**ID và Class trong CSS selectors**

CSS Selectors: là cách gọi được tới các elements (phần tử) trong

html để css

ID (identify): Định danh 1 đối tượng duy nhất

<h1 id = “1”> hi </h1>

~~<h1 id = “1> cc </h1> : không được trùng id~~

Khi select tới đối tượng sử dụng id thì sử dụng dấu #

#1 {css}

Class: Khi sử dụng thuộc tính css chung cho nhiều đối tượng khác nhau

<h1 class = “2”> hi </h2>

<h2 class = “2”> hi </h2>

<h2 class = “2”> hi </h2>

Khi select tới đối tượng sử dụng class thì sử dụng dấu .

.2 {css}

**Mức độ ưu tiên trong CSS**

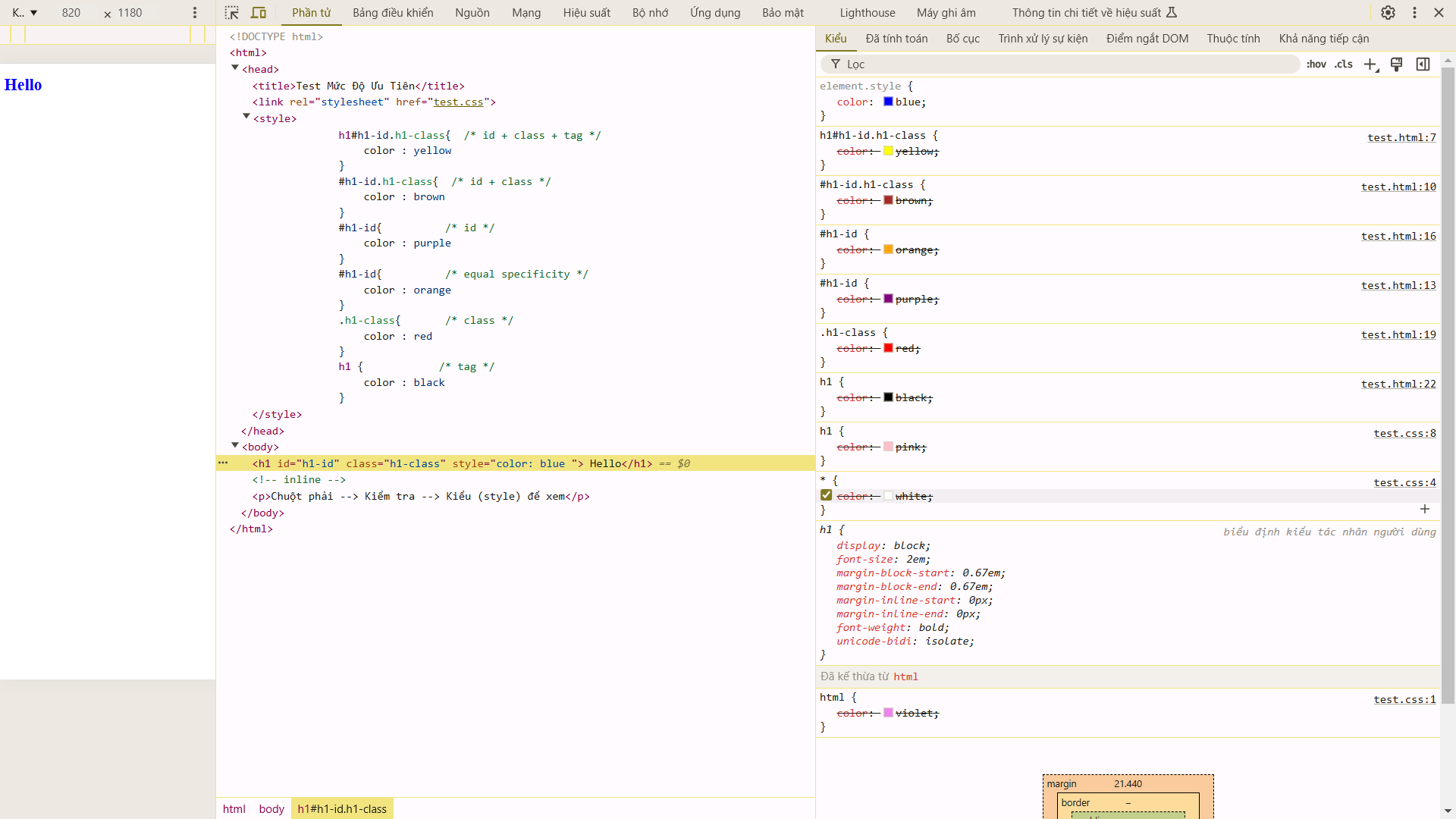
Priority : (ưu tiên)

