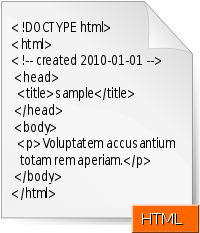
# HTML là gì

**HTML** ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh), viết tắt cho *HyperText Markup Language*, hay là "Ngôn ngữ Đánh dấu Siêu văn bản") là một [ngôn ngữ đánh dấu](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_%C4%91%C3%A1nh_d%E1%BA%A5u) được thiết kế ra để tạo nên các [trang web](https://vi.wikipedia.org/wiki/Website). Cùng với [CSS](https://vi.wikipedia.org/wiki/CSS) và [JavaScript](https://vi.wikipedia.org/wiki/JavaScript), HTML tạo ra bộ ba nền tảng kỹ thuật cho [World Wide Web](https://vi.wikipedia.org/wiki/World_Wide_Web). Phiên bản hiện tại của HTML là 5.1.2 (xuất bản ngày 03/10/2017).

**Ngôn ngữ đánh dấu** có cả văn bản và thông tin phụ về văn bản. Thông tin phụ này, có thể nói về cấu trúc của [tập tin](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin) hay cách trình bày văn bản, là mã đánh dấu được pha trộn vào văn bản chính

**Siêu văn bản** ([tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh): *hypertext*) là văn bản của một tài liệu có thể được truy tìm không theo tuần tự. Người đọc có thể tự do đuổi theo các dấu vết liên quan qua suốt tài liệu đó bằng các mối liên kết xác định sẵn do người sử dụng tự lập nên



Hình 1: Một tài liệu HTML đơn giản

Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements).

Phần tử HTML đa phần có cấu trúc thẻ đóng và mở: <tên thẻ>Nội dung</tên thẻ>

Ví dụ thẻ <html> là thẻ mở và </html là thẻ đóng>. Các văn bản muốn được đánh dấu bằng HTML sẽ được khai báo bên trong cặp thẻ.

Nhưng một số thẻ đặc biệt lại không có thẻ đóng và dữ liệu được khai báo sẽ nằm trong các thuộc tính (ví dụ như thẻ <img>).

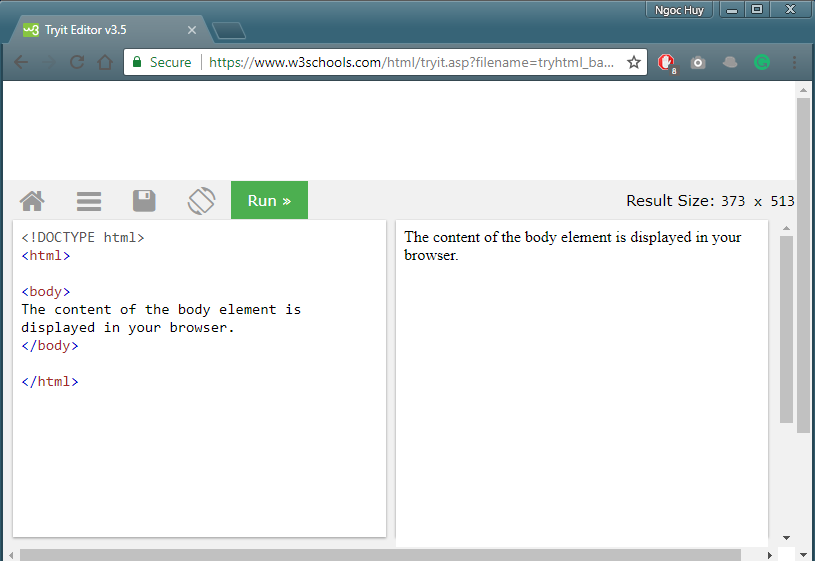
Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đuôi mở rộng là **.html** hoặc **.htm**

# Editor - Cách tạo ra tài liệu HTML

Để tạo ra một tài liệu HTML ta có thể sử dụng bất cứ chương trình gì và lưu nó dưới dạng .html và .htm

Hiện nay có rất nhiều chương trình soạn thảo online mà có thể soạn thảo và xem ngay kết quả trên web

Sau đây là cách soạn thảo HTML trên Tryit Editor – là một chương trình soạn thảo HTML ngay trên web và có thể dễ dàng kiểm tra kết quả soạn thảo.



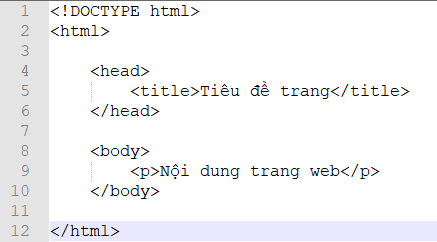
Hình 2: Chương trình soạn thảo Tryit Editor.

Sau khi soạn thảo như ý muốn. Ta có thể lấy nội dụng soạn thảo và lưu lại với đuôi file .html hoặc .htm, như vậy ta có một tài HTML theo ý muốn.

