

WEB DESIGN WITH HTML, CSS & BOOTSTRAP

PHẦN 1: GIỚI THIỆU. HTML

Contents

1. Một số khái niệm cơ bản.....	3
2. Các bước thiết lập Website	7
3. Các ngôn ngữ tạo nên một trang web là gì?	10
3.1 HTML = Hypertext Markup Language	10
3.2 CSS = Cascading Style Sheets	10
3.3 Javascript	10
4. HTML	11
4.1. Cài đặt môi trường.....	11
4.2. Định dạng tập tin html	11
4.3. Element	11
4.4. Các cài đặt thẻ cơ bản.....	12
a) Thẻ trong html	12
b) Structure of an HTML page	13
c) Thẻ html	14
d) Thẻ head & body	14
e) Thẻ heading	15
f) Thẻ paragraph <p> </p>	16
g) Thẻ break line 	16
h) Thẻ <hr> (Horizontal rules)	18
i) Định dạng văn bản	19
j) Thẻ Chú thích (Comment).....	21
k) Liên kết (link) trang.....	21
l) Thẻ image	23
m) Thẻ audio	26
n) Thẻ video	27
o) List (Danh sách)	28
p) Table (Bảng)	30
q) Thẻ span và thẻ div	37
r) Thẻ meta & title	38
s) Thẻ frame & iframe	39
t) Thẻ nút bấm (button).....	47
u) Form và Thẻ form.....	48
Giới thiệu về Form.....	48
Cấu trúc một HTML Form.....	49
Các thành phần của Form.....	50

Text field	50
Password field	50
Check box.....	51
Radio button	51
File form control	52
Nút lệnh (Button).....	53
Combo Box (Drop-down menu/ Pull-down menu).....	54
Listbox	55
Hộp nhập văn bản nhiều dòng (TextArea/ Multiline Text Field)	56
Label.....	57
Fieldset	57
Phân biệt phương thức GET/POST	57
Phương thức GET	57
Phương thức POST.....	58
v) Thẻ <marquee>.....	59
w) <figcaption> & <figure>	59
x) Thẻ <pre>.....	61
4.5 Đường dẫn (liên kết/ link) tương đối & tuyệt đối (cố định)	61
4.6 Màu sắc trong HTML	65
PHỤ LỤC.....	67

1. Một số khái niệm cơ bản

- **Internet:** mạng toàn cầu-kết nối các máy tính khắp thế giới với nhau. (LAN, nhiều LAN = WAN, nhiều WAN = Internet)
- **Trang web, web site**
 - **Trang web (Web page):**
 - Là một trang nội dung hiển thị các thông tin, dữ liệu dưới dạng văn bản, âm thanh, hình ảnh, ..
 - Một trang web có thể được viết bằng nhiều ngôn ngữ khác nhau: php, asp, aspx... nhưng kết quả khi trả về máy khách (client) là ngôn ngữ HTML (Hypertext Markup Language-Ngôn ngữ liên kết siêu văn bản).
 - **Web site:**
 - Tập hợp các trang web của các tổ chức hay cá nhân trên mạng có nội dung thống nhất phục vụ cho một mục đích nào đó.
 - Tập các trang web được liên kết với nhau, trình bày/mô tả về một chủ đề/đối tượng cụ thể nào đó, chứa nội dung như văn bản, hình ảnh, video, âm thanh, và được lưu trữ trên một máy chủ web.
- **Tại sao website lại cần Internet và ngược lại? (Mối quan hệ giữa chúng là gì?)**
 - ❖ *Website phụ thuộc vào Internet để truy cập:*
 - Một website là tập hợp các trang web chứa nội dung (văn bản, hình ảnh, video, v.v.) được lưu trữ trên các máy chủ. Để người dùng truy cập vào website, cần có một phương tiện truyền tải dữ liệu, và Internet chính là hệ thống mạng toàn cầu cho phép điều này.
 - Internet đóng vai trò như một phương tiện kết nối giữa máy chủ lưu trữ website và thiết bị của người dùng (như máy tính, điện thoại di động).
 - Thông qua Internet, nhiều người biết đến & tiếp xúc được website.
 - ❖ *Internet cần website để cung cấp nội dung:*
 - Internet bản thân chỉ là mạng lưới kết nối các thiết bị và hệ thống trên toàn cầu. Tuy nhiên, để Internet trở nên hữu ích, nó cần các dịch vụ và nội dung, và các website là nguồn cung cấp chính của các nội dung đó. Nhờ có các website, người dùng có thể tìm kiếm thông tin, giao tiếp, giải trí và thực hiện nhiều hoạt động khác.
 - Các website đóng vai trò là điểm cung cấp dịch vụ, như thương mại điện tử, mạng xã hội, giáo dục trực tuyến, và rất nhiều dịch vụ khác.
- **Mạng máy tính** là một hệ thống gồm từ hai máy trở lên, những máy tính này được kết nối với nhau nhằm mục đích trao đổi thông tin và dữ liệu.
- **Máy chủ - Máy khách:**
 - ❖ Trong mạng máy tính, máy tính được sử dụng cho những máy tính khác truy cập đến gọi là máy chủ (server) (**Máy cung cấp dịch vụ**).
 - ❖ Các máy tính truy cập vào máy chủ được gọi là máy khách (client) (**sử dụng dịch vụ**).

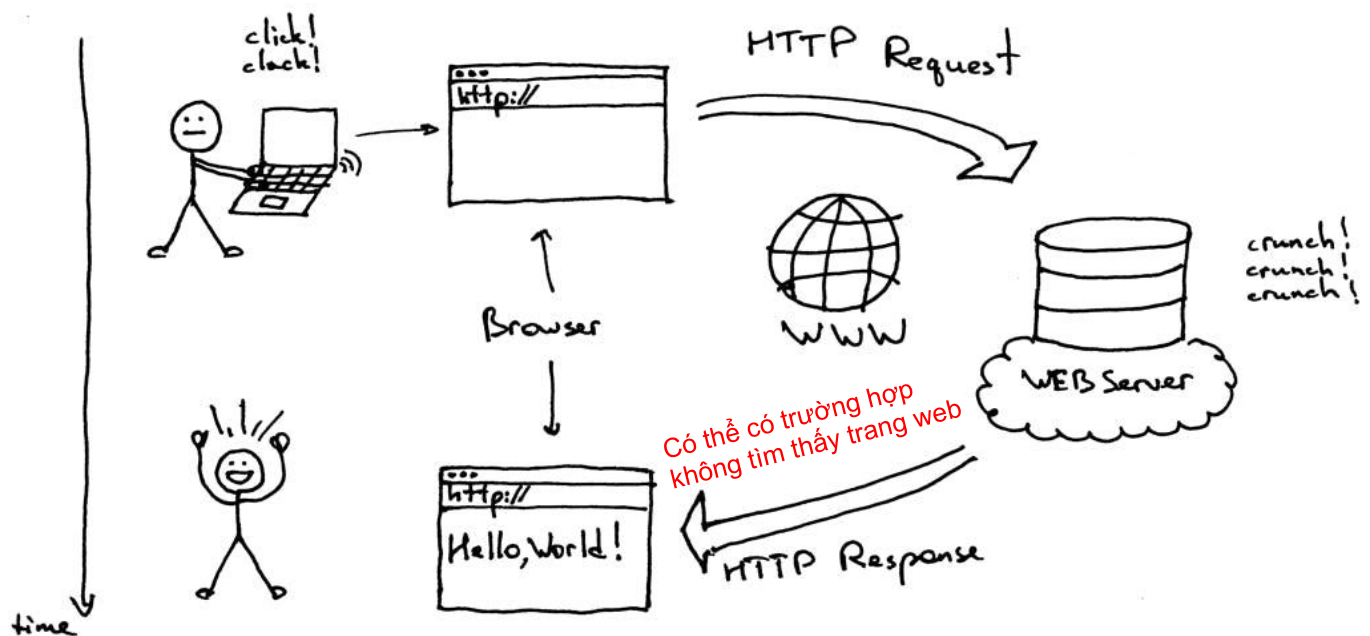
Máy chủ (Server)

- Máy chủ là máy tính chuyên cung cấp tài nguyên, dịch vụ cho máy tính khác.
- Thường được cài các phần mềm chuyên dụng để có khả năng cung cấp
- Máy chủ có cấu hình cao, khả năng hoạt động ổn định
- Một máy chủ có thể dùng cho một hay nhiều mục đích. Tên máy chủ thường gắn với mục đích sử dụng.

- Ví dụ: *File server, Application server, Mail server, Web server*
- **Web server:** *máy chủ cung cấp dịch vụ web* (cung cấp các trang web hiển thị một chủ đề nào đó: sản phẩm, bán hàng...)
- **Đề cấu hình:** Cung cấp các phần mềm Wampp, Xampp, Apache...
- * **Trang web:** Những **tập tin** có phần mở rộng là .htm, html → Web server sẽ lưu trữ những tập tin này (trong ổ cứng).

Máy khách (Client)

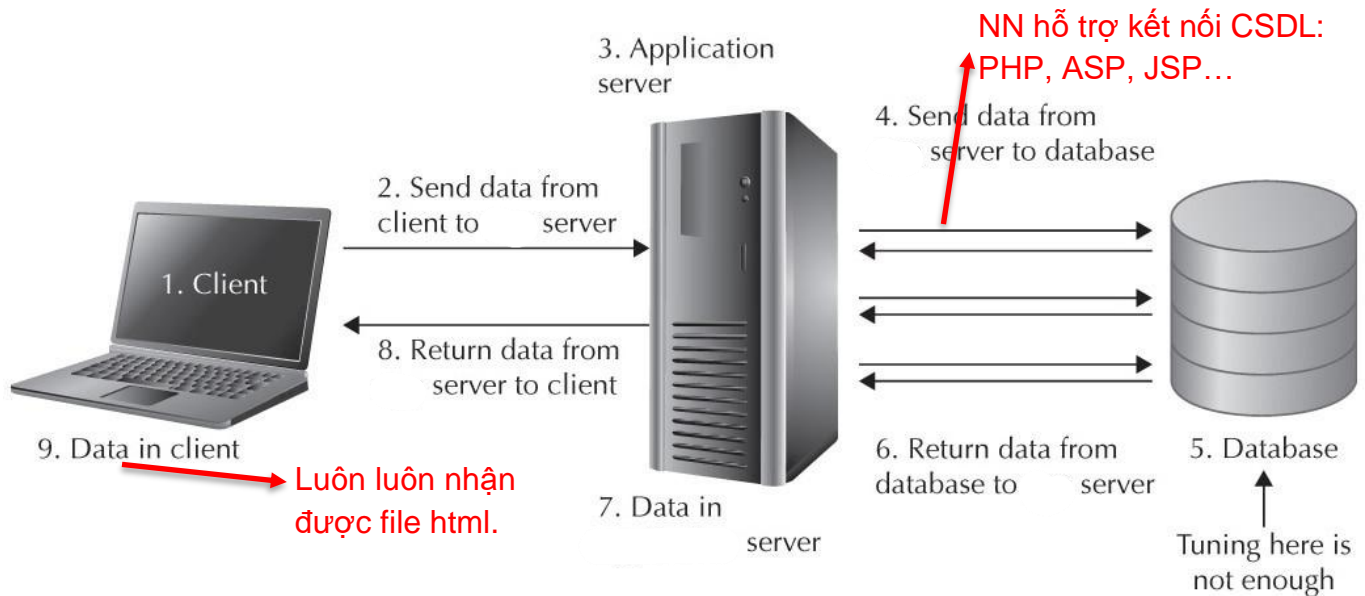
- Máy khách là **máy khai thác dịch vụ của máy chủ**
- Với mỗi dịch vụ, thường có các phần mềm chuyên biệt để hỗ trợ cho việc khai thác khai thác
- **Một máy tính có thể vừa là client vừa là server**
- Một máy tính có thể khai thác dịch vụ của chính nó.



Phân loại trang web: Dựa vào công nghệ phát triển, có 2 loại:

- **Web tĩnh:**
 - Nội dung trang web không thay đổi, muốn thay đổi nội dung cần **lập trình viên** mở source code để sửa.
 - Dễ phát triển
 - Tương tác yếu (không có sự tương tác) với người duyệt web (không có chức năng nhập bình luận và gửi đi, nhấn button để thay đổi nội dung, đăng nhập & đăng ký...)
 - Sử dụng ngôn ngữ HTML, CSS, JavaScript [HTML5, CSS3, *Jquery (dạng thư viện-OOP)*] để thiết kế (định hình mẫu web, khung sườn)
 - Người làm web tĩnh thường dùng các công cụ trực quan để tạo ra trang web như FrontPage.
 - Không nhận thông tin phản hồi từ người dùng.
 - **Không có cơ sở dữ liệu.**
- **Web động:**
 - Khó phát triển hơn
 - Tương tác mạnh với người duyệt web
 - Sử dụng nhiều ngôn ngữ khác nhau để lập trình
 - Thường phải viết nhiều mã lệnh
 - Nhận thông tin phản hồi từ người dùng.

- Có cơ sở dữ liệu.
- Sử dụng **thêm** ngôn ngữ **PHP, MySQL**
- Thay đổi dữ liệu trên trang web thông qua quản trị web (Admin, có thể không biết code như LTV, chỉ biết thao tác với dữ liệu thông qua CSDL), sau khi được QTV tác động thì trang web tự động cập nhật dữ liệu.



Giao thức mạng (Protocol): Một số giao thức mạng thông dụng:

- **TCP/IP (Transmission Control Protocol/ Internet Protocol) :**
 - Là một hệ thống giao thức - một tập hợp các giao thức hỗ trợ việc lưu truyền dữ liệu trên mạng
 - TCP/IP và các giao thức liên quan tạo ra một hệ thống hoàn chỉnh *quản lý quá trình dữ liệu được xử lý, chuyển và nhận* trên một mạng sử dụng TCP/IP. Một hệ thống các giao thức liên quan, chẳng hạn như TCP/IP, được gọi là bộ giao thức.
- **FTP: (File Transfer Protocol - Giao thức chuyển tập tin)**
 - Dùng để *truyền tải tài liệu từ một máy tính này tới một máy tính khác* qua mạng Internet.
 - Giao thức chuyển tập tin dùng để *truyền tải các tệp từ một trạm làm việc lên kho tập tin, một máy tính lưu trữ tài liệu, hoặc dùng để truyền tải các tệp tin từ kho tập tin tin xuống một trạm làm việc*. Đó là cách tập tin được chuyển đi từ một thiết bị này tới một thiết bị khác theo trình tự cho các tập tin đó.
 - FPT còn làm nhiệm vụ *tải các trang web từ người thiết kế đến các máy chủ*.
- **HTTP (HyperText Transfer Protocol – Giao thức truyền tải siêu văn bản):**
 - Công việc chính của HTTP chuyển nội dung một trang web vào một trình duyệt cho người truy cập web xem.
 - Là một giao thức dùng để chuyển các file từ một kho web vào một trình duyệt để người dùng xem được trang web đó trên mạng internet.

Tên miền (Domain name)

- Tên miền được phân chia ra thành nhiều cấp, mỗi cấp cách nhau bởi dấu "." và được đánh số cấp từ phải qua trái, bắt đầu từ cấp 1.
 - Ví dụ: uel.edu.vn
 - vn: Nước Việt Nam (cấp 1)
 - edu: Tổ chức giáo dục (cấp 2)
 - uel: Tên cơ quan (cấp 3)

- Một số tên miền cấp 1 dùng chung là:
 - .com: công ty thương mại (commercial)
 - .edu: các trường học, các tổ chức giáo dục.
 - .net: các mạng (network)
 - .int: các tổ chức quốc tế(international organizaions)
 - .org: các tổ chức khác (other organizations)
- Để sở hữu một tên miền chúng ta phải hợp đồng và trả chi phí hàng năm. Nếu hết hạn sở hữu nhưng chúng ta không tiếp tục hợp đồng và chi trả chi phí thì tên miền này có thể bị mua lại bởi một người khác.

URL - định vị các tài nguyên Internet

- URL - Uniform Resource Locator
- URL là **định danh duy nhất cho các tài nguyên** Internet
- Chỉ ra:
 - Cách truy cập
 - Vị trí tài nguyên
- Cú pháp chung:

protocol://host_name[:port_num][/path]/[file_name]

Đối với web: port_num = 80 → Người dùng không cần gõ

http://www.microsoft.com:8080/en/us/default.aspx

--	--	--	--

protocol
server name
port
directory/file name on the server

Cổng dịch vụ (Service port)

- Đây là 1 con số nguyên từ 0 đến 65535 xác định dịch vụ của máy chủ.
- Mỗi giao thức, dịch vụ sẽ mang một cổng xác định → Nếu cổng được dùng bởi một dịch vụ rồi thì sẽ bị lỗi, ta cần sử dụng port khác.
 - VD:
 - **Web:** sử dụng cổng 80
 - **Ftp:** sử dụng cổng 21.

World Wide Web (WWW)

- **WWW** là tập hợp các web site trên mạng internet.
- WWW là dịch vụ thông dụng nhất trên Internet. Để có được dịch vụ này người dùng cần phải có trình duyệt web (web browser).

Thuật ngữ

- IAP (Internet Access Provider): Nhà cung cấp đường truyền Internet.
- ISP (Internet Service Provider): Nhà cung cấp dịch vụ Internet. Một số ISP hiện nay ở Việt Nam: VDC, FPT, SaigonNet, NetNam, Viettel, ...

2. Các bước thiết lập Website

• **Xác định yêu cầu của website**

• *Mục tiêu, yêu cầu của website*

- Đề ra mục tiêu **khái quát, ngắn gọn, rõ ràng của website cần thiết kế**
- Là công cụ để đánh giá sự thành bại của website
- *Mục tiêu phải dài hạn, bao trùm toàn bộ kế hoạch phát triển.*

• *Đối tượng mà website phục vụ*

- Xác định **loại độc giả chính** của website
- Tìm hiểu: Sự hiểu biết; Trình độ; Sở thích; Kinh nghiệm sử dụng máy tính, duyệt web;
- Thiết kế web tốt = **thích hợp cho tất cả các loại độc giả có trình độ và nhu cầu khác nhau**

• *Chủ đề của website*

- Giúp định hướng cho nhà thiết kế: Giáo dục; Huấn luyện; Tin tức; Giải trí; Dẫn dắt trao đổi thảo luận; Mua bán, quản lý;...