1. Internal, External : Ưu tiên giống nhau, cách nào được gọi sau thì css đó được ưu tiên hơn
2. Inline
3. #id

Độ ưu tiên giảm dần

1. .class
2. tag (gọi thẳng thẻ h1 {css})
3. Equal specificity ( gọi selector giống nhau, lặp lại)
   1. Universal selector ( \* {}) : phổ cập chung cho tất cả thẻ
   2. Inherited: thừa hưởng (kế thừa) (html {})

Kiểm tra phần style trên web để xem phần thuộc tính ko được ưu tiên bị gạch đi



**!important**

**CSS Variable**

CSS Variable : Kỹ thuật đặt biến trong css

2 kỹ thuật: global và local

* Global:

Sử dụng lớp giả : :root{ }

Cách đặt biến: Sử dụng --tên biến : giá trị (đặt trong lớp giả)

(1 lớp giả chỉ tác dụng đối với 1 biến)

Sau đó gán biến cho thẻ bằng cách sử dụng css function – var

h1 {color : var(--tên biến)}

Công dụng của biến : để chỉnh sửa nhanh chóng bằng cách chỉ chỉnh sửa 1 nơi là ở biến

* Local : đặt biến ở trong thẻ luôn

h1 { --tên biến: giá trị;

color : var(--tên biến)}

**CSS Units**

CSS Units : là đơn vị được sử dụng trong CSS để giúp điều chỉnh kích thước của những đối tượng hiển thị trên web.

* Absolute Units (Đơn vị tuyệt đối)
* px (pixel):
* 1 px trong css tương ứng với 1 điểm ảnh trên màn hình ( với độ phân giải thấp)
* Còn đối với độ phân giải cao sẽ chiếm nhiều điểm ảnh hơn
* Thường sử dụng với thuộc tính width- height …

Ví dụ : Tạo khối:

html : <div></div>

css : width : 100 px

height : 100 px

background-color : red

Đặc tính của đơn vị tuyệt đối: kích thước sẽ cố định, không thay đổi do các yếu tố xung quanh tác động lên ( như kéo giản cửa sổ)

* Relative Units (Đơn vị tương đối): cần 1 nơi để phụ thuộc vào
* % (percent): 100 % = 16 px

Ví dụ : Tạo khối:

html : <div></div>

css : width : 50%

* Phụ thuộc vào thẻ chứa nó là thẻ body
* Thẻ body kế thừa chiều ngang của trình duyệt
* Nên thẻ body nó có chiều ngang = chiều ngang trình duyệt
* 50 % bằng nửa kích thước của thẻ body
* Khi tăng kích thước trình duyệt thì chiều ngang của khối sẽ tăng theo

Khi định nghĩa 1 đối tượng sử dụng giá trị tương đối thì đối tượng này sở hữu 1 kích thước không cố định, sẽ bị thay đổi kích thước khi đối tượng phụ thuộc thay đổi

* Rem (root em): phụ thuộc vào thuộc tính được định nghĩa ở thẻ html

Ví dụ

css : html {font – size : 40 px}

h1 {font – size : 1 rem}

* h1 có font – size là 40 px
* em (percent): phụ thuộc vào thẻ gần nhất chứa nó có thuộc tính liên quan

Ví dụ:

html :

<body>

<div style = “ font – size : 50 px”>

<h1>Hello</h1>

</div>

</body>

css : body { font –size : 100px}

h1 { font – size : 1 em}

* div chứa h1 và gần nhất nên h1 có font – size : 50px

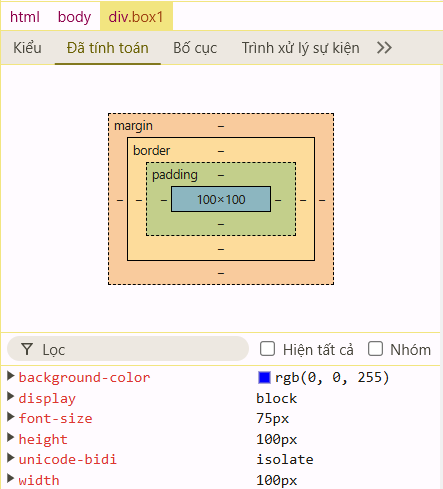
Thường sẽ sử dụng rem

* vw - vh (viewport) : khung hình của trình duyệt, không phụ thuộc vào thẻ chứa nó như %