Sau khi lưu lại ta có thể mở file HTML bằng các trình duyệt và xem kết quả.

# Cấu trúc một trang HTML

Sau đây là một trang HTML với cấu trúc đơn giản



Cấu trúc của một trang HTML đơn giản.

Trong đó:

* <!DOCTYPE html> là phần khai báo kiểu dữ liệu hiển thị là html để trình duyệt biết. Được khai báo và đặt trước tất cả các thẻ khác. Nó không phải là thẻ HTML.
* <html> và </html> là cặp thẻ nằm ngoài cùng và nó có nhiệm vụ là bao hết nội dung của trang web lại. Thẻ này là bát buộc.
* <head> và </head> là phần khai báo thông tin của trang web
* <title> và </title> nằm bên trong thẻ <head> và đây chính là khai báo tiêu đề cho trang web.
* <body> và </body> là thành phần quan trọng nhất, nó chứa nhưng đoạn mã HTML dùng để hiển thị trên website
* Các thẻ còn lại nằm trong thẻ <body> chính là các thẻ định dạng dữ liệu.

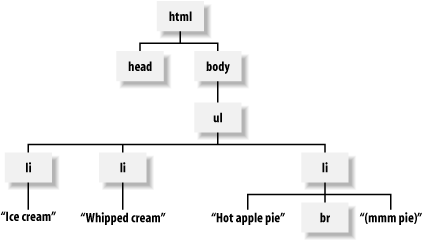
Như vậy trong một website chúng ta chia làm 2 phần chính:

* Phần head: Là nhũng khai báo thông tin cho trang web và ta đặt nó trong thẻ head.
* Phần body: Là phần hiển thị định dạng nội dung của trang web và ta đặt trong thẻ body.

# Trình duyệt đọc HTML như thế nào?

Khi mở tài liệu HTML bằng trình duyệt, trình duyệt sẽ truy tìm các cặp thẻ và chuyển nó thành văn bản và hiển thị theo ý muốn người tạo. the mô hình cây

Ví dụ, trình duyệt sẽ đọc nội dung thẻ <head></head> để đọc thông tin phụ của trang web. Khi đọc tới cặp thẻ <title></title> trình duyệt đọc nội dung và chuyển nó thành tiêu đề của trang web để người dùng biết.



Mô hình cây HTML DOM

Ví dụ hình trên trình duyệt đọc điểm đầu cuối html. Sau đó tiếp tục đọc điểm đầu cuối của head và body trong HTML. Tiếp tục như vây cho tới khi hết các thẻ. Mối quan hệ html và body là là quan hệ cha-con. Hình trên được coi như một cái cây- HTML DOM tree.

# Phân loại thẻ HTML

Số lượng thẻ HTML hiện nay hơn 190 loại. Trong đó dùng để lập trình Web thì chúng ta sẽ không sử dụng hết mà chỉ sử dụng các loại thẻ phổ biến.

## Phân loại theo thẻ loại

Với mô tả như trên, có thể phân loại HTML tags ra thành các loại:

* Thẻ theo cặp mở và đóng. Ví dụ như <title></title>
* Thẻ không có thẻ đóng: <img>, <hr>, <br>.

Một số thẻ không có thẻ đóng và sử dụng phổ biến như sau:

|  |  |
| --- | --- |
| Tên thẻ | Chức năng |
| <br> | Chèn một ngắt xuống dòng |
| <hr> | Tạo một dòng kẻ ngang |
| <img> | Chèn ảnh |
| <link> | Chèn link file cho <head></head> |
| <input> | Thành phần tạo form |
| <meta> | Chèn mô tả trong <head></head> |

## Phân loại theo chức năng

Theo mong muốn nhu cầu của các lập trình có thể phân chắc năng như sau:

* Thẻ định dang: <u>, <strong>, <i>, <b>,…
* Thẻ chèn ảnh: <img>
* Thẻ video-audio: <video>, <audio>
* Thẻ biểu mẫu: < form >
* Thẻ bảng biểu: <table>, <td>, <tr>,..
* Thẻ link: <a>
* Thẻ văn bản: <p>
* Thẻ tiêu đề: h1, h2, h3, h4,…
* Thẻ tạo menu: <ul> <li>,…

Để biết thêm các thẻ nên nhớ và dùng phổ biến chúng ta hãy xem bảng thống kê sau

# Các thẻ HTML phải biết dùng.

|  |  |
| --- | --- |
| Tên thẻ | Chức năng |
| a | Thẻ link |
| p | Thẻ định dạng văn bản |
| div | Thẻ khối |
| h1, h2, h3,… | Tiêu đề mục |
| img | Chèn ảnh |
| video | Chèn video |
| audio | Chèn audio |
| iframe | Chèn trang web nhỏ, video Youtube |
| Form | Tạo biểu mẫu |
| Button | Nút |
| Input | Là thành phần form |
| Table – td, tr, th, tbody | Chèn script |
| ul - li | Tạo menu |
|  |  |
| nav | HTML |
|  |  |

Các thẻ HTML hay được sử dụng nhiều nhất

# CSS là gì

Nếu HTML là một cơ thể, có head, có body. Thì CSS được ví như quần áo, mũ nón để làm đẹp cho cơ thể đó

**CSS** là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để**tìm và định dạng** lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). CSS đóng vai trò định dạng các phần tử trên website

# Cách nhúng mã CSS vào HTML

## Cách 1: Viết ngay trong thẻ

Tất cả các thẻ HTML đều có thuộc tính style. Thuộc tính này giúp ta viết mã CSS ngay trong thẻ

|  |
| --- |
| <h1 style="color:blue;">This is a Blue Heading</h1> |

Ưu điểm: Đây là cách viết CSS rất dễ dàng vì nó viết ngay trong thẻ HTML.

Nhược điểm: Khó sửa chữa và quản lý, Không có tính tái sử dụng mã CSS.

Lời khuyên đối với cách viết này nên hạn chế sử dụng. Vì khi mà viết như vậy, đối với trang web lớn sẽ hỗn độn mã CSS và HTML nên rất khó sửa chữa và bảo trì sau này.

## Cách 2: Sử dụng thẻ <style></style>

|  |
| --- |
| <!DOCTYPE html> <html> <head> <style> body {background-color: powderblue;} h1   {color: blue;} p    {color: red;} </style> </head> <body>  <h1>This is a heading</h1> <p>This is a paragraph.</p>  </body> </html> |

Ưu điểm: So với cách 1 thì mã CSS được gom lại một chỗ dễ sửa chữa hơn.

Nhược điểm: Vẫn rất khó quản lý và không có tính tái sử dụng CSS cho nhiều tài liệu HTML. Nếu sửa chữa phải tìm tới file HTML cần sửa.

## Cách 3: Viết mã CSS trong file .css

Cách này mã CSS viết trong file .css, khi đó ta chỉ cần dùng thẻ <link> là thẻ con của <head> gọi tới file CSS.

Mã CSS được gom chung một file và dùng thẻ link gọi tới file đó

|  |
| --- |
| <!DOCTYPE html> <html> <head>   <link rel="stylesheet" href="css/stylesHeader.css">  <link rel="stylesheet" href="css/styleSNavbar.css"> </head> <body>  <h1>This is a heading</h1> <p>This is a paragraph.</p>  </body> </html> |
|  |

Đây là cách người lập trình sử dụng phổ biến nhất và khắc phục được các nhược điểm của các cách trên

Ưu điểm: Mã CSS gom chung một hoặc nhiều file dễ dàng quản lý source code. Dễ dàng tái sử dụng lại mã CSS giữa các trang HTML

Nhược điểm: Nếu không quản lý các file tốt sẽ dẫn đến lộn xộn các file ở các thư mục khác nhau

Lời khuyên cho lập trình viên là chúng ta nên để chung các file trong cùng một thư mục, ở ví dụ trên chúng ta gom hai file “stylesHeader.css” và “styleSNavbar.css” và cùng thư mục “css”

Lưu ý: thư mục CSS ở trên là cùng cấp với file HTML (thư mục “css” và file .html cùng một folder)

# Cách viết mã CSS

Tùy cách nhúng CSS mà có cách viết khác nhau. Sau đây là cách viết phổ biến:

|  |
| --- |
| vùng chọn {     thuộc tính : giá trị;     thuộc tính: giá trị;     .....  } |

Hãy xem cách viết mã CSS với mỗi ví dụ sau

## Vùng chọn tên thẻ

Với vùng chọn là loại thẻ

|  |
| --- |
| h1 {  color:red;  } |

CSS sẽ tìm tất cả các thẻ h1 và tô đỏ chữ nội dung trong thẻ

## Vùng chọn là ID

Trong đoạn HTML có ID “firstname”cần viết mã:

|  |
| --- |
| #firstname {      background-color: yellow; } |

## Vùng chọn là Class

Trong đoạn HTML có class “intro”cần viết mã:

|  |
| --- |
| .intro {      background-color: yellow; } |

## Vùng chọn dạng A B

Với B nằm trong A (có thể B là con ruột, cháu, chắt,.. ) nằm sâu trong A. Trong tài liệu nói dễ hiểu như vậy.

## Vùng chọn dạng A->B

Với B là con trực tiếp của A (chỉ với con ruột),

## Vùng chọn với thuộc tính

Ví dụ: Viết mã CSS vớ thuộc tính type của thẻ input. Trong ví dụ sau CSS với type = text

|  |
| --- |
| input[type=text] {     width: 100%;     padding: 12px 20px;     margin: 8px 0;     box-sizing: border-box; } |

# Thuộc tính CSS đáng nhớ

|  |  |
| --- | --- |
| Thuộc tính | Gía trị phổ biến |
| color | Mã màu |
| background |  |
| border |  |
| padding |  |
| margin |  |
| clear | both |
| display | Block, none |
| position | Relative, absolute, fixed |
| float | Left, right |
| Text-aligin | Center |