• **Mua tên miền**

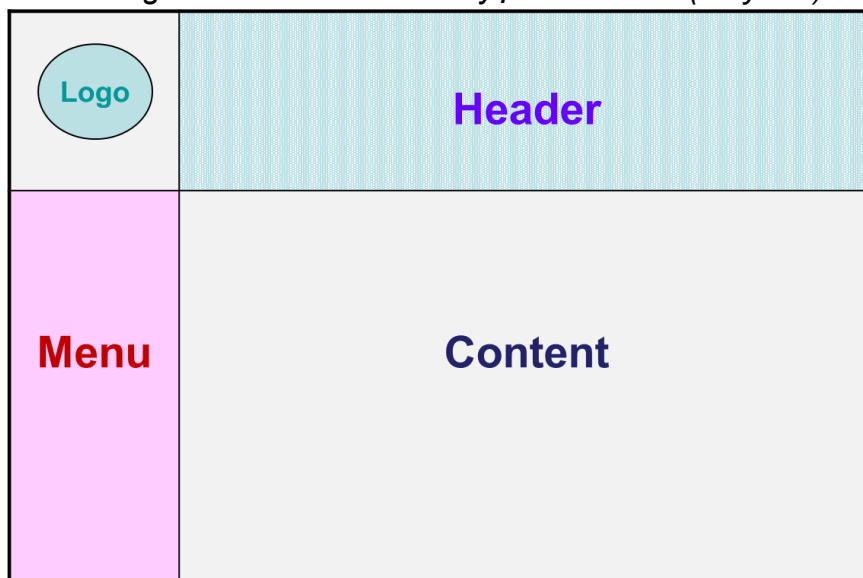
- Tên miền = tên website: Tên ngắn; Gợi nhớ; Dễ đọc; Không gây nhầm lẫn; Thể hiện được tên công ty hoặc nhãn hiệu của công ty
 - thegioididong.com
 - tuoitre.com.vn

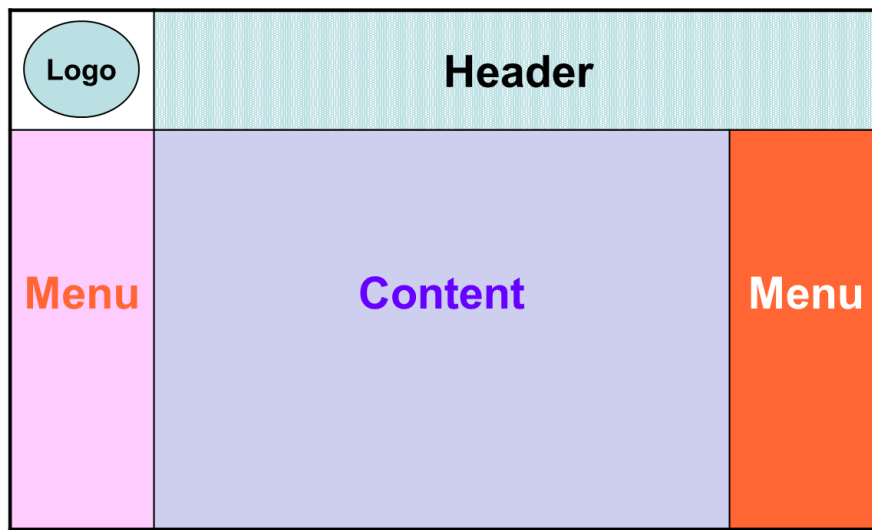
• **Thuê chỗ hosting**

• **Thiết kế website**

- Cũng giống như phân tích và thiết kế hệ thống thông tin, các bước cơ bản cho việc phát triển một website bao gồm:
 - **Đặc tả:**
 - Web để làm gì?
 - Web cho ai dùng?
 - Trình độ người dùng ra sao?
 - Nội dung, hình ảnh như thế nào?
 - **Phân tích**
 - *Mối liên quan giữa các nội dung trang web?*
 - *Thứ tự của các nội dung trang web?*
 - **Thiết kế**
 - Sơ đồ cấu trúc website
 - Giao diện
 - Web tĩnh hay web động
 - Thiết kế cơ sở dữ liệu?
 - Nội dung từng trang
 - Liên kết giữa các trang
 - **Lập trình (xây dựng website)**
 - Cấu trúc thư mục
 - Các modul dùng chung
 - ...
 - **Kiểm thử**
 - Kiểm tra trên nhiều trình duyệt
 - Kiểm tra trên nhiều cấu hình mạng
 - Kiểm tra tốc độ

- Kiểm tra các liên kết (link)
- Kiểm tra bảo mật
-
- Thiết kế giao diện website:
 - *Hướng đến người sử dụng*
 - Người sử dụng là **yếu tố chính quyết định sự thành bại của website**
 - Yêu cầu mức độ hoàn hảo của thiết kế giao diện
 - Có thể áp dụng cách: thử nghiệm website → nhận phản hồi từ độc giả
 - *Có định hướng rõ ràng*
 - Độc giả có thể trả lời được các câu hỏi sau:
 - Đang ở đâu?
 - Có thể làm được những gì?
 - Có thể tiếp tục đi đâu?
 - Các biểu tượng, hướng dẫn nên nhất quán, dễ hiểu
 - ***Không có trang cuối cùng (dead-end)***
 - Mỗi trang có ít nhất một liên kết. **Luôn tạo cho người dùng có khả năng quay lại trang chủ hoặc những trang chủ chốt của website.**
 - Không nên có trang “dead – end”, nghĩa là trang không có khả năng đi đến các trang khác.
 - Nên để liên kết trang chủ trong mọi trang.
 - *Tính ổn định của các trang Web*
 - Ổn định về cấu trúc và nội dung
 - Về cấu trúc: các thành phần được đặt đúng chỗ
 - Nội dung:
 - Các mối liên kết luôn bảo đảm có đích đến
 - Nội dung phải được cập nhật cho phù hợp với nội dung và ngữ cảnh
 - *Phản hồi và đối thoại*
 - **Cho phép độc giả đóng góp ý kiến và phản hồi đến độc giả một cách nhanh nhất**
 - Luôn cung cấp địa chỉ liên kết đến Webmaster
 - Lên kế hoạch nhân sự phụ trách lâu dài
 - *Tính tương thích trên các trình duyệt khác nhau (thay thế)*





- **Xác định kiểu chữ, màu sắc**
 - Font chữ sẽ phụ thuộc vào: Đặc điểm thông tin; Độc giả; Trình duyệt, độ phân giải; Ngôn ngữ sử dụng; Font có chân, không chân,...
 - Gam màu nên **thống nhất toàn bộ** website
- **Cấu trúc website**
 - Sitemap: **Sitemap (sơ đồ website)** là một file liệt kê các trang và tệp tin trên **website**. Danh sách liệt kê **được sắp xếp theo dạng sơ đồ phân tầng (giảm dần sự quan trọng)**



- Dùng để tổ chức các khối thông tin phức hợp.
- Là hệ thống được dùng thông dụng nhất
- Gắn với mô hình tổ chức thế giới thực
- Dễ hình dung tổ chức website
 - **Cái nhìn tổng quan từ trang chủ**
- **Đưa vào hoạt động**
- **Duy trì thông tin, bảo trì website**
- * **Quản lý thời gian**
 - Website **luôn phải được cập nhật thông tin** để độc giả luôn cảm nhận được cái mới
 - Nếu thông tin nào mới được cập nhật **ên có ký hiệu** cho độc giả biết
 - **Ghi thời gian** cập nhật
 - **Số người truy cập** website
 - Nếu website phức tạp **ên có trang hướng dẫn sử dụng**

3. Các ngôn ngữ tạo nên một trang web là gì?

3.1 HTML = Hypertext Markup Language

- NN đánh dấu siêu văn bản: Ngoài văn bản, HTML hỗ trợ nhiều loại phương tiện khác nhau, bao gồm cả hình ảnh và video.
 - Cấu trúc đơn giản nhất XD một web (thêm, xóa, sửa)
 - Hữu ích cho web dev
 - Ngôn ngữ cơ bản nhất để xây dựng các trang web.
 - It is the language of Web pages that **tells a browser how to display certain elements**, such as text and images **through the use of codes and symbols**.
 - HTML is the **standard** when it comes to creating Web pages.
 - là một hệ thống chú thích tài liệu tuân thủ XML bằng các thẻ (tags) để **định dạng cách hiển thị dữ liệu trên trang web**.
-
- XML is a **markup language** used to **store hierarchical data** in text files. It is also known as semi-structured data, like JSON. XML is machine-readable, yet can also be read and produced by people.
 - XML is **made up of elements**, sometimes casually referred to as a **tag soup**, which can themselves *contain other elements and/or text*. *Elements may also contain attributes*.
 - XML thường được sử dụng để trao đổi dữ liệu giữa các nền tảng (platforms), đặc biệt là qua internet.
 - Nó cũng ngày càng được sử dụng nhiều để lưu trữ dữ liệu bán cấu trúc (semi-structured data) trong kho dữ liệu NoSQL (cơ sở dữ liệu XML/kho tài liệu).
 - Hơn nữa, nó có tính linh hoạt trong việc xử lý dữ liệu định hướng tài liệu (document-oriented data) (text with markup), điều này khiến nó rất phổ biến trong ngành xuất bản. XML cũng được sử dụng rộng rãi cho các tệp cấu hình (configuration files).
 - HTML is sometimes called a programming language but it has no logic, so is a **markup language. (2024)**
-
- Các **trình duyệt** thường **không bảo lỗi cú pháp** cho ngôn ngữ HTML. Nếu viết sai cú pháp chỉ dẫn đến kết quả hiển thị không đúng với dự định.
 - **Tập tin HTML** có nội dung dưới **dạng text** bao gồm các thẻ (tags) nhỏ + nội dung cần hiển thị.
 - Các thẻ hiển thị nói cho trình duyệt biết nó phải hiển thị trang đó như thế nào
 - Ngôn ngữ cuối cùng hiển thị trang web là HTML, dù cho có viết ngôn ngữ khác.
 -

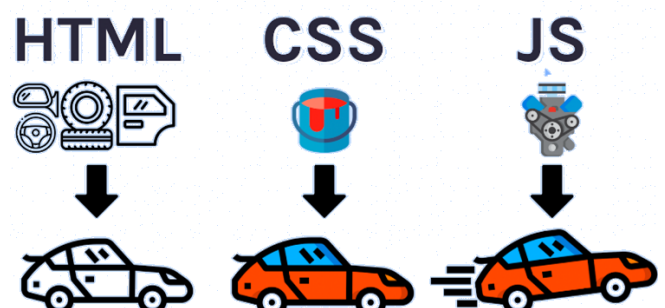
3.2 CSS = Cascading Style Sheets

Một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (HTML) = ngôn ngữ tạo phong cách cho trang web.

- Nếu HTML là phần thô của ngôi nhà, thì CSS là phần trang trí của ngôi nhà.

3.3 Javascript

~ các đường dây điện của ngôi nhà.



4. HTML

4.1. **Cài đặt môi trường**

- Browser (GG chrome...)
- Text Editor (VSC, Sublime Text, Notepad...)
 - ✎ VSC: Cài đặt các extensions (Live Server, auto close tag, auto rename tag, HTML Snippets, HTML CSS Support, Javascript code snippets...)

4.2. **Định dạng tập tin html**

- Lưu lại với tên tập tin: vidu1.**html** hoặc vidu1.**htm**
- Thông thường index.html → Do có nhiều server lựa chọn file này làm file mặc định của chúng.

4.3. **Element**

- Về mặt kỹ thuật, tài liệu HTML chỉ chứa các thẻ. Khi chúng được truy cập trên trình duyệt, các tài liệu HTML sẽ được phân tích cú pháp để chúng có thể được hiển thị bằng Mô hình đối tượng tài liệu (**Document Object Model**) (**DOM**). Các thẻ có trong tài liệu HTML nhưng các phần tử HTML chỉ xuất hiện sau khi tài liệu được phân tích cú pháp.
- Nói một cách đơn giản, thẻ HTML là ngôn ngữ đánh dấu mà bạn sử dụng trong HTML. Đây thường là thẻ bắt đầu hoặc thẻ mở và thẻ đóng hoặc kết thúc. **Thẻ được đặt trong dấu ngoặc nhọn < >.**
- Mặt khác, các phần tử HTML bao gồm nội dung.
- Vì vậy, dựa trên điều này: <p>Đây là sự khác biệt giữa các thành phần và thẻ HTML. </p>. Các thẻ ở đây là <p> và </p>, trong khi **toàn bộ được gọi là phần tử HTML (HTML element).**
- Ví dụ: **<h1>Hello World!</h1>** là một element
- An HTML element has the following syntax:
 1. An HTML element begins with **an opening or start tag, i.e., <p>.**
 2. An HTML element ends with a **closing or end tag, i.e. </p>.**
 3. The property of an HTML content is enclosed within the start and end tags, i.e., <p>This is the element content. </p>.
 4. It may also contain an attribute.

Empty Elements: Some HTML elements do have content and are simply called empty elements. These elements are **ended or closed with the opening tag**, i.e,
.

Nestled Elements

Nestled elements are **HTML elements that are found within other HTML elements**. For example, this is a nestled element:

<p>I do not know what it was.</p>

Uses of HTML elements

HTML elements serve a variety of purposes:

1. HTML elements can **describe the purpose of the text, as well as give it structure**. A good example of this type of HTML elements is `<h1>`, `<h2>` and other HTML headers.

For instance, `<h1>Introduction</h1>` communicates that "Introduction" is a first level heading.

Take note that structural HTML elements **only provide structure and do not have specific standards for display**. A browser, however, can have *default display for this type of elements*, hence (do đó) you often see *header tags being bigger than normal text on your browser*.

2. HTML elements can **dictate (quyết định) how certain content would appear (cách hiển thị một số nội dung nhất định)**.

For instance, using `` would render the text in boldface, while using `<i>` will display the text in italics.

3. HTML elements that **link your HTML document to another documents**. For instance, you can use `<a>` to link your document to another document.

4.4. Các cài đặt thẻ cơ bản

a) Thẻ trong html

- Thẻ (tags) thường đi theo cặp (gọi là **thẻ đôi**).

- Hầu hết các tag của HTML đều có thẻ bắt đầu (thẻ mở) và thẻ kết thúc (thẻ đóng) tương ứng.
- Thẻ kết thúc tương tự thẻ bắt đầu chỉ khác nhau là thêm ký tự `"/"` vào trước nó.
- Thẻ mở và thẻ đóng đều được đặt trong dấu: **`<thẻ mở> dữ liệu </thẻ đóng>`**
thẻ mở = thẻ đóng

- Thẻ mở sẽ biểu thị rằng trình duyệt sẽ xử lý văn bản tiếp theo bằng các thuộc tính của thẻ đó và thẻ đóng sẽ kết thúc văn bản đó.

- Ví dụ: thẻ mở `` và thẻ đóng `` tương ứng của nó sẽ hiển thị tất cả văn bản ở giữa được in đậm...

"My name is `Clark Kent`" will be displayed as:

My name is **Clark Kent**.

- And then there are what we call the **"empty elements"** or those that **work without a closing or opening tag**. Empty elements are often described as **self-closing tags**. (**Thẻ đơn**)

There are quite a few empty elements such as:

- | | | | | |
|---------|--------|------------|---------|-----------|
| ● area | ● base | ● basefont | ● br | ● col |
| ● frame | ● hr | ● img | ● input | ● isindex |
| ● link | ● meta | ● param | | |

All of these can stand alone. For example, line breaks in HTML documents are often denoted by **`
 or
`**. (**Hiện nay thẻ đơn yêu cầu thêm một dấu `"/"` ở phía sau**).

- Các thẻ **không phân biệt chữ hoa và thường**:

`<HTML>` tương `<Html>` đương `<html>`

- Bỏ qua các khoảng trắng thừa và các dấu ngắt dòng, xuống dòng giữa văn bản dữ liệu và các thẻ → Vì thế phải sử dụng các thẻ tương ứng để thực hiện các việc này (br).

b) Structure of an HTML page

- A browser would start reading an HTML page from the top going down, from left going right.

- It does follow a basic structure.

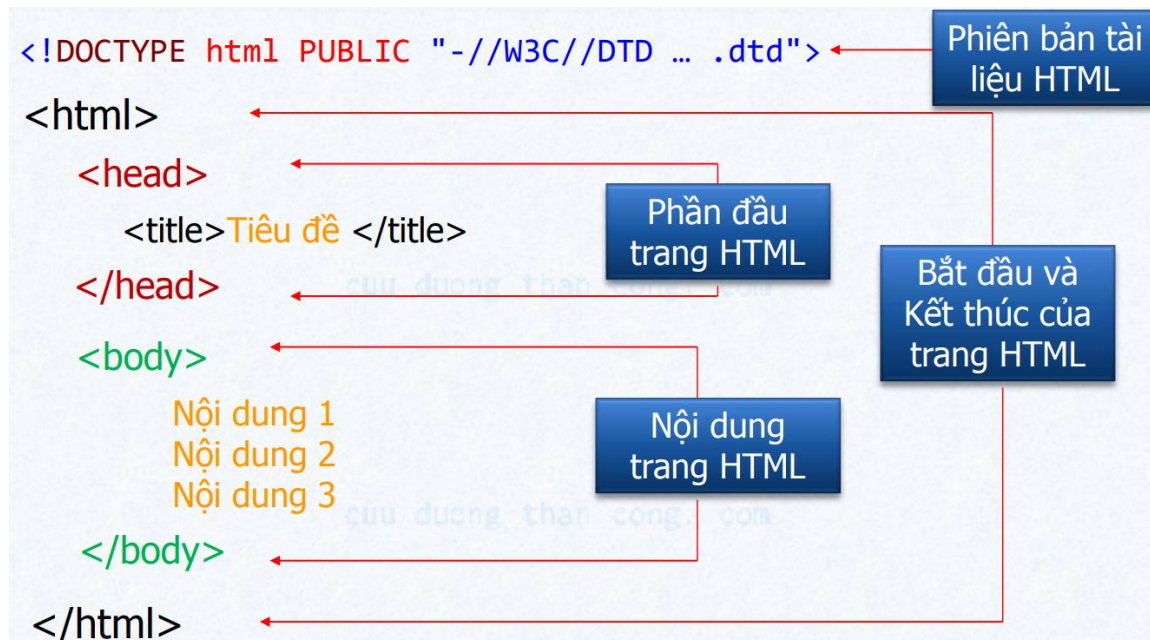
+ First, you have to **declare that the document is an HTML document**. You can do this by using the **<html>** tag.

+ Then there are the two other sections called the **<head>** and **<body>** of your document.

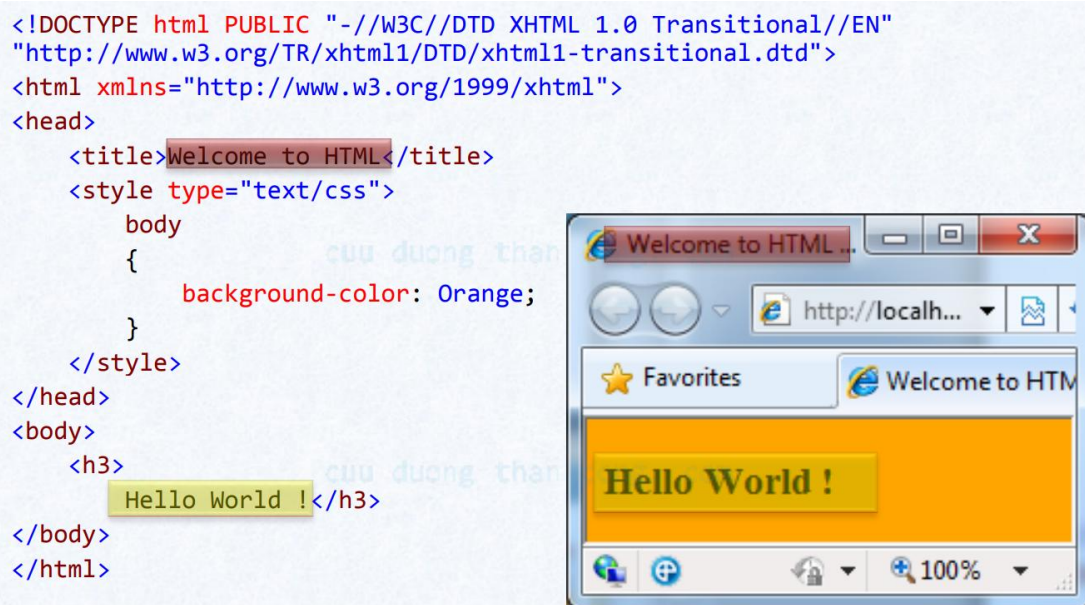
The **<head>** is where you put in all the information you want to include **about your document – là phần đầu tiên trình duyệt đọc trước, chạy trước.**

The **<body>** is basically **the content of your HTML document**. Put in another way, the **<body>** section is **what people see when they view your HTML document**.