1 vw = là 1 % chiều ngang khung hình

**Padding**

Padding: vùng đệm thêm



css : .box1{ width :100px

height : 100px}

html: <div class=”box1”></div>

100x100 (số này do css) : lớp nội dung, lúc này chính là kích thước của div (lớp nội dung và div là khác nhau)

Padding : lớp đệm ( có 4 hướng, dấu - là 0)

Khi thêm thuộc tính đệm thì kích thước của div tăng lên nhưng kích thước content ko đổi (div = padding + content)

Ví dụ : css: .box1 {padding-top: 10px}

* box1 : 100x110 , content: 100x100

Viết gọn : 1 giá trị : padding :10px ( trên dưới trái phải)

2 giá trị : padding : 10px (trên dưới) 20px (trái phải)

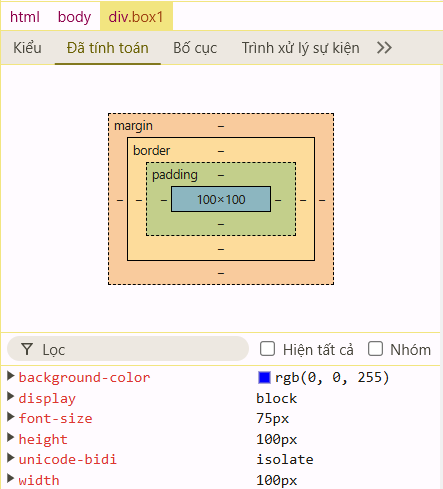
3 giá trị : padding : 10px (trên) 8px (phải trái) 20px (dưới)

4 giá trị : trên phải dưới trái ( cùng chiều kim đồng hồ)

**Border**

Border (nét liền): Lớp viền ở ngoài cùng ôm 1 elements

Margin : ko hẳn là ôm đối tượng nên k được ký hiệu là nét liền



css : .box1{ width :100px

height : 100px}

html: <div class=”box1”></div>

Thuộc tính : Độ dày border-width ( viết gọn)

Kiểu border-style : dashed (nét đứt), solid (nét liền)

inherit (kế thừa), initial (gtri ban đầu), none (hủy border

Màu border- color

Khi thêm thuộc tính đệm thì kích thước của div

Viết dài : border-top-width

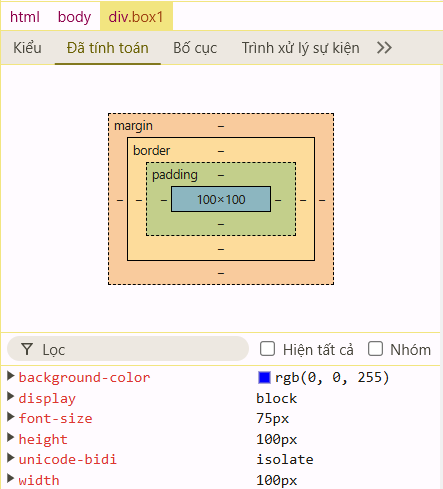
Mặc định style solid sẽ có width = 2 px

Viết gọn : border: width style color (4 hướng)

border-top: width style color ( có thể bỏ thuộc tính)

**Margin**

Margin : Giải quyết vấn đề mà lớp border & padding ko làm được là tạo khoảng cách giữa các đối tượng (border & padding chỉ làm dày bản thân)



Khi thêm thuộc tính margin thì ko làm tăng kích thước của div (lý do tại sao k được hiểu là lớp bao bọc toàn bộ elements, mà nó là chỉ là khoảng cách đẩy ra từ elements)

Để hiển thị theo chiều ngang bên trái float : left

Thuộc tính: Margin-top

Viết ngắn: y chang padding

**Tóm lại:**

Padding: để tạo khoảng cách giữ lớp context và lớp ngoài cùng (border)

Border: để tạo viền

Margin: Tạo khoảng cách cho đối tượng