

Lập trình website sử dụng c#



Mục lục

[Chương 1: CSS Display 2](#_Toc514705497)

[1. Thuộc tính display 2](#_Toc514705498)

[a. Block 2](#_Toc514705499)

[b. Inline 3](#_Toc514705500)

[c. Inline – block 3](#_Toc514705501)

[d. Table 4](#_Toc514705502)

[e. Inline – table 5](#_Toc514705503)

[f. Float 6](#_Toc514705504)

[Chương 2: Bộ chọn selector 8](#_Toc514705505)

[1. Selector là gì? 8](#_Toc514705506)

# Chương 1: CSS Display

## Thuộc tính display

Display là thuộc tính xác định kiểu hiển thị của các thành phần trong html.

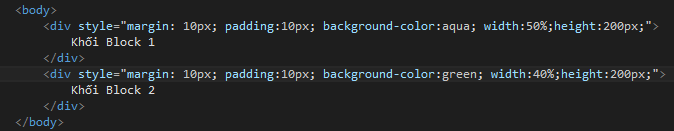
Một số kiểu hay sử dụng: block, inline, inline-block, table, inline-table.

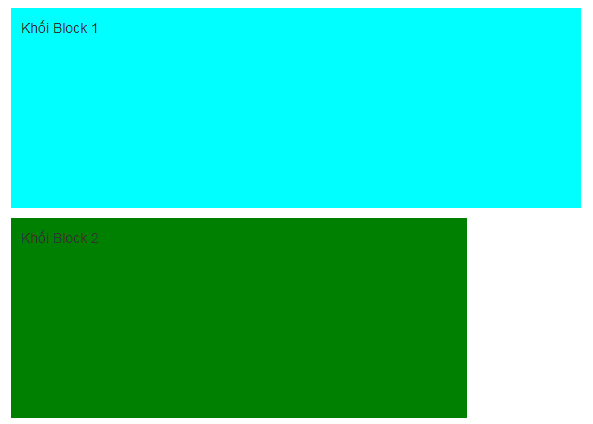
### Block

Thành phần hiển thị như một khối, khi sử dụng giá trị block, thành phần sẽ đứng một hàng độc lập so với thành phần trước và sau nó.

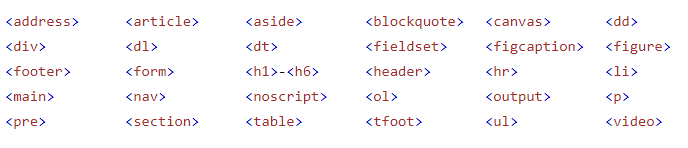
Có thể đặt giá trị tương đối (%) hoặc tuyệt đối (pt,px,..) cho thuộc tính width và height.

Thuộc tính margin và padding ảnh hưởng đến tất cả các mặt của thành phần.

Ví dụ:

Kết quả:

Các thẻ mặc định có display mặc định là block:

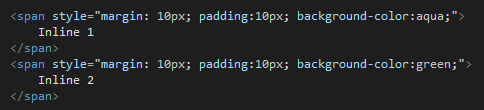


### Inline

Thành phẩn hiển thị nội tuyến, không ngắt dòng. Khi sử dụng giá trị inline các thành phần sẽ nối đuôi nhau.

Không nhận các giá trị của thuộc tính width và height.

Thuộc tính margin, padding chỉ ảnh hưởng theo chiều ngang.

Ví dụ:

Kết quả:

Các thẻ có display mặc định inline:

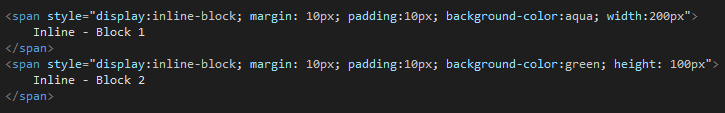
### Inline – block

Thành phần sẽ hiển thị như một khối, nhưng là một khối nội tuyến. Khi sử dụng inline – block thành phẩn các thành phần khối sẽ nối đuôi nhau.

Nhận giá trị của các thuộc tính width và height.

Thuộc tính margin và padding ảnh hưởng tất cả các mặt của thành phân.

Ví dụ:

Kết quả:

### Table

Thành phần được đối xử như một <table>, thường tích hợp các display:table-children để hiện thị các thành phần như một bảng.