So that the basic HTML document would look like this:



- DOCTYPE : Không phải là thẻ HTML, nó giúp trình duyệt biết được phiên bản HTML trang web đang sử dụng
- <html></html> : Định nghĩa phạm vi của văn bản HTML
- <head></head> : Định nghĩa các mô tả về trang HTML. Thông tin trong thẻ này không được hiển thị trên trang web
- <title></title> : Mô tả tiêu đề trang web
- <body></body> : Xác định vùng thân của trang web, nơi chứa các thông tin



- Một thẻ (tag) thường có 3 phần:
 - **Tên của thẻ:** dùng để nhận dạng chức năng của thẻ (FONT)
 - **Thuộc tính của thẻ:** dùng để nhận biết dữ liệu được hiển thị như thế nào (FACE, SIZE).
 - **Giá trị của thuộc tính** thẻ ("Arial", "4")
- Ví dụ:

Mã HTML	Hiển thị trên trình duyệt
<code> This is a webpage </code>	This is a webpage

c) Thẻ *html*

```
<html>

</html>
```

- Bao gồm một thẻ mở (không có /) & 1 thẻ đóng (có /).
- tất cả các thẻ hiển thị trên trình duyệt đều phải nằm trong 2 thẻ này.

d) Thẻ *head & body*

- Nằm trong thẻ html.
- **Tất cả thông tin của web** nằm trong thẻ head.
- **Tất cả nội dung của web** nằm trong thẻ body.

```
<html>
  <head>

  </head>
  <body>

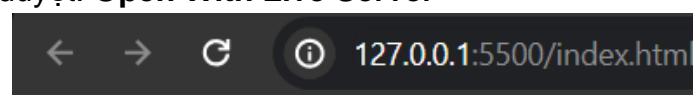
  </body>
</html>
```

Ví dụ:

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

  </head>
  <body>
    Hello World!
  </body>
</html>
```

- Để hiển thị trên trình duyệt: **Open With Live Server**



Hello World!

- Nếu có chỉnh sửa gì trong thẻ body, khi nhấn Ctrl + S (lưu) thì trình duyệt tự động cập nhật lại nội dung hiển thị trên web.

- Để không cần phải viết từng thẻ, trong VSC ta có thể nhấn **!** rồi nhấn Enter, VSC tự tạo như sau:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

e) Thẻ heading

- + **<h1> </h1>** Là thẻ tiêu đề lớn nhất trong file html.
- + Ngoài ra còn có các tiêu đề khác **<h2> <h3> <h4> <h5> <h6>**
- + Từ **<h7>** trở đi văn bản sẽ không được định dạng tiêu đề nữa.

- Mục đích

- Định dạng dòng văn bản theo phân cấp các đề mục. Kích thước của ký tự nhỏ dần từ 1 đến 6
- Sau mỗi thẻ, **văn bản tự động xuống dòng**
- Thuộc tính: **align** = "cách căn chỉnh lề": left, right, center, justify
- Tiêu đề cũng đóng vai trò quan trọng trong việc tìm kiếm: **Các công cụ tìm kiếm sẽ dò tìm từ các thẻ <h*> này trước.**

Ví dụ:

<!DOCTYPE html>	
<html>	
<head>	Hello World!
</head>	Hello
<body>	Hello
<h1>Hello World!</h1>	Hello
<h2>Hello</h2>	Hello
<h3>Hello</h3>	Hello
<h4>Hello</h4>	Hello
<h5>Hello</h5>	Hello
<h6>Hello</h6>	Hello Hello
<h7>Hello</h7>	
<h8>Hello</h8>	
</body>	
</html>	

f) Thẻ paragraph <p> </p>

- Dùng để khai báo một đoạn văn bản, chèn một ký tự xuống dòng và một dòng trống.
- Thuộc tính: **align**="cách căn chỉnh lề": left, right, center, justify
- Nếu chưa có đoạn văn, có thể tạo một đoạn văn mặc định bằng cách dùng lorem

```
<h6>Hello</h6>
<p>Paragraph, lorem</p>
</body>
</html>
```

🔑 lorem Emmet Abbreviation

- Lúc này, khi nhấn tab, xuất hiện một đoạn văn mặc định.
- Đối với thẻ này trình duyệt tự động thêm một khoảng trắng cách dòng ở trên và dưới đoạn văn.

Ví dụ:

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

</head>
<body>
  <h1>Hello World!</h1>
  <h2>Hello</h2>
  <h3>Hello</h3>
  <h4>Hello</h4>
  <h5>Hello</h5>
  <h6>Hello</h6>
  <p>Paragraph, Lorem ipsum dolor sit amet consectetur adipisicing elit.
Repudiandae ex odit, dolorem dolores rem, quod deleniti, modi iusto incidunt illum
repellendus molestias quasi. Aperiam aut assumenda dignissimos consequuntur. Amet,
tempora.</p>
</body>
</html>
```

Hello World!

Hello

Hello

Hello

Hello

Hello

Paragraph, Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae ex odit, dolorem dolores rem, quod deleniti, modi iusto incidunt illum repellendus molestias quasi. Aperiam aut assumenda dignissimos consequuntur. Amet, tempora.

g) Thẻ break line

- Thẻ này không cần thẻ đóng.
- **Công dụng:** Dùng để **xuống dòng** tại bất kỳ vị trí nào trong đoạn.

Ví dụ:

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

  </head>
  <body>
    <h1>Hello World!</h1>
    <h2>Hello</h2>
    <h3>Hello</h3>
    <h4>Hello</h4>
    <h5>Hello</h5>
    <h6>Hello</h6>
    <p>Paragraph, Lorem ipsum dolor sit amet <br> consectetur adipisicing elit.
Repudiandae ex odit, dolorem dolores rem, quod deleniti, modi iusto incidunt illum
repellendus molestias quasi. Aperiam aut assumenda dignissimos consequuntur. Amet,
tempora.</p>
  </body>
</html>
```

Hello World!

Hello

Hello

Hello

Hello

Hello

Paragraph, Lorem ipsum dolor sit amet
consectetur adipisicing elit. Repudiandae ex odit, dolorem dolores rem, quod deleniti, modi iusto incidunt illum repellendus
molestias quasi. Aperiam aut assumenda dignissimos consequuntur. Amet, tempora.

h) Thẻ <hr> (Horizontal rules)

- Chia các đoạn văn thành các section, giữa 2 section **hiển thị một đường thẳng nằm ngang** trên trình duyệt.

- Là một thẻ tự đóng, nên không cần gõ thẻ đóng trong code.

- **Thuộc tính :**

- align : Canh hàng đường kẻ ngang so với trang web
- width : Chiều dài đường kẻ ngang
- size : Bề rộng của đường kẻ ngang
- noshade : Không có bóng

```
<HR noshade size='5' align='center' width='40%'></HR>
```

```
<HR size='15' align='right' width='80%'></HR>
```

Ví dụ:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
<h1>Hello World!</h1>
```

```
<h2>Hello</h2>
```

```
<h3>Hello</h3>
```

```
<h4>Hello</h4>
```

```
<h5>Hello</h5>
```

```
<h6>Hello</h6>
```

```
<p>Paragraph, Lorem ipsum dolor sit amet <br> consectetur adipisicing elit.
```

```
Repudiandae ex odit, dolorem dolores rem, quod deleniti, modi iusto incidunt illum repellendus molestias quasi. Aperiam aut assumenda dignissimos consequuntur. Amet, tempora.</p>
```

```
<hr>
```

```
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Incidunt dignissimos, voluptate, illo eum corporis, nam facilis ea culpa magnam assumenda dolorem minima ipsum. Error deserunt odit officia esse nisi in!</p>
```

```
</body>
```

```
</html>
```

Hello World!

Hello

Hello

Hello

Hello

Hello

Paragraph, Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae ex odit, dolorem dolores rem, quod deleniti, modi iusto incidunt illum repellendus molestias quasi. Aperiam aut assumenda dignissimos consequuntur. Amet, tempora.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Incidunt dignissimos, voluptate, illo eum corporis, nam facilis ea culpa magnam assumenda dolorem minima ipsum. Error deserunt odit officia esse nisi in!

i) Định dạng văn bản

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<p>Đây là văn bản bình thường</p>

<p>Đây là văn bản in đậm</p>

<p>Đây là văn bản <i>in nghiêng</i></p>

<p>Đây là văn bản <big>lớn</big></p>

<p>Đây là văn bản <small>nhỏ</small></p>

<p>Đây là văn bản _{subscript}</p>

<p>Đây là văn bản <ins>inserted</ins></p>

<p>Đây là văn bản deleted</p>

<p>Đây là văn bản <mark>marked</mark></p>

</body>

</html>

Đây là văn bản bình thường

Đây là văn bản **in đậm**

Đây là văn bản *in nghiêng*

Đây là văn bản lớn

Đây là văn bản nhỏ

Đây là văn bản subscript

Đây là văn bản inserted

Đây là văn bản ~~deleted~~

Đây là văn bản **marked**

Thẻ mở	Thẻ đóng	Mục đích
		In đậm
<I>	</I>	In nghiêng
<U>	</U>	Gạch dưới
<TT>	</TT>	Chữ có độ rộng cố định
_		Subscript, VD: số 2 trong H ₂ O
[]	Superscript VD: số 2 trong X ²
		<p>Định dạng kích thước, màu sắc, kiểu chữ,....</p> <p>Cú pháp như sau:</p> <p>... </p> <p>VD: </p> <p><i>Không nên sử dụng thẻ này hiện tại vì nó không áp dụng được nhiều thuộc tính.</i></p>

Thẻ	Mục đích	Ví dụ	Kết quả hiển thị
	Làm đậm chữ nhưng không mang ý nghĩa nhấn mạnh.	This text is bold	This text is bold
	Nhấn mạnh văn bản với ý nghĩa quan trọng hơn .	This text is strong 	This text is strong
<big>	Tăng kích thước chữ lên so với mặc định.	<big>This text is big </big>	This text is big

	Nhấn mạnh văn bản, thường hiển thị dưới dạng in nghiêng.	 This text is emphasized 	<i>This text is emphasized</i>
<i>	In nghiêng văn bản nhưng không mang ý nghĩa nhấn mạnh.	<i> This text is italic </i>	<i>This text is italic</i>
<small>	Giảm kích thước chữ xuống so với mặc định.	<small> This text is small </small>	This text is small
<sub>	Chữ số nhỏ, nằm thấp hơn dòng chữ bình thường (chỉ số dưới).	This text contains a <sub>2</sub>	This text contains a ₂
<sup>	Chữ số nhỏ, nằm cao hơn dòng chữ bình thường (chỉ số trên).	This text contains x <sup>2</sup> = a x a	This text contains $x^2 = a \times a$
	Nhấn mạnh văn bản (giống thẻ bên trên).	 Renders as emphasized text 	<i>Renders as emphasized text</i>
<dfn>	Định nghĩa một thuật ngữ (có thể hiển thị in nghiêng tùy trình duyệt).	<DFN> Defines a definition term </DFN>	<i>Defines a definition term</i>
<code>	Hiển thị văn bản dưới dạng mã nguồn máy tính (font chữ monospaced).	<CODE> Defines computer code text </CODE>	Defines computer code text
<kbd>	Định dạng đầu vào bàn phím (thường dùng cho phím tắt).	<KBD> Defines keyboard text </KBD>	Defines keyboard text
<var>	Biểu thị một biến trong công thức hoặc chương trình.	<VAR> Defines a variable part of a text </VAR>	Defines a variable part of a text
<cite>	Định nghĩa một trích dẫn hoặc tham chiếu đến một nguồn.	<CITE> Defines a citation </CITE>	Defines a citation
<blink>	(Không còn được hỗ trợ) Làm cho chữ nhấp nháy.	<BLINK> Computer Sciences </BLINK>	<i>Không hoạt động trong trình duyệt hiện đại</i>
	Hiển thị chữ bị gạch ngang, biểu thị nội dung bị xóa.	 Computer Sciences 	Computer Sciences
<ins>	Hiển thị chữ có gạch dưới, biểu thị nội dung được thêm vào.	<INS> Computer Sciences </INS>	Computer Sciences
<samp>	Biểu diễn đầu ra của chương trình hoặc hệ thống máy tính.	<SAMP> Defines sample computer code </SAMP>	Defines sample computer code

Lưu ý

- **<blink>** đã bị loại bỏ trong HTML5 và không hoạt động trong các trình duyệt hiện đại.
- **** và **<ins>** thường được sử dụng để theo dõi sự thay đổi trong nội dung văn bản.
- **<kbd>**, **<code>**, **<samp>** và **<var>** thường dùng trong tài liệu lập trình và hướng dẫn kỹ thuật.

j) Thẻ Chú thích (Comment)

- Bao quanh giữa cặp dấu **<!-- Chú thích -->**

k) Liên kết (link) trang

- Đặc điểm của HTML có thể **liên kết các trang khác nhau hoặc các phần nằm trong cùng trang lại với nhau** bằng cách dùng **tag <A> và **
- Ta gọi:
 - Đối tượng sử dụng để **kích chuột vào** là: **Đối tượng liên kết**. Đối tượng có thể là: văn bản, hình ảnh. (**những thứ nằm giữa thẻ <A> **).
 - Địa chỉ nội dung sẽ được hiện ra là **Đích liên kết**
- Các thuộc tính:
 - **href** = “đích liên kết”: Nếu trong cùng web nên sử dụng đường dẫn **tương đối**.
 - **target** = “tên **cửa sổ** đích”. (thiết lập vị trí mở đối tượng liên kết) Có một số tên đặc biệt:
 - **_self**: mở liên kết ở cửa sổ/ tab hiện tại (**mặc định**)
 - **_blank**: mở liên kết ở cửa sổ/ tab mới
 - **_parent**: tải trang web vào cửa sổ cha của nó
 - **_top**: tải trang web vào cửa sổ cao nhất

```
<a href="https://www.google.com/" target = "_self">  
    Google  
</a>  
<a href="https://www.google.com/" target = "_blank">  
    Google  
</a>
```

○ Liên kết trang được thực hiện bằng tag <A> với nhiều cấp độ khác nhau:

▪ Liên kết với một phần khác nằm trong (thuộc) cùng trang hiện hành:

- Đặt tên ở đầu của các phần sẽ được liên kết đến:

```
<A Name = “Giới thiệu”><H1>1. Giới thiệu</H1></A>
```

“Giới thiệu” được gọi là Tên điểm neo.

- Ở mỗi điểm liên kết được đặt bằng từ khoá:

Hãy đọc trước phần giới thiệu để biết nội dung môn học.

▪ Liên kết với một phần nằm trong (thuộc) trang khác (được lưu trữ cùng thư mục với trang hiện hành):

- Đặt tên ở đầu của các phần sẽ được liên kết đến:

```
<A Name=“Giới thiệu”><H1>1. Giới thiệu </H1></A>
```

- Trong trang HTML khác ở điểm liên kết được đặt bằng từ khoá:

Hãy đọc trước phần giới thiệu để biết nội dung môn học.

- Liên kết đến địa chỉ website hay một trang cụ thể trên website (ta chỉ cần ghi trực tiếp địa chỉ của trang cần liên kết đến)

<P><A HREF="<http://www.vnuhcm.edu.vn>"> Trang web Đại học Quốc Gia TP. HCM </P>

<P><A HREF="<http://www.yahoo.com>"> Yahoo </P>

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
</head>
```

```
  <body>
```

```
    <a href="https://www.google.com/">
```

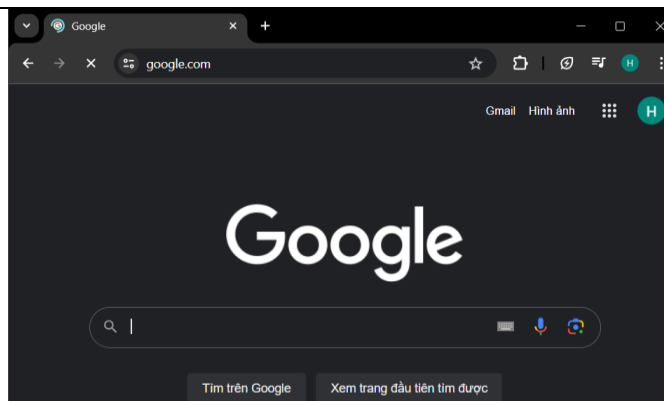
```
      Google
```

```
    </a>
```

```
  </body>
```

```
</html>
```

[Google](#)



- Liên kết đến địa chỉ Email:

Các trang web liên kết:

<P><A HREF="<mailto:htthanh@vnuhcm.edu.vn>"> Hồ Trung Thành </P>

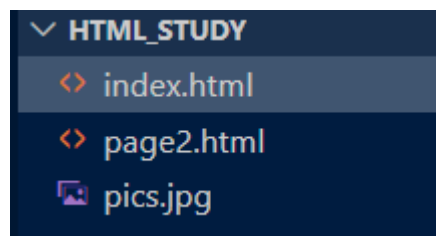
l) Thẻ image

- Dùng để chèn hình ảnh vào trang web.

- Cú pháp: ****

- Lưu ý: thẻ này không có thẻ đóng.
- Các thuộc tính của thẻ:
 - **SRC**: qui định tên tập tin ảnh (*link ảnh*)
 - **HEIGHT**: chiều cao của ảnh
 - **WIDTH**: chiều rộng của ảnh
 - **Chỉ cần chèn 1 thuộc tính để giữ nguyên tỉ lệ tấm ảnh**
 - **Nếu không có 2 thuộc tính này thì tấm ảnh có kích thước bao nhiêu sẽ giữ nguyên.**
 - **ALT**: qui định một chuỗi ký tự thay thế cho ảnh trong trường hợp ảnh không hiển thị ra
 - Search engine sẽ tìm dựa theo thuộc tính này để tìm kiếm hình ảnh, nhanh hơn tìm bằng hình ảnh.
 - **ALIGN**: canh chỉnh left, right, center **trên cùng dòng văn bản đang chứa ảnh.**
 - **Khác với canh chỉnh tấm hình so với trang web** (nếu muốn canh chỉnh hình ảnh so với trang web, ta lợi dụng thẻ <p> để đưa hình ảnh vào và dùng thuộc tính align).
 - **Position**: Top, Bottom, Middle
 - **Border**: Độ dày nét viền quanh ảnh (default=0)

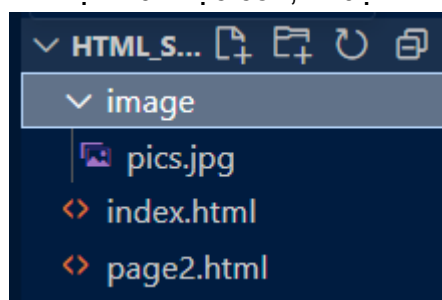
- Giả sử thư mục có chứa ảnh như sau:



- Để lấy được hình, câu lệnh được viết như sau:

```
<body>
  <img src = "pics.jpg">
</body>
```

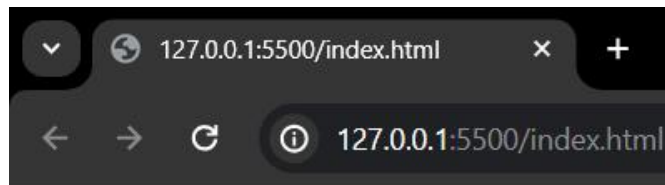
- Giả sử file ảnh này được lưu vào một thư mục con, ví dụ:



- Thì lúc này đoạn code trên không còn xuất được hình ra nữa (bị lỗi), điều cần làm là viết thêm đường dẫn của tệp ảnh vào trong thẻ img:

```
<body>
  <img src = "/image/pics.jpg">
</body>
```

Kết quả:



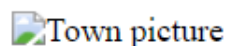
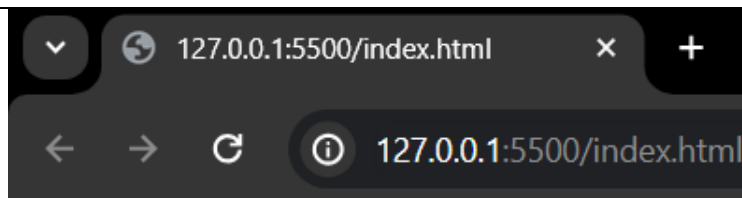
- Để chỉnh kích thước của ảnh:

```
<body>
  <img src = "/image/pics.jpg" height = 300px width = 300px>
</body>
```

Thông thường để ảnh không bị méo, móp thì chỉ cần để một thuộc tính height hoặc width, thuộc tính còn lại máy sẽ tự điều chỉnh cho cân đối.

- Để thêm chú thích của ảnh **khi ảnh bị lỗi**, ta thêm một thuộc tính là alt:

```
<body>
  <img src = "/image/pics.jp" height = 300px alt = "Town picture">
</body>
```



- Khi muốn một pop up hiện lên khi đưa trỏ chuột vào bất kỳ vị trí nào trên bức hình, ta thêm thuộc tính title:

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

  </head>
  <body>
    <img src = "/image/pics.jpg" height = 300px alt = "Town picture" title = "Thành phố">
  </body>
</html>
```



- Có thể dùng hình ảnh để chuyển hướng sang trang khác thông qua thẻ <a>

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

  </head>
  <body>
    <a href = "https://sv.ut.edu.vn/">
      <img src = "/image/pics.jpg" height = 300px alt = "Town picture" title = "Thành phố">
    </a>
  </body>
</html>
```

* **Đặt ảnh nền cho trang web:** Sử dụng thẻ <body Background='ImagePath'>

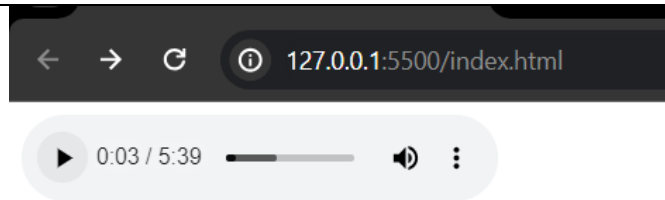
Thẻ HTML	Ý nghĩa
	Định nghĩa một hình ảnh trong tài liệu HTML
<map>	Định nghĩa một <u>ảnh bản đồ</u> trong tài liệu HTML
<area>	Định nghĩa vùng click chuột cho ảnh bản đồ

m) Thẻ audio

- Dùng để phát các tệp âm thanh.
- Sử dụng tương tự như thẻ hình ảnh (image)
- Có thể cần hoặc không cần thẻ đóng cũng được (do dùng thẻ đóng tự động)

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

  </head>
  <body>
    <audio controls src = "listening.mp3"></audio>
  </body>
</html>
```



- Thuộc tính controls để hiển thị bảng điều khiển trên giao diện.
- Để mỗi lần tải lại trang thì file âm thanh này được chạy tự động, ta dùng thêm autoplay.
- Để âm thanh đang chạy mà bị tắt đi, dùng muted.
- Khi phát hết tệp âm thanh mà muốn nó phát lại, dùng loop.

