ THÔNG TIN (ÚDPM)

TRÌNH ĐỘ: TRUNG CẤP

*(Ban hành kèm theo Quyết định số: /QĐ-CDNDL ngày ...tháng...năm...
của Hiệu trưởng Trường Cao đẳng Đà Lạt)*

LƯU HÀNH NỘI BỘ

Lâm Đồng, năm 2017

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Sau khi học xong mô đun này học viên có khả năng:

- Thiết kế được các trang web và trình bày tổng hợp trên trang web chính bao gồm thông tin, hình ảnh, âm thanh liên kết với các kiểu định dạng khác nhau.
- Sử dụng được các kỹ thuật màu sắc
- Xây dựng được hệ thống tổ chức thông tin từ các yêu cầu thực tế. Xây dựng được hệ thống Site Map phục vụ việc xây dựng trang Web
- Lập được bộ hồ sơ thiết kế về thi công trang Web

Nội dung chính của mô đun:

- Phác thảo kế hoạch xây dựng Web Site
- Thiết kế hệ thống thông tin Web Site
- Xây dựng Prototype và đánh giá hệ thống Web Site
- Tích hợp các phần tử căn bản của Web Site
- Tích hợp các phần tử Multimedia của Web Site
- Chọn công cụ và tổ chức Files
- Kiểm thử và chuyển tải Web Site lên Server.
- Sử dụng các công cụ thiết kế Web
- Tích hợp các phần tử có trong Web Site (hình ảnh, âm thanh, chữ từ các form, các database...) có hiệu quả tạo được tính logic và tính hấp dẫn của Web Site.

Thiết kế và quản trị website là mô đun bắt buộc. Mọi học viên phải học và đạt kết quả chấp nhận được đối với các bài kiểm tra đánh giá và thi kết thúc như đã đặt ra trong chương trình đào tạo.

Đà Lạt, ngày 07 tháng 07 năm 2017

Tham gia biên soạn

Chủ biên: Ths. Phạm Đình Nam

MỤC LỤC

TRANG

LỜI GIỚI THIỆU

BÀI 1: TỔNG QUAN VỀ WORLD WIDE WEB.....	1
1. Một số dịch vụ quan trọng trên Internet.....	1
1.1. Dịch vụ truyền file FTP.....	1
1.1.1. Mô hình hoạt động của FTP.	1
1.1.2. Tập hợp các lệnh FTP.	1
1.2. Dịch vụ thư điện tử Email.....	2
1.2.1 Khái niệm thư điện tử.	2
1.2.2 Hệ thống gửi thư điện tử.	2
1.2.3 Mở tài khoản gửi và nhận thư điện tử	3
1.2.4 Phương thức hoạt động của một hộp thư điện tử.	3
1.2.5 Các giao thức	4
1.3. Dịch vụ Telnet.....	4
1.4. Dịch vụ WWW / HTTP.....	5
1.4.1 WWW là gì?	5
1.4.2 Sự khác biệt của internet và www là gì?	6
1.4.3 Quá trình hình thành và phát triển của www là gì?	6
1.4.4 WWW là gì trong seo có ảnh hưởng gì tới trải nghiệm người dùng? .	7
1.5. Bộ định vị tài nguyên URL.....	8
1.5.1 URL là gì?	8
1.5.2 Định dạng và điều kiện của một URL là gì?	8
1.5.3 Định dạng URL chuẩn SEO là như nào?	8
2. Cấu trúc và hoạt động của dịch vụ WWW.....	9
2.1. Kiến trúc WWW	9
2.2. Web tĩnh và Web động	9
2.2.1. Web tĩnh là gì?	9
2.2.2. Web động là gì?	10
2.3. WebServer - WebBrowser.....	11
2.3.1 Web Server (máy chủ Web)	11
2.3.2 Web Browser (trình duyệt web)	12
BÀI 2: TẠO CÁC TRANG HTML TĨNH.....	14

1. Tạo và thực thi trang web HTML tĩnh	14
1.1 Tạo trang HTML đơn giản.....	14
1.2 Mở trang HTML	15
2. Đường dẫn tương đối và tuyệt đối	15
2.1 Liên kết là gì?	15
2.2 Đường dẫn tuyệt đối, đường dẫn tương đối.....	16
3. Thẻ HTML	17
3.1 Thẻ khai báo loại tập tin	17
3.2 Thẻ đóng mở tài liệu HTML.....	17
3.3 Thẻ đóng và mở phần thông tin website.....	17
3.4 Có thể đóng và mở phần nội dung website.....	18
4. Cấu trúc trang web HTML tĩnh.....	18
5. Định dạng văn bản.....	18
6. Heading	20
7. Siêu liên kết.....	20
8. Thành phần đa phương tiện.....	22
8.1 Chèn ảnh vào HTML	22
8.2 Chèn video	23
8.3 Chèn âm thanh – nhạc.....	23
8.4 Chèn đối tượng kỹ thuật số với thẻ <object>	23
8.5 Nhúng tài liệu HTML vào web.....	24
9. Danh sách	24
10. Bảng biểu.....	26
11. Biểu mẫu	30
BÀI 3: THIẾT KẾ GIAO DIỆN VỚI HTML VÀ CSS.....	31
1. Tổng quan về CSS.....	31
2. Thiết kế giao diện với HTML và CSS	35
2.1. CSS cơ bản.....	35
2.1.1. Kiểu chữ (font), cỡ chữ	36
2.1.2. Màu sắc	39
2.1.3. Định dạng trình bày cơ bản	41
2.2. CSS nâng cao	43
2.2.1. Siêu liên kết	43
2.2.2. Class và ID	45

2.2.3. Span và Div	47
BÀI 4: LẬP TRÌNH JAVASCRIPT	50
1. Tổng quan về Javascript.....	50
1.1. Giới thiệu chung	50
1.2. Vị trí đặt mã Javascript trong trang web HTML	50
1.3. HTML DOM.....	52
1.3.1. Thay đổi và lấy nội dung bên trong thẻ HTML	52
1.3.2. Thay đổi và lấy giá trị thuộc tính thẻ HTML bằng Javascript	53
2. Javascript cơ bản	54
2.1. Câu lệnh, khối lệnh, ghi chú	54
2.2. Kiểu dữ liệu.....	55
2.3. Các toán tử	57
2.4. Biến.....	58
2.4.1. Khai báo và gán giá trị cho biến	58
2.4.2. Sử dụng biến	60
2.4.3 Phạm vi của biến	60
2.5. Cấu trúc điều khiển	61
2.5.1. break	61
2.5.2. continue	61
2.5.3. do...while	62
2.5.4. for	63
2.5.5. for..in	64
2.5.6. function	65
2.5.7. if...else	65
2.5.8. return	66
2.5.9. switch	67
2.5.10. this	68
2.5.11. while	68
3. Hàm trong Javascript.....	69
3.1. Giới thiệu chung	69
3.2. Định nghĩa hàm Javascript.....	69
3.2.1. Định nghĩa hàm	69
3.2.2. Hàm có return và hàm không có return	70
3.2.3. Giá trị mặc định của tham số	71

3.3. Gọi hàm Javascript.....	72
3.4. Hàm định nghĩa sẵn của Javascript.....	72
3.4.1. substr	72
3.4.2. abs	72
3.4.3. atEnd	72
3.4.4. charAt	73
3.4.5. dimensions	73
3.4.6. eval	74
3.4.7. getDate	75
3.4.8. getItem	75
3.4.9. getTime	76
3.4.10. getDay	77
3.4.11. join	77
3.4.12. lbound	77
3.4.13. match	79
3.4.14. max	79
3.4.15. min	79
3.4.16. moveFirst	80
3.4.17. moveNext	80
3.4.18. replace	81
3.4.19. reverse	81
3.4.20. sort	82
3.4.21. split	82
3.4.22. substring	82
3.4.23. test	83
3.4.24. toArray	83
3.4.25. toLowerCase	84
3.4.26. toUpperCase	84
3.4.27. ubound	85
3.4.28. item	86
3.4.29. isNaN	86
4. Bài tập.....	87
BÀI 5: THIẾT KẾ VÀ QUẢN TRỊ WEBSITE VỚI CÔNG CỤ	93

1. Tổ chức và quản lý website.....	93
1.1. Phân tích cấu trúc website	93
1.2. Tạo hệ thống thư mục	94
1.3. Quản lý website	95
2.1. Làm quen với công cụ	96
2.1.1. Giao diện làm việc	96
2.1.2. Sửa thông số làm việc	98
2.2. Thiết kế trang web	100
2.2.1. Định dạng văn bản	100
2.2.2. Tạo siêu liên kết (Hyperlink) cho văn bản	102
3. Xuất bản website	104
3.1. Chọn dịch vụ hosting	104
3.1.1. Thuê host hay là mua host?	104
3.1.2 Chọn host Việt Nam hay host nước ngoài ?	104
3.2. Tải website lên host	107
3.2.1. Cách up web lên host qua File Manager của Hostinger	107
3.2.2. Cách up web lên host qua FTP	110
3.2.3. Cách up web lên host qua cPanel's File Manager	112
3.3. Chỉnh sửa các thông số hoạt động cho website	117
3.3.1. Trỏ tên miền về Name Server của Hosting	118
3.3.2. Trỏ tên miền về địa chỉ IP của Hosting	119

GIÁO TRÌNH MÔ ĐUN

Tên mô đun: Thiết kế và quản trị website

Mã mô đun: MD16

Vị trí, tính chất, ý nghĩa và vai trò của mô đun:

- Vị trí: Mô đun này học sau các môn học, mô đun cơ sở.
- Tính chất: Là mô đun chuyên môn bắt buộc.
- Ý nghĩa và vai trò của môn học/mô đun: Ngày nay internet đã phổ biến rộng khắp trên toàn thế giới, là một kênh thông tin, giải trí, mua bán, học tập được đa số dân chúng trên thế giới sử dụng. Vì vậy môn thiết kế và quản trị website là một môn không thể thiếu trong ngành tin học.

Mục tiêu của mô đun:

Về kiến thức:

- Trình bày được mô hình, cấu trúc và nguyên lý hoạt động của các website;
- Trình bày được cấu trúc một trang HTML và tính năng, cú pháp của các thẻ HTML;

Về kỹ năng:

- Áp dụng được các kỹ thuật thiết kế giao diện web với CSS;
- Áp dụng được các kỹ thuật lập trình hỗ trợ thiết kế web với ngôn ngữ javascript: xử lý dữ liệu, tương tác đối tượng;
- Xây dựng được các website có tính thẩm mỹ, trong đó có các ứng dụng phục vụ các mục tiêu tương tác dữ liệu cụ thể, có khả năng liên kết đến các trang web hay tài nguyên khác;
- Sử dụng được một hoặc một số công cụ thiết kế web như DreamWeaver, Komposer;
- Xuất bản được website lên internet và quản trị website;

Về năng lực tự chủ và trách nhiệm:

- Có khả năng tự nghiên cứu, tự học, tham khảo tài liệu liên quan đến môn học để vận dụng vào hoạt động học tập.
- Vận dụng được các kiến thức tự nghiên cứu, học tập và kiến thức, kỹ năng đã được học để hoàn thiện các kỹ năng liên quan đến môn học một cách khoa học, đúng quy định.

Nội dung của mô đun:

BÀI 1: TỔNG QUAN VỀ WORLD WIDE WEB

Mã Bài: MD16_01

Giới thiệu:

Khi xây dựng bất kỳ website nào, bước đầu tiên là xác định rõ mục đích của mình. Không có mục đích và nhiệm vụ rõ ràng, các dự định sẽ khó có thể đạt được kết quả như mong muốn. Lập kế hoạch cẩn thận kết hợp với việc đề ra mục tiêu rõ ràng là chìa khóa để xây dựng web site thành công.

Mục tiêu:

- Trình bày được môi trường hoạt động của các website;
- Trình bày được cấu trúc và các nguyên lý hoạt động của website;
- Mô tả mạch lạc được các mô hình hoạt động WWW, FTP, Email;
- Rèn luyện tính nghiêm túc, tích cực trong việc tiếp nhận kiến thức. Chủ động, tích cực tìm kiếm các nguồn tài liệu liên quan.

Nội dung chính:

1. Một số dịch vụ quan trọng trên Internet

1.1. Dịch vụ truyền file FTP

1.1.1. Mô hình hoạt động của FTP.

FTP (File Transfer Protocol) là một dịch vụ cho phép ta truyền tải file giữa hai máy tính ở xa dùng giao thức TCP/IP. FTP cũng là một ứng dụng theo mô hình client-server, nghĩa là máy làm FTP Server sẽ quản lý các kết nối và cung cấp dịch vụ tập tin cho các máy trạm. Nói tóm lại FTP Server thường là một máy tính phục vụ cho việc quảng bá các tập tin cho người dùng hoặc là một nơi cho phép người dùng chia sẻ tập tin với những người dùng khác trên Internet. Máy trạm muốn kết nối vào FTP Server thì phải được Server cấp cho một account có đầy đủ các thông tin như: địa chỉ máy Server (tên hoặc địa chỉ IP), username và password. Phần lớn các FTP Server cho phép các máy trạm kết nối vào mình thông qua account anonymous (account anonymous thường được truy cập với password rỗng). Các máy trạm có thể sử dụng các lệnh ftp đã tích hợp sẵn trong hệ điều hành hoặc phần mềm chuyên dụng khác để tương tác với máy FTP Server.

1.1.2. Tập hợp các lệnh FTP.

! Chạy chương trình command dos trên máy tính cục bộ.

? Hiển thị giúp đỡ của các lệnh Ftp, lệnh này giống với lệnh Help.

Append Chèn nội dung của một tập tin trên máy tính cục bộ vào cuối của một tập tin trên máy tính ở xa (máy FTP Server), dùng định dạng tập tin hiện tại.

Ascii Đặt loại định dạng truyền file là ASCII, giá trị này là mặc định khi khởi tạo kết nối FTP.

Bell Bật trạng thái chuông là on/off. Nếu là on thì sau mỗi lần lệnh truyền file hoàn thành thì máy phát ra tiếng chuông. Mặc định trạng thái này là off.

Binary Đặt loại định dạng truyền file là binary.

Bye Tắt kết nối với máy tính ở xa và thoát khỏi chương trình FTP.

Cd Thay đổi thư mục hiện thành trên máy ở xa(Server).

Close Ngừng phiên giao dịch với máy tính ở xa và trở về dòng lệnh của chương trình ftp.

Debug Bật trạng thái Debug on/off. Nếu là on thì mỗi lệnh gửi đến máy tính ở xa thì chương trình sẽ in ra các thông báo. Mặc định là trạng thái là off.

Delete Xóa tập tin trên máy tính ở xa.

Dir Hiển thị danh sách các tập tin và thư mục con trong thư mục hiện tại.

Disconnect Tắt kết nối với máy tính ở xa và trở về dòng lệnh FTP.

Get Chép một tập tin từ máy tính ở xa về máy tính cục bộ, dùng định dạng truyền file hiện tại.

Help Hiển thị giúp đỡ của các lệnh Ftp.

Lcd Thay đổi thư mục hiện trên máy tính cục bộ. Mặc định là thư mục đang làm việc trên máy tính cục bộ.

Ls Hiển thị danh sách các tập tin và thư mục con trong thư mục hiện tại.

Mdelete Xóa nhiều tập tin cùng trên một máy tính ở xa.

Mget Chép nhiều tập tin từ máy tính ở xa về máy tính cục bộ dùng định dạng truyền file hiện tại.

1.2. Dịch vụ thư điện tử Email

1.2.1 Khái niệm thư điện tử

Thư điện tử là dịch vụ chuyển thư dưới dạng số trên mạng máy tính thông qua các hộp thư điện tử.

1.2.2 Hệ thống gửi thư điện tử.

* Các bước gửi thư truyền thông:

B1: Người bỏ thư đã có địa chỉ chính xác của người nhận vào thùng thư.

B2: Nhân viên bưu điện tại Hà nội tập hợp mọi thư cần gửi vào TPHCM.

B3: Thư được chuyển vào TPHCM qua hệ thống vận chuyển của bưu điện.

B4: Nhân viên bưu điện tại TPHCM chuyển đến tay người nhận.

* Quá trình thực hiện gửi thư điện tử:

Các máy chủ được cài đặt phần mềm quản lý thư điện tử được gọi là máy chủ điện tử, sẽ là “bưu điện” còn hệ thống vận chuyển của “bưu điện” chính là mạng máy tính. Cả người gửi và người nhận đều sử dụng máy tính với các phần mềm thích hợp để soạn và gửi, nhận thư.

1.2.3 Mở tài khoản gửi và nhận thư điện tử

a. Mở tài khoản thư điện tử:

* Quá trình thực hiện gửi thư:

- Sử dụng Yahoo, google... để mở tài khoản thư điện tử.
- Cung cấp 1 hộp thư điện tử trên máy chủ điện tử
- Cùng với một hộp thư, người dùng có tên đăng nhập và mật khẩu dùng để truy cập thư điện tử. Hộp thư được gán với một địa chỉ thư điện tử.

* Cấu trúc chung của một hộp thư:

Một địa chỉ email sẽ bao gồm ba phần chính có dạng :

Tên_định_dạng_thêm_tên_email@tên_miền

b. Nhận và gửi thư:

* Các bước truy cập vào hộp thư điện tử:

B1: Truy cập trang web cung cấp dịch vụ thư điện tử

B2: Đăng nhập vào hộp thư điện tử bằng cách gõ tên đăng nhập (tên người dùng), mật khẩu rồi nhấn Enter (hoặc nháy vào nút đăng nhập)

* Chức năng của hộp thư điện tử:

- Mở và xem danh sách các thư đã nhận và đã lưu trong hộp thư.
- Mở và đọc nội dung của một bức thư cụ thể.
- Soạn thư hoặc gửi thư cho một hoặc nhiều người.
- Trả lời thư.
- Chuyển tiếp thư cho một người khác.

1.2.4 Phương thức hoạt động của một hộp thư điện tử.

Hoạt động của hệ thống email hiện nay có thể được minh họa qua phân tích một thí dụ như sau:

B1. Nguyễn dùng **MUA** của mình để soạn một lá thư có **địa chỉ** người nhận là Trần với địa chỉ là Tran@b.org. Nguyễn nhấn nút *Send* và phần mềm thư điện tử của Nguyễn áp dụng SMTP để gửi mẫu thông tin (lá thư) đến **MTA**, hay máy chủ thư điện tử, của Nguyễn. Trong thí dụ thì máy chủ này là smtp.a.org được cung cấp từ **dịch vụ Internet** của Nguyễn.

B2. MTA này sẽ đọc địa chỉ chỗ nhận (tran@b.org) và dựa vào **phần tên miền** nó sẽ tìm hỏi địa chỉ của tên miền này, nơi có máy chủ sẽ nhận email gửi đến, qua **Hệ thống Tên miền**.

B3. Máy chủ **DNS** của b.org là ns.b.org sẽ trả lời về một **bản ghi trao đổi thư từ**, đây là bảng ghi chỉ ra cách thức làm thế nào **định tuyến** cho email này. Trong thí dụ thì mx.b.org là máy chủ từ dịch vụ cung ứng Internet của Trần.

B4. smtp.a.org gửi mẫu thông tin tới mx.b.org dùng giao thức SMTP, điều này sẽ phân phối lá thư đến hộp thư của Trần.

B5. Khi đọc Trần ra lệnh nhận thư trên máy (MUA) của Trần, điều này tạo ra việc lấy về mẫu thông tin bằng cách áp dụng giao thức POP3.

1.2.5 Các giao thức

SMTP (từ chữ *Simple Mail Transfer Protocol*) -- hay là giao thức chuyển thư đơn giản. Đây là một **giao thức** lo về việc vận chuyển email giữa các **máy chủ** trên đường trung chuyển đến địa chỉ nhận cũng như là lo việc chuyển thư điện tử từ máy khách đến máy chủ.

IMAP (từ chữ *Internet Message Access Protocol*) -- hay là giao thức truy nhập thông điệp (từ) Internet. Giao thức này cho phép truy nhập và quản lý các mẫu thông tin về từ các máy chủ. Với giao thức này người dùng email có thể đọc, tạo ra, thay đổi, hay xoá các ngăn chứa, các mẫu tin đồng thời có thể tìm kiếm các nội dung trong hộp thư.

POP (từ chữ *Post Office Protocol*) -- hay là giao thức phòng thư. Giao thức này được dùng để truy tìm các email từ một MTA.

1.3. Dịch vụ Telnet

TELNET (viết tắt của TERminal NETwork) là một giao thức mạng (network protocol) được dùng trên các kết nối với Internet hoặc các kết nối tại mạng máy tính cục bộ LAN.

Thời gian dài trước khi xuất hiện máy tính với giao diện đồ họa tinh vi, người ta sử dụng các hệ thống dựa trên nền văn bản (text-based), thường là các đầu cuối hiển thị kết nối vào máy tính trung tâm.

Telnet cung cấp một phương pháp chuẩn để mô phỏng các thiết bị đầu cuối text-based thông qua mạng dữ liệu. Cả giao thức và phần mềm client sử dụng giao thức đều được gọi chung là Telnet. Kết nối sử dụng Telnet được gọi là phiên VTY (Virtual Terminal) hoặc kết nối VTY.

– Để hỗ trợ kết nối Telnet client, server chạy một dịch vụ gọi là Telnet daemon.

- Một kết nối VTY được thiết lập từ thiết bị đầu cuối sử dụng phần mềm Telnet client. Đa số các hệ điều hành đều cung cấp phần mềm Telnet client.
- Trong máy PC chạy windows có thể chạy Telnet từ dấu nhắc lệnh.
- Một số ứng dụng thông dụng khác có thể làm việc như Telnet client: HyperTerminal, Minicom, TeraTerm.

Telnet là một giao thức chủ/khách, quy định cách thiết lập và hủy bỏ một phiên VTY. Mỗi lệnh Telnet chứa ít nhất 2 byte:

- Byte đầu chứa một ký tự đặc biệt gọi là ký tự IAC (Interpret as Command).
- Byte còn lại là lệnh.

Một số lệnh Telnet:

- AYT (Are You There) – cho phép người dùng yêu cầu một số thông tin xuất hiện trên màn hình terminal để xác định phiên VTY đang được kích hoạt.
- EL (Erase Line) – xóa toàn bộ text từ dòng hiện thời.

Giao thức Telnet hỗ trợ xác thực người dùng nhưng không hỗ trợ truyền dữ liệu được mã hóa:

- Dữ liệu trao đổi trong các phiên Telnet được truyền dưới dạng văn bản (plaintext) qua mạng.
- Nếu cần bảo mật, giao thức SSH (Secure Shell) có thể cung cấp giải pháp an toàn cho truy cập máy chủ.
- SSH hỗ trợ xác thực mạnh hơn Telnet, hỗ trợ truyền dữ liệu các phiên sử dụng dữ liệu được mã hóa.

1.4. Dịch vụ WWW / HTTP

1.4.1 WWW là gì?

WWW là viết tắt của 3 chữ "World wide web" là một thuật ngữ trong Tiếng Anh dịch ra có nghĩa là hệ thống các trang web trên toàn thế giới và là một phần của hệ thống internet. Ví dụ: www.vietadsgroup.vn là một trang web thuộc hệ thống www (World wide web). Hoặc www.facebook.com cũng là một trang web, google.com.

Các tài liệu trên World Wide Web được lưu trữ trong một hệ thống siêu văn bản (hypertext), đặt tại các máy tính trong mạng Internet. Người dùng phải sử dụng một chương trình được gọi là trình duyệt web (web browser) để xem siêu văn bản. Chương trình này sẽ nhận thông tin (documents) tại ô địa chỉ (address) do người sử dụng yêu cầu (thông tin trong ô địa chỉ được gọi là tên miền (domain name)), rồi sau đó chương trình sẽ tự động gửi thông tin đến máy chủ (web server) và hiển thị trên màn hình máy tính của người xem. Người dùng có thể theo các liên kết siêu

văn bản (hyperlink) trên mỗi trang web để nối với các tài liệu khác hoặc gửi thông tin phản hồi theo máy chủ trong một quá trình tương tác. Hoạt động truy tìm theo các siêu liên kết thường được gọi là duyệt Web.



Hình 1: WWW là hệ thống các trang web trên toàn cầu

1.4.2 Sự khác biệt của internet và www là gì?

Internet nó là một hệ thống đồ sộ gồm các mạng và cơ sở hạ tầng mạng. Ta có thể hiểu internet giúp kết nối hàng triệu máy tính trên toàn cầu lại với nhau tạo thành một hệ thống có thể liên lạc được với nhau.

Thông tin liên lạc được truyền đi trên internet nó qua một giao thức có tên là protocol, còn www là gì? nó truy xuất thông tin trên khắp môi trường internet thông qua giao thức HTTP chỉ một trong các ngôn ngữ được sử dụng trên internet.

Vì vậy ta có thể hiểu rằng web nó chỉ là một cách thức để phổ biến thông tin trên internet. Còn bản thân internet nó cũng có khả năng giao dịch như dùng e-mail với giao thức SMTP. Có nhiều ứng dụng gọi là trình duyệt Web, như IE, Opera, Netscape... giúp bạn dễ dàng truy xuất các trang Web hơn.

1.4.3 Quá trình hình thành và phát triển của www là gì?

www được phát triển bởi Tim Berners Lee là người đứng đầu và sáng lập ra đại tổ hợp world wide web consortium hay còn gọi là W3C.

W3C có nhiệm vụ phát triển các công nghệ có khả năng hoạt động toàn cầu để phát huy hết tiềm năng của Web. Ông đã được trao giải thưởng thiên niên kỷ vì sự cống hiến thành tựu cho nhân loại.

Web chính thức đã được giới thiệu với công chúng vào năm 1991 khi mà Berners-Lee đã hình thành máy chủ đầu tiên, viết trình duyệt Web đầu tiên, và phát minh ra các giao thức dùng cho hoạt động của Web: địa chỉ URL giao thức HTTP và bộ mã HTML. Năm 2003, Berners-Lee đã được Hoàng gia Anh phong tước Hiệp sĩ.



Hình 2: Quá trình hình thành và phát triển của www

1.4.4 WWW là gì trong seo có ảnh hưởng gì tới trải nghiệm người dùng?

Gây khó khăn, mất thời gian cho người dùng khi sử dụng www gây bất tiện trong khi gõ địa chỉ website. Sử dụng www trong URL sẽ làm cho đường dẫn trở nên không thân thiện và dài hơn. Sử dụng www còn gây hiểu nhầm cho google bởi lẽ google hiểu www.abc.com và http://abc.com là 2 website có 2 địa chỉ khác nhau nhưng trùng lặp nội dung.

Nên trong trường hợp website của bạn sử dụng www rất khó seo web, trường hợp này đặc biệt lưu ý khi bạn là cá nhân hay chủ doanh nghiệp kinh doanh online bán lẻ, cá nhân bán hàng qua mạng.



Hình 3: www có tác động gì trong seo

Như vậy WWW là hệ thống các trang web trên toàn thế giới và là một phần của hệ thống internet. Các tài liệu trên World Wide Web được lưu trữ trong một hệ thống siêu văn bản (hypertext), đặt tại các máy tính trong mạng Internet. Người dùng phải sử dụng một chương trình được gọi là trình duyệt web (web browser) để xem siêu văn bản.

1.5. Bộ định vị tài nguyên URL

1.5.1 URL là gì?

URL là viết tắt tiếng Anh của (Uniform Resource Locator) được gọi là "định vị tài nguyên thống nhất" hoặc "định vị tài nguyên xác định". URL dùng để định vị địa chỉ site trên Internet, có thể là địa chỉ của một website, một webpage, một wap, wappage hoặc một bài đăng cụ thể nào đó trên Website.

Nói cách khác, URL là địa chỉ của một tài nguyên trên mạng Internet. URL có thể là một website, một trang web hay một hình ảnh,...v.v URL có cấu trúc phân cấp giống như thư mục "folder" và tài liệu "file" trên máy tính. Ví dụ URL <https://esm.vn>

URL thường xuất hiện ở trên tab trình duyệt, trên các công cụ tìm kiếm: Google, Yahoo Search, Cốc Cốc Search hay là trên chính website của bạn,...v.v

1.5.2 Định dạng và điều kiện của một URL là gì?

- URL phải ít hơn 2048 ký tự để được hiển thị trên trình duyệt Chrom, Cốc cốc, Firefox, IE,...v.v
- Sử dụng " / " để ngăn cách thư mục, dấu " - " giữa các từ.
- Không nên dùng dấu " _ " hay dấu cách
- Không nên dùng tiếng việt có dấu như: vietadsgroup.vn/quang-cao-google.aspx.

1.5.3 Định dạng URL chuẩn SEO là như nào?

Để nâng cao thứ hạng của website trong bộ máy tìm kiếm Google, có rất nhiều điều phải làm đặc biệt là việc tối ưu hóa đường dẫn URL sao cho thân thiện Google.

URL phải rõ ràng và mô tả đúng sản phẩm, dịch vụ. Các ký tự trong URL phải cách nhau bằng dấu " - " cho dịch vụ đó.

Ví dụ: Dịch vụ "quảng cáo google adwords" của VietAds phải viết sang URL chuẩn SEO là "quang-cao-google-adwords".

<https://vietadsgroup.vn/quang-cao-google-adwords.aspx>

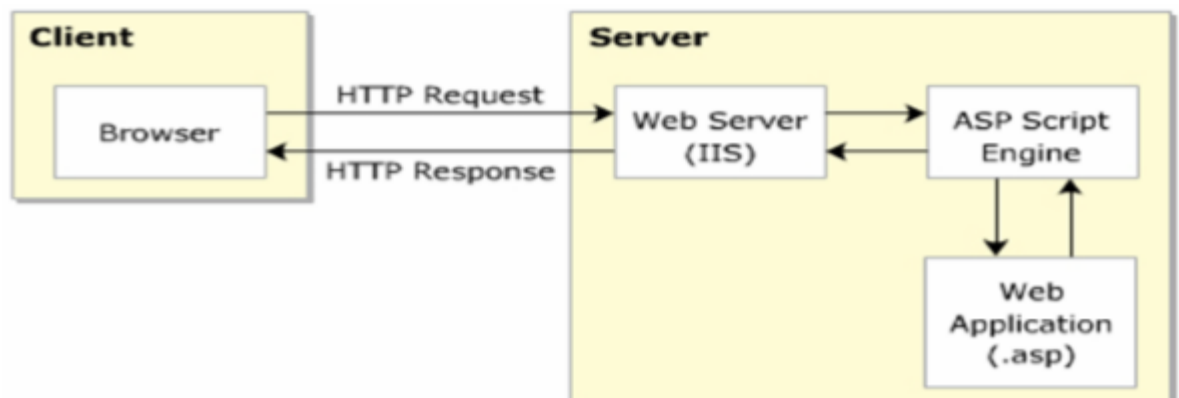
Trong SEO, URL có thể đóng vai trò là một Anchor Text của Liên kết. Bởi vậy, một URL chuẩn SEO phải luôn chứa từ khóa, ngắn gọn, có ý nghĩa, dễ copy,

dễ nhớ, không chứa những từ không cần thiết, không chứa các ký tự đặc biệt, url không có dấu,...v.v

2. Cấu trúc và hoạt động của dịch vụ WWW

2.1. Kiến trúc WWW

Dịch vụ WWW được xây dựng theo mô hình Client/Server, tức là người ta sẽ thiết lập một máy chủ phục vụ cho việc lưu trữ các tài liệu HyperText gọi là WebServer. Phía người sử dụng sẽ có một máy tính cùng với phần mềm có khả năng hiểu được các tài liệu HyperText và giao tiếp được với WebServer gọi là Web Browser hay Web Client. Khi người sử dụng có một yêu cầu đến một tài liệu nào đó bằng cách gửi WebServer một URL thì WebServer sẽ phục vụ tài liệu đó và Web Browser sẽ hiển thị tài liệu đó lên màn hình với khuôn dạng thích hợp.



2.2. Web tĩnh và Web động

2.2.1. Web tĩnh là gì?

Web tĩnh là những website không có hệ thống quản lý nội dung và người dùng không thể chỉnh sửa được hoặc có thể chỉnh sửa, thay đổi dữ liệu được. Hiện nay web tĩnh hầu như ít tồn tại hoặc chỉ tồn tại với những công ty chuyên về thiết kế website, họ sử dụng web tĩnh bởi họ có kiến thức và dễ dàng chỉnh sửa nội dung khi cần.

Về kiến thức cơ bản thì web tĩnh thường được xây dựng từ CSS, HTML, JAVASCRIPT (DHTML), hiện nay có thêm công nghệ HTML5 & CSS3. Có thể bạn nhìn một web nào đó nhìn rất lung lay, đẹp và nội dung có thể thay đổi nhưng chưa chắc nó là web động bởi vì người ta có thể sử dụng DHTML để thay đổi nội dung (tại client).

Nếu bạn chạy trên máy tính của bạn thì với web tĩnh bạn có thể đặt tại đâu cũng chạy được vì bản chất nó chỉ là một file bình thường.

Ưu điểm của web tĩnh:

- Về giao diện Designer có thể thiết kế theo kiểu mới lạ
- Tốc độ truy cập nhanh bởi nó chỉ là những file HTML
- Chi phí đầu tư thấp bởi bạn không phải trả tiền nhiều cho Coder
- Thân thiện với bộ máy tìm kiếm bởi bạn có thể đặt tên file tùy ý (ten-file.html, tieu-de-tin-tuc.html)

Nhược điểm của web tĩnh:

- Khó quản lý nội dung
- Khó nâng cấp bảo trì
- Mỗi khi thay đổi phải vào file HTML, CSS hoặc JAVASCRIPT để chỉnh sửa

Khi nào nên sử dụng web tĩnh?

- Nếu bạn là doanh nghiệp muốn tự mình làm website thì bạn có thể học các kiến thức căn bản và tự làm một Web tĩnh cho mình
- Nội dung website ít khi cập nhật và ít nên bạn muốn tiết kiệm chi phí
- Website bạn nhỏ và bạn thuê luôn người chuyên về web để quản trị

Trên thực tế thì mình thấy hiện nay chỉ có các công ty thiết kế web họ mới làm web tĩnh bởi vì họ có khả năng chỉnh sửa được chứ khách hàng thì họ không có khả năng đó. Hiện nay có các trang web ONE PAGE có các hiệu ứng slide khá độc nên cũng có thể họ sử dụng web tĩnh hoặc tĩnh một phần trong đó.

2.2.2. Web động là gì?

Ngược lại với web tĩnh, web động là những website có hệ thống quản lý nội dung và người dùng có thể chỉnh sửa nội dung được. Ví dụ như trang vnexpress.net hay 24h.com.vn thì đó là những web động bởi vì họ có thể đăng tin mới, chỉnh sửa tin mới và chỉnh sửa danh mục menu, ...

Web động sử dụng các công nghệ như HTML, CSS, HTML5, CSS3, JAVASCRIPT, ... và điều đặc biệt là có sử dụng một ngôn ngữ lập trình server như PHP, một hệ quản trị cơ sở dữ liệu như MySQL, vì vậy web động phải chạy trong máy chủ. Nếu đặt trên máy tính của bạn thì bạn phải cài các Web server ảo như vertrigo, Xampp, wampp, ...

Ưu điểm của web động:

- Dễ dàng quản lý nội dung
- Dễ dàng nâng cấp và bảo trì
- Có thể xây dựng được web lớn

- Thường sử dụng tương tác với người dùng cao
- Hiện nay vẫn SEO tốt bởi ta có thể sử dụng chức năng Rewrite URL để chuyển URL thân thiện

Nhược điểm của web động:

- Chi phí xây dựng cao
- Nếu web lớn có thể cần thêm nhân sự chuyên ngành

Khi nào nên sử dụng web động?

- Bạn làm website thương mại điện tử bán hàng
- Bạn làm web giới thiệu sản phẩm công ty
- Bạn làm web tin tức, blog cá nhân
- Web bạn tầm cỡ lớn

Nhưng nhìn chung thì hiện nay đa số nhà phát triển chọn giải pháp làm web động sẽ tốt hơn.

2.3. WebServer - WebBrowser

2.3.1 Web Server (máy chủ Web)

Web Server là máy chủ trong đó chứa thông tin dưới dạng trang Web (trang HTML có thể chứa âm thanh, hình ảnh, video, văn bản, ...). Các Web Server được kết nối với nhau thông qua mạng Internet, mỗi Server có địa chỉ duy nhất trên Internet.

Thành phần chủ chốt của Web Server là phần mềm. Mỗi phần mềm Web Server chạy trên một nền tảng phần cứng và một hệ điều hành cụ thể. Một Web Server phải có cấu hình đủ mạnh để cung cấp các dịch vụ cho các client, đáp ứng đồng thời nhiều yêu cầu từ client và có khả năng lưu trữ lớn cho tài nguyên Web.

Nói về chức năng và hiệu năng, các Web Server phân thành 4 nhóm chính:

Các máy chủ truyền thông thông thường.

Máy chủ thương mại.

Máy chủ nhóm làm việc.

Máy chủ dùng cho mục đích đặc biệt.

Các tiêu chuẩn đánh giá một Web Server:

Hiệu năng: nền tảng hệ điều hành và xử lý đa luồng.

Bảo mật: Thông qua địa chỉ IP, tên máy chủ của mạng con, thư mục ...

Truy nhập và tích hợp CSDL: Hầu hết các Web Server đều sử dụng giao diện CGI, một số khác thì dùng giao diện lập trình ứng dụng (API) hoặc ngôn ngữ hỏi đáp có cấu trúc SQL.

Quản lý và quản trị Web Server: Đặc tính quan trọng của tiêu chuẩn này là khả năng quản trị từ xa, giao diện đồ họa và điều khiển cấu hình của máy chủ.

2.3.2 Web Browser (trình duyệt web)

Web Browser là một công cụ hay chương trình cho phép truy xuất và xem thông tin trên Web. Có nhiều Web Browser để truy xuất Web, mỗi trình duyệt có những đặc điểm khác nhau và chúng hiển thị những trang Web không hoàn toàn giống nhau.



Các trình duyệt web bao gồm có Internet Explorer, Netscape Navigator Communicator, Opera, Mozilla Firefox, ... Tất cả các loại trình duyệt này đều có các phiên bản khác nhau, và các phiên bản mới nhất sẽ có nhiều tính năng hơn các phiên bản trước đó. Ngoài việc truy xuất Web, các trình duyệt còn cho phép chúng ta thực hiện các công việc khác như: gửi nhận email, tải các tập tin từ Web Server về, ... thông qua các Add-on và Plugin của trình duyệt.

BÀI 2: TẠO CÁC TRANG HTML TĨNH

Mã Bài: MD16_02

Giới thiệu:

Sau khi chúng ta xác định được mục đích và đối tượng độc giả của website, việc xây dựng hệ thống thông tin trong trang web một cách khoa học, rõ ràng và chính xác sẽ là yếu tố quyết định sự thành công của toàn bộ chu trình thiết kế web, đem lại sự nhất quán cho toàn bộ website.

Mục tiêu:

- Trình bày được cấu trúc cơ bản của một trang web HTML, cấu trúc thư mục của một website;
- Trình bày được tác dụng và cú pháp của các thẻ HTML;
- Viết được một số trang web qui mô nhỏ với các thẻ HTML;
- Thực thi được các trang HTML đó;
- Truy cập và chỉnh sửa được các đoạn mã HTML trong các trang web đã có theo các yêu cầu;
- Rèn luyện tính nghiêm túc, tỉ mỉ, khoa học trong việc tiếp thu các kiến thức lý thuyết; Tích cực, tỉ mỉ trong thực hành.

Nội dung chính:

1. Tạo và thực thi trang web HTML tĩnh

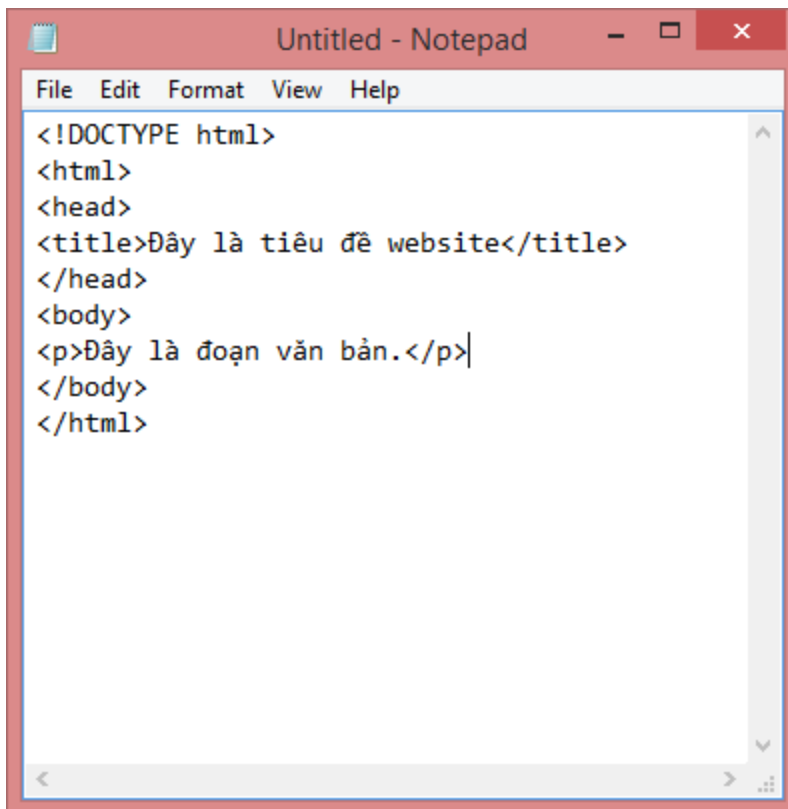
1.1 Tạo trang HTML đơn giản

HTML là một tập tin siêu văn bản nên bạn có thể dùng các chương trình soạn thảo văn bản không có chức năng định dạng văn bản để tạo ra một tập tin HTML. Trong Windows, bạn có thể dùng Notepad để tạo ra một tập tin HTML, còn trên Mac thì có thể dùng TextEdit và Vim trên các hệ điều hành Linux khác. Miễn là sau đó bạn phải lưu tập tin thành đuôi .html và sử dụng trình duyệt website để đọc nó.

Ngôn ngữ HTML là ngôn ngữ cơ bản nhất cho phép ta lập 1 trang web tĩnh, trong đó có ảnh và nội dung (chữ). Để làm web bằng html ta phải có văn bản soạn thảo (chèn ảnh, viết chữ...). Trong bài thực hành này, ta sẽ dùng **notepad** để thiết kế.

Để mở **Notepad** ta vào: **Start → Program → Accessories → Notepad**

(Hoặc phím windows + R, điền notepad)



Khi dùng **Notepad** để lưu trang web ta vào **File** → **Save**; phần **File Name** đánh tên web nhưng có đuôi là **.html** o Ví dụ: **web.html**.

* Lưu ý: Để gõ tiếng việt ở Encoding ta chọn kiểu UTF-8. Sau đó nhìn lên trên có chữ **Save In** ta có thể Save ra Desktop hoặc thư mục bất kỳ trên ổ đĩa.

1.2 Mở trang HTML

Để xem một trang HTML đã lưu trên máy tính, chúng ta có thể sử dụng bất kỳ trình duyệt web nào để mở. Cách đơn giản nhất là chỉ cần nhấp đôi chuột vào tập tin vừa lưu, hệ điều hành sẽ sử dụng trình duyệt mặc định để xem nội dung của trang HTML.

2. Đường dẫn tương đối và tuyệt đối

2.1 Liên kết là gì?

Liên kết được dùng để chuyển sang một trang web khác (hoặc một nguồn tài nguyên khác như: hình ảnh, video, file nhạc,...)

Liên kết có thể được đại diện bởi một cụm từ hoặc một hình ảnh.

Để tạo một liên kết, ta sử dụng cú pháp dưới đây:

```
<a href="URL" target="kiểu mở liên kết">Nội dung đại diện cho liên kết</a>
```

Trong đó:

- URL là địa chỉ của trang web mà bạn muốn chuyển đến khi người dùng bấm vào liên kết.
- Kiểu mở liên kết có thể là:
 - `_blank`: Mở liên kết trên một tab mới
 - `_self`: Mở liên kết ở ngay tab hiện tại (Mặc định là liên kết sẽ được mở ở ngay tab hiện tại)

Ví dụ

```
<a href="http://esm.vn">Liên kết 1</a>
<a href="http://esm.vn" target="_self">Liên kết 2</a>
<a href="http://esm.vn" target="_blank">Liên kết 3</a>
```

2.2 Đường dẫn tuyệt đối, đường dẫn tương đối

Một liên kết được xác định bởi một đường dẫn (đường dẫn hay còn được biết đến như một địa chỉ trang web).

Đường dẫn được chia làm hai loại:

- Đường dẫn tuyệt đối (là một chuỗi đầy đủ bao gồm `http://`, tên miền của trang web, đường dẫn đến tập tin)
- Đường dẫn tương đối (là một phần nhỏ của đường dẫn tuyệt đối, thông thường đường dẫn tương đối là phần đường dẫn đến tập tin)

Ví dụ:

Đường dẫn	Loại đường dẫn
<code>http://esm.vn</code>	TUYỆT ĐỐI
<code>http://esm.vn/html</code>	TUYỆT ĐỐI
<code>http://esm.vn/html/default.html</code>	TUYỆT ĐỐI
<code>http://esm.vn/html/cach-cao-mot-trang-web-html.html</code>	TUYỆT ĐỐI
<code>html</code>	tương đối
<code>html/default.html</code>	tương đối
<code>default.html</code>	tương đối
<code>html/cach-cao-mot-trang-web-html.html</code>	tương đối
<code>cach-cao-mot-trang-web-html.html</code>	tương đối

Ví dụ

Hai liên kết dưới đây đều được dùng để chuyển đến trang <http://esm.vn/html/default.html>

Tuy nhiên cách viết đường dẫn thì khác nhau:

```
<a href="http://esm.vn/html/default.html">Đường dẫn tuyệt đối</a>  
<a href="default.html">Đường dẫn tương đối</a>
```

3. Thẻ HTML

3.1 Thẻ khai báo loại tập tin

Ngay tại đoạn đầu tiên của tài liệu, chúng ta phải có một thẻ khai báo loại tập tin cho nó như thế này, cụ thể là ta sẽ khai báo rằng đây là tập tin HTML.

```
<!DOCTYPE html>
```

Với thẻ `<!DOCTYPE>` ở trên, ta có thêm một tham số đó là `html`. Tham số `html` này nghĩa là chúng ta khai báo với trình duyệt rằng đây là tài liệu HTML5 (HTML phiên bản 5), dù rằng chúng ta có thể không sử dụng HTML5 nhưng hiện tại khi khai báo tập tin HTML thì bạn cứ sử dụng tham số này vừa ngắn gọn lại vừa dễ dàng sử dụng thêm HTML5 nếu thích.

Có một điều thú vị là thẻ `<!DOCTYPE>` không phải là một thẻ của HTML, mà nó chỉ là một thẻ khai báo thông tin trên tài liệu để trình duyệt hiểu phiên bản HTML mà bạn sử dụng trên website mà thôi.

3.2 Thẻ đóng mở tài liệu HTML

Kế tiếp, ở tầng tiếp theo sẽ là thẻ `<html> </html>` có nhiệm vụ khai báo cho trình duyệt biết rằng những nội dung bên trong cặp thẻ này là HTML. Tuy nhiên, bên trong thẻ này mình có thêm một **thuộc tính** tên là **lang** với giá trị là **vi** (`<html lang="vi">`). Thuộc tính này nghĩa là chúng ta khai báo cho trình duyệt biết mã ngôn ngữ mà ta sử dụng trên website, mã vi nghĩa là Vietnamese – tiếng Việt.

Bạn nên nhớ một điều là thẻ `<html> </html>` phải bao bọc toàn bộ nội dung website, không bao gồm thẻ `<!DOCTYPE>`.

3.3 Thẻ đóng và mở phần thông tin website

Phần khai báo thông tin của website sẽ được đặt vào bên trong cặp thẻ gọi là `<head> </head>`. Nội dung bên trong thẻ này là các thẻ chuyên cho khai báo thông tin website (meta), tên website (title), khai báo CSS (style), khai báo các đoạn Javascript (script) và một số thông tin khác. Thường là các thông tin được khai báo trong đây sẽ không hiển thị trực tiếp thành siêu văn bản trên web nhưng nó sẽ có nhiệm vụ chứa các thông tin quan trọng về website.

3.4 Có thể đóng và mở phần nội dung website

Đây là cặp thẻ mà bạn sẽ tiến hành viết nội dung vào, đó là cặp thẻ `<body>` `</body>`. Cặp thẻ này là để trình duyệt xác định đây là phần thân của website, nó sẽ chứa toàn bộ các nội dung siêu văn bản hoặc media mà bạn muốn nó hiển thị lên trang web của bạn. Phần này chúng ta sẽ làm việc nhiều hơn ở các bài sau.

Bây giờ nếu bạn mở tài liệu HTML ở trên lên bằng trình duyệt thì chỉ sẽ thấy nó có mỗi đoạn nội dung nằm bên trong thẻ `<body>` `</body>` mà thôi.

4. Cấu trúc trang web HTML tĩnh

Một tài liệu hay tập tin HTML để được gọi là một tài liệu web thì sẽ được bao gồm bốn yếu tố chính đó là:

- Có thể khai báo loại tập tin/tài liệu.
- Có thể đóng và mở tài liệu HTML.
- Có thể đóng và mở phần thông tin website.
- Có thể đóng và mở phần nội dung website.

Và trong bài này, chúng ta sẽ đi tìm hiểu qua 4 phần đó để tạo ra một tài liệu website bằng HTML đúng chuẩn.

```
<!DOCTYPE html>
<html lang="vi">
  <head>
    <meta charset="utf-8" />
  </head>
  <body>
    <p>Đây là nội dung website</p>
  </body>
</html>
```

5. Định dạng văn bản

Đôi khi trong văn bản của bạn sẽ cần định dạng lại chữ viết như in đậm, tô nghiêng, gạch chân,...và đây là các thẻ HTML cho từng chức năng đó. Trước hết là hãy soạn một đoạn văn bản HTML như thế này:

```
<p>Trong tài liệu HTML, bạn có thể định dạng lại chữ viết như
<strong>tô đậm</strong> chữ viết, hoặc có thể <i>in nghiêng</i>,
hoặc cũng có thể <u>gạch chân</u> nó tùy thích. Một số trường
hợp bạn cũng có thể sẽ cần <del>gạch ngang</del> chữ
viết.</p>
<hr>
```

```
<p>Ở trên là một đoạn kẻ ngang được khai báo bằng thẻ  
<code>&lt;hr&gt;</code>.</p>  
<p>Nếu bạn cần nhấn mạnh gì đó, bạn có thể <em>viết như thế  
này</em>.</p>
```

- ****: In đậm chữ viết.
- **<i>**: In nghiêng chữ viết.
- **<u>**: Gạch chân chữ viết.
- **<strike>**: Gạch ngang chữ viết.
- ****: Nhấn mạnh câu.
- **<code>**: Định dạng cho một đoạn mã nào đó.
- **<hr>**: Thuộc kẻ ngang trên tài liệu.
- **<mark>**: Tô sáng chữ viết.

Màu chữ

Để thiết lập màu chữ, bạn có thể sử dụng thuộc tính color. Giá trị của nó là tên màu trong tiếng Anh hoặc mã màu HEX.

```
<span style="color: red">chữ màu đỏ</span>
```

Màu nền

Màu nền có cách thiết lập giống hệt màu chữ, tức là bạn có thể dùng giá trị là tên màu trong tiếng Anh hoặc mã màu HEX. Tên thuộc tính của màu nền là background-color.

```
<span style="color: white; background-color: red">Chữ có nền màu  
đỏ và màu chữ là trắng</span>
```

Kích thước chữ

Kích thước chữ bạn có thể sử dụng thuộc tính font-size và giá trị là số kèm đơn vị. Bạn có thể sử dụng đơn vị px, %, pt hoặc em tùy thích, đơn giản nhất là dùng px.

```
<span style="font-size: 32px">Chữ có kích thước 32px</span>
```

Font chữ

Nếu bạn có nhu cầu sử dụng font chữ khác so với font chữ mặc định thì hãy dùng thuộc tính font-family với giá trị là tên font chữ có trên máy tính. Một số tên font chữ phổ biến nhất là Arial, Helvetica, Time New Roman, Verdana.

```
<span style="font-family: Arial">Font chữ Arial</span>
```

Ngoài ra bạn có thể thêm font chữ dự phòng bằng cách khai báo nhiều tên font chữ khác nhau được ngăn cách bởi dấu phẩy, ví dụ:

```
<span style="font-family: Helvetica, Arial">Font chữ  
Arial</span>
```

Có nghĩa là nếu máy người đọc không có font chữ Helvetica thì nó sẽ sử dụng font chữ Arial.

Căn lề văn bản

Để canh lề, chúng ta sử dụng thuộc tính text-align với giá trị là left, center, right hoặc justify.

```
<span style="text-align: center">Canh giữa văn bản</span>
```

6. Heading

Tiêu đề (Headline) và đoạn văn bản (Paragraph) là hai thành phần mà bạn cần phân biệt để sau này sử dụng cho đúng. Bạn hãy hiểu rằng, tiêu đề văn bản nó giống như tên của từng chương trong một quyển sách vậy, nó sẽ được hiển thị to hơn nhằm nổi bật hơn.

Trong HTML, các thẻ tiêu đề sẽ được định nghĩa bằng cặp thẻ `<h>`, trong đó `n` là số tự nhiên từ 1 đến 6 tương ứng với từng cấp độ, số càng nhỏ thì cấp độ càng lớn. Để dễ hiểu hơn, hãy thử soạn thảo một tài liệu HTML như dưới đây.

```
<h1>Headline level 1</h1>  
<h2>Headline level 2</h2>  
<h3>Headline level 3</h3>  
<h4>Headline level 4</h4>  
<h5>Headline level 5</h5>  
<h6>Headline level 6</h6>
```

Còn đoạn văn bản thì nó sẽ được khai báo bằng cặp thẻ `<p>` `</p>`. Các văn bản nằm trong cặp thẻ này sẽ được hiểu là một đoạn văn bản, mỗi đoạn văn bản sẽ được xuống dòng và cách nhau với tỷ lệ nhất định. Ví dụ bên dưới:

```
<p>Đây là một đoạn văn bản, mỗi đoạn văn bản sẽ được hiển thị  
một dòng riêng và được khai báo bằng thẻ  
<code>&lt;p>&gt;</code>.</p>  
<p>Đây cũng là một đoạn văn bản.</p>
```

7. Siêu liên kết

Một trong những nét đặc trưng của siêu văn bản là có những đường liên kết (link) tới một tài liệu khác thông qua địa đường dẫn đến tài liệu hoặc địa chỉ

website. Để tạo ra các đường liên kết trong HTML ta sẽ sử dụng cặp thẻ `<a>` `</a>` chứa các tham số như sau:

```
<a href="https://thachpham.com" title="Thach Pham Blog"
target="_blank">nhập vào đây</a>
```

Mặc dù thẻ `<a>` có rất nhiều thuộc tính nhưng ở trên là các thuộc tính thường dùng nhất hiện tại.

```
<p>Một siêu văn bản sẽ có các <a href="http://thachpham.com" title="Thach Pham
Blog" target="_blank">siêu liên kết</a> để người đọc có thể truy cập một tài liệu
khác.</p>
```

Ý nghĩa các thuộc tính:

href: Địa chỉ của tài liệu muốn liên kết đến, đây có thể là một đường dẫn thư mục hoặc địa chỉ website. Nếu bạn muốn truy cập một tài liệu trên cùng một cấp thư mục thì chỉ cần ghi tên-tập-tin.định dạng (ví dụ: gioi-thieu.html).

title: Tiêu đề của liên kết, tiêu đề sẽ hiển thị như một thông tin thêm khi rê chuột vào liên kết.

target: Xác định nơi mở tài liệu, nó có các giá trị như `_blank` (mở tài liệu trên cửa sổ mới), `_self` (mở tài liệu trên cửa sổ hiện tại, nếu bạn không khai báo thuộc tính `target` thì nó sẽ sử dụng giá trị này làm mặc định), `_top` (mở tài liệu trong nội dung trang hiện tại), `_parent` (mở tài liệu trên khung trình duyệt mẹ của nó).

Và một điều bạn cần biết đó là nội dung trong cặp thẻ `<a>` có thể là một nội dung siêu văn bản nào, bao gồm các thẻ tiêu đề, hình ảnh,...

Liên kết neo (Anchor Link)

Liên kết neo nghĩa là một liên kết trong siêu văn bản sẽ dẫn đến một vị trí đặc biệt trong cùng tài liệu mà không phải tải lại hoặc mở một tài liệu mới. Một liên kết neo sẽ có hai phần:

Khu vực được neo, được khai báo bằng thẻ bất kỳ với thuộc tính `id` (ví dụ: `<p id="noi-dung"> </p>`).

Liên kết neo, được khai báo bằng thẻ `<a>` nhưng có thuộc tính là `href` nhưng giá trị là có dấu `#` và tên `id` của khu vực cần truy cập đến (ví dụ: `<a href="#noi-dung">xem nội dung</a>`).

Hãy thử xem một ví dụ sau:

```
<a href="#than-bai">Xem thân bài</a>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vestibulum aliquam mattis maximus. Phasellus facilisis felis
```

```
massa, at bibendum lorem fringilla ut. Sed ipsum sapien,  
elementum id felis ut, iaculis pulvinar elit.</p>
```

...

```
<p id="than-bai"><strong>Đây là thân bài</strong></p>
```

```
<p>Etiam efficitur felis ac dui tincidunt egestas. Vestibulum  
auctor justo tellus, ut posuere eros sagittis id. Etiam  
dignissim blandit porta. Nunc eu suscipit ipsum, vehicula mollis  
nisi. Nam non iaculis leo. Duis euismod pharetra vestibulum.  
Praesent cursus ut purus egestas cursus. Nulla eget volutpat  
neque. Aliquam erat volutpat. Sed at metus convallis, ultrices  
diam ut, iaculis neque. Nullam purus lacus, ullamcorper a libero  
sed, facilisis blandit turpis. Nam ac massa ut elit mollis  
luctus at eu dui. Nullam eu turpis quis eros iaculis convallis  
in non neque.</p>
```

...

Khi nhấp vào liên kết Xem thân bài thì nó sẽ tự động nhảy đến khu vực được gán thuộc tính id="than-bai".

8. Thành phần đa phương tiện

Các tập tin kỹ thuật số (media files) mà mình đang nói đến đó là hình ảnh (image), âm thanh (audio) và phim ảnh (video). Với HTML chúng ta có thể dễ dàng chèn những loại tập tin này vào tài liệu và nó sẽ hiển thị rất là sống động luôn.

Lưu ý rằng các tập tin kỹ thuật số được chèn vào phải có đường dẫn tuyệt đối dẫn trực tiếp đến tập tin đó (ví dụ: <http://domain.com/file/image.jpg> hoặc [audio.mp3](#), [video.mp4](#),...).

8.1 Chèn ảnh vào HTML

Để chèn ảnh vào HTML, chúng ta sẽ sử dụng thẻ `<img>` với các tham số bắt buộc, và thẻ này không có thẻ đóng. Ví dụ:

```

```

Giải thích các thuộc tính:

- **src**: Đường dẫn đến tập tin hình ảnh.
- **title**: Tiêu đề của hình ảnh.
- **alt**: Tên định danh của hình ảnh.

Ngoài ra bạn có thể bổ sung thêm thuộc tính width và height để điều chỉnh kích thước hiển thị hình ảnh.

8.2 Chèn video

Mặc dù hiện tại bạn chưa học đến HTML5 nhưng ở chức năng chèn video này chúng ta sẽ sử dụng một thẻ trong HTML để chèn video nhanh chóng đó là cặp thẻ `<video> </video>` như sau:

```
<video width="320" height="240" controls>
  <source src="https://www.w3schools.com/tags/movie.mp4"
type="video/mp4">
  Trình duyệt của bạn không hỗ trợ HTML5.
</video>
```

Nghĩa là trong cặp thẻ `<video> </video>` sẽ có một thẻ `<source>` với các thuộc tính nhằm khai báo đường dẫn tập tin video và loại tập tin. Các thuộc tính bên trong cặp thẻ này có vẻ không có gì là khó hiểu.

Lưu ý: Để đảm bảo tất cả các trình duyệt có thể đọc được, chỉ nên chèn video định dạng MP4.

8.3 Chèn âm thanh – nhạc

Giống như thẻ `<video>`, để chèn âm thanh vào tài liệu HTML thì bạn có thể sử dụng thẻ `<audio>` và thẻ này cũng là HTML5.

```
<audio controls>
  <source src="https://www.w3schools.com/tags/horse.mp3"
type="audio/mp3">
  Trình duyệt của bạn không hỗ trợ HTML5
</audio>
```

8.4 Chèn đối tượng kỹ thuật số với thẻ `<object>`

Ngoài các thẻ đặc trưng cho từng loại tập tin ở trên thì bạn còn có một cách khác để chèn các đối tượng kỹ thuật số vào tài liệu HTML đó là dùng thẻ `<object>`, đây là một thẻ có thể giúp bạn chèn các loại đối tượng kỹ thuật số như Flash, Java, Audio, Video, PDF, ActiveX. Nhưng thông thường thì các loại mã nhúng của một số website cho phép sử dụng mã nhúng như Youtube họ sẽ dùng thẻ này để bạn chèn đối tượng vào web.

8.5 Nhúng tài liệu HTML vào web

Nếu bạn có nhu cầu nhúng thẳng một trang nào đó vào tài liệu HTML của bạn thì có thể sử dụng thẻ `<iframe>` như sau:

```
<iframe src="http://thachpham.com" width="500px" height="800px"
scrolling="auto"></iframe>
```

Trong đó, thuộc tính `src` là đường dẫn của trang bạn muốn nhúng vào, `width` và `height` là kích thước của khung hiển thị (gọi là frame).

Ngoài ra, có một cách nữa để chèn một liên kết và ép nó mở bằng frame trên website đó là sử dụng thuộc tính `name` trong thẻ `<iframe>` và thuộc tính `target` trong thẻ `<a>`, xem ví dụ:

```
<p><a target="window" href="http://thachpham.com">Truy
cập</a></p>
<iframe src="https://www.w3schools.com/" name="window"
width="500px" height="800px"></iframe>
```

Trong đó, giá trị của thuộc tính `target` trong thẻ `<a>` phải trùng với giá trị trong thuộc tính `name` của thẻ `<iframe>`.

9. Danh sách

- Thẻ `<DL>` `</DL>`

Cú pháp:

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

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

```
<html>
```

```

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

```

→ Kết quả hiển thị trong trình duyệt:

```

Đề mục danh sách
Danh sách mức 1
Danh sách mức 1
  Danh sách mức 2
  Danh sách mức 2
Danh sách mức 1
  Danh sách mức 2

```

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

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

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

```

<UL>
<LI>Đây là mục 1 trong danh sách.
<LI>Đây là mục 2 trong danh sách.

```


Kết quả trình duyệt như sau:

Đây là mục 1 trong danh sách.

Đây là mục 2 trong danh sách.

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

- Định nghĩa OL:

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

START=n: chỉ định chỉ số bắt đầu

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

A: sử dụng ký tự lớn A,B,C,...

a: sử dụng ký tự nhỏ a,b,c,...

I: sử dụng số La Mã lớn I,II,III,...

i: sử dụng số La Mã nhỏ i,ii,iii,...

1: sử dụng số 1,2,3,...

Ví dụ:

<OL TYPE=I>

Nghị thức HTTP

Nghị thức FTP

Nghị thức SMTP

Kết quả trình duyệt như sau:

I. Nghị thức HTTP

II. Nghị thức FTP

III. Nghị thức SMTP

10. Bảng biểu

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

Ví dụ:

<TABLE>

<TR><TD>Ô 11<TD>Ô 12

<TR><TD>Ô 21<TD>Ô 22

</TABLE>

Kết quả trình duyệt như sau:

Ô 11 Ô 12

Ô 21 Ô 22

Để đóng khung cho bảng, sử dụng thuộc tính BORDER

Ví dụ:

<TABLE BORDER=1>

<TR><TD>Ô 11<TD>Ô 12

<TR><TD>Ô 21<TD>Ô 22

</TABLE>

Kết quả trình duyệt như sau:

Ô 11	Ô 12
Ô 21	Ô 22

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

Ví dụ:

<TABLE BORDER=1 WIDTH=100%>

<TR><TD>Ô 11<TD>Ô 12

<TR><TD>Ô 21<TD>Ô 22

</TABLE>

Kết quả trình duyệt như sau:

Ô 11	Ô 12
Ô 21	Ô 22

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

Ví dụ:

<TABLE BORDER=1 WIDTH=100%>

<CAPTION> Tiêu đề của bảng</CAPTION>

<TR><TD>Ô 11<TD>Ô 12

```
<TR><TD>Ô 21<TD>Ô 22
</TABLE>
```

Kết quả trình duyệt như sau:

Tiêu đề của bảng

Ô 11	Ô 12
Ô 21	Ô 22

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

Ví dụ:

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

Kết quả trình duyệt như sau:

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

- Định nghĩa TABLE(bảng):

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

Trong đó:

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

BACKGROUND=url: chỉ định ảnh nền của bảng

BGCOLOR=color-type: màu nền của bảng

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

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

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

- Định nghĩa TR(dòng):

<TR

ALIGN=align-type

BACKGROUND=url

BGCOLOR=color-type

VALIGN=v-align-type>

Trong đó:

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

BACKGROUND=url: chỉ định ảnh nền của dòng

BGCOLOR=color-type: màu nền của dòng

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

- Định nghĩa TD (ô):

<TD

ALIGN=align-type

BACKGROUND=url

BGCOLOR=color-type

COLSPAN=n

ROWSPAN=n

VALIGN=v-align-type>

Trong đó:

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

BACKGROUND=url: chỉ định ảnh nền cho ô

BGCOLOR=color-type: màu nền của ô

COLSPAN=n: ô trải rộng trên n cột

ROWSPAN=n: ô trải dài trên n hàng

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

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

11. Biểu mẫu

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

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

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

- Cú pháp:

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

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

<input type=submit value = “Nút Submit”>

</FORM>

- Phương thức: gồm phương thức POST và GET.

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

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

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

BÀI 3: THIẾT KẾ GIAO DIỆN VỚI HTML VÀ CSS

Mã Bài: MD16_03

Giới thiệu:

CSS (Cascading Style Sheets) là một ngôn ngữ dùng để định dạng cho các phần tử HTML (Ví dụ: chỉnh kích cỡ chữ, chỉnh font chữ, màu chữ, màu nền, hình nền, đường viền,...) dựa trên các cặp thuộc tính: giá trị thuộc tính.

Với việc sử dụng CSS, ta có thể định dạng ra các phần tử HTML thật đặc biệt và chuyên nghiệp.

Mục tiêu:

- Trình bày được nguyên lý hoạt động của CSS;
- Xác định được các selector tương ứng cho từng đối tượng, nhóm đối tượng HTML;
- Áp dụng được CSS cho các thành phần, đối tượng HTML;
- Rèn luyện tính nghiêm túc, tích cực trong việc tiếp nhận kiến thức. Chủ động, tích cực tìm kiếm các nguồn tài liệu liên quan.

Nội dung chính:

1. Tổng quan về CSS

Có ba cách để sử dụng CSS: Inline, Internal, External

Cách 1) Inline CSS

Ta đặt thuộc tính style vào thẻ mở của phần tử HTML, giá trị của thuộc tính style là các cặp thuộc tính định dạng CSS (Bạn có thể tham khảo cách sử dụng CSS bằng Inline trong bài Thuộc tính Style trong HTML).

Ví dụ:

```

```

Cách 2) Internal CSS

Ta đặt các cặp thuộc tính định dạng CSS vào bên trong cặp thẻ <style type="text/css"></style>

Cặp thẻ <style type="text/css"></style> thì được đặt bên trong cặp thẻ <head></head>

Ví dụ:

```
<!DOCTYPE html>
<html>
```

```

<head>
  <style type="text/css">
    img{
      width:400px;
      height:400px;
    }
  </style>
</head>
<body>
  
</body>
</html>

```

Để định dạng cho một phần tử, ta sử dụng cú pháp:

```

element {
  property1 : value;
  property2 : value;
  property3 : value;
  ....
  propertyN : value;
}

```

Ví dụ:

```

<!DOCTYPE html>
<html>
  <head>
    <style type="text/css">
      h1{
        color:red;
      }
      h2{
        color:blue;
        font-size:50px;
        background-color:gray;
      }
      h3{

```

```

        color:green;
        font-size:70px;
        font-family:Courier;
        text-align:center;
    }
</style>
</head>
<body>
    <h1>heading 1</h1>
    <h2>heading 2</h1>
    <h3>heading 3</h1>
</body>
</html>

```

Cách 3) External CSS

Với Internal CSS:

- Ta đặt các thuộc tính định dạng vào bên trong cặp thẻ `<style type="text/css"></style>`
- Rồi cặp thẻ `<style type="text/css"></style>` thì đặt bên trong cặp thẻ `<head></head>`

Với External CSS:

- Ta đặt các thuộc tính định dạng vào bên trong tập tin CSS
- Trong cặp thẻ `<head></head>` của tập tin HTML, ta dùng thẻ `<link rel="stylesheet" type="text/css" href="đường dẫn đến tập tin CSS">` để nhúng tập tin CSS vào trang web.

Tôi có một tập tin dinh dang.css (cùng cấp với tập tin HTML) có nội dung là

```

h1 {
    color:red;
    font-size:50px;
    font-family:Courier;
}

```

Ví dụ

Khi thực thi tập tin HTML có nội dung:

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="dinh dang.css">
  </head>
  <body>
    <h1>Hello HTML</h1>
  </body>
</html>
```

Màn hình trình duyệt sẽ hiển thị là

Hello HTML



Cách tạo tập tin CSS giống với cách tạo tập tin HTML (thay .html bằng .css)
Ta có thể nhúng vào trang web nhiều tập tin CSS bằng cách viết nhiều thẻ

```
<link rel="stylesheet" type="text/css" href="đường dẫn đến tập tin CSS">
```

Một số thuộc tính định dạng CSS quan trọng:

Thuộc tính	Ý nghĩa
color	Thiết lập màu chữ cho phần tử. Giá trị của thuộc tính color có thể xác định theo: Tên màu (Ví dụ: green, blue, red, yellow, pink, black, white,...) Giá trị Hex (Ví dụ: #00FF00, #99FF66, #66CC99,...) Giá trị RGB (Ví dụ: rgb(255,0,0), rgb(120,0,150),...)
background-color	Thiết lập màu nền cho phần tử. Giá trị của thuộc tính background-color có thể xác định theo tên màu, giá trị Hex, giá trị RGB giống như thuộc tính color.
font-size	Thiết lập kích cỡ chữ cho phần tử. Giá trị của thuộc tính font-size có thể xác định theo: px em (1em = 16px)
font-family	Thiết lập kiểu chữ cho phần tử. Dưới đây là một số kiểu chữ phổ biến: Times New Roman

	Book Antiqua Arial Black Lucida Sans Unicode Palatino Linotype Georgia Lucida Console Arial Tahoma Impact Verdana monospace Courier
text-align	Canh lề cho nội dung của phần tử. Thuộc tính text-align có ba giá trị: left (canh cho nội dung nằm bên trái) center (canh cho nội dung nằm ở giữa) right (canh cho nội dung nằm bên phải)
border	Tạo đường viền bao xung quanh phần tử. Giá trị của thuộc tính border có 3 tham số: Tham số thứ nhất là độ dày của đường viền Tham số thứ hai là kiểu đường viền (Có một số kiểu như: groove, ridge, inset, outset, none, hidden, dotted, dashed, solid, double,...) Tham số thứ ba là màu của đường viền.
width, height	Thiết lập chiều rộng và chiều cao cho phần tử. Có thể xác định theo đơn vị px hoặc %

2. Thiết kế giao diện với HTML và CSS

2.1. CSS cơ bản

Để tìm hiểu cú pháp CSS chúng ta hãy thử xem một ví dụ sau.

Ví dụ: Để định màu nền cho một trang web là xanh nhạt (light cyan) chúng ta dùng code sau:

+ Trong HTML: `<body bgcolor="#00BFF3">`

+ Trong CSS: `body { background-color:#00BFF3; }`

Nhìn qua ví dụ trên ít nhiều chúng ta cũng thấy được mối tương đồng giữa các thuộc tính trong HTML và CSS cho nên nếu bạn đã học qua HTML thì cũng sẽ rất dễ dàng tiếp thu CSS. Đó là một chút lợi thế của câu chuyện hành trình mà Pearl đã nói ở bài trước. Nhưng không sao cả, bây giờ hãy nhìn vào ví dụ của chúng ta và các bạn xem nó có giống với cấu trúc sau không nhé.

Cú pháp CSS cơ bản: Selector { property:value; }

Trong đó:

+ *Selector*: Các đối tượng mà chúng ta sẽ áp dụng các thuộc tính trình bày. Nó là các tag HTML, **class** hay **id** (chúng ta sẽ học về 2 thành phần này ở bài học sau). Ví dụ: body, h2, p, img, #title, #content, .username,...

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

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

Nên trong trường hợp này, nếu bạn có ý dùng CSS đó chỉ riêng phần ảnh thì chỉ nên ghi là img .visitors thôi.

Một lỗi viết tên selector nữa đó là dựa trên tên các thuộc tính có trong

HTML. Ví dụ trong HTML ta có đoạn mã như vậy: **<input name="Search" type="Text" value="Key Word">**. Để áp dụng thuộc tính CSS cho riêng ô tìm kiếm này chúng ta sẽ dùng selector **input[name="Search"]**.

2.1.1. Kiểu chữ (font), cỡ chữ

2.1.1.1 Thuộc tính font-family:

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

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

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

+ Generic families: Tên của một họ gồm nhiều font. Ví dụ: sans-serif, serif,... Khi lên danh sách font dùng để hiển thị một trang web bạn sẽ chọn những font mong muốn trang web sẽ được hiển thị để đặt ở các vị trí ưu tiên. Tuy nhiên, có thể những font này sẽ không thông dụng lắm nên bạn cũng cần chỉ định thêm một số font thông dụng dự phần như Arial, Tohama hay Times New Roman và bạn cũng được đề nghị đặt vào danh sách font của mình một generic families (thường thì nó sẽ có độ ưu tiên thấp nhất). Thực hiện theo cách này thì sẽ đảm bảo trang web của bạn có thể hiển thị tốt trên bất kỳ hệ thống nào.

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

```
body { font-family:"Times New Roman",Tohama,sans-serif } h1, h2, h3 { font-family:arial,verdana,serif }
```

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

Chú ý: Đối với các font có khoảng trắng trong tên như Times New Roman cần được đặt trong dấu ngoặc kép.

2.1.1.2. Thuộc tính font-style:

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

```
h1 { font-style:italic; }
h2 { font-style:oblique; }
```

2.1.1.3. Thuộc tính font-variant:

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

Một font small-caps là một font sử dụng chữ in hoa có kích cỡ nhỏ hơn in hoa chuẩn để thay thế những chữ in thường. Nếu như font chữ dùng để hiển thị không có sẵn font small-caps thì trình duyệt sẽ hiện chữ in hoa để thay thế.

Trong ví dụ sau chúng ta sẽ sử dụng kiểu small-caps cho phần h1

```
h1 { font-variant:small-caps
```

}

2.1.1.4. Thuộc tính font-weight:

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

Thử in đậm phần p:

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

2.1.1.5. Thuộc tính font-size:

Kích thước của một font được định bởi thuộc tính font-size.

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

Ở ví dụ sau trang web sẽ có kích cỡ font là 20px, h1 là 3em = 3 x 20 = 60px, h2 là 2em = 40px.

```
body {  
    font-size:20px  
} h1 { font-size:3em  
} h2 { font-size:2em  
}
```

Thuộc tính font rút gọn

Tương tự như các thuộc tính background, chúng ta cũng có thể rút gọn các thuộc tính font lại thành một thuộc tính đơn như ví dụ sau:

```
h1 {  
    font-style: italic; font-variant:small-caps; font-weight: bold;  
font-size: 35px;  
font-family: arial,verdana,sans-serif;  
}
```

Thành

```
h1 {
```

```
font: italic bold 35px arial,verdana,sans-serif;
```

```
}
```

Cấu trúc rút gọn cho các thuộc tính nhóm font:

Font :<font-style> | < font-variant> | <font-weight> | <font-size> |< font-family>

Đi qua bài học này, bạn đã nắm bắt được cách định font chữ cho một thành phần trang web cũng như cách sử dụng các kiểu font in nghiêng, in đậm, font small-caps và cách qui định kích thước font. Trong bài học kế chúng ta sẽ được tìm hiểu thêm về các thuộc tính CSS về định dạng văn bản.

2.1.2. Màu sắc

2.1.2.1. Màu nền (thuộc tính background-color):

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

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

```
body {          background-color:cyan
}
```

```
h1 {          background-color:red
} h2 {          background-color:orange
}
```

2.1.2.2. Màu chữ

Ngoài việc viết tên selector cụ thể, chúng ta cũng có thể dùng một selector đại diện như *** { color:red }** sẽ tác động đến tất cả các thành phần có trên trang web làm cho chúng có text màu đỏ.

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

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

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

Để dễ đọc hơn, bạn nên viết mỗi thuộc tính CSS ở một dòng. Tuy nhiên, nó sẽ làm tăng dung lượng lưu trữ CSS của bạn.

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

Đối với một trang web có nhiều thành phần có cùng một số thuộc tính, chúng ta có thể thực hiện gom gọn lại như sau:

```
h1 { color:#0000FF; text-transform:uppercase } h2 {  
color:#0000FF;  
text-transform:uppercase;  
}  
h3 { color:#0000FF;  
text-transform:uppercase;  
}  
⇒ h1, h2, h3 {  
color:#0000FF;  
text-transform:uppercase;  
}
```

+ *Value*: Giá trị của thuộc tính. Ví dụ: như ví dụ trên value chính là #FFF dùng để định màu trắng cho nền trang.

Đối với một giá trị có khoảng trắng, bạn nên đặt tất cả trong một dấu ngoặc kép. Ví dụ: `font-family:"Times New Roman"`

Đối với các giá trị là đơn vị đo, không nên đặt một khoảng cách giữa số đo với đơn vị của nó. Ví dụ: `width:100 px`. Nó sẽ làm CSS của bạn bị vô hiệu trên Mozilla/Firefox hay Netscape.

Chú thích trong CSS:

Cũng như nhiều ngôn ngữ web khác. Trong CSS, chúng ta cũng có thể viết chú thích cho các đoạn code để dễ dàng tìm, sửa chữa trong những lần cập nhật sau.

Chú thích trong CSS được viết như sau `/* Nội dung chú thích */` *Ví dụ:*

```
/* Màu chữ cho trang web */ body {  
color:red  
}
```

2.1.2.3. Bảng mã và tên màu

Đơn vị màu sắc

Đơn vị	Mô tả
Color-name	Tên màu tiếng Anh. Ví dụ: black, white, red, green, blue, cyan, magenta,...
RGB (r,g,b)	Màu RGB với 3 giá trị R, G, B có trị từ 0 – 255 kết hợp với nhau tạo ra vô số màu.
RGB (%r,%g,%b)	Màu RGB với 3 giá trị R, G, B có trị từ 0 – 100% kết hợp.
Hexadecimal RGB	Mã màu RGB dạng hệ thập lục. Ví dụ: #FFFFFF: trắng, #000000: đen, #FF00FF: đỏ tươi.

2.1.3. Định dạng trình bày cơ bản

Định dạng và thêm vào các kiểu định dạng đặc biệt cho phần nội dung một trang web là một vấn đề quan trọng cho bất cứ nhà thiết kế web nào. Như đã nói, ở bài học này chúng ta sẽ được tìm hiểu về các thuộc tính CSS về định dạng văn bản.

Màu chữ (thuộc tính color):

Để định màu chữ cho một thành phần nào đó trên trang web chúng ta sử dụng thuộc tính color. Giá trị của thuộc tính này là các giá trị màu CSS hỗ trợ.

Ví dụ sau chúng ta sẽ viết CSS để định màu chữ chung cho một trang web là đen, cho tiêu đề h1 màu xanh da trời, cho tiêu đề h2 màu xanh lá chúng ta sẽ làm như **body {**

```
    color:#000
    }
h1 {    color:#0000FF
    } h2 {    color:#00FF00
    }
```

Thuộc tính text-indent :

Thuộc tính text-indent cung cấp khả năng tạo ra khoảng thụt đầu dòng cho dòng đầu tiên trong đoạn văn bản. Giá trị thuộc tính này là các đơn vị đo cơ bản dùng trong CSS.

Trong ví dụ sau chúng ta sẽ định dạng thụt đầu dòng một khoảng 30px cho dòng văn bản đầu tiên trong mỗi đoạn văn bản đối với các thành phần <p>

```
p { text-indent:30px }
}
```

Thuộc tính text-align :

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

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

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

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

Thuộc tính letter-spacing:

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

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

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

Thuộc tính text-decoration:

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

Ví dụ sau chúng ta sẽ định dạng gạch chân cho thành phần h1, gạch đầu thành phần h2

```
h1 { text-decoration:underline }
h2 { text-decoration:overline }
}
```

Thuộc tính text-transform:

Text-transform là thuộc tính qui định chế độ in hoa hay in thường của văn bản mà không phụ thuộc vào văn bản gốc trên HTML.

Thuộc tính này có tất cả 4 giá trị: uppercase (in hoa), lowercase (in thường), capitalize (in hoa ở ký tự đầu tiên trong mỗi từ) và none (không áp dụng hiệu ứng – mặc định).

Trong ví dụ dưới đây chúng ta sẽ định dạng cho thành phần h1 là in hoa, h2 là in hoa đầu mỗi ký tự.

```
h1 {      text-transform:uppercase
    } h2 {      text-transform:capitalize
    }
```

2.2. CSS nâng cao

2.2.1. Siêu liên kết

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

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

Ví dụ 1: Ví dụ này chúng ta sẽ áp dụng 4 màu sắc khác nhau cho từng trạng thái liên kết: các liên kết chưa thăm có màu xanh lá; các liên kết mouse over sẽ có màu đỏ tươi; các liên kết đã thăm sẽ có màu đỏ và các liên kết đang kích hoạt có màu tím.

```
a:link {      color:#00FF00
    }
a:hover {      color:#FF00FF
    }
a:visited {      color:#FF0000
    }
a:active { color:#662D91 }
```

Ví dụ 2: Tạo các hiệu ứng tương ứng với trình trạng liên kết: các liên chưa thăm có màu xanh lá, kích cỡ font 14px; liên kết mouse over có màu đỏ tươi, kích

cỡ font 1.2em, hiệu ứng nhấp nháy; liên kết đã thăm sẽ có màu xanh da trời, không có đường gạch chân; các liên kết đang kích hoạt có màu tím và font dạng small-caps.

```
a:link {          color:#00FF00;          font-size:14px
}
```

```
a:hover {          color:#FF00FF;          font-size:1.2em;
text-decoration:blink
}
```

```
a:visited {          color:#FF0000;          text-decoration:none
}
```

```
a:active {          color:# 662D91;          font-variant:small-caps
}
```

Ví dụ 3: Ví dụ này cũng tạo cho liên kết hiệu ứng màu sắc giống ví dụ 2 nhưng sẽ có thêm 1 số hiệu ứng: các liên kết sẽ có khung viền màu đen, kích cỡ font 14px; liên kết mouse over có nền light cyan; các liên kết đã thăm có nền light yellow.

```
a {      border:1px solid #000;          font-size:14px
}
```

```
a:link {
          color:#00FF00;
}
```

```
a:hover {          background-color:#00BFF3;          color:#FF00FF;
font-size:1.2em;          text-decoration:blink
}
```

```
a:visited {          background-color:#FFF568;          color:#FF0000;
text-decoration:none
}
```

}

a:active { color:#662D91; font-variant:small-caps }

Ba ví dụ trên chỉ là một tí gợi ý về khả năng kết hợp các thuộc tính CSS với pseudo-classes để tạo nên nhiều hiệu ứng hấp dẫn cho trang web. Cũng xin nói luôn là các ví dụ Pearl trình bày trong bài này cũng như những bài khác thật ra trông không dễ nhìn, đó là do Pearl không có nhiều thời gian để chăm chút các ví dụ của mình. Cái mà Pearl muốn nói chỉ là làm thế nào các bạn hiểu tác dụng một thuộc tính nào đó để có thể vận dụng cho trang web của chính mình.

2.2.2. Class và ID

Trong các bài học trước, chúng ta đã được học các thuộc tính CSS về background, color, font,... Tuy nhiên, bạn cũng nhận ra là khi áp dụng một thuộc tính CSS cho một thành phần nào đó ví dụ như h1, h2, p, a, img,... thì toàn bộ các thành phần này trong trang web đều nhận thuộc tính này. Vậy có các nào để nhóm lại một số thành phần nào đó để áp dụng một thuộc tính đặc biệt. Ví dụ như bạn muốn các liên kết trên menu trang web sẽ được in hoa, và có kích cỡ lớn hơn so với liên kết trong nội dung thì phải làm thế nào? Đây chính là vấn đề mà chúng ta sẽ cùng giải quyết trong chương này.

Nhóm các phần tử với class :

Ví dụ chúng ta có một đoạn mã HTML sau đây :

```
<p>Danh Sách Các Tỉnh, Thành Phố Của Việt Nam</p>
<ul>
<li>Hà Nội</li>
<li>TP. Hồ Chí Minh</li>
<li>Đà Nẵng</li>
<li>Thừa Thiên Huế</li>
<li>Khánh Hòa</li>
<li>Quảng Ninh</li>
<li>Tiền Giang</li>
</ul>
```

Yêu cầu đặt ra là làm thế nào để tên các thành phố là màu đỏ và tên các tỉnh là màu xanh da trời. Để giải quyết vấn đề này chúng ta sẽ dùng một thuộc tính HTML gọi là class để tạo thành 2 nhóm là thành phố và tỉnh. Ta sẽ viết lại đoạn HTML sau thành như thế này:

```

<p>Danh Sách Các Tỉnh, Thành Phố Của Việt Nam</p>
<ul>
<li class="tp">Hà Nội</li>
<li class="tp">TP. Hồ Chí Minh</li>
<li class="tp">Đà Nẵng</li>
<li class="tinh">Thừa Thiên Huế</li>
<li class="tinh">Khánh Hòa</li>
<li class="tinh">Quảng Ninh</li>
<li class="tinh">Tiền Giang</li>
</ul>

```

Với việc dùng class để nhóm các đối tượng như trên thì công việc của chúng ta sẽ trở nên đơn giản hơn nhiều:

```

li.tp {          color:FF0000
    } li.tinh {   color:0000FF
    }

```

Lưu ý: Không nên đặt tên class với ký tự đầu là chữ số, nó sẽ không làm việc cho Firefox.

Nhận dạng phần tử với id:

Ví dụ:

Cũng với đoạn HTML như ví dụ về class. Nhưng yêu cầu đặt ra là Hà Nội sẽ có màu đỏ sậm, TP. Hồ Chí Minh màu đỏ, Đà Nẵng màu đỏ tươi còn các tỉnh màu xanh da trời. Để giải quyết vấn đề này chúng ta sẽ sử dụng thuộc tính HTML là id để nhận dạng mỗi thành phố và dùng class để nhóm các tỉnh. Đoạn HTML của chúng ta bây giờ sẽ là :

```

<p>Danh Sách Các Tỉnh, Thành Phố Của Việt Nam</p>
<ul>
<li id="hanoi">Hà Nội</li>
<li id="hcmc">TP. Hồ Chí Minh</li>
<li id="danang">Đà Nẵng</li>
<li class="tinh">Thừa Thiên Huế</li>
<li class="tinh">Khánh Hòa</li>
<li class="tinh">Quảng Ninh</li>

```

```
<li class="tinh">Tiền Giang</li>
</ul>
```

Và đoạn CSS cần dùng sẽ là :

```
#hanoi { color:# 790000 }
#hcmc { color:#FF0000 }
#danang { color:#FF00FF } .tinh { color:#0000FF }
```

Lưu ý: Không nên đặt tên id với ký tự đầu là chữ số, nó sẽ không làm việc cho Firefox.

Trả qua hai ví dụ trên chúng ta có thể rút ra những kết luận sau:

Class dùng để nhóm các đối tượng có cùng thuộc tính, do tính chất đó nó có thể được sử dụng nhiều lần.

Id dùng để nhận dạng một đối tượng đặc trưng, id có tính duy nhất.

Trong bài học này, chúng ta đã được học về các sử dụng class và id để áp dụng các đặc tính đặc biệt cho một thành phần web. Ở bài kế tiếp chúng ta sẽ được học thêm về hai thẻ <div> và trong HTML và ý nghĩa 2 thẻ này đối với việc viết CSS.

2.2.3. Span và Div

Như đã hứa, ở bài học này chúng ta sẽ cùng tìm hiểu về 2 thẻ và <div> trong HTML và xem chúng có lợi ích gì cho công việc viết CSS của chúng ta.

Nhóm phần tử với thẻ :

Thẻ trong HTML thật ra là một thẻ trung hòa, nó không thêm hay bớt bất cứ một thứ gì vào một tài liệu HTML cả. Nhưng chính nhờ tính chất trung hòa này mà nó lại là một công cụ đánh dấu tuyệt vời để qua đó chúng ta có thể viết CSS định dạng cho các phần tử mong muốn.

Ví dụ: Chúng ta có đoạn HTML sau trích dẫn câu nói của chủ tịch Hồ Chí Minh

```
<p>Không có gì quý hơn độc lập, tự do.</p>
```

Yêu cầu ở đây là chúng ta hãy dùng CSS tô đậm 2 từ độc lập, tự do. Để giải quyết vấn đề này, chúng ta sẽ thêm thẻ vào đoạn HTML như sau:

```
<p>Không có gì quý hơn <span class="nhanmanh">độc lập</span>, <span class="nhanmanh">tự do</span>.</p>
```

Và bây giờ chúng ta có thể viết CSS cho yêu cầu trên:

```
.nhanmanh {
```

```
font-weight:bold }
```

Rất đơn giản phải không nào.

Nhóm khối phần tử với thẻ <div>:

Cũng như , <div> cũng là một thẻ trung hòa và được thêm vào tài liệu HTML với mục đích nhóm các phần tử lại cho mục đích định dạng bằng CSS. Tuy nhiên, điểm khác biệt là dùng để nhóm một khối phần tử trong khi đó <div> có thể nhóm một hoặc nhiều khối phần tử.

Trở lại ví dụ về danh sách tỉnh, thành trong phần class bài trước chúng ta sẽ giải quyết vấn đề bằng cách nhóm các phần tử với <div> như sau:

```
<p>Danh Sách Các Tỉnh, Thành Phố Việt Nam:</p>
```

```
<ul>
```

```
<div id="tp">
```

```
<li>Hà Nội</li>
```

```
<li>TP. Hồ Chí Minh</li>
```

```
<li>Đà Nẵng</li>
```

```
</div>
```

```
<div id="tinh">
```

```
<li>Thừa Thiên Huế</li>
```

```
<li>Khánh Hòa</li>
```

```
<li>Quảng Ninh</li>
```

```
<li>Tiền Giang</li>
```

```
</div>
```

```
</ul>
```

Và đoạn CSS cho mục đích này sẽ là:

```
#tp { color:#FF0000  
}
```

```
#tinh { color:0000FF  
}
```

Trong hai bài học trên, chúng ta đã được học qua về id, class, <div>, và lợi ích của nó trong việc nhóm, đánh dấu phần tử để có thể dùng CSS để tạo ra các kiểu trình bày đặc biệt. Có thể một số bạn chưa học qua HTML sẽ khó khăn trong việc nắm bắt các kiến thức này. Tuy nhiên, yêu cầu mà Pearl đưa ra ở đây chỉ là các bạn có thể vận dụng các phần tử này trong CSS.

BÀI 4: LẬP TRÌNH JAVASCRIPT

Mã Bài: MD16_04

Giới thiệu:

Javascript đóng vai trò rất quan trọng khi bạn xây dựng website. Nhiệm vụ của Javascript là xây dựng các hiệu ứng động giúp website trở nên thân thiện với người dùng hơn.

Để học Javascript tốt thì đòi hỏi bạn phải biết CSS căn bản kết hợp với HTML để tạo nên giao diện và kết hợp chúng với nhau.

Mục tiêu:

- Trình bày được nguyên lý hoạt động của Javascript trong trang web;
- Sử dụng được các kiểu dữ liệu cơ bản trong lập trình Javascript;
- Xây dựng được hàm javascript để xử lý dữ liệu, xử lý yêu cầu;
- Rèn luyện tính nghiêm túc, tích cực trong việc tiếp nhận kiến thức. Chủ động, tích cực tìm kiếm các nguồn tài liệu liên quan.

Nội dung chính:

1. Tổng quan về Javascript

1.1. Giới thiệu chung

Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa ra, javascript được sử dụng rộng rãi trong các ứng dụng Website. Javascript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, ... thậm chí các trình duyệt trên thiết bị di động cũng có hỗ trợ.

Nếu bạn đã biết qua về HTML5 thì bạn thấy có các khái niệm như sessionStorage hay localStorage, đây là hai đối tượng được tạo nên từ Javascript nên rõ ràng trong HTML5 cũng có sử dụng nó. Với những ứng dụng đó thì bạn thấy javascript không thể chết trong các ứng dụng website được.

Hay thậm chí có những ứng dụng Webgame người ta sử dụng javascript để xử lý các thao tác trên Client, nếu không có nó thì thông thường chọn Flash để xây dựng nhưng lại gặp vấn đề load chậm nên có một số người chọn Javascript để làm.

1.2. Vị trí đặt mã Javascript trong trang web HTML

Bước đầu tiên để học Javascript là bạn phải chuẩn bị một phần mềm để viết mã Javascript, mình thì thích dùng Netbeans nhưng nếu bạn là người mới thì nên sử dụng Notepad++ vì nó rất nhẹ và dễ cài đặt.

Tất cả các trình duyệt đều hỗ trợ javascript nên ta không cần phải download thư viện gì cả mà thông qua những đoạn mã `script` giúp trình duyệt nhận diện được đó là Javascript.

Cặp thẻ mở và thẻ đóng

Tất cả những đoạn mã Javascript đều phải đặt trong cặp thẻ mở `<script>` và thẻ đóng `</script>`, ví dụ:

Cặp thẻ script

```
<script language="javascript">
    alert("Hello World!");
</script>
```

Đặt thẻ script ở đâu?

Chúng ta có ba cách đặt thường được sử dụng sau đây:

Cách 1: Internal - viết trong file html hiện tại

Thông thường chúng ta sẽ viết những đoạn mã javascript trên phần `head`, tuy nhiên đó không phải là điều kiện bắt buộc, nghĩa là bạn có thể đặt ở đâu tùy thích miễn là những đoạn mã đó phải được bao lại bằng thẻ `script`.

Ví dụ đặt trong thẻ head

```
<html>
  <head>
    <title></title>
    <script language="javascript">
      alert("Hello World!");
    </script>
  </head>
  <body>
  </body>
</html>
```

Ví dụ đặt trong thẻ body

```
<html>
  <head>
    <title></title>
  </head>
  <body>
    <script language="javascript">
      alert("Hello World!");
    </script>
```

```
</body>
</html>
```

Cách 2: External - viết ra một file js khác rồi import vào

Bạn có thể viết những đoạn mã javascript ở một file có phần mở rộng là `.js`, sau đó dùng thẻ `script` để import vào (giống CSS vậy). Ví dụ file JS của mình có tên là `demo.js` thì lúc này mình import vào như sau:

Code import JS

```
<script language="javascript" src="demo.js"></script>
```

Lúc này bên trong file `demo.js` bạn không đặt thẻ `script` nữa nhé vì nó là file có phần đuôi là `.js` rồi nên trình duyệt tự nhận diện đây là file chứa mã Javascript.

Cách 3: Inline - viết trực tiếp trong thẻ HTML

Inline nghĩa là bạn sẽ viết những đoạn mã Javascript trực tiếp trong thẻ HTML luôn. Như ví dụ dưới đây là mình đang viết dạng inline vì đoạn mã `alert(1)` được đặt trong sự kiện `onclick` của thẻ `button`.

Code bên trong sự kiện click

```
<input type="button" onclick="alert(1)" value="Click Me"/>
```

1.3. HTML DOM

1.3.1. Thay đổi và lấy nội dung bên trong thẻ HTML

Để lấy nội dung bên trong một thẻ HTML thì chúng ta sử dụng cú pháp như sau:

```
var html = document.getElementById("content").innerHTML
```

Và để thay đổi nội dung cho một thẻ HTML thì ta dùng cú pháp sau:

```
var html = document.getElementById("content").innerHTML = "<h1>Nội dung</h1>";
```

Ví dụ: Trong ví dụ này ta sẽ viết chương trình thay đổi nội dung HTML của một thẻ DIV và lấy nội dung bên trong của một thẻ DIV

```
<html>
<body>
  <script language="javascript">
    // Hàm thiết lập nội dung cho thẻ div#content
    function set_content()
    {
      document.getElementById("content").innerHTML = "<h1>Nội dung đã được thay đổi</h1>";
    }

    // Hàm lấy nội dung cho thẻ div#content
    function get_content()
```

```

    {
        var html = document.getElementById("content").innerHTML;
        alert("Nội dung cần lấy là: " + html);
    }

</script>
<div id="content">Nội dung của thẻ DIV</div>
<input type="button" value="Lấy nội dung" id="get_content"
onclick="get_content()" />
<input type="button" value="Thay đổi nội dung" id="set_content"
onclick="set_content()" />
</body>
</html>

```

1.3.2. Thay đổi và lấy giá trị thuộc tính thẻ HTML bằng Javascript

Để thay đổi giá trị của một thuộc tính HTML bất kì thì ta sử dụng cú pháp sau:

```
document.getElementById("element").attributeName = "new value";
```

Để lấy giá trị của một thuộc tính HTML ta sử dụng cú pháp sau:

```
var value = document.getElementById("element").attributeName;
```

Quá đơn giản phải không nào, rất giống với cách thay đổi và lấy nội dung bên trong thẻ HTML. Từ đây có thể suy ra rằng trong Javascript để thiết lập (set) và lấy (get) thì sử dụng chung một cú pháp, chỉ khác nhau ở chỗ **gán bằng** và **không có gán bằng**.

Ví dụ: Xây dựng chương trình khi click vào một **button** thì chuyển nó thành **textbox**, và tiếp tục click vào **textbox** thì sẽ đổi thành **button**

```

<html>
<body>
    <script language="javascript">
        function change()
        {
            // Lấy đối tượng
            var object = document.getElementById("object");

            // lấy thuộc tính type
            var type = object.type;

            // kiểm tra thuộc tính type và thay đổi
            if (type == "button"){

```

```

        object.type = 'text';
    }
    else{
        object.type = "button";
    }
}

</script>
<input type="button" value="Click me" onclick="change()" id="object"
/>

</body>
</html>

```

2. Javascript cơ bản

2.1. Câu lệnh, khối lệnh, ghi chú

Một câu lệnh trong javascript được kết thúc bởi dấu chấm phẩy (;).

Một khối lệnh gồm nhiều câu lệnh được đặt trong cặp dấu { và }.

Ghi chú (Comment): là các dòng mô tả thêm của người lập trình sẽ được bỏ qua trong quá trình biên dịch và chạy.

Cú pháp 1:

Chú thích 1 dòng:

// chú thích

Cú pháp 2

Chú thích nhiều dòng:

*/**

chú thích

**/*

Ví dụ:

```

function myfunction(arg1, arg2)
{
    /* This is a multiline comment that
       can span as many lines as necessary. */
    var r;
    // This is a single line comment.
    r = arg1 + arg2; // Sum the two arguments.
    return(r);
}

```

2.2. Kiểu dữ liệu

Kiểu động

JavaScript là một ngôn ngữ định kiểu yếu hay động. Điều đó nghĩa là không cần phải khai báo kiểu của các biến trước khi dùng. Kiểu sẽ được xác định tự động trong khi chương trình được thực thi. Điều đó cũng có nghĩa là một biến có thể chứa giá trị của các kiểu dữ liệu khác nhau:

```
var foo = 42; // foo là một số
```

```
var foo = "bar"; // foo là một chuỗi
```

```
var foo = true; // foo là một boolean
```

Tiêu chuẩn ECMAScript mới nhất xác định bảy sáu kiểu sơ khai:

Kiểu boolean

Kiểu của null

Kiểu của undefined

Kiểu số

Kiểu chuỗi

Kiểu Symbol (mới trong ECMAScript 6)

và Kiểu đối tượng

Tất cả các kiểu trừ đối tượng đều được xác định giá trị bất biến (giá trị, không có khả năng thay đổi). Ví dụ và không giống C, một chuỗi là bất biến. Ta gọi chúng là "giá trị sơ khai" ("primitive").

Kiểu boolean

Kiểu boolean đại diện có hai giá trị logic: true, và false.

Kiểu null

Có duy nhất một giá trị: null. Xem null và Null để biết thêm chi tiết.

Kiểu undefined

Một biến chưa được gán giá trị có giá trị undefined. Xem undefined và Undefined để biết thêm chi tiết.

Kiểu số

Theo tiêu chuẩn ECMAScript, chỉ có duy nhất một kiểu số: the double-precision 64-bit binary format IEEE 754 value (có giá trị từ $-(2^{53}-1)$ đến $2^{53}-1$). Không có kiểu số nguyên. Ngoài việc có thể chứa giá trị dấu phẩy động, kiểu số có ba giá trị biểu tượng: +Infinity, -Infinity, and NaN (not-a-number).

Để kiểm tra lớn hơn hay nhỏ hơn +/-Infinity, bạn có thể xem Number.MAX_VALUE hoặc Number.MIN_VALUE và bắt đầu từ ECMAScript

6, bạn cũng có thể kiểm tra một số có nằm trong khoảng double-precision floating-point bằng cách dùng `Number.isSafeInteger()` cũng như `Number.MAX_SAFE_INTEGER` và `Number.MIN_SAFE_INTEGER`.

Ngoài phạm vi này, một số trong Javascript không còn an toàn nữa.

Có một số nguyên duy nhất có hai đại diện: 0 được đại diện bởi -0 và +0. ("0" là một bí danh của +0). Trong thực tế, điều này hầu như không có tác động. Ví dụ `+0 === -0` là true. Tuy nhiên, có thể nhận thấy điều này khi chia một số cho không:

> 42 / +0

Infinity

> 42 / -0

-Infinity

Mặc dù một số thường chỉ đại diện cho giá trị của nó, JavaScript cung cấp một vài toán tử nhị phân. Chúng có thể được sử dụng như một chuỗi boolean bằng cách dùng bit masking. Điều này thường được xem như là một cách tệ, tuy nhiên, JavaScript không cung cấp bất kỳ phương tiện nào khác để trình bày một tập hợp các boolean (như một mảng các boolean hay một đối tượng với các thuộc tính boolean). Bit masking cũng có xu hướng làm mã khó đọc, hiểu, và duy trì hơn. Nó có thể cần thiết trong một môi trường rất hạn chế, giống như khi cố gắng để đối phó với hạn chế lưu trữ cục bộ hoặc trong trường hợp nặng khi mỗi chút so với đếm mạng. Kỹ thuật này chỉ nên được xem xét khi nó là biện pháp cuối cùng có thể được thực hiện để tối ưu hóa kích thước.

Kiểu chuỗi

Kiểu chuỗi được dùng để biểu diễn dữ liệu dạng văn bản. Nó là một dãy "các phần tử" số nguyên 16-bit. Mỗi phần tử có một vị trí trong chuỗi. Phần tử đầu tiên có chỉ số 0, tiếp theo là 1, Độ dài của chuỗi là số phần tử của nó.

Không giống với những ngôn ngữ như C, Chuỗi trong Javascript là bất biến. Nghĩa là một khi chuỗi được tạo thì không thể chỉnh sửa. Tuy nhiên, vẫn có thể tạo một chuỗi mới dựa vào các thao tác trên chuỗi cũ. Ví dụ:

Tạo một chuỗi con của chuỗi ban đầu bằng cách ghép từng ký tự hoặc dùng `String.substr()`.

Nối hai chuỗi bằng toán tử (+) hoặc hàm `String.concat()`.

Cẩn thận với mã "stringly-typing"!

Chuỗi có thể được dùng để biểu diễn dữ liệu với cấu trúc phức tạp. Điều này mang tới một vài lợi ích ngắn hạn:

Rất dễ để xây dựng một chuỗi bằng phép nối.

Dễ debug (những gì bạn thấy khi in luôn là tất cả những thứ có trong chuỗi).

Chuỗi là mẫu số chung của rất nhiều API (nhập, local storage values, XMLHttpRequest phản hồi khi dùng responseText, ...) và điều này có thể khiến việc chỉ làm việc với chuỗi được yêu thích.

Chuỗi có thể biểu diễn bất kì kiểu dữ liệu nào. Những đây không được xem là một ý hay. Ví dụ, đối với một separator, có thể bắt trước một chuỗi (trong khi một mảng sẽ thích hợp hơn). Thật không may, khi separator được dùng trong một "danh sách" các phần tử, danh sách bị hỏng. Một escape character có thể được chọn, Tất cả những điều này yêu cầu một quy ước và tạo ra gánh nặng bảo trì không cần thiết.

Chỉ nên dùng chuỗi để lưu trữ dữ liệu văn bản. Khi biểu diễn một cấu trúc phức tạp, phân tích chuỗi thành các cấu trúc dữ liệu với mức trừu tượng cao hơn.

Kiểu Symbol

Kiểu Symbol là một kiểu mới trong Javascript tiêu chuẩn ECMAScript 6. Mỗi Symbol là một giá trị sơ khai đơn nhất và bất biến và có thể được dùng như một khóa của một Object (xem bên dưới). Trên một số ngôn ngữ lập trình, Symbol còn được gọi là "atom" (nguyên tử). Ta cũng có thể so sánh với các enumeration (enum) trong C. Xem Symbol và Symbol để biết thêm chi tiết.

2.3. Các toán tử

Toán tử	Ký hiệu
Cộng	+
Gán giá trị	=
AND	&
NOT	~
OR	
XOR	^
Chia	/
So sánh bằng	==
S/s lớn hơn	>

S/s lớn hơn hoặc bằng	>=
Giảm 1 đơn vị	--
Tăng 1 đơn vị	++
So sánh không bằng	!=
S/s nhỏ hơn	<
S/s nhỏ hơn hoặc bằng	<=
AND	&&
NOT	!
OR	
Chia lấy phần dư	%
Nhân	*
Tạo object mới	new
Trừ	-

Lưu ý: Giá trị logic trả về:

Đúng	true
Sai	false

2.4. Biến

2.4.1. Khai báo và gán giá trị cho biến

2.4.1.1. Khai báo biến trong javascript

Để khai báo một biến ta sử dụng từ khóa `var tenbien`.

Khai báo biến

```
var username;
```

Tuy nhiên không phải muốn đặt tên gì cho biến cũng được mà phải tuân theo quy tắc sau đây:

Tên biến phải là các chữ không dấu viết hoa hoặc viết thường, các chữ số từ 0-9 và dấu gạch dưới (_).

Tên biến bắt đầu phải là chữ hoặc dấu gạch dưới (_), nếu bắt đầu bằng số là sai

Tên biến có thể đặt dài hay ngắn tùy vào lập trình viên.

Ví dụ các cách khai báo biến

```
// Đúng
var username;

// Đúng
var _username;

// Đúng
var __username;

// Đúng
var username90;

// SAI
var 90thehalfheart;
```

Ngoài cách khai báo như trên ta còn có cách khai báo nhiều biến cùng một lúc bằng cách viết chúng liên tiếp với nhau và cách nhau bởi dấu phẩy.

Ví dụ khai báo nhiều biến

```
var username, _username, __username, username90;
```

2.4.1.2. Gán giá trị cho biến trong javascript

Để gán giá trị cho biến ta dùng dấu bằng (=) để gán về phải vào về trái, ví dụ gán giá trị **thehalfheart** vào biến **username** thì ta viết là `username = 'thehalfheart'`.

Có hai cách gán thông dụng đó là vừa khai báo biến vừa gán giá trị và khai báo rồi mới gán giá trị:

Vừa khai báo vừa gán giá trị

```
var username = 'thehalfheart';
```

Khai báo xong mới gán giá trị

```
var username;

username = 'thehalfheart';
```

Trong Javascript rất hạn chế về kiểu dữ liệu nhưng bù lại cơ chế xử lý kiểu dữ liệu của nó rất linh hoạt giúp lập trình viên có thể chuyển đổi một cách dễ dàng, bởi mọi kiểu dữ liệu đều có thể quy về đối tượng và mỗi đối tượng ta có thể bổ sung các phương thức xử lý riêng.

Giống như php, để xác định biến có kiểu dữ liệu là gì thì ta dựa vào giá trị mà nó đang có, có nghĩa khi bạn gán một con số không có dấu chấm động thì nó sẽ là kiểu INT, nếu gán một chuỗi thì là kiểu String.

Gán kiểu giá trị cho biến

```
// Biến website đag kiểu String
```

```
var website = 'thehalfhart';  
// Biến website chuyển sang kiểu INT  
website = 12;  
// Biến website chuyển sang kiểu float  
website = 12.5;
```

Trong Javascript bạn sẽ có các kiểu dữ liệu thông dụng như kiểu chuỗi (*String*), số (*Number*), mảng (*Array*), đối tượng (*Object*). Chi tiết từng kiểu dữ liệu chúng ta sẽ được học trong các bài tiếp theo.

2.4.2. Sử dụng biến

Để in giá trị của biến hoặc một chuỗi nào đó ra trình duyệt thì ta sử dụng hàm `document.write(value)`.

Ví dụ: In giá trị của biến

```
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
    <script language="javascript">  
      var website = 'freetuts.net';  
      document.write(website);  
    </script>  
  </body>  
</html>
```

2.4.3 Phạm vi của biến

2.4.3.1. Biến cục bộ trong Javascript

Một biến được gọi là cục bộ khi bạn khai báo nó nằm bên trong một hàm cụ thể nào đó, lúc này biến đó sẽ không sử dụng được ở bên ngoài **hàm**.

Ví dụ: Trong ví dụ này đoạn code `alert()` biến `comment` ở ngoài function `add_comment()` sẽ thông báo lỗi vì nó chưa được định nghĩa

```
function add_comment()  
{  
  var comment = "Nội dung comment";  
  // Đoạn code này đúng vì biến comment đã tồn tại  
  alert(comment);  
}  
  
// Đoạn code này sẽ sai vì không tồn tại biến comment
```

```
alert(comment);
```

2.4.3.2. *Biến toàn cục trong Javascript*

Biến toàn cục là biến mà bạn khai báo bên ngoài và không nằm bên trong một hàm cụ thể nào cả.

Ví dụ: Trong ví dụ này không có đoạn code nào sai cả

```
// Biến toàn cục
```

```
var comment = "Nội dung comment bên ngoài";
```

```
// Hàm có sử dụng biến toàn cục
```

```
function add_comment()
```

```
{
```

```
    alert(comment);
```

```
}
```

```
// In biến toàn cục
```

```
alert(comment);
```

2.5. *Cấu trúc điều khiển*

2.5.1. *break*

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

Ví dụ:

```
function BreakTest(breakpoint)
```

```
{
```

```
    var i = 0;
```

```
    while (i < 100)
```

```
    {
```

```
        if (i == breakpoint)
```

```
            break;
```

```
            i++;
```

```
    }
```

```
    return(i);
```

```
}
```

2.5.2. *continue*

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

Ví dụ:

```

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

```

2.5.3. *do...while*

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

Cú pháp

do

statement

while (*expression*) ;

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

Ví dụ:

```

function GetDriveList()
{
var fso, s, n, e, x;
fso = new ActiveXObject("Scripting.FileSystemObject");
e = new Enumerator(fso.Drives);
s = "";
do
{

```

```

x = e.item();
s = s + x.DriveLetter;
s += " - ";
if (x.DriveType == 3)
    n = x.ShareName;
else if (x.IsReady)
    n = x.VolumeName;
else
    n = "[Drive not ready]";
s += n + "<br>";
e.moveToNext();
}
while (!e.atEnd());
return(s);
}

```

2.5.4. *for*

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

Cú pháp

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

statement

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

Ví dụ:

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

```

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

```

2.5.5. *for..in*

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

Cú pháp

```

for (variable in [object | array])
    statement

```

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

Ví dụ:

```

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

```

```
}
```

2.5.6. *function*

Khai báo 1 function mới.

Cú pháp

```
function functionname([argument1 [, argument2 [, ...argumentn]]])
```

```
{  
  statements  
}
```

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

Ví dụ:

```
function myfunction(arg1, arg2)
```

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

2.5.7. *if...else*

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

Cú pháp

```
if (condition)
```

```
  statement1
```

```
[else
```

```
  statement2]
```

Phần	Diễn giải
<i>condition</i>	Là biểu thức boolean. Nếu <i>condition</i> là null hoặc undefined , <i>condition</i> được gán là false.

<i>statement1</i>	Nhóm lệnh được thực hiện nếu <i>condition</i> = true.
<i>statement2</i>	Nhóm lệnh được thực hiện nếu <i>condition</i> = false.

Ví dụ:

```

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

```

Ví dụ:

```

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

```

Ví dụ:

```

if (x == 5)
    z = 7;
    q = 42;
else
    z = 19;

```

2.5.8. return

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

Cú pháp

```
return [expression];
```

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

Ví dụ:

```

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

```

```
return(r);
}
```

2.5.9. switch

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

Cú pháp

```
switch (expression) {
case label :
    statementlist
case label :
    statementlist
...
default :
    statementlist
}
```

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

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

Ví dụ:

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

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

2.5.10. *this*

Tham khảo đến object hiện tại.

Cú pháp

this.property

Ví dụ:

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

2.5.11. *while*

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

Cú pháp

while (expression)

statement

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

Ví dụ:

```
function BreakTest(breakpoint)
{
  var i = 0;
  while (i < 100)
```

```

{
if (i == breakpoint)
    break;
    i++;
}
return(i);
}

```

3. Hàm trong Javascript

3.1. Giới thiệu chung

Hàm là một phương pháp lập trình truyền thống và thường được ứng dụng trong các phương pháp lập trình thủ tục, lập trình hướng module,... Với các ngôn ngữ lập trình bậc cao như C++ thì việc dùng hàm để code ứng dụng là người ta không thích tại vì khó quản lý, nâng cấp và bảo trì, nhưng đối với javascript và PHP thì ta rất hay dùng hàm.

3.2. Định nghĩa hàm Javascript

3.2.1. Định nghĩa hàm

Hàm sẽ gom một số đoạn code vào một khối xử lý và khi cần thì gọi ra dùng.

Giả sử mình viết chương trình kiểm tra số chẵn hay lẻ như sau:

```

var number = 2;
if (number % 2 == 0){
    alert('Số chẵn');
}
else {
    alert('Số lẻ');
}

```

Giả sử mình cần kiểm tra cho 100 số, như vậy mình cứ viết if else cho 100 lần quá nản phải không nào :D. Để giải quyết nó ta tìm hiểu hàm đã nhé.

Cú pháp:

```

function name_of_function(var1, var2, var3, ...)
{
    // Some code
}

```

Trong đó:

function: là từ khóa của javascript nên bắt buộc phải như vậy

name_of_function: là tên của function, thông thường chúng ta tạo những tên có ý nghĩa như find_max, find_min, ...

var1, var2 var3, ... là các tham số truyền vào hàm. Ví dụ viết hàm kiểm tra số chẵn hay lẻ thì ta sẽ có một tham số đó là số cần kiểm tra.

Ví dụ: Viết hàm kiểm tra một số chẵn hay lẻ.

```
// Tạo hàm
function check_number(number)
{
    if (number % 2 == 0){
        alert(number + ' là số chẵn');
    }
    else {
        alert(number + 'Số lẻ');
    }
}
// Sử dụng hàm kiểm tra cho 5 số
check_number(1);
check_number(2);
check_number(3);
check_number(4);
check_number(5);
```

Các bạn thấy mình đã tạo một hàm với tham số truyền vào có tên là number. Như vậy khi sử dụng nếu ta truyền số 1 vào thì lúc này biến number trong hàm sẽ có giá trị là 1, tương tự cho 2, 3,4,5.

Lưu ý với các bạn trong javascript không tồn tại khái niệm con trỏ và tham chiếu trong hàm

3.2.2. Hàm có return và hàm không có return

Hàm có return là hàm có sử dụng từ khóa `return` để đặt ở cuối hàm với mục đích trả kết quả về để sử dụng tiếp ở những đoạn code bên ngoài. Ví dụ ta cần viết một hàm tính tổng của hai số a và b thì hàm này phải trả về giá trị là tổng của hai số a, b. Xem ví dụ sau:

```
// Khai báo hàm
function tinh_tong(a, b)
{
    // trả về kết quả là a + b
    return a + b;
}
// Sử dụng
var so1 = 1;
var so2 = 2;
```

```
// truyền so1 và so2 vào hàm
var ketqua = tinh_tong(so1, so2);
alert(ketqua);
```

Hàm không có return là hàm không có sử dụng từ khóa return đặt trong hàm. Ví dụ viết chương trình in ra tổng của hai số a và b.

```
// Khai báo hàm
function tinh_tong(a, b)
{
    document.write('Tổng là ' + (a + b));
}

// Sử dụng
var so1 = 1;
var so2 = 2;

// truyền so1 và so2 vào hàm
tinh_tong(so1, so2);
```

Như vậy tùy vào mục đích mà ta dùng có return hay không có return. Nhưng thông thường ta sử dụng return ở những trường hợp cần lấy kết quả đó để xử lý tiếp, như ở ví dụ trên đó là mình lấy kết quả để in thông báo.

3.2.3. Giá trị mặc định của tham số

Có một số trường hợp bạn muốn một tham số nào đó có thể được truyền hoặc không cần truyền vào đều được, lúc này chúng ta phải sử dụng nó như một tham số mặc định. Javascript không có cú pháp gán giá trị mặc định như PHP mà thay vào đó chúng ta sử dụng toán tử `||`.

Ví dụ: Hàm hiển thị một thông báo.