Các thành phần con:

+ table-cell

+ table-column

+ table-colgroup

+ table-caption

+ table-row

+ table-row-group

+ table-header-group

+ table-footer-group

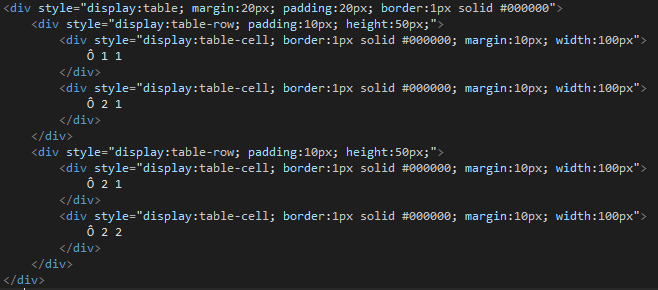
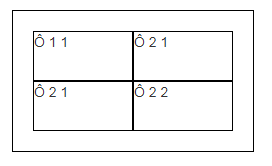
Display:table nhận các giá trị width, height và margin, padding như block.

Display:table-row không nhận giá trị width chỉ nhận giá trị heght và không nhận giá trị padding chỉ nhận margin.

Display:table-cell nhận giá trị width và height, không nhận giá trị margin chỉ nhận giá trị padding

Các bảng sẽ bắt đầu bằng dòng mới như block.

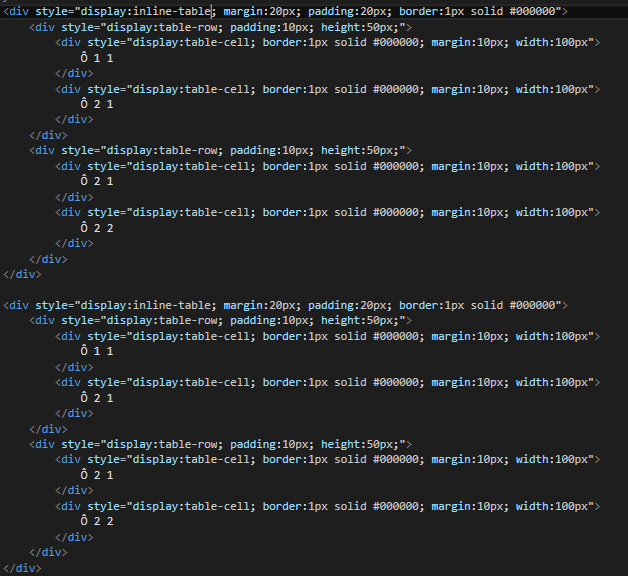
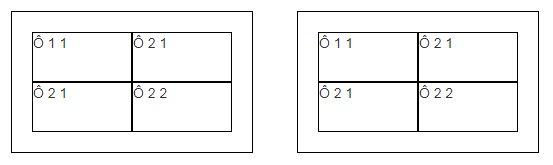
Ví dụ:

Kết quả:

### Inline – table

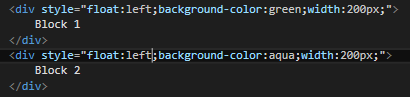
Giống với table nhưng các table sẽ được hiển thị nội tuyến cho phép các bảng nối đuôi nhau.

Ví dụ:

Kết quả:

### Float

Thành phần được hiển thị trôi nổi, cho phép các thành phần khác chiếm vùng không gian của nhau.

Ví dụ:

Kết quả:

# Chương 2: Bộ chọn selector và các hàm truy xuất selector trong jQuery

## Selector là gì?

Selector là các chuỗi mẫu được sử dụng để chọn các thành phần mà người dùng muốn tác động.

## Một số selector thường gặp

|  |  |  |
| --- | --- | --- |
| Selector | Ví dụ | Mô tả |
| \* | \* | Lấy tất cả các thành phần |
| #id | #myId | Lấy thành phần có id=’myId’ |
| .class | .content | Lấy tất cả các thành phần có class=’content’ |
| element | div | Lấy tất cả các thành phần div |
| element, element | div, p | Lấy tất cả các thành phần div và p |
| element element | div p | Lấy tất cả các thành phần p nằm trong thành phần div |
| element>element | div > p | Lấy tất cả các thành phần p có cha là div |
| [attribute] | [target] | Lấy tất cả các thành phần có thuộc tính target |
| [attribute=value] | [type=checkbox] | Lấy tất cả các thành phần có thuộc tính type = checkbox |
| :checked or :not:checked | input:checked | Lấy tất cả các thành phần input đã check |
| :disabled or :enabled | input:disabled | Lấy tất cả các thành phần input đã disabled |
| :empty | p:empty | Lấy tất cả các thành phần p không có children (kể cả text) |
| :first-child or :last-child | p:first-child | Lấy thành phần p đầu tiên trong thành phần cha |
| :nth-child | p:nth-child(2) | Lấy thành phần p là con thứ 2 của thành phần cha |
| tag:eq() | p:eq(2) | Lấy thành phần p có index = 2 trong mảng các thành phần p |
| tag:gt() | p:gt(2) | Lấy các thành phần p có index > 2 trong mảng các thành phần p |
| tag:lt() | p:lt(2) | Lấy các thành phần p có index <2 trong mảng các thành phần p |
| tag:even | li:even | Lấy các thành phần li ở vị trí lẻ |
| tag:odd | li:odd | Lấy các thành phần li ở vị trị chẵn |
| tag:first | p:first | Lấy thành phần p ở vị trí đầu tiên |
| tag:last | p:last | Lấy thành phần p ở vị trí cuối cùng |
| tag:first-of-type | li:first-of-type | Lấy thành phần con đầu tiên hoặc duy nhất trong thành phần cha |
| tag:last-of-type | p:last-of-type | Lấy thành phần con cuối cùng hoặc duy nhất trong thành phần cha |
| tag:parent | div:parent | Lấy thành phần có ít nhất một thành phần con (bao gồm cả text) |
| tag:empty | div:empty | Lấy thành phần không có thành phần con (kể cả text) |