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

  </head>
  <body>
    <audio loop autoplay muted controls src = "listening.mp3"></audio>
  </body>
</html>
```

- Tạo tệp âm thanh dự phòng:

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

  </head>
  <body>
    <audio loop autoplay muted controls>
      <source src = "listening.mp3" type = "audio/mpeg">
      <source src = "listening.wav" type = "audio/mpeg">
    </audio>
  </body>
</html>
```

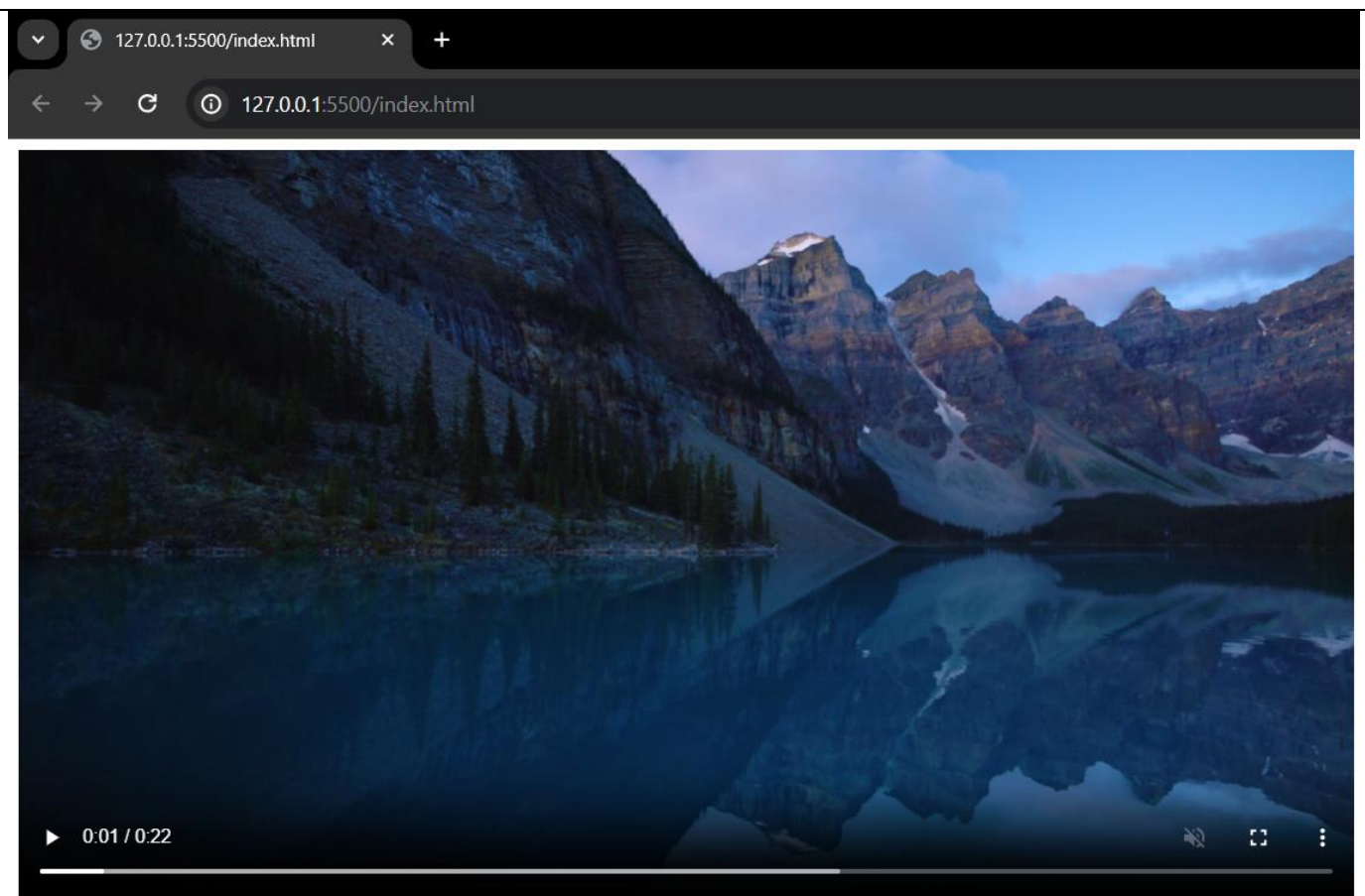
Khi tệp .mp3 lỗi thì nó sẽ chạy tệp .wav bên dưới.

n) Thẻ video

- Thực hiện tương tự như thẻ audio.
- Có thể chỉnh tỷ lệ của video phù hợp với kích thước cửa sổ.

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

  </head>
  <body>
    <video controls muted width= 1000px>
      <source src = "video.mp4">
    </video>
  </body>
</html>
```



- Tạo các sources dự phòng:

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

  </head>
  <body>
    <video controls muted width= 1000px>
      <source src = "video.mp4">
      <source src = "video.webm">
      <source src = "video.ogv">
    </video>
  </body>
</html>
```

o) List (Danh sách)

- HTML hỗ trợ việc trình bày các đề mục với dạng các nút hay đánh số thứ tự

- Có 3 loại:

* Unordered list (Dạng các nút)

- + Làm việc với thẻ ``
- + Đánh dấu đầu dòng bằng các **bullet**.
- + Mỗi mục danh sách bắt đầu bằng ``

```
<h2>Unordered list</h2>
```

```
<ul>
  <li>Toán</li>
  <li>Văn</li>
  <li>Anh</li>
</ul>
```

Unordered list

- Toán
 - Văn
 - Anh
- có thể dùng thuộc tính **Type** để đổi dạng khác
 - Cú pháp: `<UL Type = kiểu danh sách>`

Kiểu danh sách	Kiểu bullet dùng đánh dấu
Square	Bullet là hình vuông đầy.
Circle	Bullet là hình tròn rỗng.
Disc (mặc định)	Bullet là hình tròn đầy.

* Ordered list (Dạng đánh số thứ tự)

- + Làm việc với thẻ ``
- + Đánh dấu đầu dòng bằng **số thứ tự**.
- + Mỗi mục danh sách bắt đầu bằng ``

```
<h2>Ordered list</h2>
```

```
<ol>
  <li>Toán</li>
  <li>Văn</li>
  <li>Anh</li>
</ol>
```

Ordered list

1. Toán
2. Văn
3. Anh

* Có thể thay đổi số thứ tự bằng kiểu khác, sử dụng type: (Cú pháp: `<OL Type = kiểu đánh số>`)

```
<ol type = "">
  <li>Toán 1
  <li>Văn A
  <li>Anh a
</ol>
I
i
```

Kiểu đánh số	Kiểu ký tự dùng để đánh số thứ tự
I	Dùng chữ số la mã I, II, III, ...
i (mặc định)	Dùng chữ số thường 1,2,3,...
A	Dùng ký tự A, B, C, ...
a	Dùng ký tự a,b,c, ...

* Description list (danh sách các đề mục):

- Danh sách đề mục có diễn giải **bắt đầu là <DL> kết thúc </DL>** trong đó mỗi mục trong danh sách **<DT> là từ khoá, <DD> là diễn giải.**

```
<h2>Discription list</h2>
<dl>Danh sách môn học</dl>
<dd>Toán, Văn, Anh</dd>
```

Discription list

Danh sách môn học

Toán, Văn, Anh

```
<body>
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
</body>
```

```
Coffee
  - black hot drink
Milk
  - white cold drink
```

* Nested list (lồng nhau)

```
<h2>Unordered list</h2>
<ul>
  <li>Toán</li>
  <li>Văn
    <ul>
      <li>Văn xuôi</li>
      <li>Văn nghị luận</li>
    </ul>
  </li>
  <li>Anh</li>
</ul>
```

Unordered list

- Toán
- Văn
 - Văn xuôi
 - Văn nghị luận
- Anh

Các kiểu list khác tương tự.

p) Table (Bảng)

- Sử dụng thẻ `<table>`/`</table>` để tạo một bảng
- Các dòng trong bảng: `<tr>`/`</tr>` (**table row**), mỗi thẻ `<tr>` là một dòng của bảng
 - + Dòng tiêu đề của bảng: `<th>`/`</th>` (**table header**) (*ND tự in đậm, canh giữa*)
 - + Nội dung từng dòng của bảng (**tạo ô**): `<td>`/`</td>` (**table data**)
- **Tổng số thẻ `<td>` và `<th>` bằng số ô của bảng.** Dòng có bao nhiêu ô thì có bấy nhiêu thẻ `<td>` và/hoặc `<th>` *nằm trong cặp thẻ `<tr>...</tr>` tương ứng*
 - Chú ý: Để có được **một ô trống trong bảng** (ô không có dữ liệu) thì cần đặt nội dung ô là: **` `**; (**Thẻ khoảng trắng**)

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

</head>
  <body>
    <table>
      <tr>
        <th>STT</th>
        <th>Họ & Tên</th>
        <th>Ngành</th>
      </tr>

      <tr>
        <td>001</td>
        <td>Nguyễn Văn A</td>
        <td>Information Technology</td>
      </tr>
    </table>
  </body>
</html>
```

STT	Họ & Tên	Ngành
001	Nguyễn Văn A	Information Technology

- + Để đổi màu cho một dòng/ tô viền cho bảng, dùng thuộc tính **bgcolor** (background color)

```
<table bgcolor = "black">
  <tr bgcolor = "grey">
    <th>STT</th>
    <th>Họ & Tên</th>
    <th>Ngành</th>
  </tr>

  <tr bgcolor = "lightgrey">
    <td>001</td>
    <td>Nguyễn Văn A</td>
    <td>Information Technology</td>
  </tr>
```

STT	Họ & Tên	Ngành
001	Nguyễn Văn A	Information Technology

● Các thuộc tính:

- **<caption>**: định nghĩa tiêu đề cho bảng
- **BORDER**="số": kích thước đường viền. Đặt bằng 0 (mặc định): không có đường viền.

```
<table border="5px">
  <caption>Tiêu đề của bảng</caption>
  <tr>
    <th>Cột 1</th>
    <th>Cột 2</th>
  </tr>
  <tr>
    <td>Dữ liệu 1</td>
    <td>Dữ liệu 2</td>
  </tr>
</table>
```

Tiêu đề của bảng

Cột 1	Cột 2
Dữ liệu 1	Dữ liệu 2

- **WIDTH**="rộng", **HEIGHT**="cao": độ rộng và độ cao của bảng. Có thể đặt theo 2 cách:
 - n: (n là số) Quy định độ rộng, cao là n pixels
 - n%: Quy định độ rộng, cao là n% độ rộng, cao của đối tượng (element) chứa bảng. (Kích thước tương đối)
 - Lưu ý: Dữ liệu nhiều lên thì bảng giãn ra, dù cho ta cố định kích thước.
- **CELLSPACING**="số": Khoảng cách giữa 2 ô liên tiếp
- **CELLPADDING**="số": Khoảng cách từ các cạnh của ô đến nội dung ô
- **BGCOLOR**="màu": màu nền của bảng
- **BACKGROUND**="địa_chỉ_ảnh": Địa chỉ của file ảnh làm nền cho bảng. Nên sử dụng đường dẫn tương đối nếu có thể.

● Thuộc tính rowspan.

- Thuộc tính **rowspan** dùng để **gộp 2 hoặc nhiều hàng** vào làm một.
- Cú Pháp: **rowspan="n"**. Trong đó n là số lượng hàng muốn gộp.

Ví dụ:

```
<table border="1" cellpadding="2" cellspacing="2">
  <thead>
    <tr>
      <th>Tiêu đề cột 1 hàng 1</th>
      <th>Tiêu đề cột 2 hàng 1</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="2">Phần thân cột 1 hàng 2</td>
      <td>Phần thân cột 2 hàng 2</td>
    </tr>
    <tr>
```

```

        <td>Phần thân cột 2 hàng 3</td>
    </tr>
</tbody>
</table>

```

Tiêu đề cột 1 hàng 1	Tiêu đề cột 2 hàng 1
Phần thân cột 1 hàng 2	Phần thân cột 2 hàng 2
	Phần thân cột 2 hàng 3

- **Thuộc tính colspan.**

- Thuộc tính **colspan** dùng để **gộp 2 hoặc nhiều cột** vào làm một.
- Cú Pháp: **colspan="n"**. Trong đó n là số lượng cột muốn gộp.

Ví dụ:

```

<table border="1" cellpadding="2" cellspacing="2">
    <thead>
        <tr>
            <th>Tiêu đề cột 1 hàng 1</th>
            <th>Tiêu đề cột 2 hàng 1</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td colspan="2">Phần thân cột 1 hàng 2</td>
        </tr>
        <tr>
            <td>Phần thân cột 1 hàng 3</td>
            <td>Phần thân cột 2 hàng 3</td>
        </tr>
    </tbody>
</table>

```

Tiêu đề cột 1 hàng 1	Tiêu đề cột 2 hàng 1
Phần thân cột 1 hàng 2	
Phần thân cột 1 hàng 3	Phần thân cột 2 hàng 3

- Khi cấu trúc bảng có hơi phức tạp và nhiều thông tin thì bạn có thể sử dụng các thuộc tính **<thead>**, **<tbody>** và **<tfoot>** để đánh dấu phần đầu của cột, phần cuối cũng như nội dung chính cho bảng.
- Bên cạnh đó, <thead> còn có chức năng giúp quản lý thông tin trong bảng một cách dễ dàng hơn, cố định phần tiêu đề khi bảng có chứa dữ liệu lớn.
- Hay <tbody> giúp tự động tạo scroll, tách biệt với phần <thead> để dễ dàng tiếp cận thông tin và <tfoot> khi số lượng dòng vượt quá chiều cao của bảng.
- Để sử dụng các thuộc tính này trong table HTML, bạn có thể tham khảo cách như sau:

- **<thead>** có nhiệm vụ gom nhóm tiêu đề bảng, thường sẽ là hàng đầu tiên. Nếu bạn có đang sử dụng thuộc tính **<col>** hoặc **<colgroup>** thì **<thead>** sẽ ở ngay bên dưới thuộc tính này.
- Đối với **<tfoot>** sẽ bao gồm các nội dung ở phía cuối của bảng, thường sẽ là các cột mang nội dung thành tiền hoặc là đếm tổng.
- **<tbody>** sẽ bao bọc các phần nội dung của bảng, sẽ xuất hiện ở ngay bên dưới nội dung của **<thead>**.

Lưu ý:

- ✎ Trong code table HTML dù bạn có thay đổi vị trí của các thẻ **<tbody>** hoặc **<thead>** (ví dụ bạn viết **<tbody>** trên cùng, còn **<thead>** ở giữa) thì khi hiển thị trên trang web, nội dung của thẻ **<tbody>** vẫn sẽ nằm đúng ở vị trí ngay giữa.
- ✎ **Điều này nghĩa là nếu bạn đặt các nhóm này ở đâu thì khi hiển thị vẫn theo đúng trình tự <thead>, <tbody> và <tfoot>.**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Table with Scrollable tbody</title>
</style>
table {
  width: 100%;
  border-collapse: collapse;
}
th, td {
  border: 1px solid black;
  padding: 8px;
  text-align: left;
}
thead, tfoot {
  background-color: #f2f2f2;
}
tbody {
  display: block;
  height: 150px; /* Giới hạn chiều cao của tbody */
  overflow-y: auto; /* Tạo thanh cuộn dọc */
  width: 100%;
}
thead, tfoot {
  display: table;
  width: 100%;
  table-layout: fixed; /* Đảm bảo các cột khớp nhau */
}
tbody tr {
  display: table;
  width: 100%;
  table-layout: fixed;
}
</style>
```

```
</head>
<body>

<table>
  <thead>
    <tr>
      <th>Header 1</th>
      <th>Header 2</th>
      <th>Header 3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Row 1 - Cell 1</td>
      <td>Row 1 - Cell 2</td>
      <td>Row 1 - Cell 3</td>
    </tr>
    <tr>
      <td>Row 2 - Cell 1</td>
      <td>Row 2 - Cell 2</td>
      <td>Row 2 - Cell 3</td>
    </tr>
    <tr>
      <td>Row 3 - Cell 1</td>
      <td>Row 3 - Cell 2</td>
      <td>Row 3 - Cell 3</td>
    </tr>
    <tr>
      <td>Row 4 - Cell 1</td>
      <td>Row 4 - Cell 2</td>
      <td>Row 4 - Cell 3</td>
    </tr>
    <tr>
      <td>Row 5 - Cell 1</td>
      <td>Row 5 - Cell 2</td>
      <td>Row 5 - Cell 3</td>
    </tr>
    <tr>
      <td>Row 6 - Cell 1</td>
      <td>Row 6 - Cell 2</td>
      <td>Row 6 - Cell 3</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Footer 1</td>
      <td>Footer 2</td>
      <td>Footer 3</td>
    </tr>
  </tfoot>
</table>
```

</table>

</body>

</html>

Header 1	Header 2	Header 3
Row 3 - Cell 1	Row 3 - Cell 2	Row 3 - Cell 3
Row 4 - Cell 1	Row 4 - Cell 2	Row 4 - Cell 3
Row 5 - Cell 1	Row 5 - Cell 2	Row 5 - Cell 3
Row 6 - Cell 1	Row 6 - Cell 2	Row 6 - Cell 3
Footer 1	Footer 2	Footer 3

- ✎ **CSS cho <tbody>**: Được đặt display: block; để giới hạn chiều cao và tạo thanh cuộn khi nội dung vượt quá giới hạn đó (height: 150px;).
- ✎ **CSS cho <thead> và <tfoot>**: Được giữ nguyên với display: table; để đảm bảo chúng luôn hiển thị cố định ở trên và dưới bảng.
- ✎ **border-collapse: collapse;** là một thuộc tính trong CSS được sử dụng để **loại bỏ khoảng trống giữa các ô**, các đường viền của các ô liền kề sẽ "chồng lên nhau" (collapsing borders), tạo thành một đường viền chung duy nhất.
 - Thông thường, nếu không sử dụng thuộc tính này, mỗi ô của bảng sẽ có một đường viền riêng, tạo ra khoảng cách giữa các ô.
- ✎ **border-collapse: separate;** (mặc định): Các ô sẽ có đường viền riêng và có khoảng cách giữa các đường viền.

• Tag <colgroup>

- được sử dụng để định dạng cho một nhóm cột trong <table>
- hữu ích trong việc định dạng chung cho toàn bộ cột, thay vì định dạng riêng, lặp lại cho mỗi hàng.
- **Note:** The <colgroup> tag must be a **child of a <table> element**, **after any <caption> elements** and **before any <thead>, <tbody>, <tfoot>, and <tr> elements**.
- Được dùng chung với thẻ <col>

Attribute	Value	Description
span	number	Specifies the number of columns a column group should span (số lượng cột cần định dạng)

```

<head>

</head>

<body>
  <table>
    <colgroup>
      <col span="2" style="background-color:red">
      <col style="background-color:yellow">
    </colgroup>
    <tr>
      <th>ISBN</th>
      <th>Title</th>
      <th>Price</th>
    </tr>
    <tr>
      <td>3476896</td>
      <td>My first HTML</td>
      <td>$53</td>
    </tr>
  </table>

</body>

```

ISBN	Title	Price
3476896	My first HTML	\$53

- **The <col> tag**

- specifies column properties for each column within a [<colgroup>](#) element.
- is useful for applying styles to entire columns, instead of repeating the styles for each cell, for each row.

Attribute	Value	Description
span	<i>number</i>	Specifies the number of columns a <col> element should span

q) Thẻ span và thẻ div

- div gom phần tử theo khối (block), nhóm các phần tử thành một khối; **có \n – không cách dòng như <p>.**

- span gom các phần tử theo dòng (inline), nhóm các phần tử thành một dòng; **(không \n)**
Định dạng với một từ/cụm từ trong văn bản.

- Công dụng: có thể **định dạng 1 đoạn văn bản trong một văn bản** mà không ảnh hưởng đến các nội dung khác trong văn bản.

Ví dụ:

```
<body>
  <p><span>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dolores
perspiciatis aut deserunt quis et sit officiis molestias dolor quidem obcaecati cum reprehenderit,
minima corporis est magni a ut neque tempora.</p>

  <p><div>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vero esse
necessitatibus consectetur obcaecati est libero corporis iure? Dolore dolorem provident illo,
optio maxime sunt. Consequuntur sapiente aliquid facere quo maxime.</p>
</body>
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dolores perspiciatis aut deserunt quis et sit officiis molestias dolor quidem obcaecati cum reprehenderit, minima corporis est magni a ut neque tempora.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vero esse necessitatibus consectetur obcaecati est libero corporis iure? Dolore dolorem provident illo, optio maxime sunt. Consequuntur sapiente aliquid facere quo maxime.

```
<body>
  <p><span style = "color: red">Lorem ipsum dolor sit amet, consectetur
adipisicing</span> elit. Dolores perspiciatis aut deserunt quis et sit officiis molestias dolor
quidem obcaecati cum reprehenderit, minima corporis est magni a ut neque tempora.</p>

  <p><div style = "background-color: yellow;">Lorem ipsum dolor sit amet, consectetur
adipisicing</div> elit. Vero esse necessitatibus consectetur obcaecati est libero corporis iure?
Dolore dolorem provident illo, optio maxime sunt. Consequuntur sapiente aliquid facere quo
maxime.</p>
</body>
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Dolores perspiciatis aut deserunt quis et sit officiis molestias dolor quidem obcaecati cum reprehenderit, minima corporis est magni a ut neque tempora.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Vero esse necessitatibus consectetur obcaecati est libero corporis iure? Dolore dolorem provident illo, optio maxime sunt. Consequuntur sapiente aliquid facere quo maxime.

r) Thẻ meta & title

- Thường đặt trong **thẻ head** (phần đầu của một trang web).
- **Thẻ title:** Dùng để ghi tên của trang web trên thanh tab.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

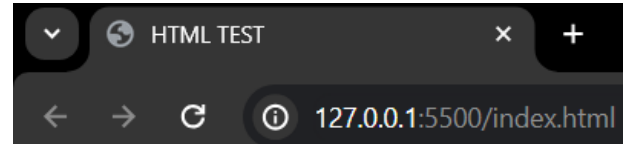
```
    <title>HTML TEST</title>
```

```
  </head>
```

```
  <body>
```

```
    </body>
```

```
</html>
```



- **Thẻ meta:** Dùng để **thêm các chi tiết của webpage** khi được tìm kiếm trên một trình duyệt nào đó

Gives the browser some metadata (siêu dữ liệu: dạng dữ liệu mô tả thông tin chi tiết về dữ liệu, chẳng hạn thuộc tính, đặc điểm và các phần tử khác) about the document.

The **charset** attribute *declares the character encoding*.

Modern HTML documents **should always use UTF-8**, even though it is not a requirement. In HTML, the `<meta>` tag **does not require a closing tag**.

<https://seongon.com/blog/seo/the-meta-can-thiet-cho-seo.html>

<meta name="title" content="mô tả" />

Thẻ Meta Title khi tìm kiếm trên Google nó chỉ hiển thị 65-70 ký tự, trong trường hợp title của bạn >70 ký tự thì cũng không nên cut (...) tiêu đề đó. Chúng ta có thể chấp nhận độ dài 65-100 ký tự để hiển thị đầy đủ tiêu đề có ý nghĩa.

<meta name="description" content="mô tả" />

Meta Description là thẻ mô tả **tóm tắt nội dung của website**, nội dung mô tả này được google hiển thị khi tìm kiếm là khoảng 160 ký tự, nhưng bạn cũng không nên cut (...) mô tả nếu nó dài hơn 160 ký tự. Chúng ta có thể chấp nhận độ dài trong khoảng 160-250 ký tự để hiển thị đầy đủ mô tả.

<meta name="viewport" content="

Thẻ này thông báo cho trình duyệt biết **cách hiển thị một trang Web trên thiết bị di động**. Bạn có thể dùng thẻ này khi muốn Google biết rằng nó thân thiện với thiết bị di động.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>HTML TEST</title>
```

```
    <meta charset = "utf-8">
```

```
    <meta name = "Home discription" content = "Home">
```

```
    <meta name = "keywords" content="Home HTML CSS Tutorial"> <!--trình duyệt dựa
```

vào các từ khóa để đề xuất cho người dùng khi người dùng tìm kiếm-->

```
    <meta name = "author" content = "Văn A"> <!--Tên tác giả-->
```

```
    <meta name = "viewport" content = "width=device-width, initial-scale=1.0">
```

```
    <meta http-equiv="refresh" content = "2"> <!--sau mỗi 2s trang tải lại-->
```

```
  </head>
```

```
  <body>
```

```
    </body>
```

```
</html>
```

s) Thẻ **frame** & **iframe**

- Cho phép chia một trang web làm nhiều phần, mỗi phần dùng để nhúng một tài liệu nào đó (một trang web khác, tập tin pdf, tấm hình,) vào trang web hiện tại.

- Đối với thẻ **frame**:

* **Thẻ `frameset`**:

Not Supported in HTML5.

- Thẻ **<frameset>** dùng để xác định một bộ khung
- Trong một bộ khung sẽ có một hoặc nhiều khung, **mỗi khung là một phần tử `<frame>`**
- Phần tử **<frameset>** chỉ định trong một bộ khung sẽ có bao nhiêu hàng, bao nhiêu cột, và kích thước của mỗi phần đó.

- Cần thay cặp thẻ **<body>...</body>** bằng **<frameset>...</frameset>** (Nếu không thì sẽ không hiển thị kết quả trên cửa sổ trình duyệt).

Ví dụ 1:

- Phần tử `<frameset>` bên dưới xác định bộ khung sẽ có hai **cột** (tương ứng hai phần tử `<frame>`)
 - o Phần tử `<frame>` thứ nhất có **chiều rộng** bằng 35% chiều rộng của phần tử `<frameset>`
 - o Phần tử `<frame>` thứ hai có **chiều rộng** bằng 65% chiều rộng của phần tử `<frameset>`

```
<frameset cols="35%,65%">
  <frame src="http://webcoban.vn/html/demo?file=1434"/>
  <frame src="http://webcoban.vn/html/demo?file=1435"/>
</frameset>
```

Ví dụ 2:

- Phần tử `<frameset>` bên dưới xác định bộ khung sẽ có ba hàng (tương ứng ba phần tử `<frame>`)
 - o Phần tử `<frame>` thứ nhất có chiều cao bằng 15% chiều cao của phần tử `<frameset>`
 - o Phần tử `<frame>` thứ hai có chiều cao bằng 50% chiều cao của phần tử `<frameset>`
 - o Phần tử `<frame>` thứ ba có chiều cao bằng 35% chiều cao của phần tử `<frameset>`

```
<frameset rows="15%,50%,35%">
  <frame src="http://webcoban.vn/html/demo?file=1434"/>
  <frame src="http://webcoban.vn/html/demo?file=1435"/>
  <frame src="http://webcoban.vn/html/demo?file=1436"/>
</frameset>
```

- Ngoài ra, chúng ta có thể đặt một bộ khung bên trong một bộ khung khác để tăng tính đa dạng của bộ khung.

Ví dụ 3:

```
<frameset rows="50%,50%">
  <frame src="http://webcoban.vn/html/demo?file=1434"/>
  <frameset cols="30%,70%">
    <frame src="http://webcoban.vn/html/demo?file=1435"/>
    <frame src="http://webcoban.vn/html/demo?file=1436"/>
  </frameset>
</frameset>
```



Các thuộc tính của thẻ <frameset>

Cols	Xác định số lượng cột và chiều rộng mỗi cột trong phần tử <frameset>
Rows	Xác định số lượng hàng và chiều cao mỗi hàng trong phần tử <frameset>
noresize	Ngăn chặn hành động kéo thả làm thay đổi kích thước của các phần tử <frame> bên trong phần tử <frameset>
frameborder	Đường viền cho frame, có giá trị “yes” hoặc “no” (hoặc Nó nhận giá trị 1 hoặc 0)
framespacing	= “n” (px): Khoảng cách giữa 2 khung tính bằng pixels

• Thuộc tính cols

- Xác định **số lượng cột và chiều rộng mỗi cột** trong phần tử <frameset> (Thuộc tính **cols** có bao nhiêu giá trị thì sẽ tương ứng với bấy nhiêu cột, và mỗi giá trị đó cũng xác định chiều rộng của mỗi cột. Lưu ý, các giá trị được ngăn cách bởi dấu phẩy)
- Giá trị của thuộc tính cols được xác định bởi một trong ba đơn vị: px, %, * (tức là bằng chiều rộng của phần tử <frameset> trừ đi tổng chiều rộng của những cột khác)

Ví dụ:

- Phần tử <frameset> bên dưới xác định bộ khung sẽ có ba cột (tương ứng ba phần tử <frame>)
 - o Phần tử <frame> thứ nhất có chiều rộng bằng 20% chiều rộng của phần tử <frameset>
 - o Phần tử <frame> thứ hai có chiều rộng bằng 30% chiều rộng của phần tử <frameset>
 - o Phần tử <frame> thứ ba có chiều rộng bằng 50% chiều rộng của phần tử <frameset>

```
<frameset cols="20%,30%,*">
  <frame src="http://webcoban.vn/html/demo?file=1434"/>
  <frame src="http://webcoban.vn/html/demo?file=1435"/>
  <frame src="http://webcoban.vn/html/demo?file=1436"/>
</frameset>
```

• Thuộc tính rows

- Xác định **số lượng hàng và chiều cao mỗi hàng** trong phần tử <frameset> (Thuộc tính **rows** có bao nhiêu giá trị thì sẽ tương ứng với bấy nhiêu hàng, và mỗi giá trị đó cũng xác định chiều cao của mỗi hàng. Lưu ý, các giá trị được ngăn cách bởi dấu phẩy)
- Giá trị của thuộc tính rows được xác định bởi một trong ba đơn vị: px, %, * (tức là bằng chiều cao của phần tử <frameset> trừ đi tổng chiều cao của những hàng khác)



Ví dụ:

- Phần tử <frameset> bên dưới xác định bộ khung sẽ có ba hàng (tương ứng ba phần tử <frame>)
 - o Phần tử <frame> thứ nhất có chiều cao bằng 20% chiều cao của phần tử <frameset>

- Phần tử <frame> thứ hai có chiều cao bằng 30% chiều cao của phần tử <frameset>
- Phần tử <frame> thứ ba có chiều cao bằng 50% chiều cao của phần tử <frameset>

```
<frameset rows="20%,30%,*">
  <frame src="http://webcoban.vn/html/demo?file=1434"/>
  <frame src="http://webcoban.vn/html/demo?file=1435"/>
  <frame src="http://webcoban.vn/html/demo?file=1436"/>
</frameset>
```

• Thuộc tính noresize

- Mặc định, khi ta **rê chuột vào các cạnh** của một phần tử <frame> thì nó sẽ hiện lên con trỏ giống như  hoặc  giúp chúng ta có thể kéo thả để thay đổi kích thước của phần tử <frame>

- Thuộc tính noresize dùng để xác định việc **ngăn chặn hành động kéo thả làm thay đổi kích thước của phần tử <frame>**

Ví dụ:

```
<frameset rows="20%,30%,50%" noresize>
  <frame src="http://webcoban.vn/html/demo?file=1434"/>
  <frame src="http://webcoban.vn/html/demo?file=1435"/>
  <frame src="http://webcoban.vn/html/demo?file=1436"/>
</frameset>
```

* Thẻ frame:

Not Supported in HTML5.

```
<frameset>
  các khung...
  <frame src="đường dẫn đến tài liệu mà bạn muốn nhúng vào trang web"/>
</frameset>
<noframes>
  nội dung trong trường hợp trình duyệt không hỗ trợ khung
</noframes>
```

Ví dụ:

```
<frameset cols="40%,60%">
  <frame src="http://webcoban.vn/file/ban-khai-nhan-khau.pdf"/>
  <frame src="http://webcoban.vn/css/default.html"/>
</frameset>
```

Các thuộc tính của thẻ <frame>

Src	Xác định đường dẫn đến tài liệu mà bạn muốn đặt vào khung
scrolling	Xác định việc có nên hiển thị các thanh scroll hay không
noresize	Ngăn chặn hành động kéo thả làm thay đổi kích thước của phần tử <frame>
Name	Thuộc tính này cho phép bạn đặt tên cho một Frame . Nó được sử dụng để chỉ rõ Frame nào của tài liệu sẽ được tải trong đó . Nó đặc biệt quan trọng khi bạn muốn tạo các link trong một Frame mà tải các trang web trong một Frame khác, trong trường hợp đó, Frame thứ hai cần một tên để xác định chính nó như là đích tới của link. (Giống thẻ liên kết <a>)
frameborder	Thuộc tính xác định có hay không Border trong Frame đã được chỉ; nhận giá trị 1 (Yes) hoặc 0 (No) .

marginwidth	Thuộc tính này cho phép bạn xác định độ rộng của khoảng cách giữa bên trái và phải của Border Frame với nội dung trong Frame. Giá trị là pixel. Ví dụ marginwidth="10".
marginheight	Thuộc tính cho phép bạn xác định chiều cao của khoảng cách giữa trên và dưới của Border Frame và nội dung của nó. Giá trị là pixel. Ví dụ marginheight="10".
longdesc	Thuộc tính này cho phép bạn cung cấp một đường link tới trang web khác chứa một sự miêu tả dài trong nội dung của Frame . Ví dụ longdesc="framedescription.jsp". (Dựng lại một trang có chứa giới thiệu chi tiết content của hình ảnh thông qua tham số URL.)

• Thuộc tính src

- Xác định đường dẫn đến tài liệu mà bạn muốn đặt vào khung. (Đường dẫn đến tài liệu có thể là đường dẫn **tương đối** hoặc đường dẫn **tuyệt đối**)

Ví dụ:

```
<frameset cols="40%,60%">
  <frame src="http://webcoban.vn/file/ban-khai-nhan-khau.pdf"/>
  <frame src="../css/default.html"/>
</frameset>
```

• Thuộc tính scrolling

- Nếu **tài liệu được nhúng vào có kích thước lớn hơn phần tử <frame>**, thì khi đó mặc định phần tử <frame> sẽ **hiển thị các thanh scroll (ngang hoặc dọc)** giúp ta kéo lên xuống hoặc sang trái phải để có thể xem hết nội dung của tài liệu được nhúng vào.

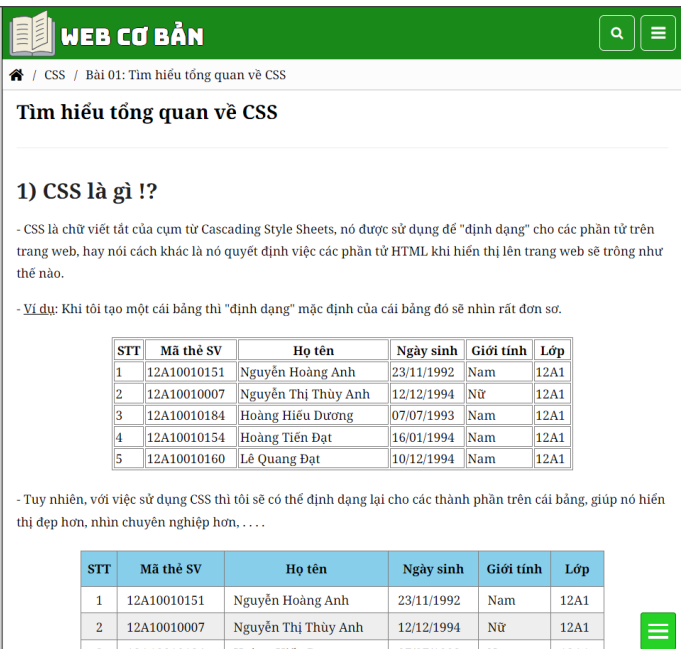
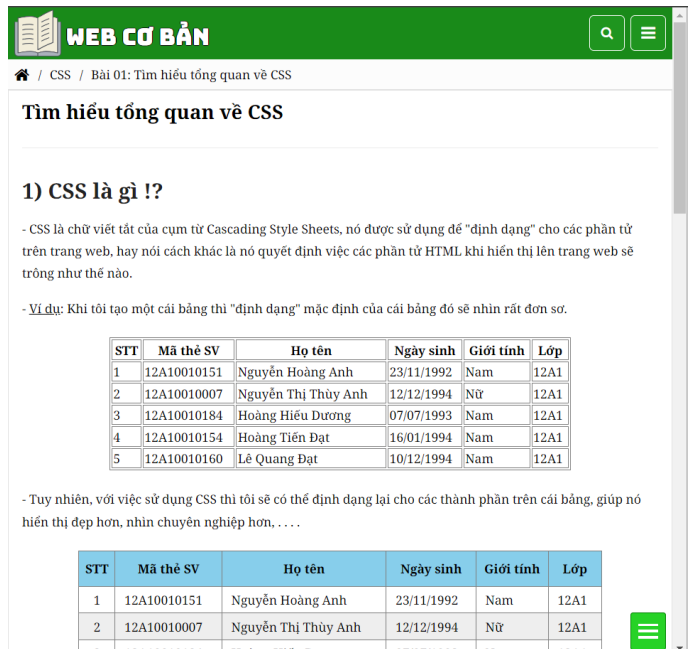
- Thuộc tính scrolling giúp xác định việc có nên hiển thị các thanh scroll hay không.

- Thuộc tính scrolling có thể có một trong ba giá trị: **auto, yes, no**



- Giá trị **auto** (đây là giá trị **mặc định**): thanh scroll sẽ **tự động hiển thị** khi nào kích thước của tài liệu được nhúng vào lớn hơn kích thước của phần tử <frame>
- Giá trị **yes**: thanh scroll sẽ **luôn luôn hiển thị mặc cho có cần thiết hay không**.
- Giá trị **no**: thanh scroll sẽ **không bao giờ được hiển thị**.

Ví dụ:

```
<frameset cols="50%,50%">
  <frame src="http://webcoban.vn/css/default.html" scrolling="yes"/>
  <frame src="http://webcoban.vn/css/default.html" scrolling="no"/>
</frameset>
```



• Thuộc tính noresize

- Mặc định, khi ta rê chuột vào các cạnh của một phần tử <frame> thì nó sẽ hiện lên con trỏ giống như  hoặc  giúp ta có thể kéo thả để thay đổi kích thước của phần tử <frame>
- Từ đây, thuộc tính noresize dùng để xác định việc ngăn chặn hành động kéo thả làm thay đổi kích thước của phần tử <frame>

Ví dụ:

```
<frameset cols="50%,50%">
  <frame src="http://webcoban.vn/css/default.html" noresize/>
  <frame src="http://webcoban.vn/css/default.html" noresize/>
</frameset>
```

- Đối với thẻ iframe:

An inline frame is used to embed another document within the current HTML document.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML TEST</title>
  </head>
  <body>
    <iframe src = "https://tenten.vn/tin-tuc/frame-html/"></iframe>
  </body>
</html>
```

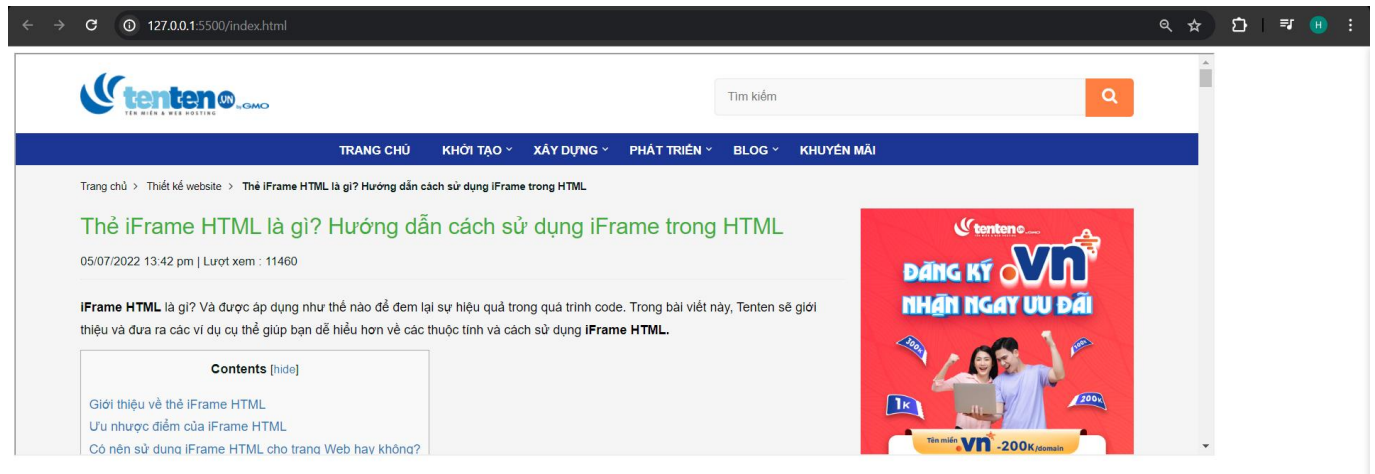


- Ta cần chỉnh độ rộng của cửa sổ thông qua width


```

<!DOCTYPE html>
<html>
  <head>
    <title>HTML TEST</title>
  </head>
  <body>
    <iframe width = 1500px height = 500px src = "https://tenten.vn/tin-tuc/frame-
html/"></iframe>
  </body>
</html>

```



- Có thể nhúng một file **page2.html** khác. Giả sử có file như sau:

```

<!DOCTYPE html>
<html lang="en">
<head>

</head>
<body>
  <h1 style="color: red; background: yellow">Đây là page 2.</h1>
</body>
</html>

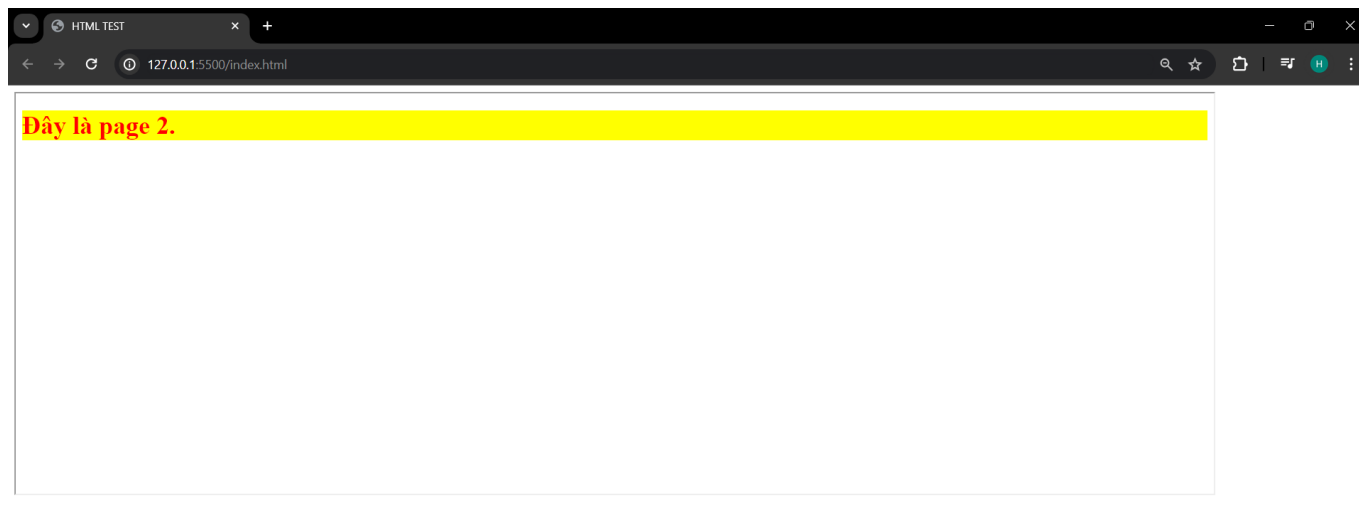
```

Ở file **index.html**, ta viết như sau:

```

<!DOCTYPE html>
<html>
  <head>
    <title>HTML TEST</title>
  </head>
  <body>
    <iframe width = 1500px height = 500px src = "page2.html"></iframe>
  </body>
</html>

```

Các thuộc tính của iFrame HTML

- **src**: Thuộc tính này dùng để chèn một file vào iFrame HTML. Nếu là một URL, được chỉ định webpage sẽ được load bên trong iFrame.
- **name**: Thuộc tính name thường được dùng để định danh một iFrame. Sẽ rất hữu ích nếu bạn muốn tạo một link để mở một website khác.
- **allowfullscreen**: Thuộc tính này cho phép **hiển thị frame ở định dạng full-width**. Bạn cần set giá trị của **thuộc tính này thành true** để nó có tác dụng.
- **frameborder**: Thuộc tính này cho phép bạn khai báo có hiện hoặc không hiện border của frame. Khi set giá trị là 1, border sẽ được thể hiện, nếu set giá trị là 0, border của frame sẽ không được thể hiện.
- **marginwidth**: Cho phép bạn khai báo khoảng cách của hai bên trái & phải so với nội dung của frame.
- **marginheight**: Thuộc tính này cho phép bạn khai báo khoảng cách phía trên & phía dưới so với nội dung của frame.
- **scrolling**: thuộc tính này dùng để kiểm soát việc thanh cuộn (scrollbar) có hiển thị hay không hiển thị vào iFrame. Các giá trị bao gồm là Yes hoặc No hoặc Auto.
- **height**: Được sử dụng để xác định **chiều cao cho frame** (Được xác định bằng % hoặc tính bằng pixel).
- **width**: Được sử dụng để xác định **chiều rộng cho frame** (Được xác định bằng % hoặc tính bằng pixel).
- **Longdesc**: Với sự trợ giúp của thuộc tính này, bạn có thể liên kết một trang khác với mô tả dài về nội dung trong frame của bạn.

Ví dụ về thuộc tính name:

```
<!DOCTYPE html>
<html>

<body>

  <h1>The iframe name attribute</h1>

  <iframe src="demo_iframe.htm" name="iframe_a">
    <p>Your browser does not support iframes.</p>
  </iframe>

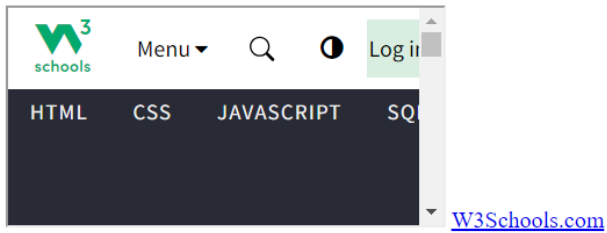
  <a href="https://www.w3schools.com" target="iframe_a">W3Schools.com</a>
```

<p>Note: Because the target of the link matches the name of the iframe, the link will open in the iframe.</p>

</body>

</html>

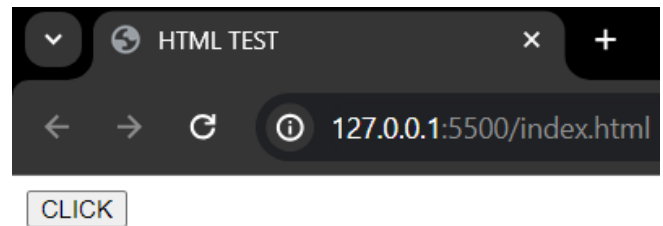
The iframe name attribute



Note: Because the target of the link matches the name of the iframe, the link will open in the iframe.

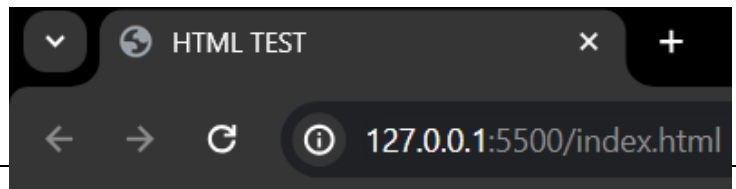
t) Thẻ nút bấm (button)

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML TEST</title>
  </head>
  <body>
    <button>
      CLICK
    </button>
  </body>
</html>
```



- Ta có thể chỉnh sửa giao diện của nút bấm cho phù hợp. Ví dụ:

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML TEST</title>
  </head>
  <body>
    <button style = "border-radius: 5px; background-color: aqua;">
      CLICK
    </button>
  </body>
</html>
```



- Thêm hàm Javascript:

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML TEST</title>
  </head>
  <body>
    <button onclick = "changeText()" style = "border-radius: 5px; background-color: aqua;">
      CLICK
    </button>

    <h1 id = "h1id">Hello</h1>

    <script>
      changeText = () =>{
        document.getElementById("h1id").innerHTML = "Goodbye" //Khi nhấn vào nút
        thì nội dung h1 sẽ thay đổi
      }
    </script>
  </body>
</html>
```

CLICK

CLICK

Hello Sau khi nhấn vào button: **Goodbye**

u) Form và Thẻ form

Giới thiệu về Form

- Cho phép **người sử dụng nhập dữ liệu trên trang web**. Dữ liệu này có thể được **gửi về server để xử lý**.
- Người sử dụng nhập dữ liệu thông qua các điều khiển (controls). Có nhiều loại control:
 1. Form
 2. Textbox
 3. Checkbox
 4. Radio Button
 5. Button
 6. Combo box (drop-down menu)
 7. Listbox
 8. Hộp nhập văn bản nhiều dòng (TextArea)
 9. ...
- Tất cả các điều khiển đều có tên được quy định qua thuộc tính **name**. Tuy nhiên có một số điều khiển thì name không quan trọng (các điều khiển mà sau này không cần lấy dữ liệu)
- Các điều khiển từ số 2. đến số 5 được định nghĩa nhờ thẻ **<input>** và thuộc tính **type** sẽ xác định là điều khiển nào sẽ được tạo ra.
- Form chứa các thành phần nhập liệu khác.

```
<FORM NAME="..." ACTION="..." METHOD="...">  
    <!-- các thành phần của Form -->  
</FORM>
```

- Các thuộc tính của **</FORM>**
 - **NAME** : tên FORM
 - **ACTION** : chỉ định trang web sẽ nhận xử lý dữ liệu từ FORM này khi có sự kiện click của button SUBMIT.
 - **METHOD** : Xác định phương thức chuyển dữ liệu (**POST,GET**)
 -

```
<html>  
  <body>  
    <form Name="Dangnhap"  
      Action="/admin/xlDangnhap.php"  
      Method="Post">  
      .....  
    </form>  
  </body>  
</html>
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>HTML TEST</title>  
  </head>  
  <body>  
    <form method="POST" action=""> <!-- gửi đến nơi nào đó -->  
      <h1>Đăng nhập</h1>  
      <div>  
        <label for = "firstname">first name:</label>  
        <input type = "text" id = "fname"><br></div>
```

```

</div>

<div>
  <label for = "lastname">last name:</label>
  <input type = "text" id = "lname"> <br></br>
</div>

<div>
  <label for = "password">password:</label>
  <input type = "password" id = "password"> <br></br> <!--type = password: khi nhập, text hiển thị
dấu chấm...-->
</div>

<div>
  <label for = "date">date:</label>
  <input type = "date" id = "date"> <br></br>
</div>

<div>
  <label for = "number">date:</label>
  <input type = "number" id = "no."> <br></br> <!--chỉ nhập được số.-->
</div>

<input type = "submit"> <!--tất cả dữ liệu được nhập sẽ chuyển đi đâu đó & biến mất khỏi giao diện
màn hình-->
</form>
</body>
</html>

```

Đăng nhập

first name:

last name:

password:

date: 

date:

Cấu trúc một HTML Form

```

<FORM
  ACTION      = url
  METHOD      = GET | POST
  NAME        = name
>
<!-- Các thành phần của Form -->
<INPUT ...>
<INPUT ...>
</FORM>

```

<INPUT

TYPE = BUTTON | CHECKBOX | FILE | IMAGE | PASSWORD | RADIO | RESET | SUBMIT | TEXT

NAME = string

VALUE = string

ALIGN = LEFT | CENTER | RIGHT

>

Các thành phần của Form

- Text field
- Password field
- Hidden Text field
- Check box
- Radio button
- File Form Control
- Submit Button, Reset Button, Generalized Button
- Multiple-line text field
- Label
- Pull-down menu
- Scrolled list
- Field Set

Text field

- **<input type="text" /> định nghĩa textbox**

- Sử dụng để nhập các văn bản ngắn (trên 1 dòng) hoặc mật khẩu
- Thẻ: **<input>**
- Thuộc tính:
 - **name="tên"**: quan trọng
 - **value="giá trị mặc định"**

```
<form>
  First name: <input type="text" name="firstname" /><br />
  Last name: <input type="text" name="lastname" />
</form>
```

Hiển thị trên trình duyệt

First name:

Last name:

Password field

- **<input type="password" /> định nghĩa một Password Field**

```
<form>
  Password: <input type="password" name="pwd" />
</form>
```

Hiển thị trên trình duyệt

Password:

**** Text khi gõ vào Password Field sẽ được mã hóa**

Check box

● **<input type="checkbox" /> định nghĩa một checkbox**

- Cho phép **chọn nhiều lựa chọn** trong một nhóm lựa chọn được đưa ra bằng cách đánh dấu ("tích").
- Thẻ: **<input>**: mỗi ô nhập cần 1 thẻ
- Thuộc tính:
 - ✓ **name="tên"**: quan trọng
 - ✓ **checked**: mặc định được chọn
 - ✓ **value="giá trị"**: đây là giá trị chương trình sẽ nhận được nếu người dùng chọn ô này.

```
<form>
  <input type="checkbox" name="vehicle" value="Bike" /> I have a bike<br />
  <input type="checkbox" name="vehicle" value="Car" /> I have a car
</form>
```

Hiển thị trên trình duyệt

☐ I have a bike
☐ I have a car

```
<html>
<body>
  Check box group : <br>
  Anh van: <input type="checkbox" name="Languages1" value="En"><br>
  Hoa: <input type="checkbox" name="Languages2" value="Chz" checked><br>
  Nhut: <input type="checkbox" name="Languages3" value="Jp"><br>
</body>
</html>
```

Check box group :
Anh van: ☐
Hoa: ☒
Nhut: ☐

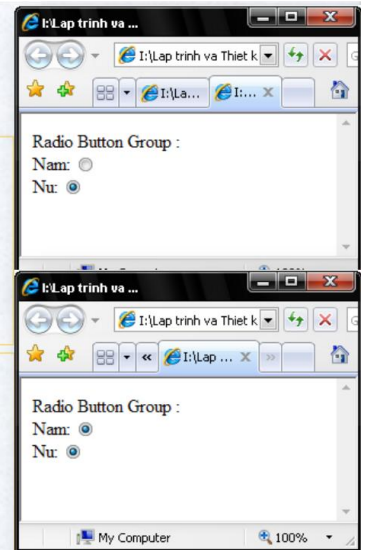
Radio button

● **<input type="radio" /> định nghĩa radio button**

- Cho phép **chọn một** lựa chọn **trong một nhóm** lựa chọn được đưa ra.
- Trên 1 form có thể có nhiều nhóm lựa chọn kiểu này.
- Thẻ: **<input>**: Mỗi ô cần 1 thẻ
- Thuộc tính:
 - **name="tên"**: quan trọng. **Các đối tượng cùng tên thì thuộc cùng nhóm.**
 - **type="radio"**
 - **value="giá trị"**: đây là giá trị chương trình sẽ nhận được nếu người dùng chọn ô này.
 - **checked**: mặc định được chọn

```
<html>
<body>
  Radio Button Group : <br>
  Nam: <input type="radio" name="sex" value="nam" checked><br>
  Nu: <input type="radio" name="sex" value="nu" checked ><br>
</body>
</html>
```

```
<html>
<body>
  Radio Button Group : <br>
  Nam: <input type="radio" name="sex1" value="nam" checked><br>
  Nu: <input type="radio" name="sex2" value="nu" checked ><br>
</body>
</html>
```



```
<div> <!--cùng name để chỉ lựa chọn được 1 option-->
  <input type = "radio" name = "role" value = "Student">Student
  <input type = "radio" name = "role" value="Teacher">Teacher
  <input type="radio" name = "role" value="Staff">Staff
</div>
```

☐ Student ☐ Teacher ☒ Staff

File form control

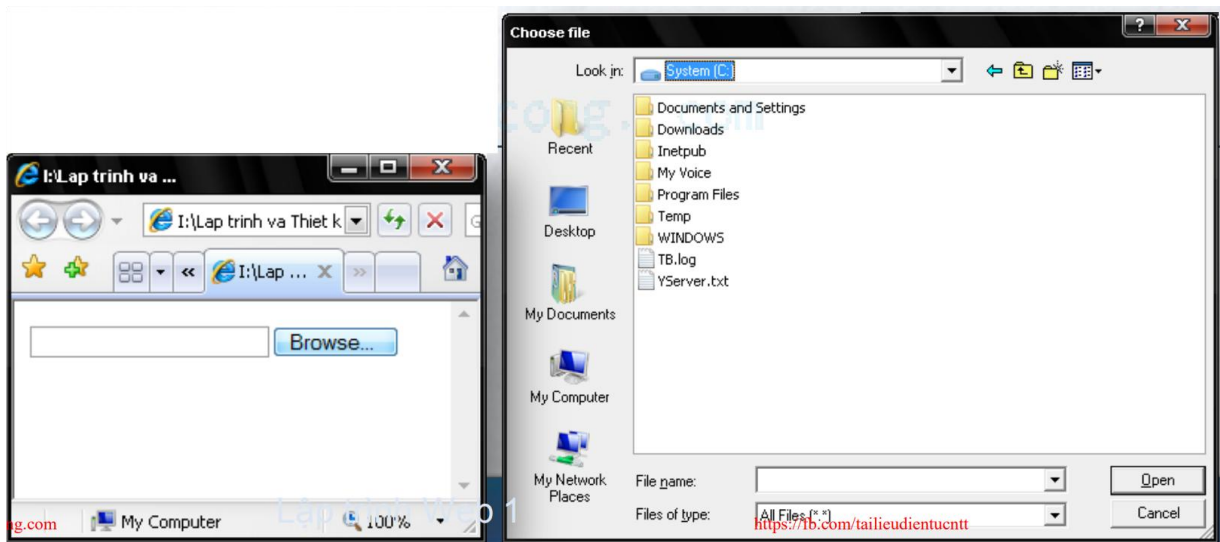
• <input type="file" /> file form control

- Dùng để upload 1 file lên server
- Cú pháp

```
<form action="..." method="post" enctype="multipart/form-data" name="...">
  <input TYPE="FILE" NAME="...">
</form>
```

Ví dụ

```
<html>
<body>
  <form name="frmMain" action="POST" enctype="multipart/form-data">
    <input type="file" name="fileUpload">
  </form>
</body>
</html>
```

Nút lệnh (Button)

- Sử dụng để người dùng **ra lệnh** thực hiện công việc.
- Trên web có 3 loại nút:
 - submit**: Tự động ra lệnh gửi dữ liệu
 - reset**: đưa mọi dữ liệu về trạng thái mặc định
 - normal**: người lập trình tự xử lý
- Thẻ: **<input>**
- Thuộc tính:
 - name**=“tên_đối tượng”: thường **không quan trọng**
 - type**=“**submit**”: nút submit
 - type**=“**reset**”: nút reset
 - type**=“**button**”: nút thông thường (normal)
 - value**=“**tiêu đề nút**”



Submit button

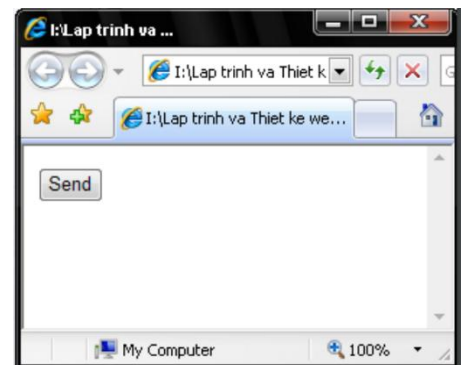
Nút phát lệnh và gửi dữ liệu của form đến trang xử lý. Mỗi form chỉ có một nút submit và nút này được viền đậm

Cú pháp:

```
<input TYPE="submit" name="..." value="...">
```

Ví dụ:

```
<input type="submit" name="btnSend" value="Send">
```



Reset button

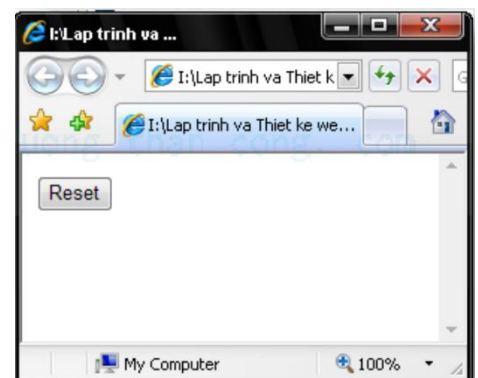
Dùng để trả lại giá trị mặc định cho các control khác trong form

Cú pháp

```
<input TYPE="reset" name="..." value="...">
```

Ví dụ

```
<input type="reset" name="btnReset" value="Reset">
```



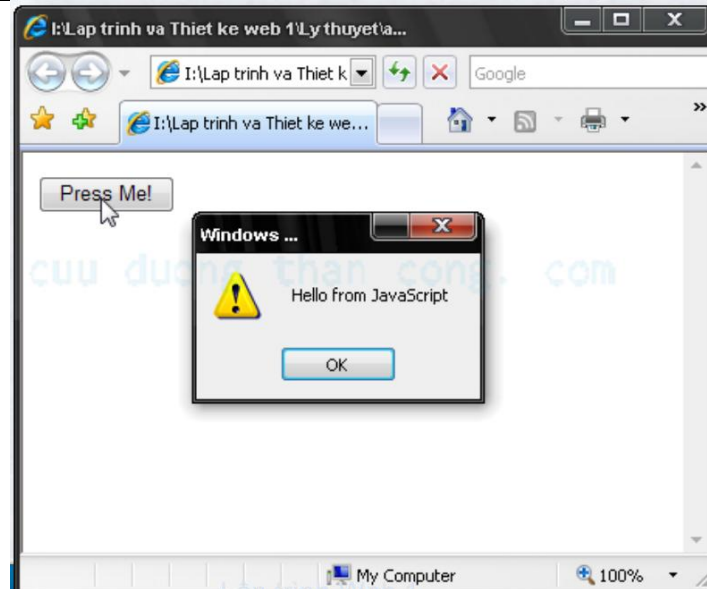
Generalized button

Cú pháp

```
<input type="button" name="..." value="..." onclick="script">
```

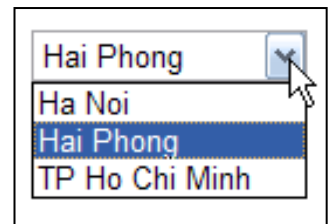
Ví dụ

```
<input type="button" name="btnNormal" value="Press Me!"  
onclick="alert('Hello from JavaScript');" >
```



Combo Box (Drop-down menu/ Pull-down menu)

- Bao gồm **một danh sách có nhiều phần tử**. Tại một thời điểm chỉ có 1 phần tử được chọn
- Người dùng có thể chọn 1 phần tử trong danh sách bằng cách kích vào mũi tên bên phải hộp danh sách.



- Thẻ tạo **hộp danh sách**:

```
<select>Danh sách phần tử</select>
```

- Thuộc tính: **name**="tên_đối tượng"

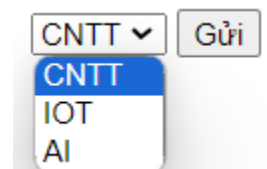
- Thẻ tạo **1 phần tử trong danh sách**:

```
<option>Tiêu đề phần tử</option>
```

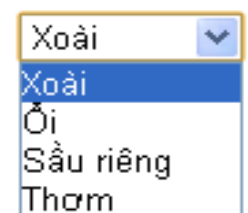
- Thuộc tính:

- value**="giá trị": giá trị chương trình nhận được nếu phần tử được chọn
- selected**: mặc định được chọn

```
<select name = "major"> <!--major sẽ nhận giá trị từ value sau khi nhấn gửi-->  
  <option value = "CNTT">CNTT</option>  
  <option value = "IOT">IOT</option>  
  <option value = "AI">AI</option>  
</select>
```



```
<form>  
  <select name="traicay">  
    <option value="Xoai">Xoài</option>  
    <option value="oi">Ổi</option>  
    <option value="saurieng">Sầu riêng</option>  
    <option value="thom">Thơm</option>  
  </select>  
</form>
```



```

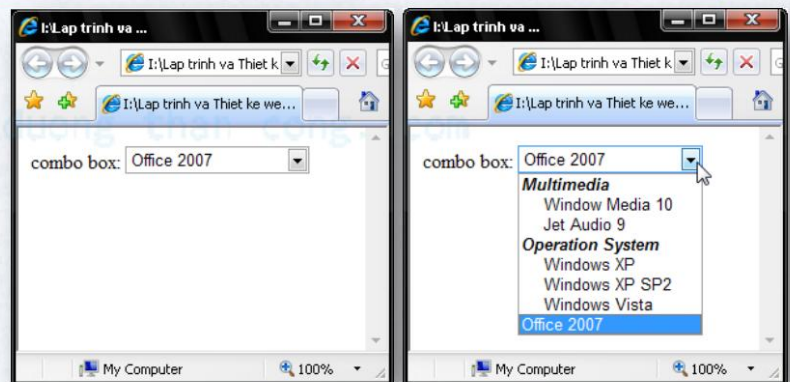
<Select name="...">
  <optgroup label="...">
    <option [selected] value="..." >.....</option>
    .....
  </optgroup>
  <option [selected] value="..." >.....</option>
  .....
</select>

```

```

<html>
  <body>
    combo box:
    <select name="DSSoftware">
      <optgroup label="Multimedia">
        <option value="WM10">Window Media 10</option>
        <option value="JA9">Jet Audio 9</option>
      </optgroup>
      <optgroup label="Operation System">
        <option value="WXP">Windows XP</option>
        <option value="WXPSP2">Windows XP SP2</option>
        <option value="WVT">Windows Vista</option>
      </optgroup>
      <option selected value="Office07">Office 2007</option>
    </select>
  </body>
</html>

```



Listbox

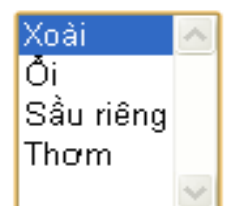
- Tương tự như Combo box, tuy nhiên **có thể nhìn thấy nhiều phần tử cùng lúc**, có thể **lựa chọn nhiều phần tử**
- Thẻ: <select>...</select>
- Thuộc tính: tương tự của combo tuy nhiên có 2 thuộc tính khác:
 - **size="số dòng"**
 - **multiple**: cho phép lựa chọn nhiều phần tử cùng lúc
- Thẻ <option>...</option> tương tự của combo box



```

<form action="">
  <select name="traicay" multiple=true size=5 >
    <option value="Xoai" selected=true>Xoài</option>
    <option value="oi">Ổi</option>
    <option value="saurieng">Sầu riêng</option>
    <option value="thom">Thơm</option>
  </select>

```



Hộp nhập văn bản nhiều dòng (TextArea/ Multiline Text Field)

- Cho phép nhập văn bản dài trên nhiều dòng.
- Thẻ:

<textarea>

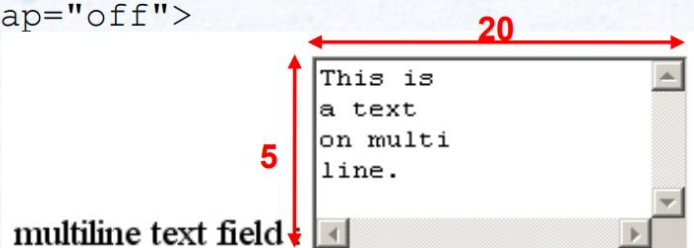
Nội dung mặc định

</textarea>

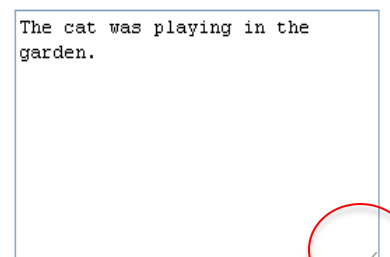
- Thuộc tính:

Attribute	Value	Description
autofocus	autofocus	Specifies that a text area should automatically get focus when the page loads
cols	number	Specifies the visible width of a text area
dirname	textarea.name.dir	Specifies that the text direction of the textarea will be submitted
disabled	disabled	Specifies that a text area should be disabled
form	form_id	Specifies which form the text area belongs to
maxlength	number	Specifies the maximum number of characters allowed in the text area
name	text	Specifies a name for a text area
placeholder	text	Specifies a short hint that describes the expected value of a text area
readonly	readonly	Specifies that a text area should be read-only
required	required	Specifies that a text area is required/must be filled out
rows	number	Specifies the visible number of lines in a text area
wrap	Specifies how the text in a text area is to be wrapped when submitted in a form (kiểm soát cách văn bản tự động xuống dòng khi nhập nội dung vào ô nhập liệu)	
	off	Không tự động xuống dòng. Người dùng phải nhấn Enter để xuống dòng. Khi gửi biểu mẫu, văn bản giữ nguyên định dạng gốc.
	soft	Văn bản tự động xuống dòng trong giao diện nhưng không thêm ký tự xuống dòng vào dữ liệu khi gửi biểu mẫu.
	hard	Văn bản tự động xuống dòng và thêm ký tự xuống dòng (\n) vào dữ liệu khi gửi biểu mẫu.

```
<textarea cols="7" rows="5" wrap="off">
  This is a text on multiline.
</textarea>
```



```
<textarea rows="10" cols="30">
  The cat was playing in the garden.
</textarea>
```



*** Để khung không thể kéo dẫn hay thu lại bởi người dùng:**

Dùng css **resize: none;**

Label

- Dùng để **gán nhãn cho một Form Field**, đồng thời nhận chuyển các sự kiện nhận được đến form field.
- Tag <label> không hiển thị bất cứ gì đặc biệt cho người dùng, tuy nhiên nó cung cấp một cái thiện cho người sử dụng chuột, **nếu click chuột vào nhãn, sẽ đưa con trỏ chuột vào vùng <input />.**
- Muốn sử dụng hiệu quả <label>, cần thiết phải cho giá trị id của <input /> và giá trị for của <label>, hai giá trị này phải trùng nhau.

Cú pháp

```
<LABEL  
    FOR = IDString  
    CLASS=string  
    STYLE=string  
>
```

Ví dụ

```
<label for="Languages">Anh văn: </label>  
<input type="checkbox" name="Languages" id="Languages" value="Eng"> Anh văn: ☐
```

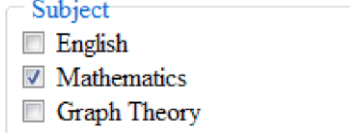
Fieldset

- Dùng để tạo ra Group box, **nhóm các thành phần nhập liệu trong form**
- Cú pháp**

```
<fieldset>  
    <legend>GroupBox"s Name</legend>  
    <input .....>  
    ...  
</fieldset>
```

- Ví dụ**

```
<html>  
<body>  
    <fieldset>  
        <legend>Subject</legend>  
        <input type="checkbox" name="Subjects" value="Eng"> English<br>  
        <input type="checkbox" name="Subjects" value="Math" checked> Mathematics<br>  
        <input type="checkbox" name="Subjects" value="GraphTheory"> Graph Theory<br>  
    </fieldset>  
</body>  
</html>
```



Phân biệt phương thức GET/POST

Phương thức GET

- Các **đối số của Form được ghi kèm theo vào đường dẫn URL của thuộc tính Action** trong tag <Form> → DL truyền đi hiển thị trên đường link trên trang web - người dùng sẽ thấy
- Ví dụ:** localhost:3000/.../tên_đối_tượng_1 = giá_trị_1&tên_đối_tượng_2 = giá_trị_2
- Khối lượng dữ liệu đối số được truyền đi của Form bị giới hạn bởi chiều dài tối đa của một URL trên Address bar. (tối đa của một URL là 2048 bytes)
- Thiết kế một form đơn giản bao gồm 2 textbox và 1 button dạng submit

```
<form action="form_action.asp" method="get">  
  First name: <input type="text" name="fname" value="Mickey"/><br />  
  Last name: <input type="text" name="lname" value="Mouse"/><br />  
  <input type="submit" value="Submit" />  
</form>
```

First name:

Last name:

- Khi người dùng chọn vào nút Submit thì dữ liệu trong 2 textbox sẽ được truyền theo phương thức get.

Phương thức POST

- Các **đối số của Form** được truyền “ngầm” bên dưới
- **Truyền DL Dạng ẩn - người dùng sẽ không thấy** - Dùng cho form đăng nhập/ đăng ký, các form chứa nhiều dữ liệu (bài báo, tin tức...)
- Khối lượng dữ liệu đối số được truyền đi của Form không phụ thuộc vào URL → **Không bị giới hạn**
- Chỉ sử dụng được phương thức truyền POST khi Action chỉ định đến trang web thuộc dạng xử lý trên Server
- Thiết kế một form đơn giản bao gồm 2 textbox và 1 button dạng submit

```
<form action="form_action.asp" method="post">  
  First name: <input type="text" name="fname" value="Mickey"/><br />  
  Last name: <input type="text" name="lname" value="Mouse"/><br />  
  <input type="submit" value="Submit" />  
</form>
```

- Khi người dùng chọn vào nút Submit thì dữ liệu trong 2 textbox sẽ được truyền theo phương thức post.

v) Thẻ <marquee>

- **Không tương thích với HTML5.**
- Thẻ <marquee> sẽ làm chạy hầu hết các đối tượng trong thẻ như: chữ, hình ảnh, các định dạng khác.
- Cú pháp như sau: **<marquee>** đối tượng chạy **</marquee>**
- Thuộc tính **direction** (hướng chạy):
 - **Mặc định** các đối tượng sẽ chạy từ **phải sang trái**. Tuy nhiên ta có thể thay đổi hướng chạy.
 - Các giá trị của thuộc tính: **down, left, right, up**
- Thuộc tính **behavior** (kiểu chạy):
 - **Slide**: chạy 1 lần rồi dừng
 - **Scroll**: chạy bình thường
 - **Alternate**: chạy qua rồi chạy lại
- Thuộc tính **scrollamount** (tốc độ chạy):
 - Giúp tăng hay giảm tốc độ chạy, thuộc tính này nhận giá trị chạy là số nguyên
- Thuộc tính **loop** (chế độ lặp):
 - Lặp 1 lần: loop = 1
 - Lặp liên tục: loop = -1

Ngoài ra còn có các thuộc tính:

- **onmouseover**: Thuộc tính này cho phép bạn tạm dừng chuyển động của nội dung khi người dùng di chuột qua vùng chứa <marquee>. Điều này rất hữu ích trong trường hợp bạn muốn người dùng có thể dừng lại và đọc nội dung một cách dễ dàng.
- **onmouseout**: Khi kết hợp với onmouseover, thuộc tính này giúp tiếp tục chuyển động của nội dung khi người dùng di chuyển chuột ra khỏi vùng chứa <marquee>.
- **width**: Định nghĩa chiều rộng của marquee theo pixel hoặc %.
- **height**: Định nghĩa chiều cao của marquee theo pixel hoặc %.
- **hspace**: Xác định khoảng cách ngang giữa vùng chứa <marquee> và các phần tử khác trên trang. Giá trị có thể được xác định bằng pixel hoặc phần trăm.
- **vspace**: Tương tự như hspace, nhưng thuộc tính này điều chỉnh khoảng cách theo chiều dọc.
- **scrolldelay**: Định nghĩa độ trễ khi cuộn theo giây.
- **bgcolor**: Định nghĩa màu nền của marquee.

<body>

```
<marquee direction = "right" behavior = "scroll" loop = -1 scrollamount = 10> Chạy từ  
phải qua, kiểu chạy scroll, lặp liên tục, tốc độ chạy 10 </marquee>  
<marquee direction = "left" behavior = "slide" loop = 1 scrollamount = 10> Chạy từ  
trái qua, kiểu chạy slide, lặp 1 lần, tốc độ chạy 10 </marquee>  
<marquee direction = "down" behavior = "scroll" loop = -1 scrollamount = 10> <p  
align="center"> Chạy từ trên xuống, kiểu chạy scroll, lặp liên tục, tốc độ chạy 10  
</p></marquee>  
<marquee direction = "up" behavior = "alternate" loop = 1 scrollamount = 10> <p  
align="center"> Chạy từ dưới lên, kiểu chạy alternate, lặp 1 lần, tốc độ chạy 10  
</p></marquee>
```

</body>

w) <figcaption> & <figure>

Để thêm một phụ đề hoặc văn bản dưới hình ảnh trong HTML, bạn có thể sử dụng thẻ <figcaption> kết hợp với thẻ <figure>. Dưới đây là cách thực hiện:

Ví dụ Sử dụng <figure> và <figcaption>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Image with Caption Example</title>
  <style>
    figure {
      text-align: center; /* Căn giữa hình ảnh và phụ đề */
      margin: 20px; /* Khoảng cách xung quanh figure */
    }

    img {
      max-width: 100%; /* Đảm bảo hình ảnh responsive */
      height: auto; /* Giữ tỷ lệ hình ảnh */
    }

    figcaption {
      font-size: 14px; /* Kích thước chữ cho phụ đề */
      color: #666; /* Màu chữ cho phụ đề */
      margin-top: 5px; /* Khoảng cách giữa hình ảnh và phụ đề */
    }
  </style>
</head>
<body>

  <figure>
    
    <figcaption>This is a caption for the image.</figcaption>
  </figure>

</body>
</html>
```

Giải thích:

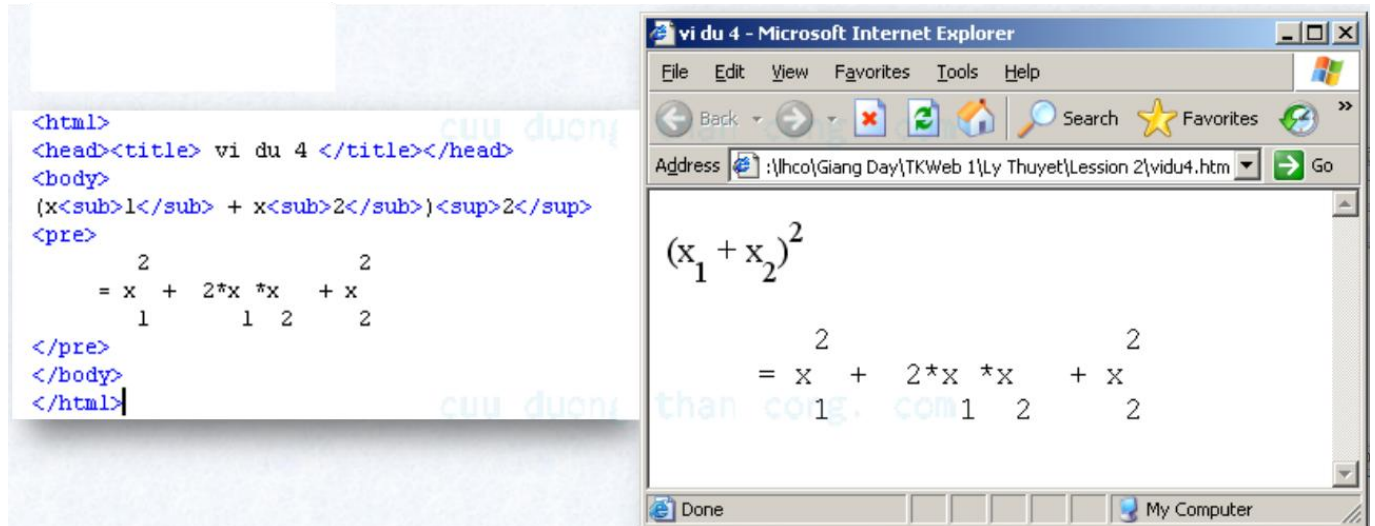
1. **Thẻ <figure>**: Được sử dụng để nhóm hình ảnh và phụ đề lại với nhau. Nó cũng giúp định nghĩa một mối quan hệ giữa hình ảnh và văn bản phụ đề.
2. **Thẻ <figcaption>**: Dùng để thêm văn bản phụ đề cho hình ảnh. Văn bản này sẽ được hiển thị dưới hình ảnh.
3. **CSS**:
 - text-align: center;: Căn giữa cả hình ảnh và phụ đề.
 - max-width: 100%;: Đảm bảo hình ảnh hiển thị responsive, không vượt quá chiều rộng của phần tử chứa nó.
 - margin-top: 5px;: Tạo khoảng cách giữa hình ảnh và phụ đề.

Kết quả:

- Hình ảnh sẽ hiển thị cùng với phụ đề ngay dưới nó, được căn giữa và có khoảng cách hợp lý.

x) Thẻ <pre>

Hiển thị đúng dạng văn bản đã soạn thảo (khoảng trắng, xuống dòng, thẻ,...)

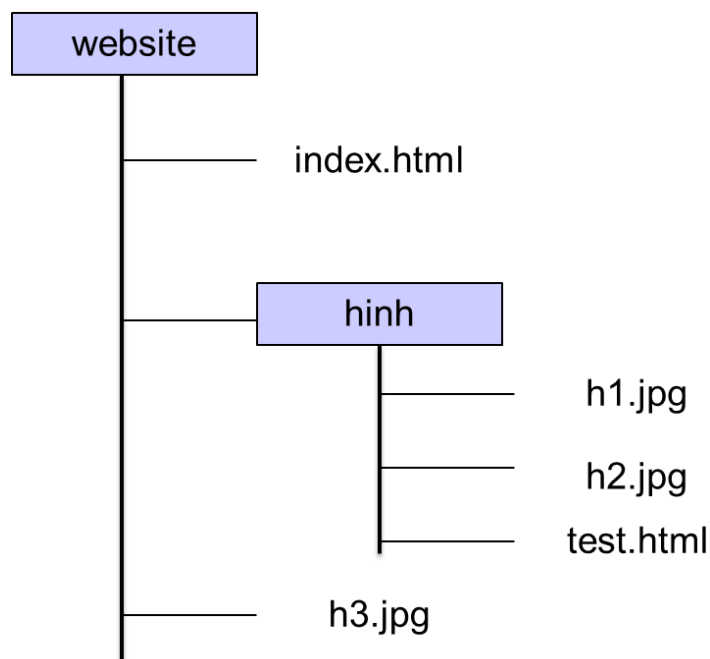


4.5 Đường dẫn (liên kết/ link) tương đối & tuyệt đối (cố định)

- Đường dẫn tuyệt đối (là một chuỗi đầy đủ bao gồm http://, tên miền của trang web, đường dẫn đến tập tin)
 - VD: <https://www.ut.edu.vn/abc/hinh.jpg> - Cho biết tấm hình “hinh” đang nằm trong thư mục nào (abc) của máy chủ nào (ut.edu.vn).
- 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)
 - Đi vào một thư mục: Tên_thư_mục/
 - Đi ra một thư mục: ../

Ký hiệu	Ý nghĩa
/	Trở về thư mục gốc của website
./	Thư mục hiện tại của trang web sử dụng link (mặc định)
../	Quay ra thư mục cha / đi ngược lại 1 cấp thư mục

Ví dụ 1:



Từ trang index.html, cần chèn hình h1.jpg vào:

Từ trang index.html, cần chèn hình h3.jpg, h2.jpg vào:

;

Ví dụ 2:

Vào ổ đĩa D tạo một thư mục có tên là laptrinhweb. Vào thư mục laptrinhweb, lần lượt tạo các thư mục và tập tin như sau:

 laptrinhweb

trang1.html

 folder1

trang2.html

 folder3

trang4.html

 folder7

trang5.html

 folder8

trang6.html

 folder4

 folder2

trang3.html

 folder5

trang7.html

 folder9

trang8.html

Giả sử bạn đang ở trang1.html muốn đến trang2.html thì đường đi sẽ là:

1. Đi vào folder1
2. Đi vào trang2.html

`Liên kết đến trang 2`

Giả sử bạn đang ở trang1.html muốn đến trang6.html thì đường đi sẽ là:

1. Đi vào folder1
2. Đi vào folder3
3. Đi vào folder8
4. Đi vào trang6.html

`Liên kết đến trang 6;`

Giả sử bạn đang ở trang4.html muốn đến trang2.html thì đường đi sẽ là:

1. Đi ngược về thư mục cha (Để đi ngược về thư cha, ta dùng dấu hai chấm ..)
2. Đi vào trang2.html

`Liên kết đến trang 2`

Giả sử bạn đang ở trang4.html muốn đến trang8.html thì đường đi sẽ là:

1. Đi ngược về thư mục cha
2. Đi ngược về thư mục cha
3. Đi vào folder2
4. Đi vào folder5

5. Đi vào folder9
6. Đi vào trang8.html

Liên kết đến trang 8

Giả sử bạn đang ở trang5.html muốn đến trang6.html thì đường đi sẽ là:

1. Đi ngược về thư mục cha
2. Đi vào folder8
3. Đi vào trang6.html

Liên kết đến trang 6

Giả sử bạn đang ở trang3.html muốn đến trang4.html thì đường đi sẽ là:

1. Đi ngược về thư mục cha
2. Đi vào folder1
3. Đi vào folder3
4. Đi vào trang4.html

Liên kết đến trang 4

Giả sử bạn đang ở trang1.html muốn đến trang8.html thì đường đi sẽ là:

1. Đi vào folder2
2. Đi vào folder5
3. Đi vào folder9
4. Đi vào trang8.html

Liên kết đến trang 8

Ví dụ 3:

- Giả sử tồn tại một file **page2.html** có nội dung sau:

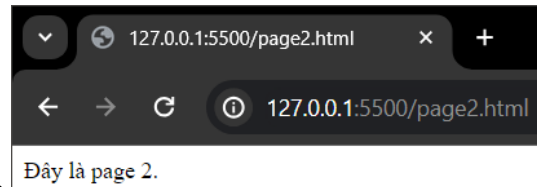
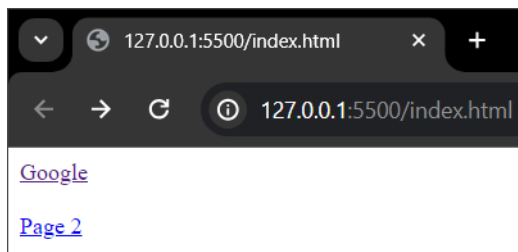
```
<!DOCTYPE html>
<html lang="en">
<head>

</head>
<body>
    Đây là page 2.
</body>
</html>
```

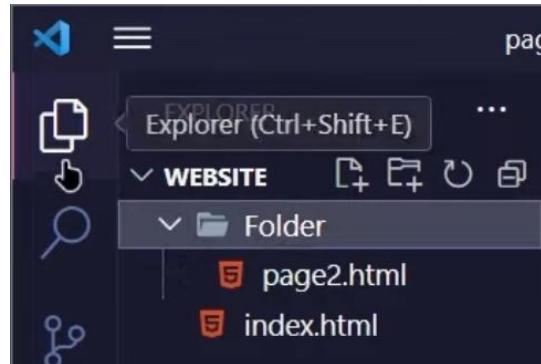
- Để **chuyển hướng từ index.html sang page2.html**, làm như sau:

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

    </head>
    <body>
        <a href="https://www.google.com/" target = "_blank">Google </a>
        <br><br>
        <a href="page2.html">
            Page 2
        </a>
    </body>
</html>
```



* **Lưu ý:** Nếu 2 file này nằm ở 2 folder khác nhau (không nằm cạnh nhau)



Thì ta cần viết đường dẫn như sau

```
<body>
  <a href="https://www.google.com.vn/" target="_"
    Google
  </a>
  <br>
  <a href="/Folder/page2.html">
    Page2
  </a>
</body>
```

Nếu page2.html nằm bên ngoài cả Folder, thì ta cần viết đường dẫn:

```
<a href="../page2.html">
  Page2
</a>
```

Ưu điểm và nhược điểm của hai loại đường dẫn

Loại đường dẫn	Ưu điểm	Nhược điểm
Đường dẫn tuyệt đối	Dễ dàng xác định được đường dẫn đến một trang web (hoặc tài nguyên) nào đó.	<ul style="list-style-type: none"> Không thể sử dụng cho các tên miền khác. Ví dụ, bạn đang xem bài hướng dẫn này ở trang webcoban.vn, tôi có một liên kết đến trang Cách tạo một trang web HTML với đường dẫn là <code>http://webcoban.vn/html/cach-tao-mot-trang-web-html.html</code> Nếu tôi up toàn bộ mã nguồn của trang web này cho tên miền webcoban.com thì khi bấm vào liên kết Cách tạo một trang web HTML nó vẫn chuyển đến trang <code>http://webcoban.vn/html/cach-tao-mot-trang-web-html.html</code> chứ không phải là <code>http://webcoban.com/html/cach-tao-mot-trang-web-html.html</code>
Đường dẫn tương đối	Có thể sử dụng trên các tên miền khác nhau	Nếu chưa quen cách sử dụng đường dẫn tương đối thì việc xác định đường dẫn đến một trang web (hoặc tài nguyên) nào đó sẽ rất khó khăn và dễ bị sai

Khi nào sử dụng địa chỉ tương đối/ tuyệt đối?

- Từ một website cần liên kết đến các file thuộc website **khác** → Sử dụng đường dẫn tuyệt đối.
- Liên kết các thành phần trong **cùng một website** → Sử dụng đường dẫn tương đối.

4.6 Màu sắc trong HTML

- Giá trị màu được ghi bằng **tổ hợp**: #RRGGBB (RED_GREEN_BLUE)
 - RR: red
 - GG: green
 - BB: blue
- Mỗi thành phần màu được ghi bằng **số thập lục phân** (hệ đếm 16) từ **00 đến FF**
 - Giá trị thấp nhất là #000000 là màu đen
 - Giá trị cao nhất là #FFFFFF là màu trắng
 - Hoặc ta có thể dùng các tên màu có định nghĩa sẵn như: red (đỏ), yellow (vàng),...
 - Có một số mã màu được làm gọn: **#2F5 = #22FF55**

http://www.w3schools.com/tags/ref_colornames.asp

- Màu nền, màu chữ được đặt tương ứng nhờ vào các biến thuộc tính BGCOLOR, TEXT.

```
<HTML>
<!--Tạo trang web đầu tiên-->
<HEAD>
<TITLE> "HTML Example 2" </TITLE>
</HEAD>
<BODY BGCOLOR="yellow" TEXT = "blue">
    <H1> This is my HTML page </H1>
    <P> - First paragraphs </P>
    <P> - Second paragraphs </P>
</BODY>
</HTML>
```

- Ngoài ra ta có thể đặt hình nền cho trang nhờ vào biến thuộc tính BACKGROUND
 - Lưu ý: chép ảnh cần dùng làm hình nền vào cùng tập tin HTML.

```
<HTML>
<!--Tạo trang web đầu tiên-->
<HEAD>
<TITLE> "HTML Example 2" </TITLE>
</HEAD>
<BODY background = "Autumn.jpg" TEXT="Yellow"
  <H1> This is my HTML page </H1>
  <H1> Use Image for Background </H1>
  <P> - First paragraphs </P>
  <P> - Second paragraphs </P>
</BODY>
</HTML>
```

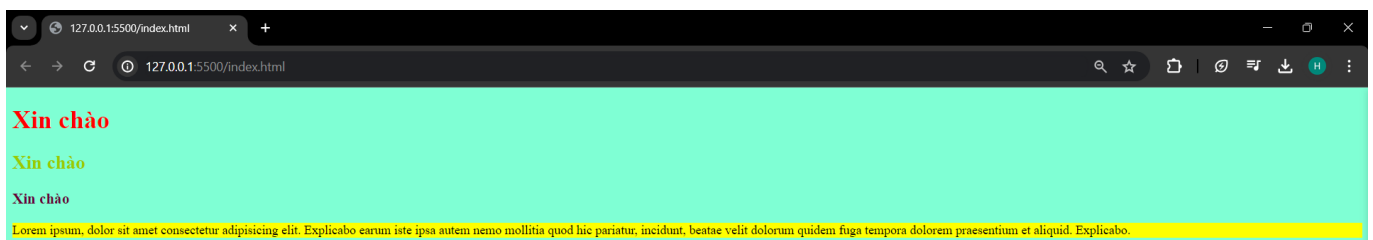
* Sử dụng thuộc tính style.

- Nếu chưa biết màu gì, có thể dùng rgb hoặc dùng mã hexa của màu tương ứng.

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

  </head>
  <body style=" background-color: aquamarine">
    <h1 style = "color: red">Xin chào</h1>
    <h2 style = "color: rgb(153,204,0)">Xin chào</h2>
    <h3 style = "color: #660033">Xin chào</h3>

    <p style = "background-color: yellow;">Lorem ipsum, dolor sit amet consectetur
    adipisicing elit. Explicabo earum iste ipsa autem nemo mollitia quod hic pariatur, incidunt,
    beatae velit dolorum quidem fuga tempora dolorem praesentium et aliquid. Explicabo.</p>
  </body>
</html>
```



PHỤ LỤC

***** Trang chủ:** bao quát nội dung của website: đó là web gì, có chức năng nào.

xd khung website → tùy biến cho các trang thành phần

tập trung mô tả chính cho trang homepage (định hình trước homepage rồi các trang khác tuân theo)

Trên cùng: Banner (Bảng hiệu - mục đích/tên/chủ thể của website) + Logo (tăng độ tin tưởng cho NSD) + Các thành phần khác (Chuyển đổi ngôn ngữ (dùng biểu tượng nếu ít ngôn ngữ/ danh mục nếu nhiều ngôn ngữ), giao diện sáng tối, Thanh tìm kiếm, Đăng nhập/Đăng ký)

Bên trái: Chứa danh mục (Liên kết đến các thành phần bên trong website) - Chứa các liên kết đa cấp (xổ xuống nhiều liên kết - một sp nhiều loại) - Hiện nay chèn thêm quảng cáo để lấp đầy.

Thanh menu: Chứa liên kết đơn (đơn cấp - Trang chủ, Giới thiệu, Liên hệ, Hướng dẫn, Câu hỏi thường gặp) - Nằm ở banner/dưới banner - (Liên kết đến các thành phần bên trong website)

Ở giữa: Nội dung: Chứa tổng quan (nếu là trang chủ)/ Nội dung của các trang thành phần - Chiếm độ rộng lớn nhất

Bên phải: Mở rộng: Bài viết/Sản phẩm/Khuyến mãi mới... Gây chú ý cho NSD/ Thông tin đăng nhập/đăng ký/Giỏ hàng/Bảng giá vàng/Thời tiết/Chèn quảng cáo/Kênh liên hệ

Bên trái và bên phải có thể thay thế nhau: Chứa mục lục bài viết (loại, năm...)

Footer: Chứa thông tin liên hệ với chủ web (Địa chỉ cụ thể của cơ quan/cửa hàng, SĐT, Email, Hotline, Mã số thuế, Bản quyền: Mã số đk... - Công khai minh bạch) + Tùy biến: Nhúng thêm gmap để NSD tìm đường

Tùy tình huống, 1 số thành phần này có thể không cần thiết (hoặc kết hợp với các thành phần khác) → Tùy biến để bố trí cho hợp lý.

Có bao nhiêu cách để xây dựng layout như trên: 2 (frame - chia 3 hàng, trên hàng 2 chia thành 3 cột, table-3 dòng-3 cột, vận dụng trộn ô), hoặc div css

Dùng div: dùng kích thước tương đối - biến đổi cách trình bày phù hợp từng loại màn hình (responsive)

<https://www.youtube.com/watch?v=seR5JFRdx5A>

<https://webcoban.vn/html/lien-ket-duong-dan-tuyet-doi-duong-dan-tuong-doi-trong-html.html>

<https://webcoban.vn/html/cach-su-dung-the-frame-trong-html.html>

https://vietjack.com/html/tao_frame_trong_html.jsp

https://tenten.vn/tin-tuc/frame-html/#Caacutec_thu7897c_tiacutenh_c7911a_iFrame_HTML

<https://rdsic.edu.vn/blog/blog-3/marquee-la-vi-cb.html#3>

https://itviec.com/blog/table-html/#Mot_so_the_tag_pho_bien_cua_Table_HTML

<https://www.w3schools.com/tags>

https://www.w3schools.com/tags/tag_textarea.asp