```
function showMessage(message)
{
    // Nếu message không được truyền vào hoặc giá trị nó là rỗng
    // thì sẽ được thay thế bằng giá trị 'Không có tin nhắn'
    message = message || 'Không có tin nhắn <br/>';
    document.write(message);
}

// Cách 1: không truyền tham số
showMessage();

// Cách 2: Truyền tham số
showMessage('Chào mừng bạn đến với freetuts.net');
```

Toán tử `||` gồm hai vế, trong đó nếu vế phải có giá trị rỗng (*undefined*, *null*, *false*, ...) kết quả sẽ trả về vế trái, ngược lại nó sẽ trả về vế phải.

3.3. Gọi hàm Javascript

Để gọi hàm chúng ta sử dụng cú pháp:

<tên hàm>(danh sách tham số nếu có);

Ví dụ:

```
var ketqua = tinh_tong(so1, so2);
```

3.4. Hàm định nghĩa sẵn của Javascript

3.4.1. substr

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

Ví dụ

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

3.4.2. abs

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

Ví dụ:

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

3.4.3. atEnd

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

Ví dụ:

```

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

```

3.4.4. *charAt*

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

Ví dụ:

```

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

```

3.4.5. *dimensions*

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

Ví dụ:

```

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

```

3.4.6. eval

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

Ví dụ:

```

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

```

3.4.7. *getDate*

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

Ví dụ:

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

3.4.8. *getItem*

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

Ví dụ:

```
<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Function CreateVBArray()
    Dim i, j, k
    Dim a(2, 2)
    k = 1
    For i = 0 To 2
        For j = 0 To 2
            a(i, j) = k
            document.writeln(k)
            k = k + 1
        Next
        document.writeln("<BR>")
    Next
    CreateVBArray = a
End Function
-->
</SCRIPT>
```

```

<SCRIPT LANGUAGE="JScript">
<!--
function GetItemTest(vbarray)
{
    var i, j;
    var a = new VBArray(vbarray);
    for (i = 0; i <= 2; i++)
    {
        for (j =0; j <= 2; j++)
        {
            document.writeln(a.getItem(i, j));
        }
    }
}-->
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JScript">
<!--
    GetItemTest(CreateVBArray());
-->
</SCRIPT>
</BODY>

```

3.4.9. *getTime*

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

Ví dụ:

```

function GetTimeTest()
{
    var d, s, t;
    var MinMilli = 1000 * 60;
    var HrMilli = MinMilli * 60;
    var DyMilli = HrMilli * 24;
    d = new Date();
    t = d.getTime();
    s = "It's been "
    s += Math.round(t / DyMilli) + " days since 1/1/70";
}

```

```
return(s);  
}
```

3.4.10. *getDay*

objDate.getDay(): Trả về thứ trong tuần với:

0 = Sunday

1 = Monday

2 = Tuesday

3 = Wednesday

4 = Thursday

5 = Friday

6 = Saturday

Ví dụ:

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

3.4.11. *join*

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

Ví dụ:

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

3.4.12. *lbound*

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

Ví dụ:

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

<SCRIPT LANGUAGE="JScript">
<!--
function VBArrayTest(vba)
{
    var i, s;
    var a = new VBArray(vba);
    for (i = 1; i <= a.dimensions(); i++)
    {
        s = "The lower bound of dimension ";
        s += i + " is ";
        s += a.lbound(i)+ "<BR>";
        return(s);
    }
}
-->
</SCRIPT>
</HEAD>

<BODY>
<SCRIPT language="jscript">
```

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

3.4.13. match

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

Ví dụ:

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

3.4.14. max

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

Ví dụ:

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

3.4.15. min

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

Ví dụ:

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

3.4.16. moveFirst

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

Ví dụ:

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

3.4.17. moveNext

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

Ví dụ:

```
function ShowDriveList()
{
```

```

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

```

3.4.18. replace

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

Ví dụ:

```

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

```

3.4.19. reverse

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

Ví dụ:

```

function ReverseDemo()
{

```

```

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

```

3.4.20. *sort*

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

Ví dụ:

```

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

```

3.4.21. *split*

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

Ví dụ:

```

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

```

3.4.22. *substring*

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

Ví dụ:

```

function SubstringDemo()
{
  var s, ss;

```

```

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

```

3.4.23. *test*

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

Ví dụ:

```

function TestDemo(re, s)
{
var s1;
// Test string for existence of regular expression.
if (re.test(s))
    s1 = " contains ";
else
    s1 = " does not contain ";
// Get text of the regular expression itself.
return(s + s1 + re.source);
}

```

3.4.24. *toArray*

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

Ví dụ:

```

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

```

```

        document.writeln("<BR>")
    Next
    CreateVBArray = a
End Function
-->
</SCRIPT>
<SCRIPT LANGUAGE="JScript">
<!--
function VBArrayTest(vbarray)
{
    var a = new VBArray(vbarray);
    var b = a.toArray();
    var i;
    for (i = 0; i < 9; i++)
    {
        document.writeln(b[i]);
    }
}
-->
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JScript">
<!--
    VBArrayTest(CreateVBArray());
-->
</SCRIPT>
</BODY>

```

3.4.25. toLowerCase

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

Ví dụ:

```

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

```

3.4.26. toUpperCase

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

Ví dụ:

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

3.4.27. ubound

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

Ví dụ:

```
<HEAD>  
<SCRIPT LANGUAGE="VBScript">  
<!--  
Function CreateVBArray()  
    Dim i, j, k  
    Dim a(2, 2)  
    k = 1  
    For i = 0 To 2  
        For j = 0 To 2  
            a(j, i) = k  
            k = k + 1  
        Next  
    Next  
    CreateVBArray = a  
End Function  
-->  
</SCRIPT>  
  
<SCRIPT LANGUAGE="JScript">  
<!--  
function VBArrayTest(vba)  
{  
    var i, s;  
    var a = new VBArray(vba);  
    for (i = 1; i <= a.dimensions(); i++)  
    {  
        s = "The upper bound of dimension ";  
        s += i + " is ";  
        s += a.ubound(i) + ".<BR>";  
        return(s);  
    }  
}
```

```
-->
</SCRIPT>
</HEAD>
  <BODY>
<SCRIPT language="jscript">
  document.write(VBArrayTest(CreateVBArray()));
</SCRIPT>
</BODY>
```

3.4.28. *item*

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

Ví dụ:

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

3.4.29. *isNaN*

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

Cú pháp

isNaN(numvalue)

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

Ví dụ:

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

4. Bài tập

Bài tập 01 : Viết chương trình như mẫu sau và chạy website

```
<body>
  <script language="javascript">
    //dữ liệu cơ bản
    var x = 10;
    document.write("x = " + x + "<br/>");
    var s = "Chuyên viên công nghệ web";
    alert("Chuỗi s = " + s.toUpperCase() + "<br/>");
    var y = 5.5;
    document.write("y = " + y + "<br/>");
    var t = x + y;
    alert("Tổng = " + t + "<br/>");
  </script>
</body>
```

Bài tập 02 : Viết chương trình nhập vào một mảng và xuất ra tổng của mảng, sắp xếp mảng tăng dần.

```

<script language="javascript1.5" type="text/javascript">
    var n = prompt("Nhập số phần tử của mảng");
    if(!isNaN(n))
    {
        var A = new Array(n);
        var s = 0;
        for(i=0;i<n;i++)
        {
            A[i] = prompt("Nhập phần tử thứ " + i,0);
            s = s + A[i];
        }
        for(i=0;i<n-1;i++)
            for(j=i+1;j<n;j++)
                if(A[i]>A[j])
                {
                    t = A[i];
                    A[i] = A[j];
                    A[j] = t;
                }
        document.write('Các phần tử sau khi sắp xếp:');
        for(i in A)
            document.write(i + " ");
        document.write("<br/>Tổng của mảng = " + s);
    }
    else
        document.write("Bạn nhập chưa đúng vui lòng refresh lại trang web");
</script>

```

Bài tập 03 : Viết chương trình in ra giờ theo giờ địa phương và cập nhật từng giây

```

<body>
    <div id="clock">
    </div>
    <script language="JavaScript">
        var now = new Date();
        localtime = now.toString();
        document.write("Local time: " + localtime + "<BR>");
        function InNgayGio() {
            var now = new Date();
            hours = now.getHours();
            mins = now.getMinutes();
            secs = now.getSeconds();
            theDiv = document.getElementById("clock");
            theDiv.innerHTML = hours + ":" + mins + ":" + secs;
            window.setInterval(InNgayGio,1000);
        }
        InNgayGio();
    </script>
</body>

```

Bài tập 04 : Tạo trang web chứa các tùy chọn cho phép thay đổi màu nền trang web như sau :

```

<script language="javascript">
    function ChangeColor(color)
    {
        document.bgColor = color;
    }
</script>
<body>
<form method="post">
    <input type="radio" value="Red" name="choose" onclick="ChangeColor(this.value)"/> Red <br/>
    <input type="radio" value="Green" name="choose" onclick="ChangeColor(this.value)"/> Green <br/>
    <input type="radio" value="Blue" name="choose" onclick="ChangeColor(this.value)"/> Blue <br/>
    <input type="radio" value="Yellow" name="choose" onclick="ChangeColor(this);"/> Yellow <br/>
</form>
</body>

```

Bài tập 5: Thiết kế trang đăng ký thành viên và kiểm tra dữ liệu trong trang:

ĐĂNG KÝ THÀNH VIÊN

Họ và Tên:(*)	Dương Thành Phết	
Giới Tính :	Nam	
Ngày sinh :(*)	Ngày: 20 Tháng: 08 Năm: 1976	
Email (*)	phetitc	
Tên Đăng Nhập :(*)	phetitc	
Mật khẩu :(*)		
Nhập lại Mật khẩu :(*)		
Đăng ký Bỏ qua		

ĐĂNG KÝ THÀNH VIÊN

Họ và Tên:(*)	Dương Thành Phết	
Giới Tính :	Nam	
Ngày sinh :(*)	Ngày: 20 Tháng: 08 Năm: 1976	
Email (*)	phetitc@ymail.com	
Tên Đăng Nhập :(*)	phetitc	
Mật khẩu :(*)		
Nhập lại Mật khẩu :(*)		
Đăng ký Bỏ qua		

<html>

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

<head>

<script language="javascript">

```

    function kiểmtra(){
        var
        ht=window.document.dangky.txthoten.value;
        na=window.document.dangky.nam.value;
        email=window.document.dangky.em.value;
        tdn=window.document.dangky.tdn.value;

```

```

mk=window.document.dangky.mk.value;
nlmk=window.document.dangky.nlmk.value;
    alert("vui lòng nhập họ ten");
    window.document.dangky.txthoten.focus();
        return false;
    }
    if(na.length==0){
        alert("hãy điền năm sinh");
        window.document.dangky.nam.focus();
        return false;
    }
    if(isNaN(na)==true){
        alert("Day khong phai la so");
        window.document.dangky.nam.value="";
        document.dangky.nam.focus();
        return false;
    }
    re=/\w+@\w+\.\w+/;        if (email==""){
        alert("Bạn chưa nhập email") ;
        window.document.dangky.em.value="";
        document.dangky.em.focus(); return false
    }
    else if(re.test(email)==false){        alert("Email không đúng
định dạng!");        document.dangky.em.focus();        return
false
    }
    if(tdn.length==0){        alert("hãy điền tên đăng nhập");
        window.document.dangky.tdn.focus();        return
false;
    }
    if(mk.length==0){        alert("hãy nhập mật khẩu");
        window.document.dangky.mk.focus();
        return false;
    }

```

```

        if(nlmk.length==0) {
            alert("hãy nhập lại mật khẩu");
window.document.dangky.nlmk.focus();
            return false;
        }
        if(mk != nlmk){
            alert("mật khẩu và nhập lại mật khẩu ko trùng nhau");
window.document.dangky.nlmk.focus();
            return false;
        }
        alert("Đăng ký thành công! Chúc mừng bạn");
            return true;

    }
</script>
</head>
<body>
<h3 align="center"> ĐĂNG KÝ THÀNH VIÊN </h3>
<form action="dangkythanhvien.htm" name="dangky" method="post"
onSubmit="return kiemtra()">
    <table border="0" align="center" width="500">
    <tr>
    <td width="150">Họ và Tên:(*)</td>
    <td width="350"><input type="text" name="txthoten" > </td> </tr>
    <tr>
    <td >Giới Tính :</td>
    <td ><select name="gt" size="1">
    <option value="0">Nam
    <option value="1">Nữ
    </select></td>
    </tr>
    <tr>
    <td>Ngày sinh :(<td >Ngày:<select name="ngay">
    <option value=1>01<option value="2>02<option value=3>03
    <option value=4>04<option value=5>05<option value=6>06

```

```

<option value=7>07<option value=8>08<option value=9>09
. . . . .
<option value=31>31
</select>
Tháng:<select name="thang">
<option value=1>01<option value=2>02<option value=3>03
<option value=4>04<option value=5>05<option value=6>06
. . . .
</select>
Năm:<input type=text size="3" name="nam"> </td>
</tr>
<tr>
<td>Email (*)</td>
<td><input name="em" type="text" maxlength="40" size=20></td> </tr>
<tr>
<td>Tên Đăng Nhập :(<td><input type=text name=tdn size=20></td>
</tr>
<tr>
<td>Mật khẩu :(<td><input type=password name=mk size=20 ></td> </tr>
<tr>
<td>Nhập lại Mật khẩu :(<td><input type=password name=nlmk size=20 ></td> </tr>
<tr>
<td></td>
<td><input type=submit value="Đăng ký"><input type=reset value="Bỏ
qua"></td> </tr>
</table></form>
</body>
</html>

```

BÀI 5: THIẾT KẾ VÀ QUẢN TRỊ WEBSITE VỚI CÔNG CỤ

Mã Bài: MD16_05

Giới thiệu:

Khi xây dựng bất kỳ website nào, bước đầu tiên là xác định rõ mục đích của mình. Không có mục đích và nhiệm vụ rõ ràng, các dự định sẽ khó có thể đạt được kết quả như mong muốn. Lập kế hoạch cẩn thận kết hợp với việc đề ra mục tiêu rõ ràng là chìa khóa để xây dựng web site thành công.

Mục tiêu:

- Tổ chức được một website một cách hợp lý;
- Sử dụng được một hoặc một số công cụ như phần mềm KompoZer, DreamWeaver,... để thiết kế và quản lý website;
- Cài đặt được website lên máy chủ web;
- Rèn luyện tính nghiêm túc, tích cực trong việc tiếp nhận kiến thức. Chủ động, tích cực tìm kiếm các nguồn tài liệu liên quan.

Nội dung chính:

1. Tổ chức và quản lý website

1.1. Phân tích cấu trúc website

Sau khi đã xác định được mục tiêu của mình, tìm hiểu về khán giả, tổ chức trang và thiết kế khung trang chủ, một thành phần cuối cùng của bản kế hoạch còn lại là kiểm tra các công cụ và vật liệu để tiến hành xây dựng web. Giai đoạn này bao gồm cả việc chúng ta đã có các văn bản đã soạn thảo và chỉnh sửa, các file đồ họa đã thay đổi kích cỡ phù hợp, các bức ảnh số hoặc được quét, và các công cụ để sắp xếp tất cả các mục trên trang web. Như vậy chúng ta thấy các tác vụ trong giai đoạn này rất cụ thể và đơn giản nhưng để hoàn thành nhìn chung mất nhiều thời gian vì vậy chúng ta phải có kế hoạch cụ thể. Về cơ bản trước khi tạo trang web chúng ta cần phải liệt kê tất cả các công cụ như liệt kê danh sách kiểm tra công cụ dưới đây:

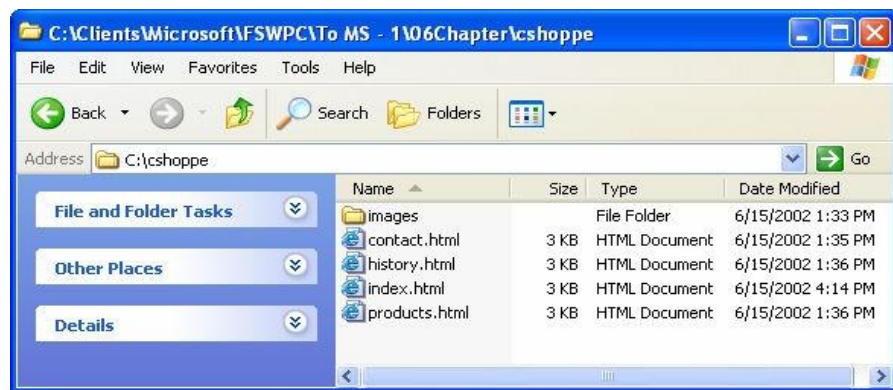
Danh sách kiểm tra công cụ:

- Văn bản đã sửa lỗi kiểm tra chính tả và nội dung
- Ảnh, đồ họa và các minh họa (bao gồm các nút, thanh tiêu đề, logo chất lượng cao)
- Khung phác thảo và khuôn mẫu các trang

- Trình soạn thảo HTML, trình soạn thảo văn bản, hoặc các công cụ tạo trang web như frontpage, Dreamwave...
- Chương trình đồ họa như Photoshop, Corel...
- Tên miền (hoặc mua, hoặc được phân bổ)
- Không gian máy chủ

1.2. Tạo hệ thống thư mục

Như chúng ta đã biết trang web thường bao gồm một số ít tập tin kết hợp với nhau để tạo ra diện mạo cho một trang. Hơn nữa một website bao gồm nhiều trang web. Do đó tính chất của các trang web và website là gồm nhiều tập tin nên chúng ta phải lập kế hoạch cho việc đặt tên và tổ chức các tập tin. Đối với phần lớn các trường hợp, một website tiêu chuẩn có thể gồm cấu trúc đơn giản như cho thấy hình bên dưới:



Hình 5.1: Tổ chức file đơn giản của một website

Thư mục chính: chứa các tập tin html, danh mục ảnh. Chúng ta có thể đặt tên bất kỳ cho danh mục này khi thiết kế các trang trên máy tính. Khi tải các trang này lên dịch vụ máy chủ, chúng ta sẽ đặt nội dung của các thư mục chứa danh mục chính vào một thư mục trực tuyến có tên là web và sao lưu toàn bộ thư mục ảnh, âm thanh, video vào thư mục web.

Thư mục phụ: chứa các tập tin ảnh gif và jpg được sử dụng trong trang web. Thư mục này thường được lưu trữ trong thư mục chính.

Trong hình trên thư mục chính bao gồm 4 tập tin HTML - contact.html, history.html, index.html, products.html - mỗi tập tin cho một trang chính của website. Chúng ta phải nhớ rằng mỗi tập tin HTML là tên tập tin xuất hiện trên URL của trang web. Ví dụ muốn ghé thăm trang sản phẩm của website Curiosity Shop, chúng ta sẽ gõ vào <http://creativeonguide.com/cshoppe/products.html>. Như vậy có thể thấy, địa chỉ URL này bao gồm tên miền (www.creationguide.com), tên thư mục chứa tập tin (cshoppe), và tên tập tin products.html.

Đối với hầu hết các trang chủ và trang con, chúng ta thấy rằng không cần gõ vào tên tập tin. Nếu không có tên tập tin HTML được chỉ ra sau tên miền hay thư mục, hầu hết các máy chủ theo mặc định sẽ hiển thị một tên tập tin đặt biệt – đa số là `index.html`, hoặc một số máy chủ hỗ trợ một số tên đặt biệt như `index.htm`, `default.html`. Hãy hỏi dịch vụ cung cấp cho biết chúng ta nên dùng tên gì cho trang chủ của mình, thông thường nhất vẫn là `index.html`.

Bởi vì tên tập tin HTML xuất hiện trong URL nên chúng ta cần tuân theo một số quy tắc cơ bản sau khi đặt tên cho tập tin:

- Tên tập tin phải đơn giản, ngắn và có ý nghĩa. Người sử dụng có thể muốn truy cập trực tiếp đến các trang phụ, vì vậy nên tạo ra các tên URL dễ nhớ và dễ gõ.
- Tránh dùng các ký hiệu và các dấu chấm câu. Hầu hết mọi người đều thấy rằng nếu dùng ký hiệu và dấu chấm câu sẽ làm chậm đi tốc độ gõ và khả năng sai sót cao.
- Sử dụng dấu gạch dưới để chỉ thị một khoảng trống trong tập tin.
- Sử dụng hoàn toàn các chữ thường.

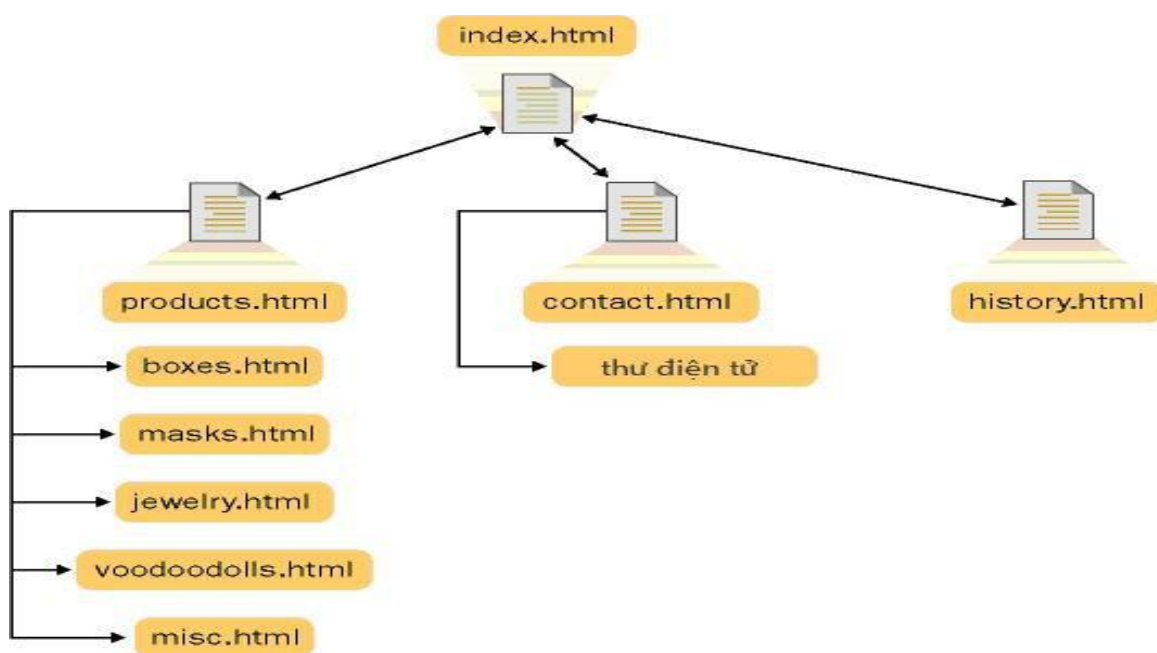
Đặt tên cho hình ảnh

Bên cạnh việc đặt tên cho các tập tin HTML, chúng ta cũng cần đặt tên cho các tập tin ảnh. Nói chung người sử dụng không truy cập trực tiếp đến các tập tin ảnh mà các trang HTML tham chiếu đến các tập tin ảnh bất kỳ lúc nào mà chúng cần đến. Vì vậy sẽ khó khăn hơn khi đặt tên cho các tập tin ảnh. Cách đặt tên ảnh logic và khoa học sẽ giúp chúng ta thuận tiện hơn trong việc tìm kiếm và truy xuất đến chúng. Cụ thể chúng ta nên đặt trước các tên tập tin ảnh chữ `p_`, `b_`, `t_`. Trong đó `p_` là tên của một bức tranh, ví dụ `p_mask` chỉ ra bức tranh về cái mặt nạ. Hình ảnh `b_` là nút bấm, ví dụ `b_products` đó là bức tranh về nút bấm chuyển hướng products. Và hình ảnh `t_` ý nói là thanh tiêu đề, ví dụ `t_contacts` là hình vẽ thanh tiêu đề được sử dụng trong trang contacts.

1.3. Quản lý website

Đến bây giờ chúng ta thấy rằng việc tổ chức file và các đối tượng đóng vai trò quan trọng trong lập kế hoạch cho website. Vì vậy chúng ta cần phải xem cần phải hệ thống hoá các tập tin như thế nào, bằng cách nào cho logic nhất.

Chúng ta nên tạo hồ sơ cấu trúc trang có một bố cục rõ ràng bao gồm tất cả các thông tin dưới một định dạng có thể truy cập dễ dàng. Ví dụ xem hình dưới:



Hình 5.2: Một trong các hồ sơ ban đầu về cấu trúc và cách tổ chức file của một trang web2. Thiết kế và quản trị website với công cụ (KomboZer, DreamWeaver, ...)

2.1. Làm quen với công cụ

2.1.1. Giao diện làm việc

Dreamweaver là một chương trình Visual Editor chuyên nghiệp để tạo và quản lý các trang web. Dreamweaver cung cấp các công cụ phác thảo trang web cao cấp, hỗ trợ các tính năng **DHTML** (Dynamic HTML) mà không cần viết các dòng lệnh giúp các bạn không biết lập trình web cũng có thể thiết kế được các trang web động một cách dễ dàng, trực quan. Với Dreamweaver bạn có thể dễ dàng nhúng các sản phẩm của các chương trình thiết kế web khác như Flash, Fireworks, Shockwave, Generator, Authorwave vv...

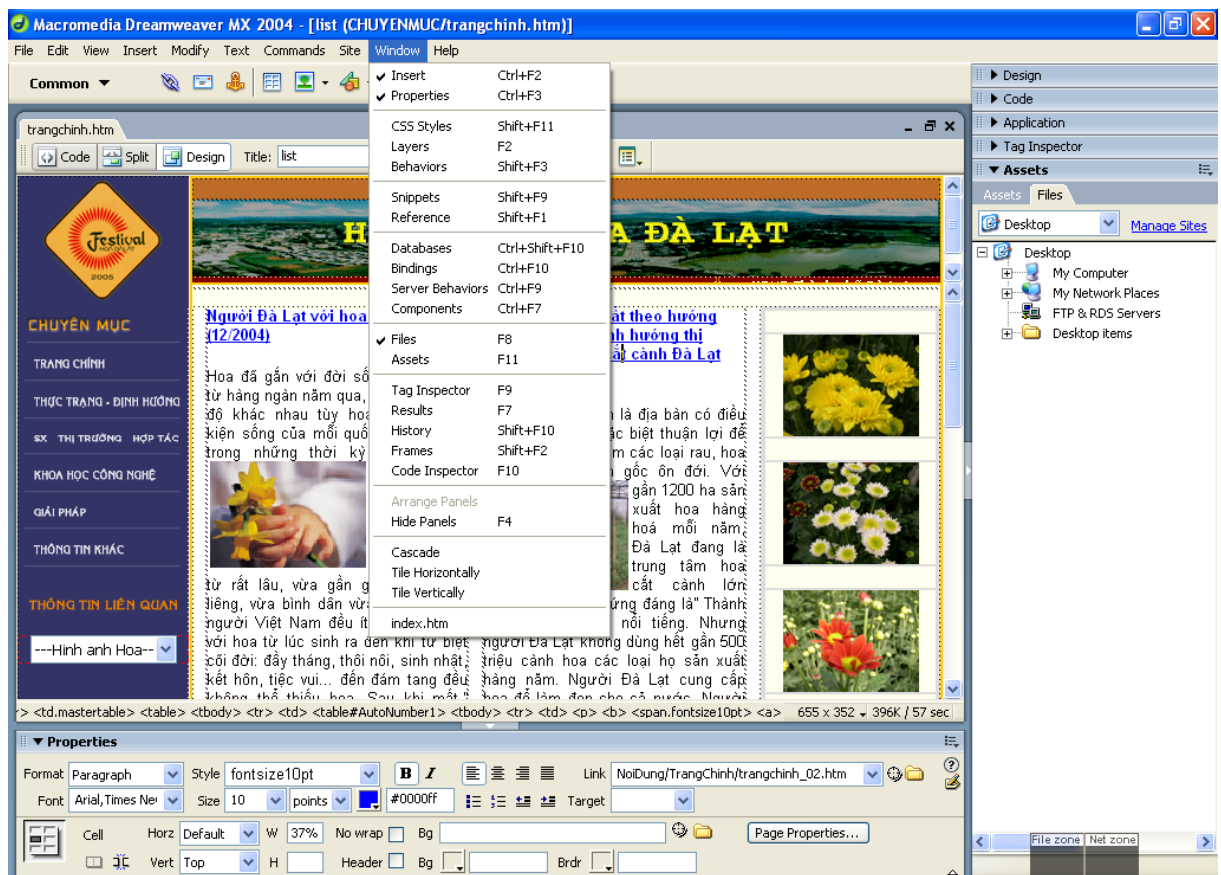
Với Dreamweaver bạn có thể quản lý các Local và Remote site giúp cho việc quản lý các trang web trong các site cục bộ và các website điều khiển từ xa có thể đồng bộ.

Ngoài ra Dreamweaver còn cho phép bạn chỉnh sửa trực tiếp HTML. Với Code View bạn có thể nhanh chóng bổ sung hoặc xóa bỏ một HTML mà không cần thoát khỏi cửa sổ tài liệu. Chế độ soạn thảo trang web bằng HTML giúp các bạn có thể thiết kế trang trực tiếp bằng ngôn ngữ HTML.

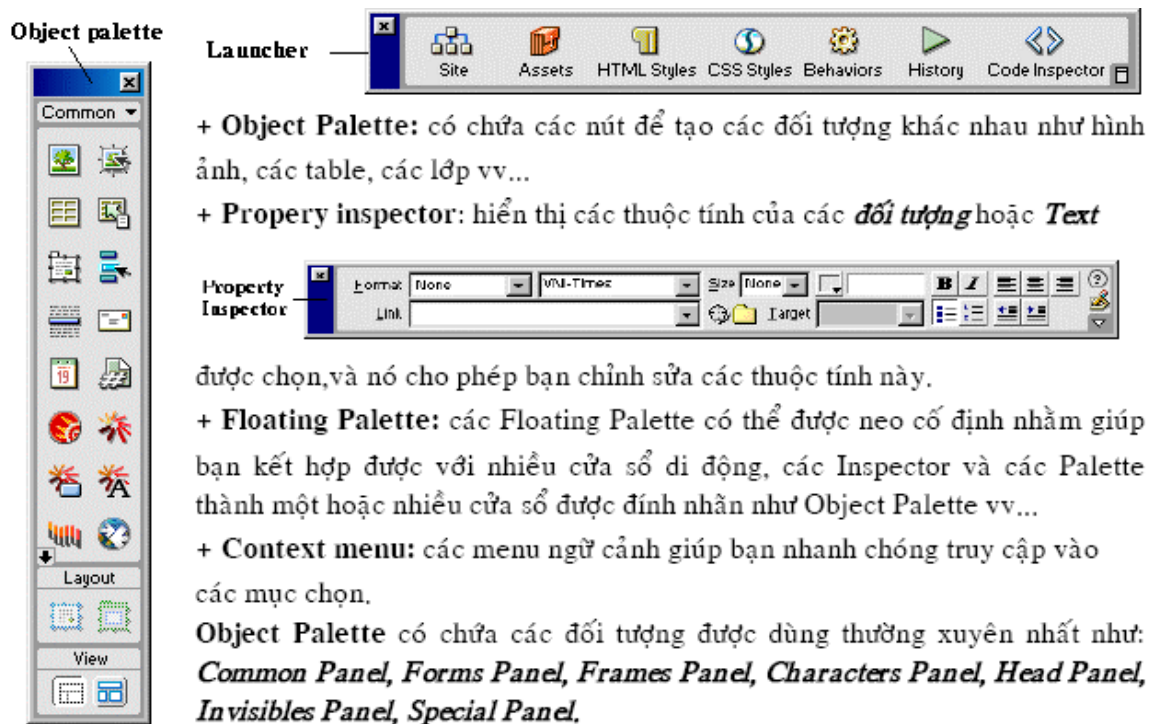
Dreamweaver còn hỗ trợ các **HTML Styles** và **Cascading Style Sheet** giúp bạn định dạng trang web nhằm tăng tính hấp dẫn khi duyệt các trang web này.

Vùng làm việc của chương trình Deamweaver rất linh động và dễ sử dụng gồm các thành phần như sau: Document, Launcher, Object Palette, Property Inspector,

Floating Palette và các context menu (menu ngữ cảnh) ... để mở các thành phần này bạn có thể vào menu **Window-> chọn tên của thành phần cần hiển thị.**



+ **Launcher:** có chứa các nút để mở và đóng các Inspector và các Palette được dùng thường xuyên nhất. Bạn cũng có thể truy cập Launcher tại **Mini Launcher** tại góc phải của cửa sổ Document.



2.1.2. Sửa thông số làm việc

Sử dụng các tham chiếu **Fonts/Encoding** để cài đặt việc tạo mã font mặc định dành cho các trang tài liệu mới và cài đặt các font mà Dreamweaver dùng để hiển thị mỗi một chế độ tạo mã font. Việc tạo mã cho tài liệu (trang web) sẽ xác định cách mà trang web được hiển thị trong trình duyệt.

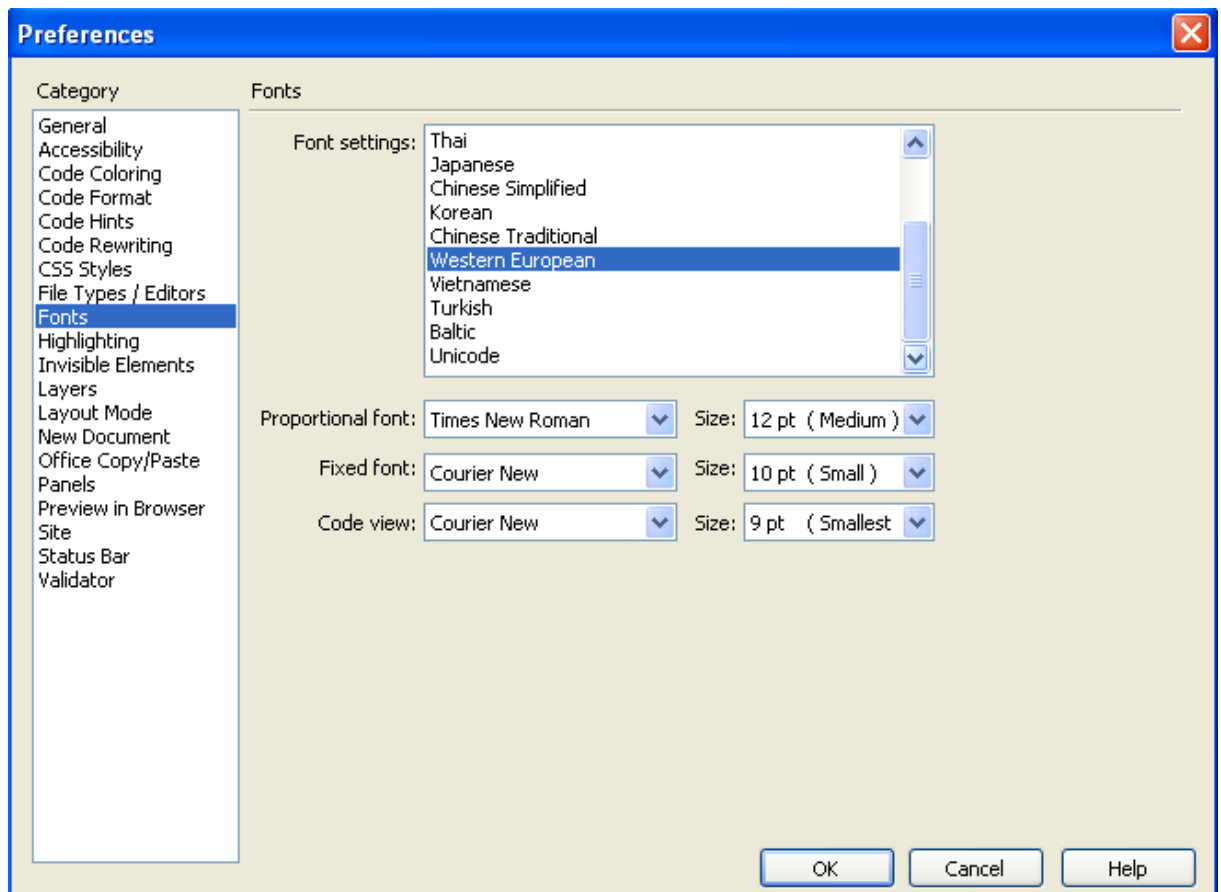
Chọn Edit->Preferences, chọn một chế độ tạo mã từ select box Default Encoding.

Chế độ tạo mã mặc định được lưu cùng lúc với tài liệu trong thẻ <META> được chèn ở đầu tài liệu. Ví dụ nếu bạn chỉ định Western European thì thẻ <meta> phát sinh bên source code ngôn ngữ HTML là :

`<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">`

Để chỉ định các font dùng cho US English và tất cả các thứ tiếng Tây Âu hãy chọn **Western European**.

Để cài đặt font cho các thứ tiếng Việt hãy chọn **Vietnamese**.

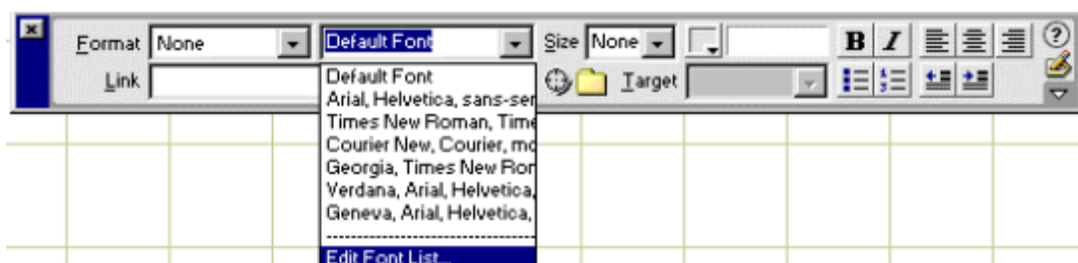


Hãy chọn kiểu tạo mã chẳng hạn như Western European hoặc Vietnamese từ Fonts.

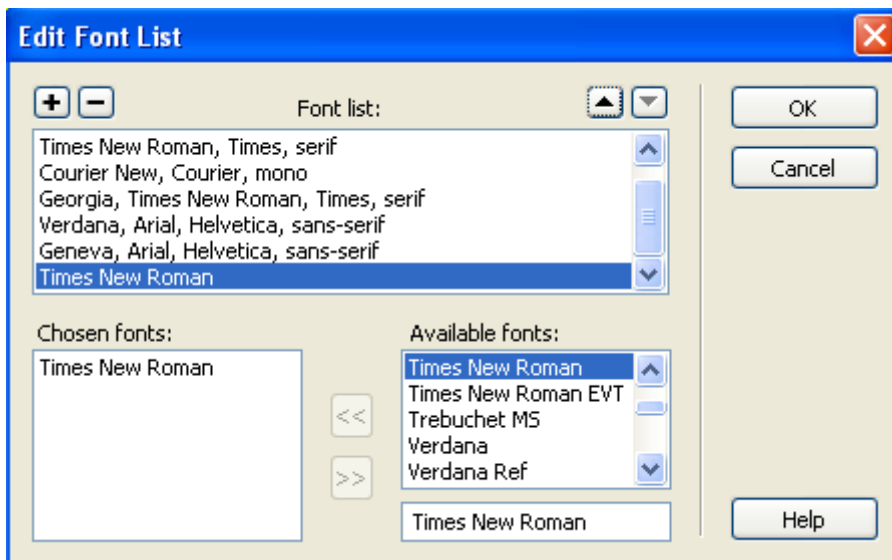
Tại Proportional Font chọn font và kích thước hiển thị mặc nhiên khi soạn thảo trang web (ví dụ: Times New Roman, size=12pt).

+ Fixed Font là font mà Dreamweaver dùng nhằm hiển thị văn bản bên trong các thẻ <PRE>, <CODE>, <TT>

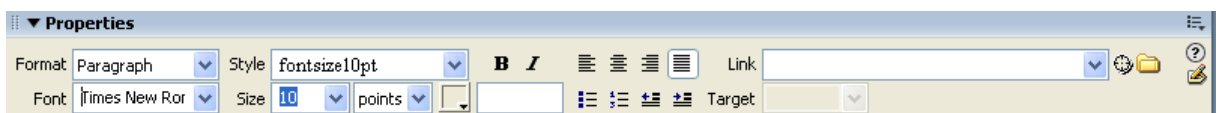
+ Code View là font được dùng cho tất cả các text (văn bản) bên trong HTML



+ Tại hộp thoại Property Inspector tại select box Default font hãy chọn Edit font list hoặc chọn menu Text->Font->Edit Font list sẽ xuất hiện hộp thoại sau:



+ Tại Available Fonts hãy chọn fonts thường dùng (ví dụ font Times New Roman) sau đó Click nút <<. Fonts được chọn sẽ hiện trong hộp Chosen Fonts và sẽ hiện trong hộp danh sách các Font List thường dùng. Lưu ý, hiện nay các Font Unicode được làm chuẩn cho Internet, tại select box chọn font trong hộp thuộc tính Property Inspector thì mới phát sinh được thuộc tính `face="Times New Roman"` ... > thì mới hiện được font tiếng việt khi duyệt trang web trong trình duyệt.

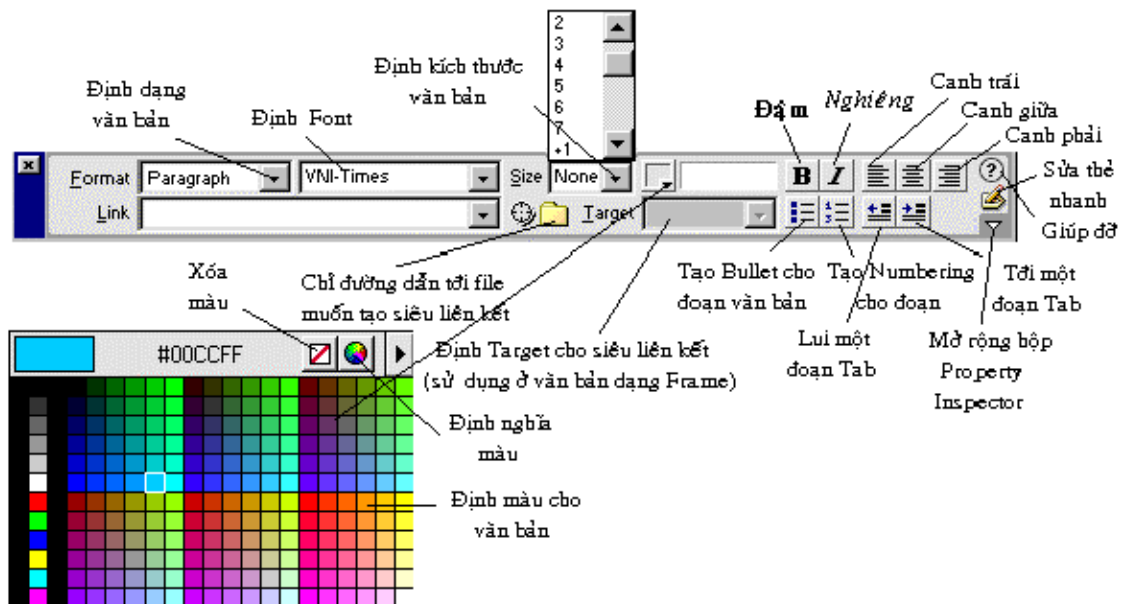


Lưu ý: Bạn nên Edit Font List mỗi lần một font và bạn thực hiện việc Edit Font list này cho các font bạn thường dùng.

2.2. Thiết kế trang web

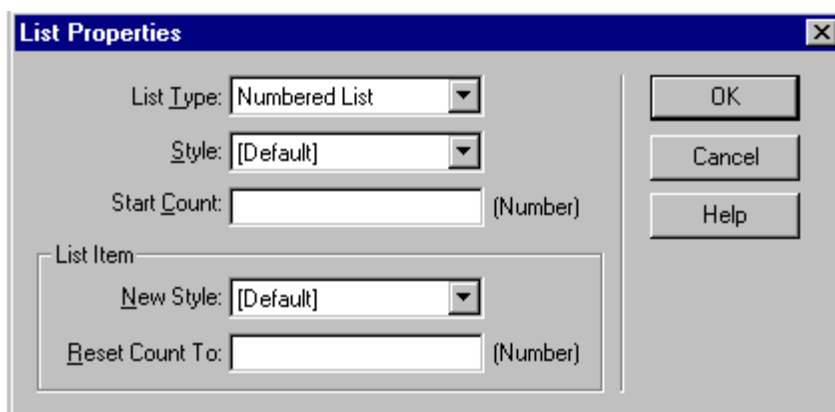
2.2.1. Định dạng văn bản

Bạn có thể định dạng text trong cửa sổ tài liệu Document bằng cách cài đặt các thuộc tính trong hộp Property Inspector. Để định dạng văn bản trước tiên hãy bôi đen văn bản muốn định dạng rồi sau đó chọn các thay đổi trên hộp thuộc tính Property Inspector. Nếu hộp Property Inspector không mở hãy chọn menu Window->Properties



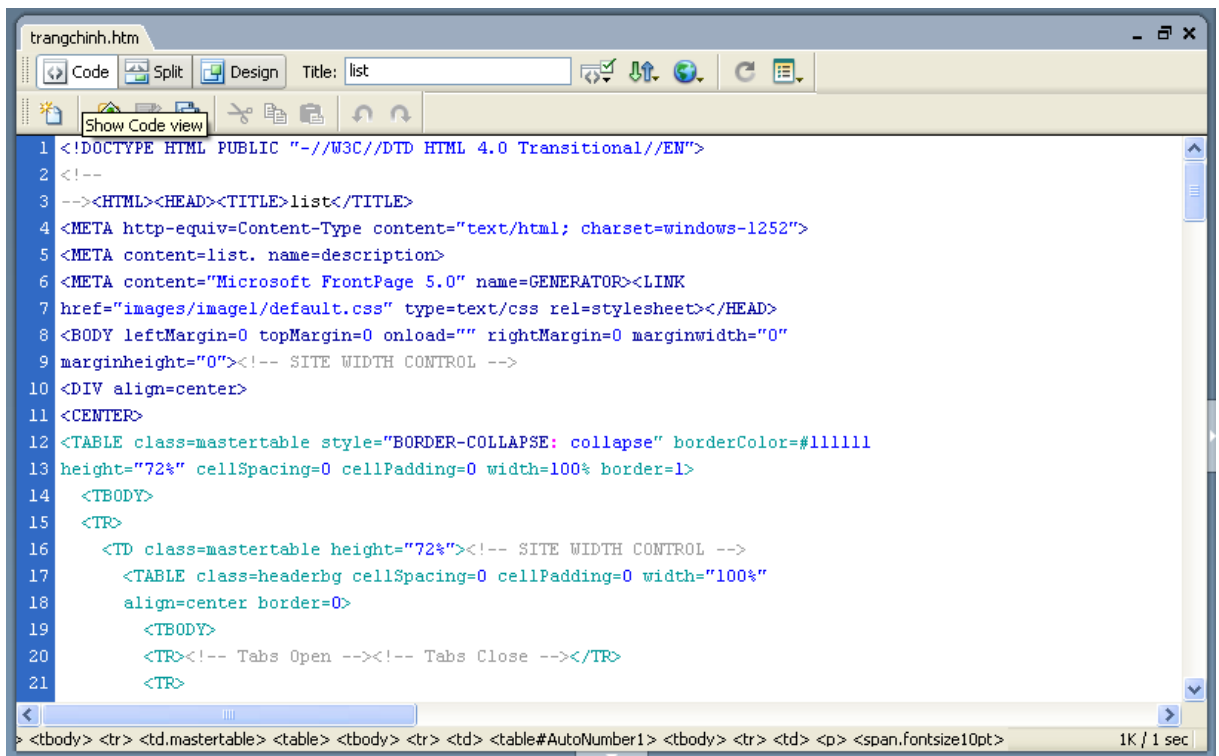
Lưu ý: Trong lúc định dạng văn bản bạn có thể sử dụng hộp Property Inspector hoặc bạn có thể sử dụng menu Text để định dạng văn bản vì có những chức năng mà hộp thuộc tính Property Inspector không xây dựng sẵn cho bạn (ví dụ như văn bản được gạch dưới, gạch ngang hoặc bạn có thể sử dụng chế độ thiết kế web bằng ngôn ngữ HTML để thiết kế thêm những thuộc tính cần thiết (ví dụ như bạn muốn văn bản được canh đều `<p align="justify" ...>`

Nếu bạn đang thao tác với các Bullets hay Numbering bạn có thể Click nút List Item trên thanh Property Inspector để cài đặt một số thông số sau:



+ Chọn kiểu List Type là Bullet hay Numbering, chọn kiểu Style. Nếu bạn đang thao tác với numbering bạn có thể cho số tăng từ mấy tại Start Count. Tại Reset Count to bạn có thể nhập vào giá trị số trở về mặc nhiên (thường là 1).

Để hiện chế độ thiết kế trang web ở ngôn ngữ HTML bạn có thể click nút `<>` Show Code View.



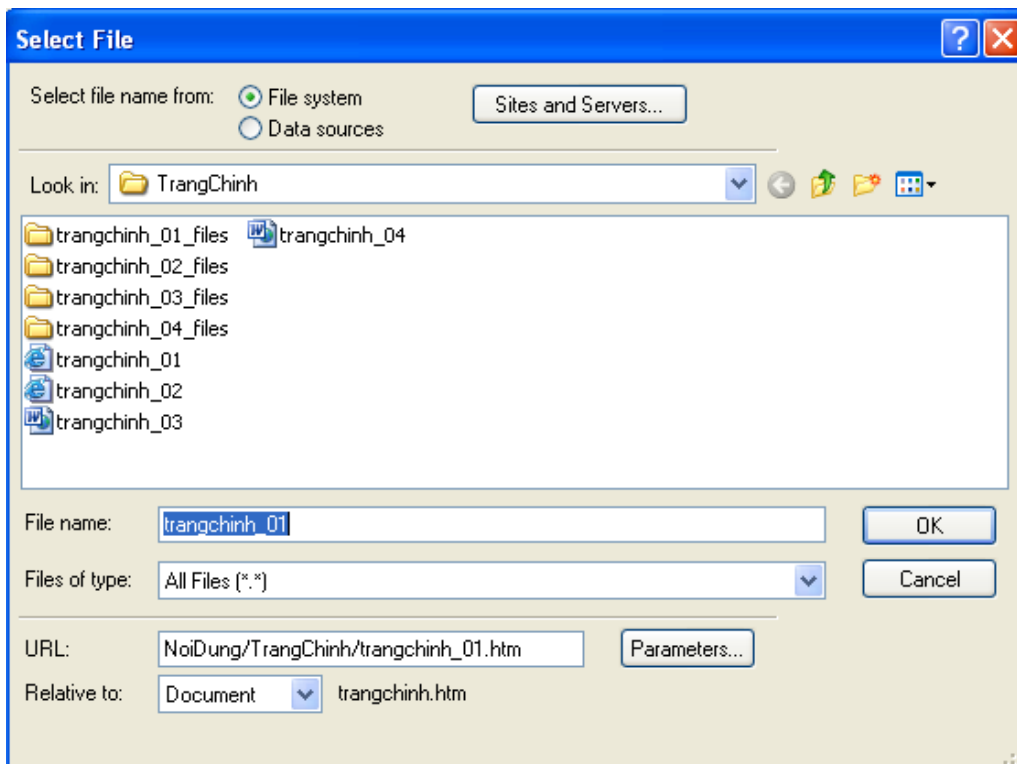
```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2 <!--
3 --><HTML><HEAD><TITLE>list</TITLE>
4 <META http-equiv=Content-Type content="text/html; charset=windows-1252">
5 <META content=list. name=description>
6 <META content="Microsoft FrontPage 5.0" name=GENERATOR><LINK
7 href="images/imagel/default.css" type=text/css rel=stylesheet></HEAD>
8 <BODY leftMargin=0 topMargin=0 onload="" rightMargin=0 marginwidth="0"
9 marginheight="0"><!-- SITE WIDTH CONTROL -->
10 <DIV align=center>
11 <CENTER>
12 <TABLE class=mastertable style="BORDER-COLLAPSE: collapse" borderColor=#111111
13 height="72%" cellSpacing=0 cellPadding=0 width=100% border=1>
14 <TBODY>
15 <TR>
16 <TD class=mastertable height="72%"><!-- SITE WIDTH CONTROL -->
17 <TABLE class=headerbg cellSpacing=0 cellPadding=0 width="100%"
18 align=center border=0>
19 <TBODY>
20 <TR><!-- Tabs Open --><!-- Tabs Close --></TR>
21 <TR>
```

2.2.2. Tạo siêu liên kết (Hyperlink) cho văn bản

Với Dreamweaver bạn có nhiều cách để tạo siêu liên kết đến các trang web khác hoặc các tập tin tài liệu, hình ảnh, âm thanh (Sound), Phim(Video) hoặc các chương trình có thể tải xuống (Download).

Thường có các loại siêu liên kết sau:

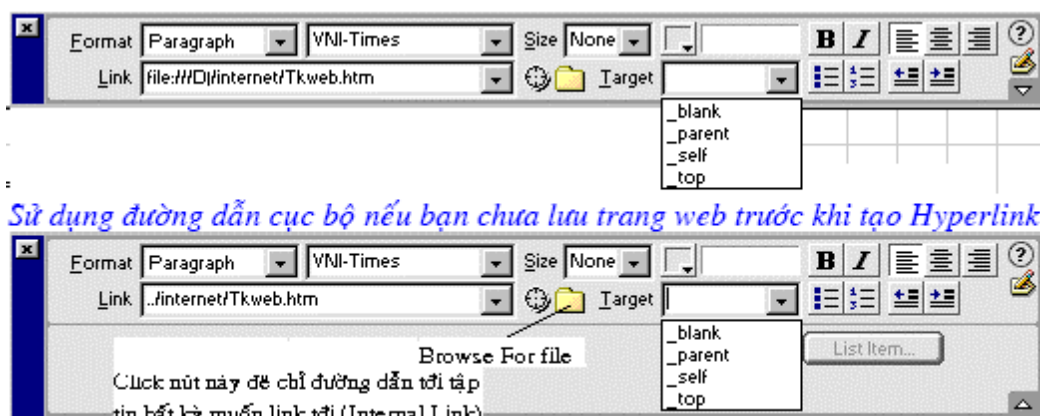
- 1) Internal Link :Liên kết đến các tài liệu trong cùng Website (Local Site) .
- 2) External Link :Liên kết các tài liệu bên ngoài Local website (ví dụ như link tới các WebServer khác trên Internet, FTP Server khác trên Internet, các Telnet Server, các News Server vv...)
- 3) E-mail Link :tạo một siêu liên kết cho phép người sử dụng gửi Email.
- 4) Link to Name Anchors (còn có thể gọi là Bookmark) :Tạo siêu liên kết nhảy đến một vị trí nào đó trong trang web hiện hành.



Nguyên tắc chung khi bạn tạo một siêu liên kết cho văn bản(text), hình ảnh(image), Hotpot của bản đồ hình ảnh(hotpot of Image map): hãy chọn văn bản hoặc hình ảnh hoặc Hotpot muốn tạo siêu liên kết sau đó click nút Browse For file (tạo Internal Link) trên hộp thuộc tính Property Inspector của đối tượng tương ứng đang chọn.

Khi bạn chỉ đường dẫn tới trang web bạn sẽ thấy địa chỉ của trang web này hiện tại địa chỉ URL như hình vẽ.

Tại hộp link của hộp Property Inspector xuất hiện đường dẫn của tập tin bạn đang liên kết tới theo kiểu cục bộ(trên đĩa cứng) nhưng nếu bạn đã save lại có tên bạn sẽ có kiểu đường dẫn là tương đối. Bạn nên lưu trang web lại trước khi tạo siêu liên kết để không hiện thông báo nhắc nhở mỗi khi tạo siêu liên kết.



Sử dụng đường dẫn kiểu tương đối

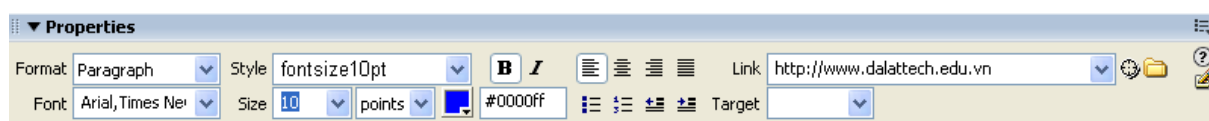
Bạn có thể định Target cho siêu liên kết (nơi hiển thị nội dung của trang web khi click vào).

Ta có các loại Target sau:

- `_blank` : nội dung trang web sẽ hiện ra ở cửa sổ mới (New Window).
- `_parent`: nội dung sẽ hiện ở trang cha
- `_self`: nội dung hiện ở trang hiện hành
- `_top`: nội dung hiện ở toàn trang (Whole page)

Chú ý: Nếu bạn tạo một External link tới một Web Server, FTP Server, News Server vv.. bạn không cần Click nút Browse For File mà hãy nhập vào địa chỉ của Server (máy chủ) mà bạn đang muốn liên kết tới.

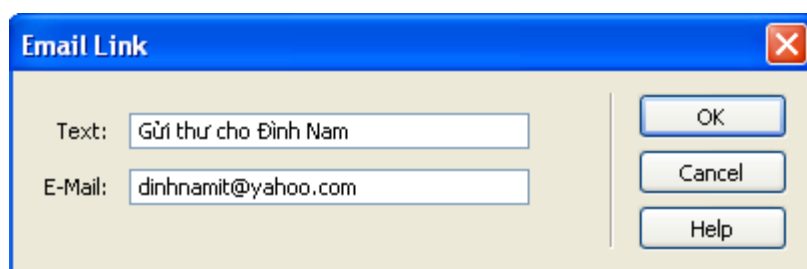
Ví dụ: <http://www.dalatttech.edu.vn/> , <http://www.dalat.gov.vn/rauhoatl>, vv...



Tạo siêu liên kết để Reply Email:

(ví dụ: dinhnamit@yahoo.com, admin@dalatttech.edu.vn)

+ Hãy Click nút Insert Email Link bạn sẽ thấy hộp thoại sau:



+ Tại hộp thoại Text hãy nhập vào chuỗi hiển thị.

+ Tại hộp E-mail: hãy nhập vào địa chỉ Email muốn gửi đến.

3. Xuất bản website

3.1. Chọn dịch vụ hosting

3.1.1. Thuê host hay là mua host?

Cái này chỉ là vấn đề nhỏ thôi nhưng việc biết nó cũng ít nhiều gì cũng có ích.

Nếu nói về host, thì bạn có thể gọi là mua host hoặc thuê host gì cũng được, nhưng chính xác nhất thì là thuê host. Sở dĩ chúng ta gọi là thuê bởi vì bạn phải trả tiền sử dụng theo hàng tháng vì chi phí cho các thứ như mạng, cơ sở hạ tầng của datacenter, tủ rack đựng máy chủ, máy chủ vật lý đều là của họ.

3.1.2 Chọn host Việt Nam hay host nước ngoài ?

Host Việt Nam là như thế nào?



Hình 5.3: Datacenter của Viettel IDC Sóng Thần – Bình Dương

Host Việt Nam không phải hoàn toàn là nhà cung cấp đó tại Việt Nam, mà một yếu tố nữa đó là các máy chủ được đặt tại một (hoặc nhiều) datacenter tại Việt Nam.

Theo mình biết thì Việt Nam ta đang có khoảng 5 datacenter trên cả nước. Trong đó 1 datacenter tại Bình Dương và 4 datacenter tại Hà Nội, cụ thể:

[IDC Viettel Datacenter](#) (Bình Dương)

[FPT Datacenter](#) (Hà Nội)

[GDS Hà Nội – Thăng Long](#)

[Hanel – CSF Datacenter](#) (Hà Nội)

[Telehouse International Vietnam](#) (Hà Nội)

Ưu điểm của Host Việt Nam

- Tốc độ truy cập trong nước rất nhanh vì sử dụng chung một hạ tầng, băng thông nội bộ và khoảng cách địa lý từ người dùng đến datacenter thấp.
- Dễ mua, dễ thanh toán.
- Dễ yêu cầu hỗ trợ.
- Không bất đồng ngôn ngữ.

Nhược điểm của Host Việt Nam

- Giá hơi cao do mặt bằng băng thông cao so với Châu Âu hay Mỹ.
- Gây cản trở khi truy cập từ nước ngoài nếu có đứt cáp.
- Một số dịch vụ Hosting hỗ trợ sau bán hàng không được tốt.

Host Nước Ngoài là như thế nào?



Bên trong một Datacenter tại Dallas, Texas. Mỹ

Ngắn gọn là các nhà cung cấp host này có trụ sở và máy chủ không đặt tại Việt Nam. Hiện nay một số nơi đặt datacenter thông dụng nhất trên thế giới cho các dịch vụ hosting là:

Dallas, Texas, Mỹ.

Amsterdam, Hà Lan.

Michigan, Mỹ.

Singapore

London, Anh.

Stockholm, Thụy Điển.

Berlin, Đức.

Đó chỉ là nơi đặt server, còn trụ sở datacenter thì hiện nay rất nhiều nhà cung cấp đều có một datacenter riêng của họ chứ ít khi thuê của ai.

Ưu điểm của host nước ngoài

- Tác phong phục vụ chuyên nghiệp.
- Nhiều chương trình khuyến mãi do độ cạnh tranh cao.
- Tốc độ truyền tải đi toàn cầu rất tốt. Ở Việt Nam nếu ping sang host nước ngoài thì chỉ khoảng từ 250 đến 300ms. Nhưng ở nước ngoài ping về VN thì ít nhất là 450ms.

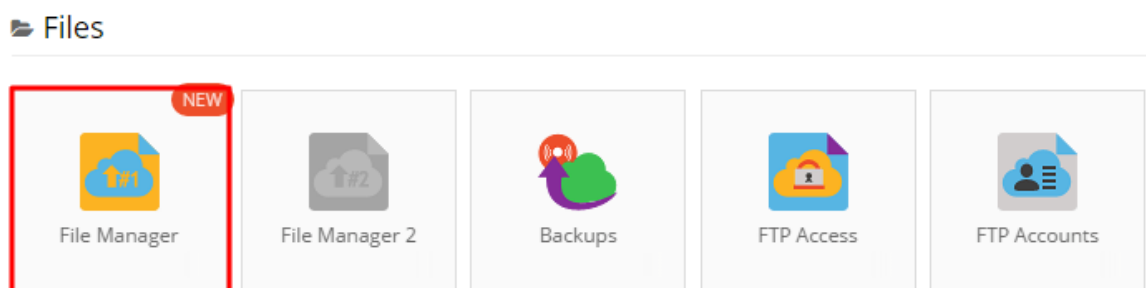
Nhược điểm của host nước ngoài

- Bất đồng ngôn ngữ.
- Khó khăn khi mua và thanh toán. Khi mua họ cần phải chứng thực qua CMND hoặc gọi điện. Và bạn chỉ có thể thanh toán qua Credit Card (Visa/Mastercard) hoặc PayPal.
- Số lượng quá nhiều khó chọn.
- Nhiều chỗ không chào đón người Việt cho lắm.

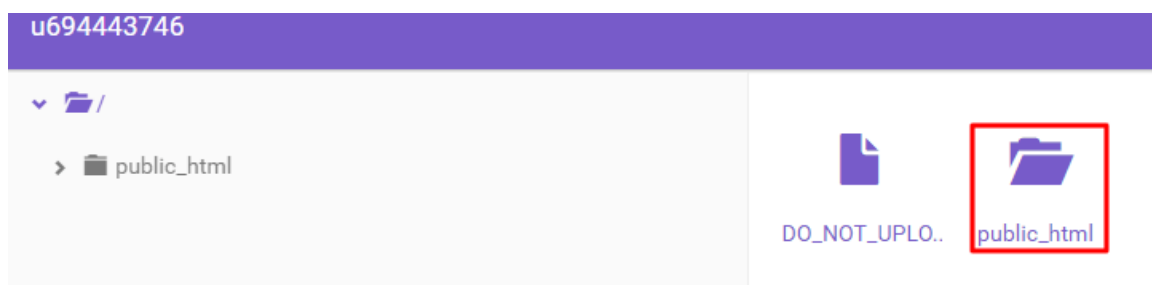
3.2. Tải website lên host

3.2.1. Cách up web lên host qua File Manager của Hostinger

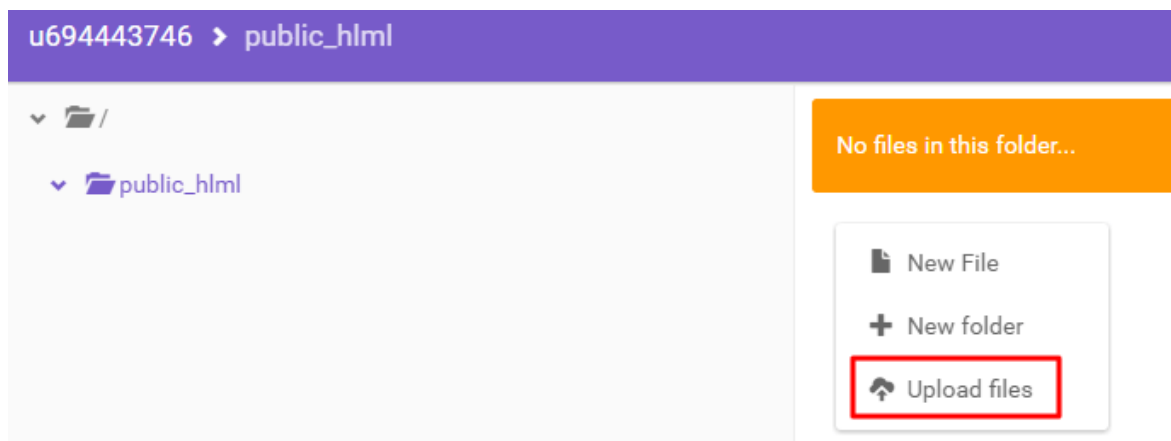
B1. Chuyển tới mục **File** và mở **File Manager**



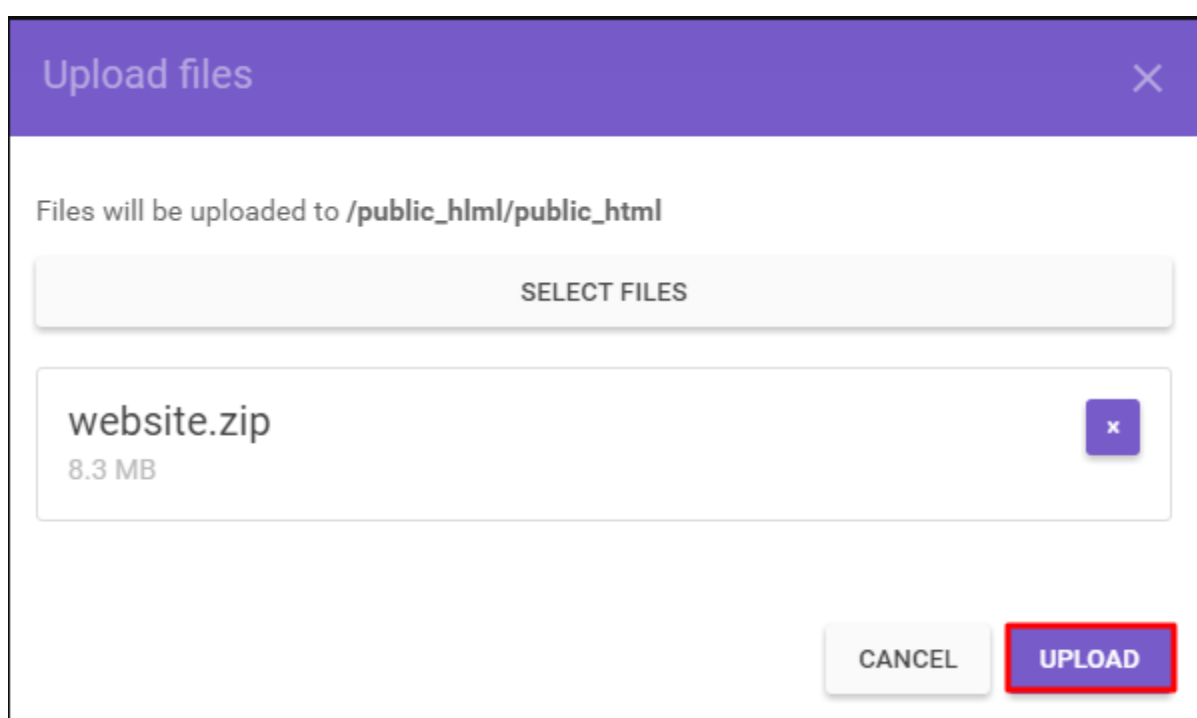
B2. Sau đó, vào thư mục **public_html**, là **root directory** của domain của bạn (ví dụ: <http://www.domainname.com/>). File của bạn sẽ được upload ở đó. Thư mục này cũng có thể thay đổi dựa trên URL nào mà bạn muốn dùng để truy cập từ. Trong trường hợp bạn muốn nội dung hiện lên tại <http://www.domainname.com/something/> nội dung phải được đặt trong thư mục **public_html/something**.



B3. Để upload, chuột phải và nhấn nút **Upload files**.



B4. Tìm backup được nén của bạn qua mục **Select Files**, nhấn **Upload** để tiến hành và chờ upload hoàn tất.

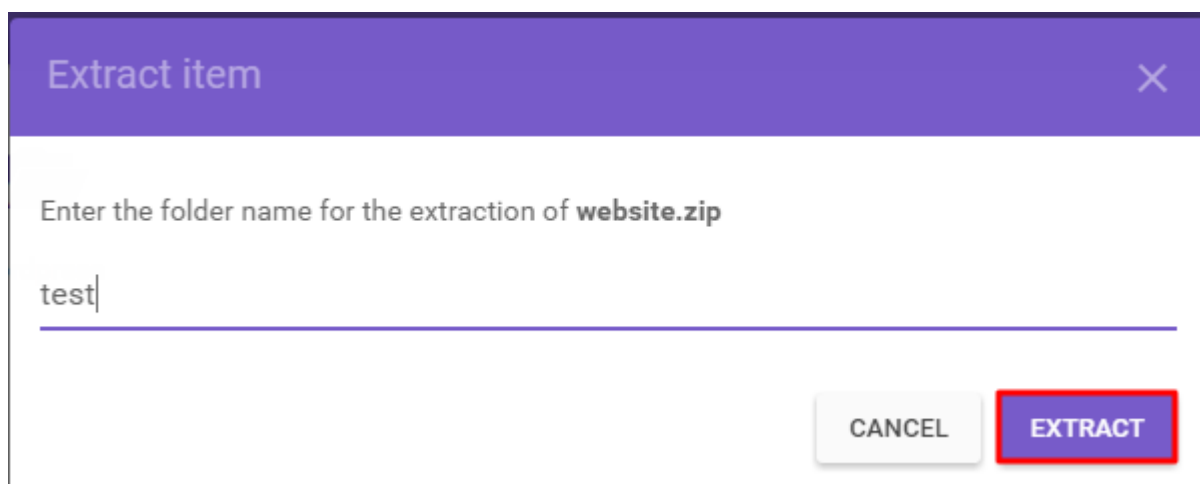


B5. Để unzip, bạn chuột phải và nhấn nút **Extract**.

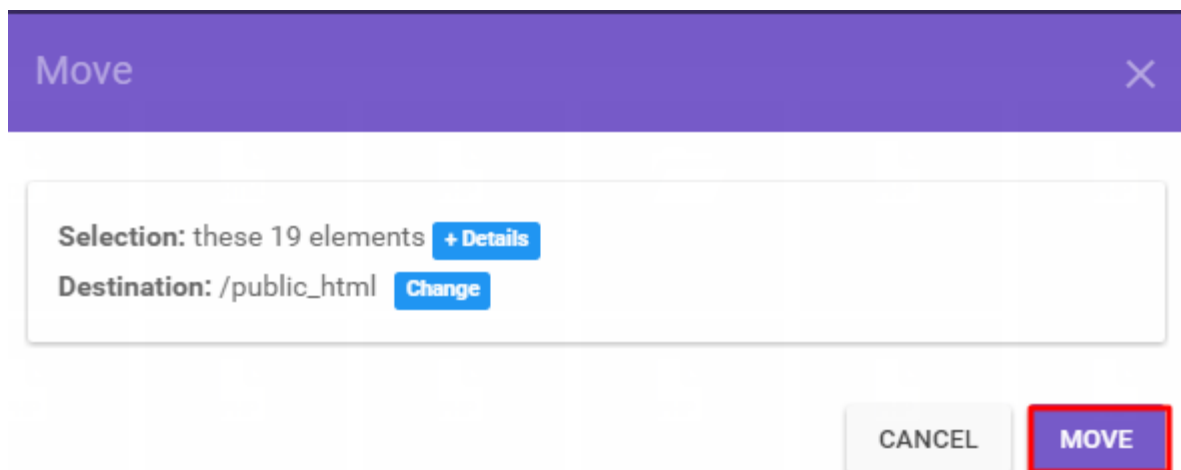


QUAN TRỌNG! Nếu các bạn muốn giải nén file lớn mà file manager không hỗ trợ, hãy xem qua bài hướng dẫn [Giải nén file lớn tại Hostinger bằng SSH](#)

B6. Bạn sẽ cần chọn thư mục được extract. Trong ví dụ này, file cài đặt sẽ được extract vào thư mục **public_html/test**.



B7. Để di chuyển files, chọn hết và chuột phải sau đó nhấn nút **Move**. Trong hướng dẫn này, chúng ta sẽ di chuyển tất cả nội dung từ **public_html/test** sang **public_html**.

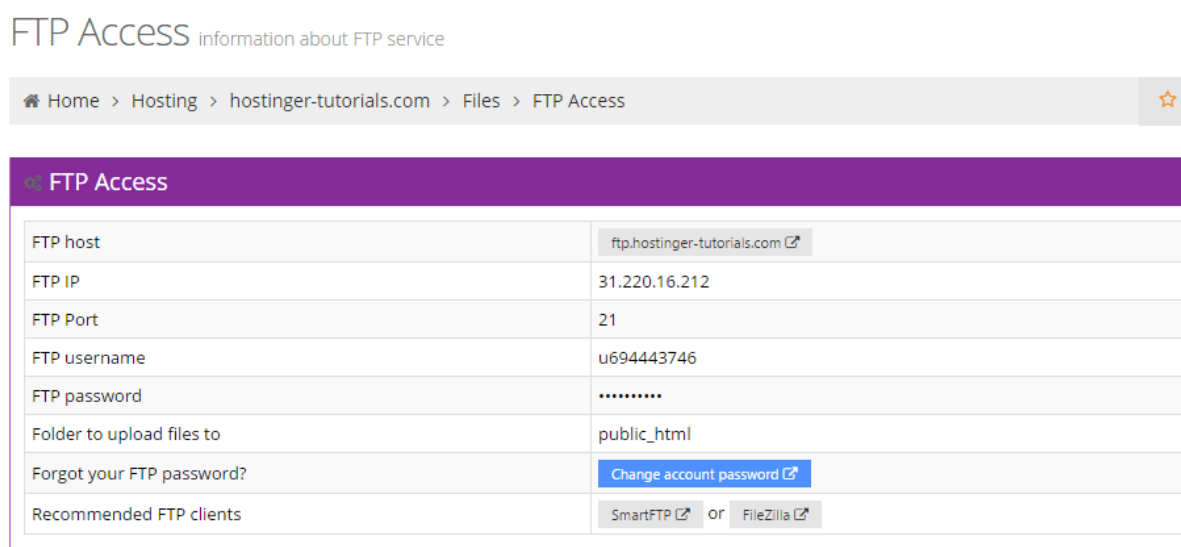


B8. Khi dữ liệu đã được chuyển, bạn sẽ có thể truy cập nó qua trình duyệt.

3.2.2. Cách up web lên host qua FTP

Một số người dùng thích up web lên host qua FTP, như [FileZilla](#), [SmartFTP](#), [CoreFTP](#), hoặc bất kỳ trình phân mềm nào khác để upload website. Trong ví dụ này, chúng tôi sẽ dùng Filezilla.

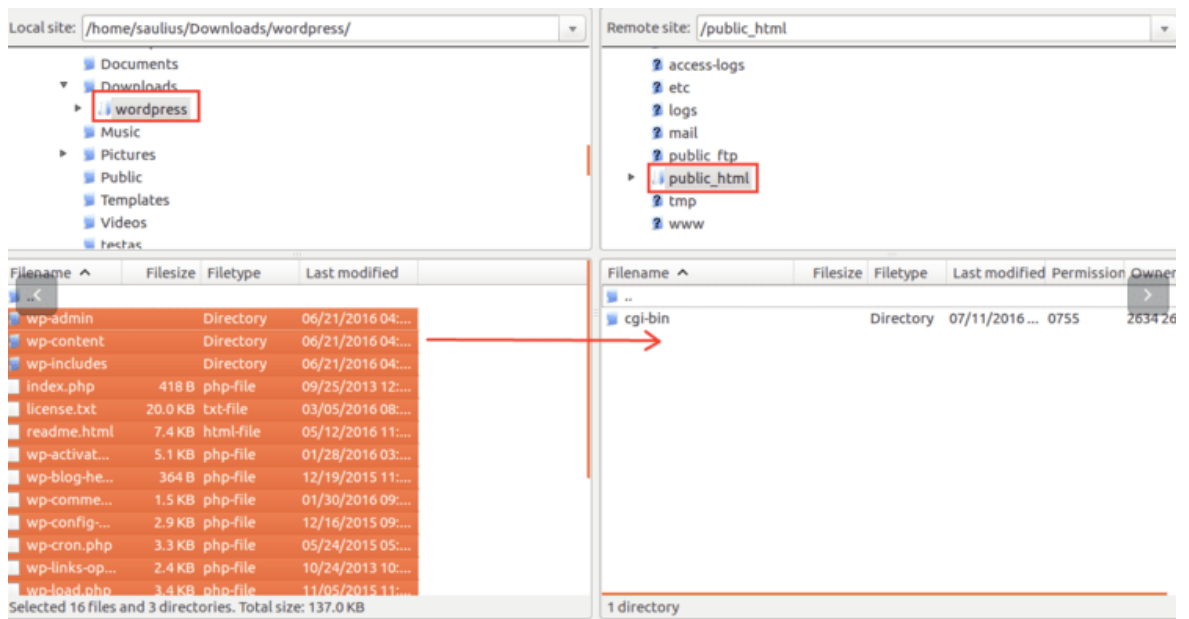
B1. Đầu tiên, bạn sẽ cần lấy thông tin FTP. Trên Hostinger, bạn có thể lấy chúng qua mục **FTP Access**. Nếu bạn không chắc password FTP là gì, bạn có thể dễ dàng đặt password mới qua mục **Change account password**.



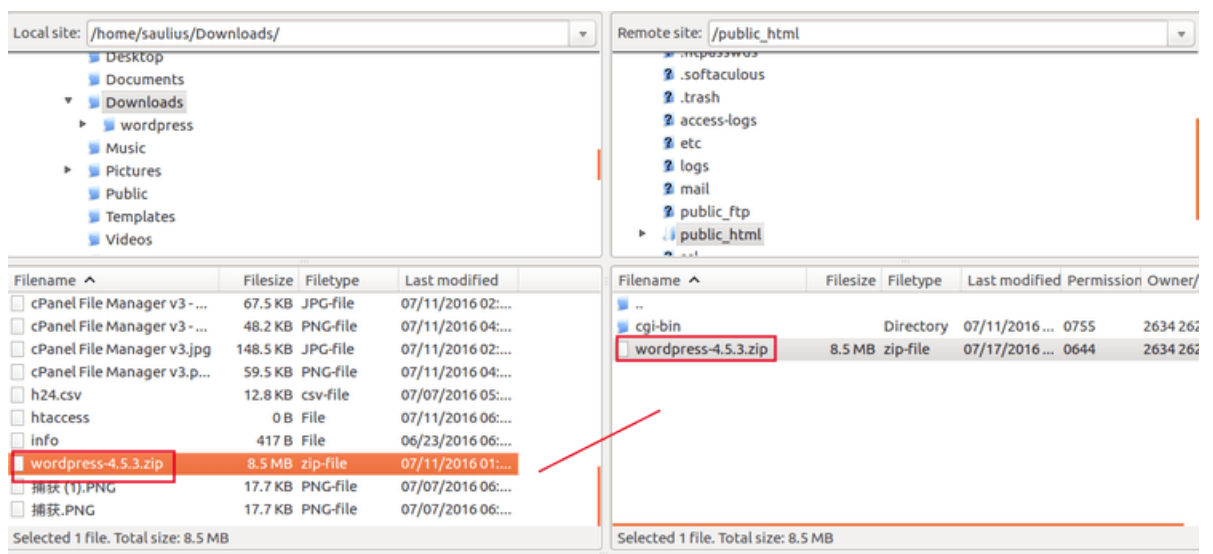
B2. Mở FileZilla và điền thông tin FTP để truy cập và nhấn nút **Quickconnect**.



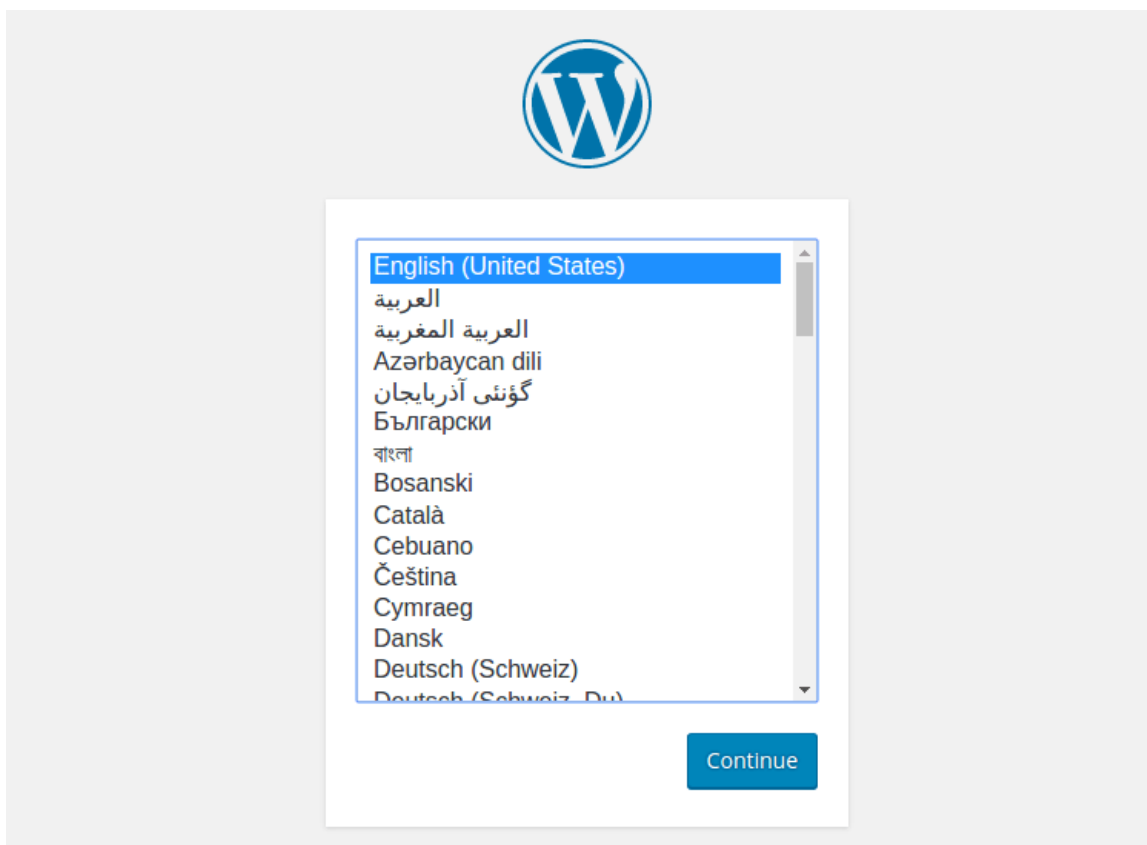
B3. Khi bạn đã kết nối, hãy tìm và kéo thả dữ liệu website từ bên **Trái** của phần mềm vào bên **Phải**, với thư mục đích là thư mục “**public_html**”.



(Tùy chọn) Tương tự, bạn có thể upload file nén qua FTP, bằng cách kéo thả từ trái qua phải. Bạn sẽ cần extract nó qua File Manager. Hãy xem lại “**Lựa chọn 1 — Upload website của bạn qua File manager**” ở phần trên để biết cách extract file nén bằng File Manager và SSH.



B4. Website của bạn có thể được truy cập qua thư mục public_html bây giờ. Hãy truy cập thử bằng trình duyệt để xem kết quả như thế nào.

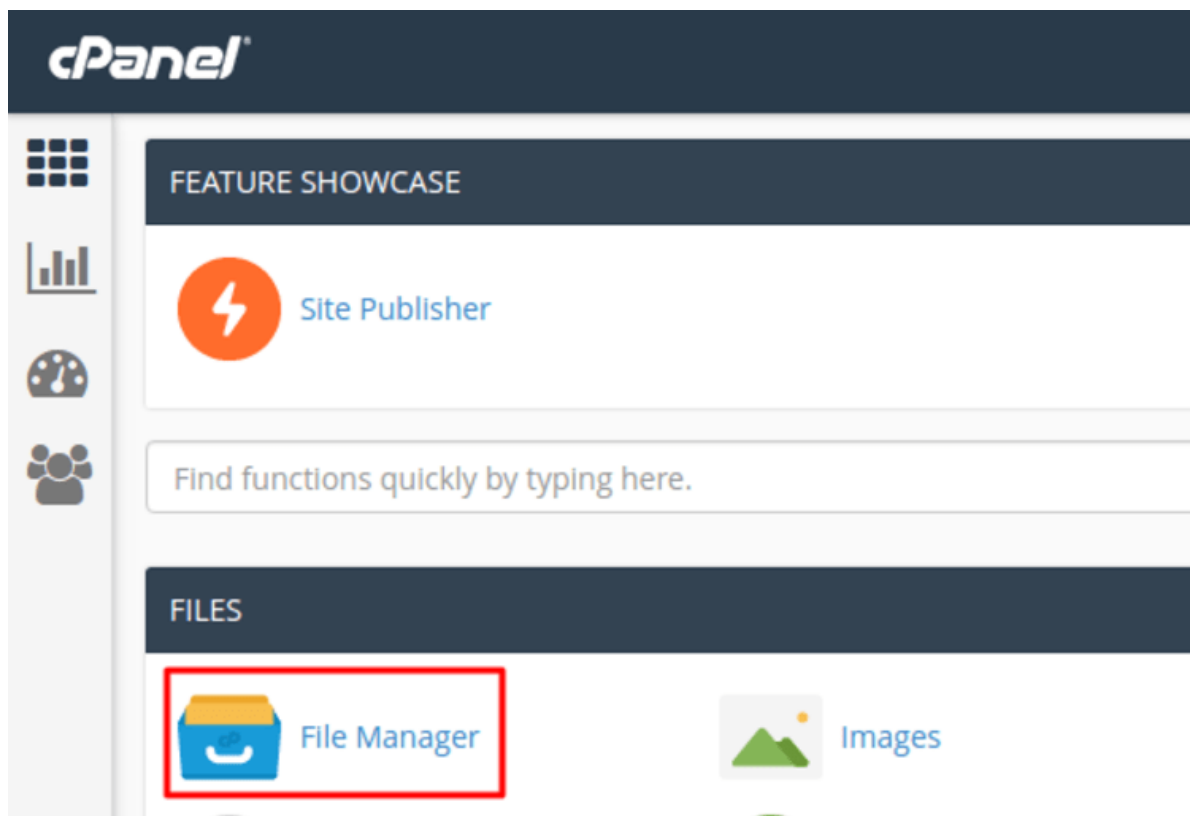


B5. Thành công! Chúng ta đã nhìn thấy trang cài đặt mặc định WordPress có nghĩa là cách up web lên host đã hoàn tất và bạn có thể tiến hành cài đặt WordPress.

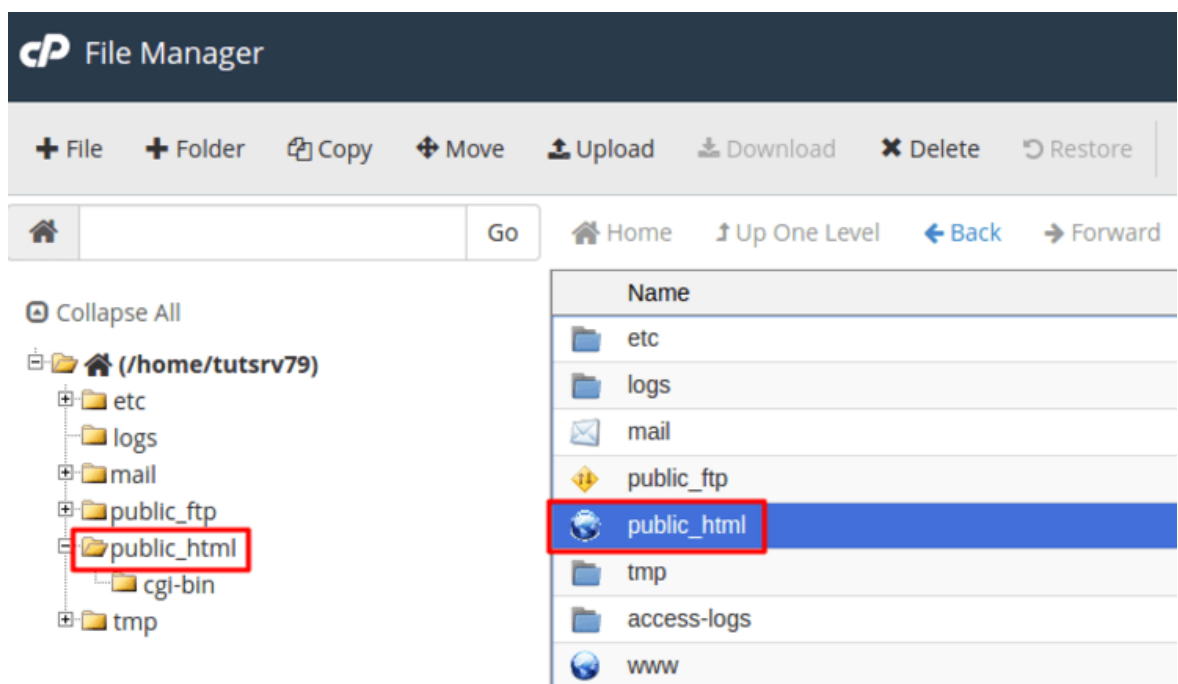
3.2.3. Cách up web lên host qua cPanel's File Manager

Truy cập vào **cPanel** của tài khoản hosting của bạn và làm theo hướng dẫn bên dưới:

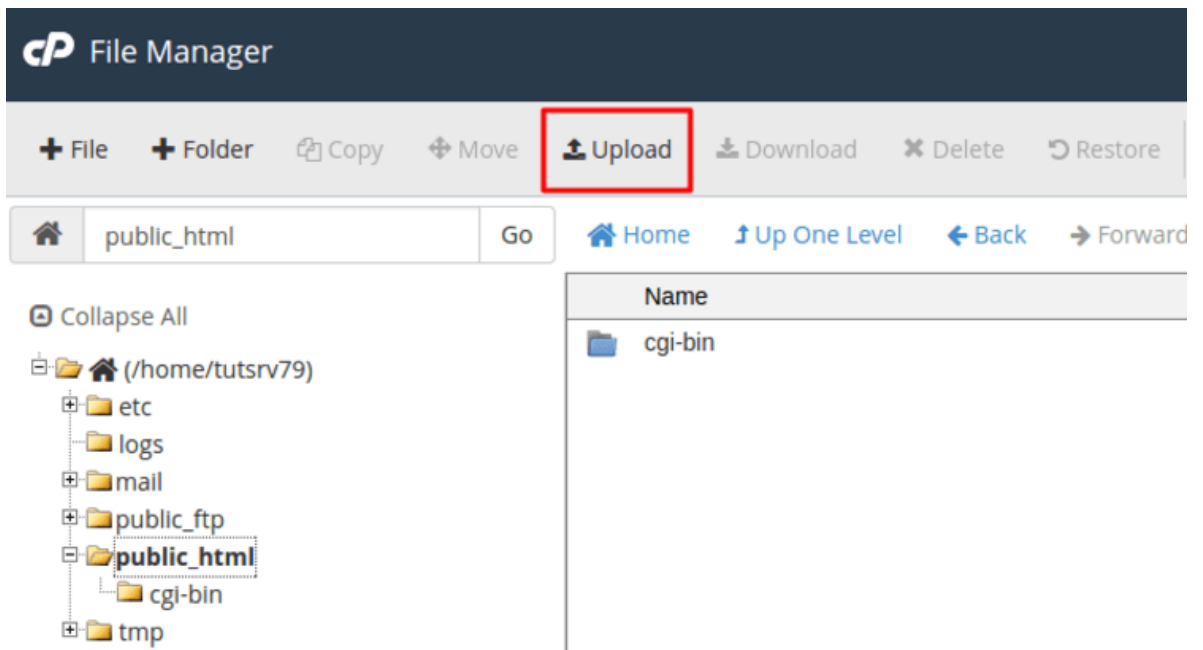
B1. Click vào icon **File Manager**, đặt bên dưới mục **Files**.



B2. Khi đã vào trong **File Manager**, mở thư mục **public_html**.

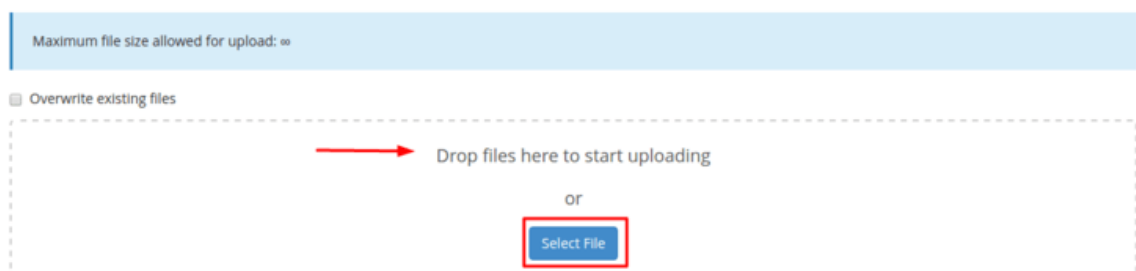


B3. Click **Upload** sau khi truy cập vào thư mục **public_html**.

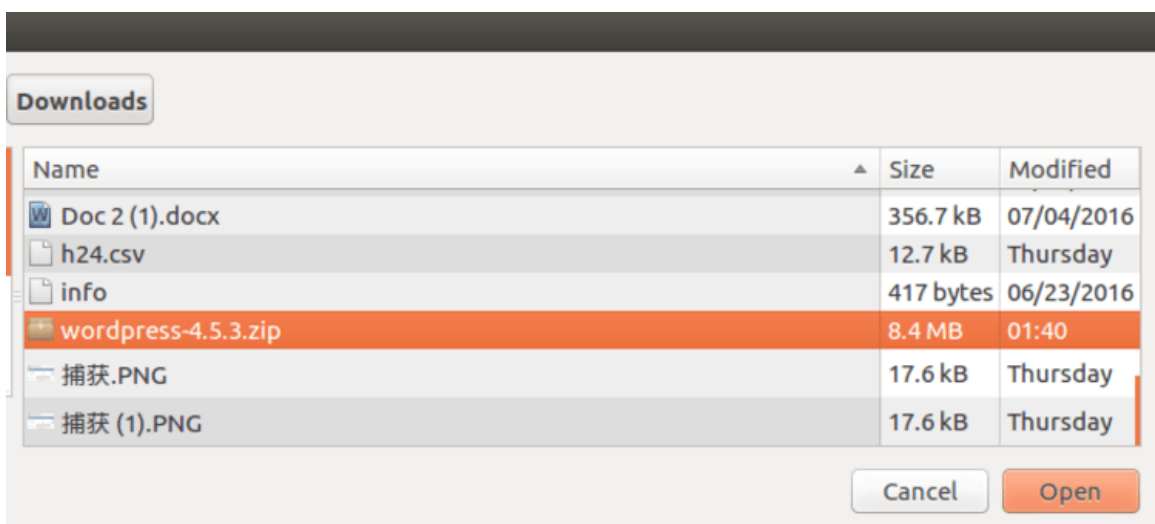


B4. Bạn có thể click vào nút “Select File” để chọn từng file hoặc cứ kéo thả vào vùng nhận file.

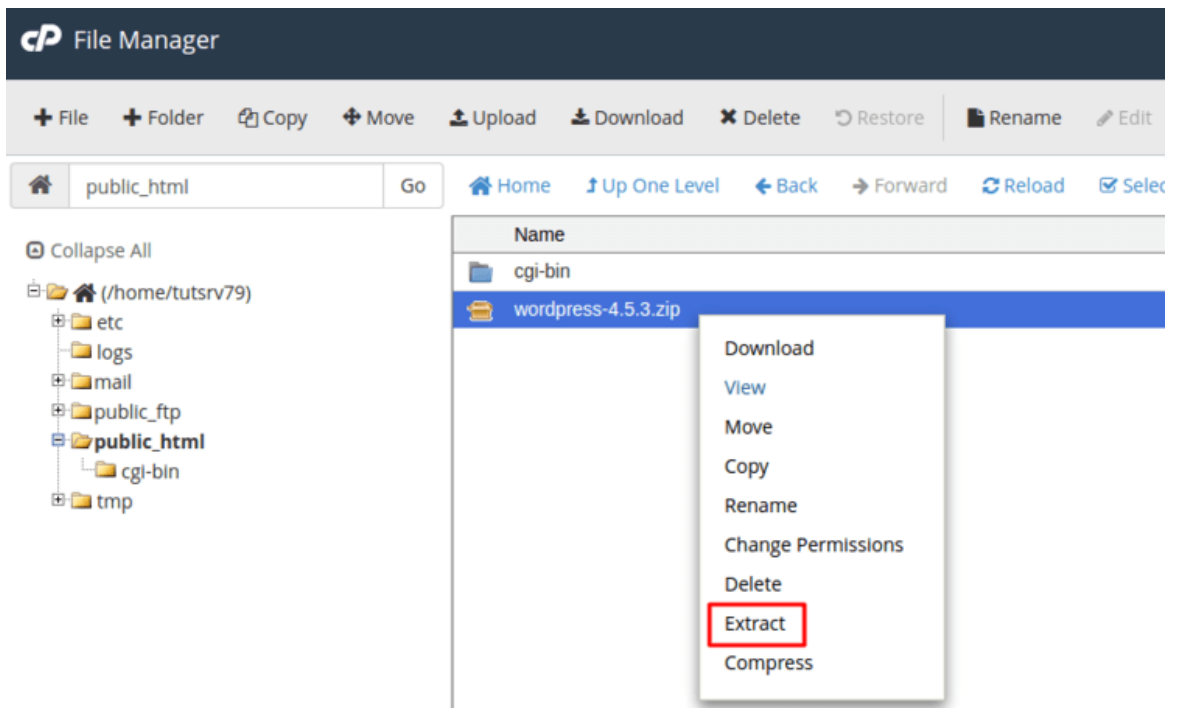
Select the file you want to upload to “/home/tutsv79/public_html”.



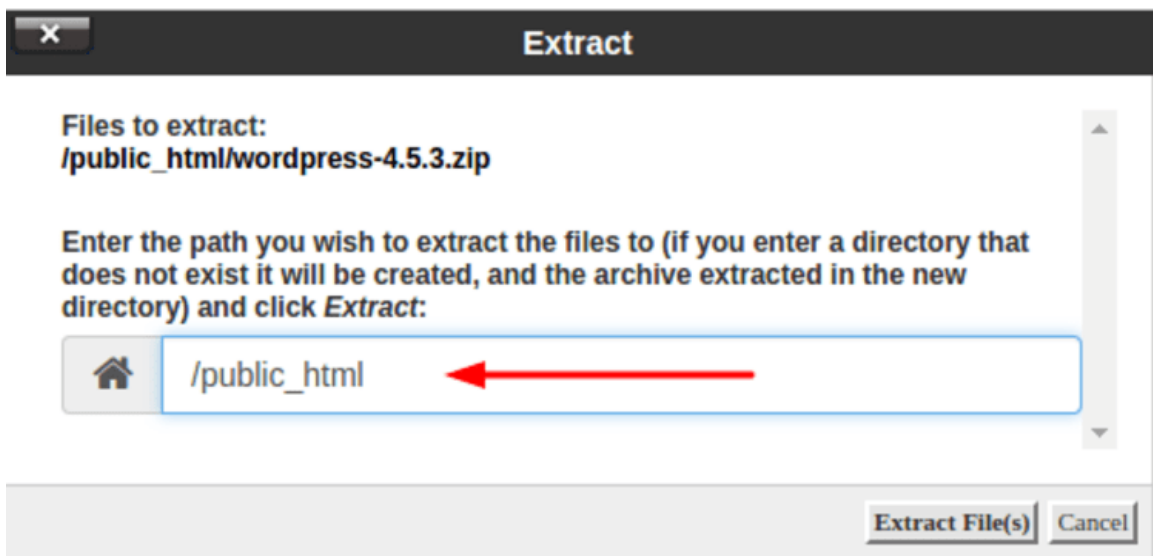
B5. Trong ví dụ này, chúng tôi đã kéo thả file cài wordpress: wordpress.zip.



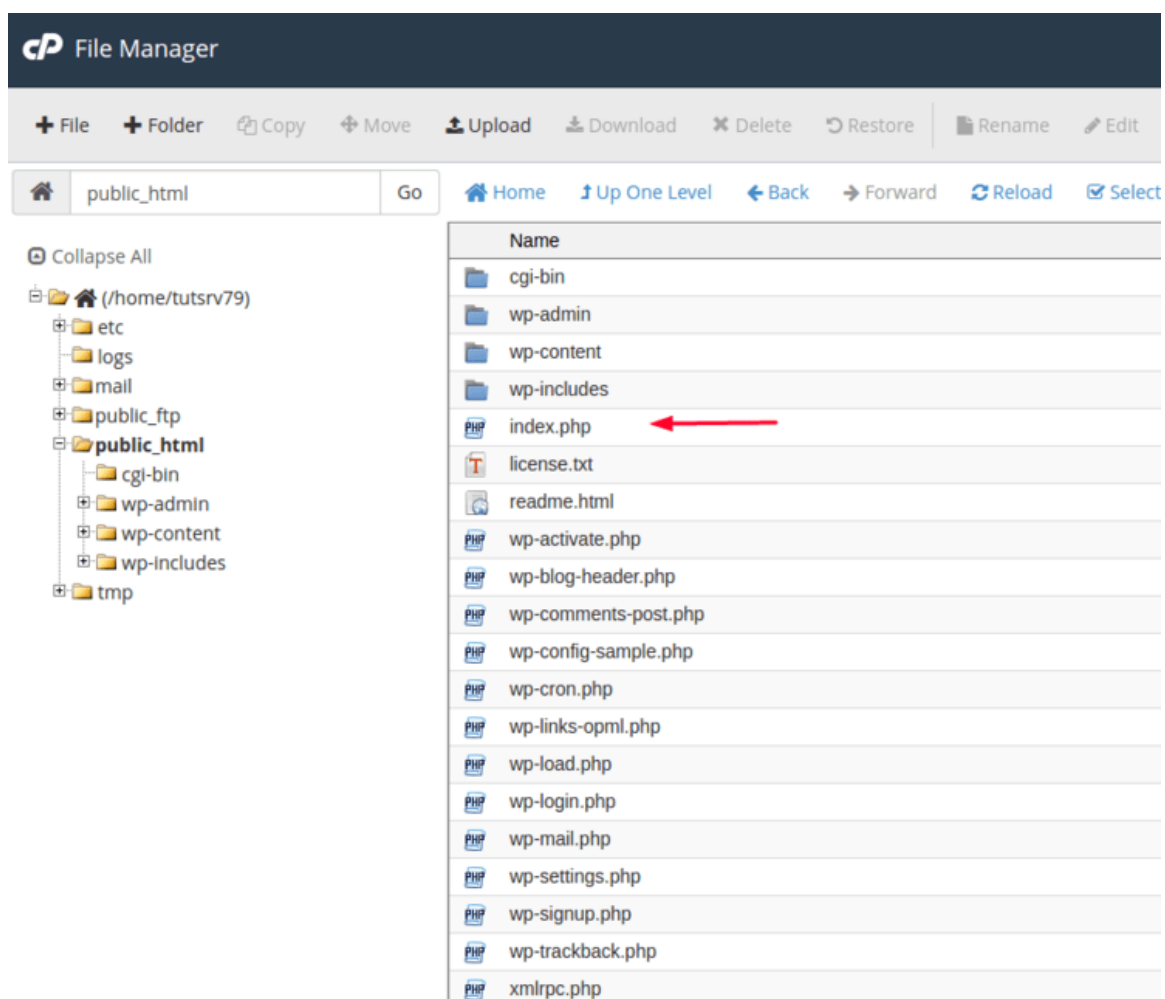
B6. Khi upload xong, trở lại với **File Manager** để thấy file archive đã xuất hiện trong thư mục **public_html**. Chuột phải và nhấn **Extract** để extract file archive.



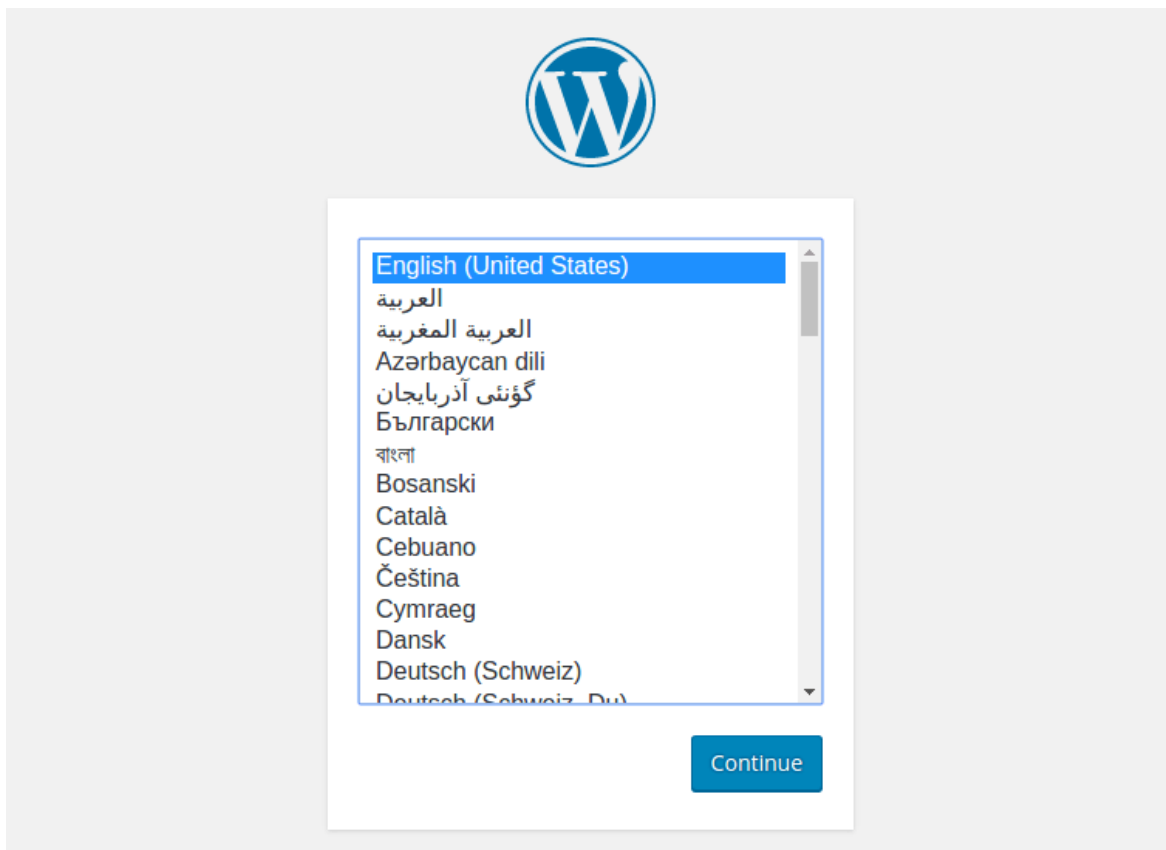
B7. Nếu bạn thấy cửa sổ hỏi vị trí file archive cần extract tới, hãy đảm bảo bạn chọn `/public_html`.



B8. File archive đã được extract, tất cả file bây giờ đã xuất hiện trong **public_html**. Chuyển lại thư mục **public_html** để xem các files giờ đã được chuyển hoàn toàn.



B9. Website của bạn bây giờ đã được upload hoàn toàn, và có thể truy cập qua URL của website. Để kiểm tra bạn chỉ cần gõ URL trên trình duyệt, ví dụ như <http://www.domainname.com/>, xem nó có hiện đúng không. Trong ví dụ này, chúng tôi đã thấy trang cài đặt WordPress



3.3. Chỉnh sửa các thông số hoạt động cho website

Nếu muốn có một website trên mạng, bạn bắt buộc phải có 2 thứ, đó là **Tên miền** và **Hosting**. Sau đó, để sử dụng được, bạn cần kết nối 2 thứ đó lại với nhau, hay còn gọi là trỏ Tên miền về Hosting.

Tên miền là địa chỉ website, để người khác có thể xem được thông tin giới thiệu, sản phẩm, dịch vụ mà bạn cung cấp. Hosting là nơi chứa toàn bộ dữ liệu website của bạn.



Có 2 cách để trỏ tên miền về hosting, bạn chỉ cần làm 1 trong 2 cách là được:

Trỏ tên miền về Name Server của Hosting

Trỏ tên miền về địa chỉ IP của Hosting

Cách đầu tiên thì nhanh gọn và đơn giản hơn nhiều, bạn chỉ cần trỏ tên miền về name server xong là website sẽ hoạt động ngay lập tức, toàn bộ service khác cũng sẽ được tự động kích hoạt theo.

Nếu bạn không chuyên, hãy ưu tiên lựa chọn phương án số 1.

3.3.1. Trỏ tên miền về Name Server của Hosting

Sau khi Hosting được kích hoạt, bạn sẽ nhận được một email chứa toàn bộ thông tin cần thiết để sử dụng, trong đó có **2 địa chỉ Name server**, cái mà chúng ta cần trỏ tên miền đến. Tiêu đề địa chỉ email thường là “*New Account Information*”.

Ví dụ mail của [Hawk Host](#):

New Account Information:

Domain: [canhme.com](#)
Username: canhmeco
Password: 
Hosting Package: Primary
Shared/Dedicated IP: Refer to your control panel
Nameserver 1: [ns9.hawkhost.com](#)
Nameserver 2: [ns10.hawkhost.com](#)



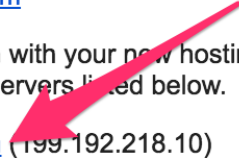
Ví dụ mail của [StableHost](#):

Server Information

Server Name: [phx14.stablehost.com](#)

If you are using an existing domain with your new hosting account, you will need to update the nameservers to point to the nameservers listed below.

Nameserver 1: [ns1.stablehost.com](#) (199.192.218.10)
Nameserver 2: [ns2.stablehost.com](#) (199.192.219.10)



Giờ chỉ cần vào trang quản lý tên miền, thay đổi name server, đợi vài phút cho tên miền cập nhật là xong.

Mình có một số bài hướng dẫn quản lý tên miền ở các nhà cung cấp phổ biến, trong đó có phần thay đổi name server, mời bạn tham khảo thêm:

[GoDaddy](#)

[Namecheap](#)

[Domain.com](#), [MyDomain](#), [Dotster](#) và [Netfirms](#)

[Namesilo](#)

[Exabytes](#)

[BigRock](#), [Whois.com](#)

[1&1](#)

[Google Domains](#)

[Z.com](#)

[Name.com](#)

Để kiểm tra tên miền đã thay đổi name server xong chưa, bạn có thể Whois ở <https://who.is>, thông tin update nhanh, không bị cache.

Lưu ý: với hosting sử dụng cPanel, sau khi thay đổi name server, nếu muốn thay đổi các record, bạn hãy vào cPanel, click *Advanced Zone Editor* để cấu hình.

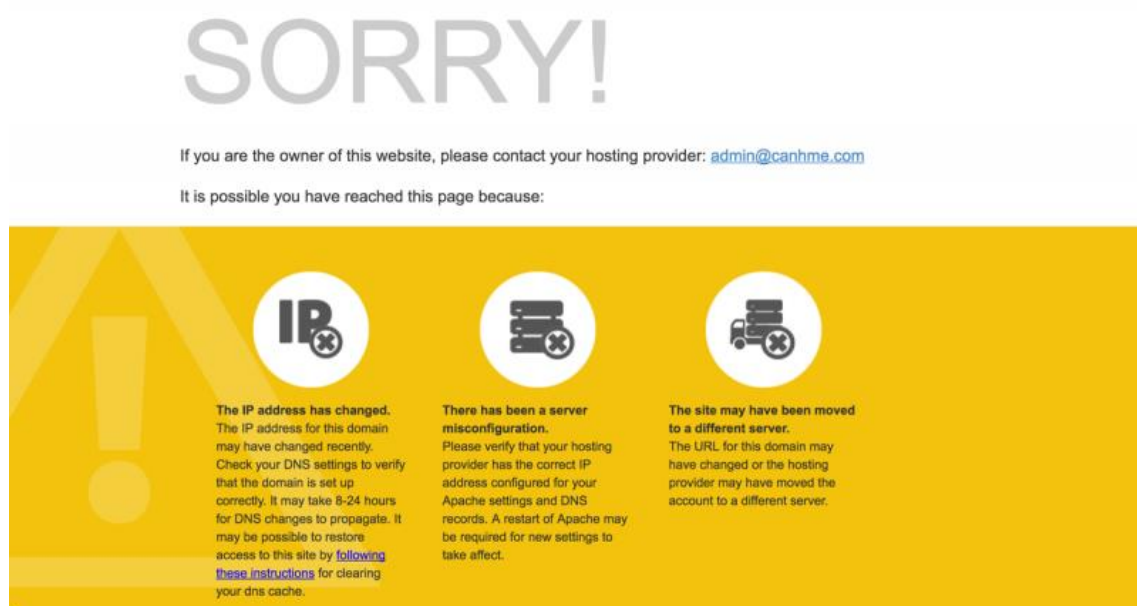
3.3.2. Trỏ tên miền về địa chỉ IP của Hosting

Cách này phức tạp hơn một chút so với cách trên, tuy nhiên bạn có toàn quyền lựa chọn name server, thao tác quản lý các record thuận tiện hơn, record cập nhật nhanh, tận dụng được một số dịch vụ cao cấp của name server.

3.3.2.1. Xác định địa chỉ IP của Hosting

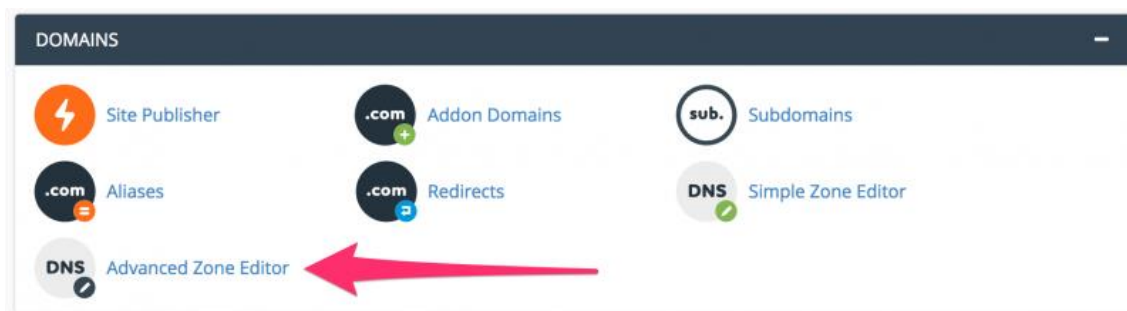
Đầu tiên, cần phải xác định địa chỉ IP của gói Hosting mà bạn đã đăng ký.

Nhiều bạn sử dụng luôn địa chỉ IP có trong mail thông tin Hosting, tuy nhiên, đôi lúc đây không phải địa chỉ IP để bạn trỏ tới, dẫn đến gặp thông báo lỗi *defaultwebpage.cgi* như hình bên dưới:



Lưu ý: thông báo lỗi Default Web Site Page hay bị cache, nên dù bạn đã update lại IP chuẩn, nhưng vẫn gặp lỗi. Lúc này hãy xóa hết browser cache hoặc sử dụng trình duyệt ẩn danh để truy cập lại vào website.

Để lấy IP chuẩn, bạn hãy login vào cPanel, trong phần *DOMAINS* nhấn vào *Advanced Zone Editor*.



Địa chỉ IP sẽ xuất hiện như bên dưới:

| Zone File Records | | | | | | |
|--------------------------|-------|-------|-------|--|---------|--------|
| Name | TTL | Class | Type | Record | Actions | |
| canhme.com. | 14400 | IN | A | 198.252.101.181 | Edit | Delete |
| mail.canhme.com. | 14400 | IN | CNAME | canhme.com | Edit | Delete |
| www.canhme.com. | 14400 | IN | CNAME | canhme.com | Edit | Delete |
| ftp.canhme.com. | 14400 | IN | CNAME | canhme.com | Edit | Delete |
| canhme.com. | 14400 | IN | TXT | v=spf1 +a +mx +ip4:216.12.199.6 +include:_spf.arandomserver.com ~all | Edit | Delete |
| webdisk.canhme.com. | 14400 | IN | A | 198.252.101.181 | Edit | Delete |
| cp calendars.canhme.com. | 14400 | IN | A | 198.252.101.181 | Edit | Delete |
| cpcontacts.canhme.com. | 14400 | IN | A | 198.252.101.181 | Edit | Delete |

Sau khi có được IP, giờ chúng ta **trở tên miền về IP**.

Do không dùng name server của nhà cung cấp Hosting, do đó bạn cần lựa chọn name server cho tên miền, có 2 lựa chọn:

Dùng name server mặc định của nhà cung cấp tên miền

Dùng name server trung gian, như [CloudFlare](#), [PowerDNS](#) của [Namecheap](#)...

3.3.2.2. Nếu dùng name server mặc định của nhà cung cấp tên miền

Thông thường, khi đăng ký mặc định tên miền sẽ sử dụng luôn name server của nhà cung cấp. Nếu như bạn đã lỡ trở sang name server khác thì cần phải trở lại, trong giao diện quản lý tên miền kiểu gì cũng có lựa chọn cho bạn thay đổi về record default này.

Nếu bạn tìm không thấy, hãy để lại tên nhà cung cấp bằng cách comment bên dưới mình sẽ hướng dẫn chi tiết.

Sau khi chắc chắn tên miền đang trở về name server mặc định, tùy từng nhà cung cấp sẽ có hướng dẫn để bạn trở về IP Hosting khác nhau.

Đợi một lúc để hệ thống cập nhật IP mới là xong. Một số nơi update rất lâu, có thể tới 48 giờ, ví dụ như 1&1, hoặc Z.com cũng chậm, bạn lưu ý kiên nhẫn chờ.

Có 2 record cần thiết bắt buộc phải có để tên miền hoạt động được bình thường, đó là **record @** (hay domain.com) và **record www** (hay

www.domain.com). Bạn lưu ý trở cả 2 record này về IP Hosting, kiểm tra bằng cách ping.

3.3.2.3. Nếu dùng name server trung gian

Nếu bạn dùng name server trung gian, không dùng của nhà cung cấp tên miền, tùy từng dịch vụ mà có hướng dẫn riêng.

TÀI LIỆU THAM KHẢO

- [1] Bùi Huy Quỳnh (2004), *Giáo trình thiết kế web*, Đại học sư phạm Kỹ Thuật TP HCM.
- [2] Dương Thành Phết (2017), *Bài thực hành thiết kế web cơ bản với ngôn ngữ javascript*, <http://thayphet.net>.