Bảng 2.1: Bảng các selector thường gặp

Ngoài ra còn rất nhiều selector khác dành cho css, javascript, jquery, tham khảo thêm tại [bộ chọn Selector](https://hocwebchuan.com/reference/jquery/jquery_selectors.php).

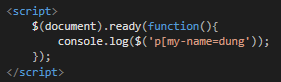
Nguyên lý viết selector:

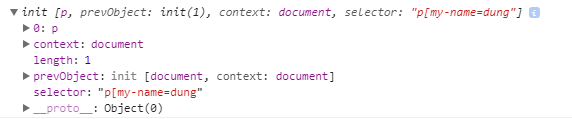
Ta có selector: element element … element thứ n

Ví dụ: ‘html body div p’

Nguyên lý từ phải qua trái là con đến cha. Theo ví dụ trên, nghĩa là p là con của div, body, html, còn div là con của body, html và body là con của html.

Ta có thể thay thế element bằng các selector trên như id, class, tên thẻ. Các thành phần không còn sự phân cấp (không có con) thường đặt ở sau cùng như các selector liên quan đến thuộc tính hoặc các thuộc tính tự định nghĩa.

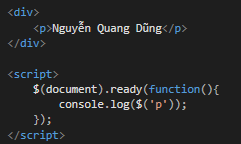
Ví dụ:

Kết quả:

## Các hàm truy xuất selector trong jQuery

### Hàm cơ sở

$() hay jQuery() là hàm cơ sở để lấy thành phần theo bộ selector truyền vào.

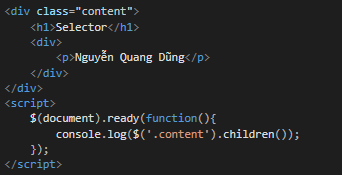
Ví dụ:

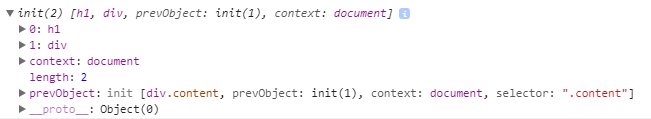
### Các hàm truy xuất đến thành phần con và truy xuất ngược đến thành phần cha

Sau khi lấy được đối tượng, ta có thể truy xuất đến các thành phần con hoặc truy xuất ngược đến các thành phần cha thông qua các hàm sau:

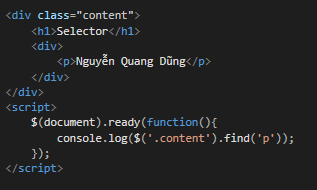
* Hàm children(): Lấy tất cả thành phần con cách cha 1 bậc.

Ví dụ:

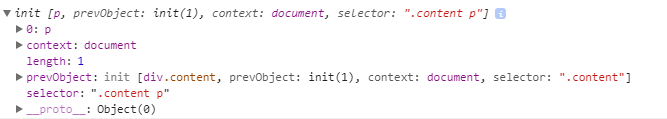
Kết quả: Lấy được 2 thành phần là h1 và div, không lấy thẻ p



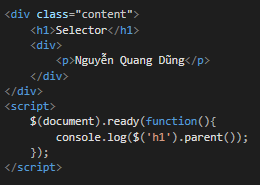
* Hàm find(‘selector’): Lấy tất cả các thành phần con phù hợp với bộ selector truyền vào

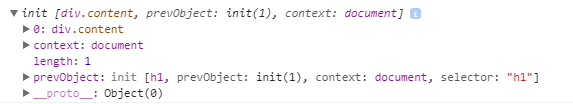
Ví dụ:

Kết quả: Lấy được thành phần p

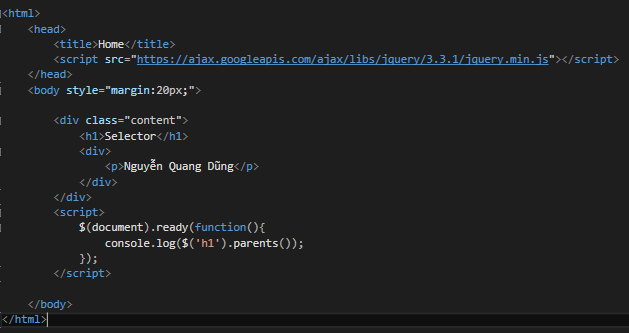


* Hàm parent(): Lấy thành phần cha cách thành phần con 1 bậc.

Ví dụ:

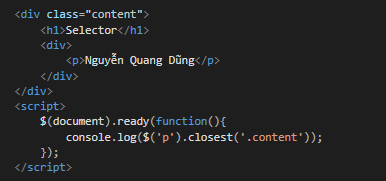
Kết quả: Lấy đc thẻ div gần nhất

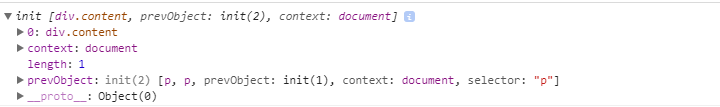
* Hàm parents(): Lấy tất cả các thành phần cha

Ví dụ:

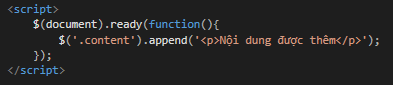
Kết quả: Lấy được tất cả thành phần cha

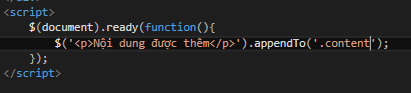
* Hàm closest(‘selector’): Lấy thành phần cha gần nhất phù hợp với selector truyền vào

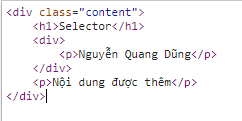
Ví dụ:

Kết quả:

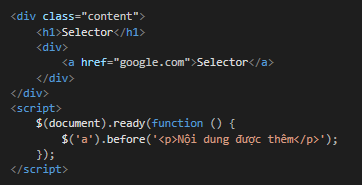
* Hàm appent(): Thành phần được chèn thêm nội dung, nội dung này được sắp xếp ở dưới cùng
* Hàm appentTo(): Chèn thêm nội dung ở dưới cùng vào thành phần đã có

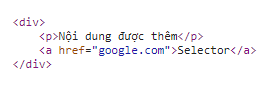
Ví dụ:



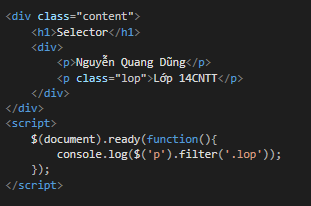
Kết quả:

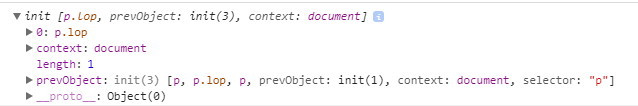
* Hàm before(): Thêm thành phần vào ngay trước thành phần đã có

Ví dụ:

Kết quả:

* Hàm filter(): Lọc các thành phần

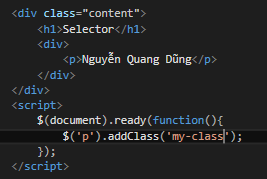
Ví dụ:

Kết quả:

### Các hàm xử lý thuộc tính của thành phần

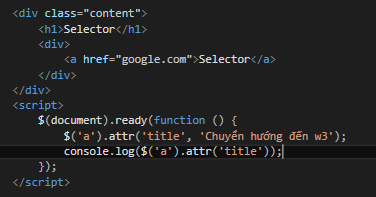
Sau khi lấy được các thành phần, ta có thể thay đổi các giá trị thuộc tính, hoặc thêm, xóa các thuộc tính của các thành phần đó.

* Hàm addClass() / removeClass(): Thêm / Xóa thuộc tính class của thẻ.

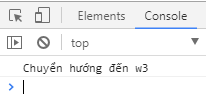
Ví dụ:

Kết quả:

* Hàm attr(‘attribute’): Lấy thuộc tính của thành phần (get)
* Hàm attr(‘attribute’,’value’): Đặt giá trị cho thuộc tính của thành phần (set)

Ví dụ:

Kết quả:



* Hàm css(): Được dùng để thêm một hoặc nhiều style cho thành phần, ngoài ra còn có thể sử dụng .css() để lấy giá trị style của thành phần.

Ví dụ: