

Tổng Quan Selectors Cơ Bản Để Chọn Phần Tử Css

Một selector được sử dụng để nhắm mục tiêu phần tử HTML cụ thể

Chọn theo type hay còn gọi là tag name

Giống như tên gợi ý của nó, nó khớp với type của HTML element

```
p {  
  color: green;  
}
```

Chọn Theo Class

Css không bị giới hạn chọn các phần tử theo giới hạn type của chúng. Như bạn biết html có thể có các attribute. Khi làm việc với html css thì class attribute là một trong những cách phổ biến nhất để chọn phần tử. Hãy xem xét ví dụ sau:

```
<p class='brand'>Sole Shoe Company</p>
```

Thẻ p trong ví dụ trên có class attribute ở trong thẻ mở p. class attribute được set đến brand. Để select elemnt này sử dụng css chúng ta có thể tạo bộ quy tắc với bộ chọn lớp bắt đầu dấu .

```
.brand {  
  
}
```

Để select một html element bởi class của nó sử dụng css thì chúng ta dùng . trước value class

Chúng ta có thể sử dụng class để chọn ra phần tử html áp dụng style nhưng nếu các phần tử html chỉ có 1 class thì chúng ta sẽ áp dụng rất nhiều style cho một class và làm cho code chúng ta rất khó tái sử dụng lại được, may mắn thay chúng ta có thể thêm nhiều class đến các phần tử html và các class cách nhau space. Xem ví dụ bên dưới áp dụng style đến thẻ vừa có màu xanh vừa đậm

```
.green {  
  color: green;  
}  
  
.bold {
```

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

```
<h1 class='green bold'> ... </h1>
```

Chọn Theo ID

Nếu một phần tử html cần được tạo kiểu duy nhất, chúng ta có thể có thể thêm một id(identified) attribute

```
<h1 id='large-title'> ... </h1>
```

Trái ngược với class có nhiều giá trị và có thể sử dụng lại trong trang html, một id của phần tử chỉ có một giá trị duy nhất và có thể chỉ được sử dụng một lần trên một trang. Để select phần tử theo id, chúng ta thêm trước value id dấu #. Lấy ví dụ html trên được select theo id như sau:

```
#large-title {
}
```

Hiểu Đơn Vị px

px là đơn vị độ dài tuyệt đối, chúng không phụ thuộc vào bất cứ điều gì khác, và thường được coi là có cùng kích thước. Ví dụ về đơn vị tuyệt đối như m, cm... Hầu hết các đơn vị tuyệt đối hữu ích khi được sử dụng để in thay vì xuất ra màn hình, px sử dụng là đơn vị đo lường nội dung xuất ra màn hình.

Tìm hiểu value của màu sắc

Các cách thể hiện value của màu sắc

1. Color keywords

Chính là tên màu sắc bằng tiếng anh như màu đỏ: red, xanh(green), cách này không phải là cách thông dụng khi phát triển web trên môi trường production bởi vì nó chỉ dễ hiểu cho người dùng và gây khó khăn khi sử dụng nó trong code.

2. Hexadecimal RGB values

Loại value tiếp theo bạn có thể gặp là mã thập lục phân. Hệ thập lục phân sử dụng 16 kí tự bao gồm **0123456789abcdef**. Mỗi color value sẽ bao gồm dấu # đầu tiên và 6 kí tự sau đó(2 kí tự đầu đại diện cho màu đỏ (R), 2 kí tự tiếp theo đại diện cho màu xanh lá cây(G), 2 kí tự cuối cùng đại diện cho màu xanh da

trời(B)); Mỗi kí tự có thể là 1 trong 16 kí tự thập lục phân vì vậy sẽ có $16 \times 16 = 256$ cặp kí tự đại diện cho R hoặc G hoặc B.

3. RGB Function values

Để tạo trực tiếp giá trị RGB, hàm rgb nhận 3 giá trị tham số đại diện cho red, green, blue. Sự khác biệt với biểu thị bằng kí tự thập lục phân là mỗi tham số đại diện cho màu sắc không biểu thị bằng 2 kí tự thập lục phân mà được biểu thị bằng số từ 0 đến 255

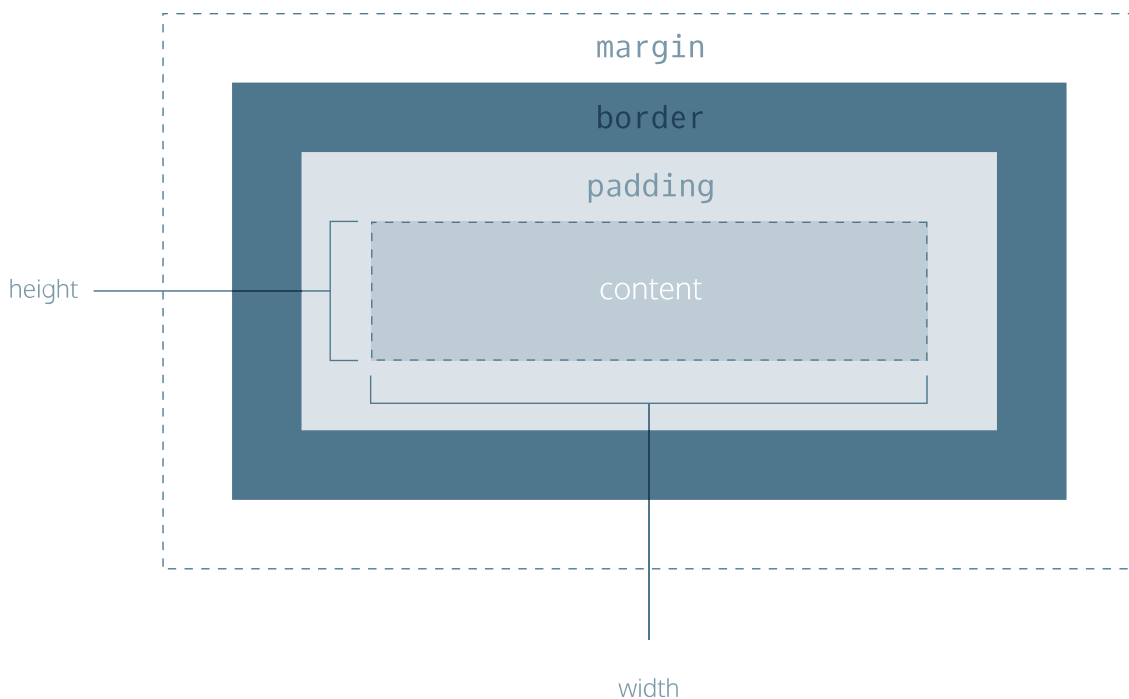
4. HSL Function values

H: Màu cơ bản của màu sắc, cái này nhận từ 0 đến 360 S: Màu sắc bão hòa như thế nào? nhận giá trị từ 0 đến 100% L: Màu sắc sáng hay nhạt? nhận giá trị từ 0 đến 100%, 0 là không có ánh sáng (màu đen hoàn toàn), 100 là sáng đầy đủ (màu trắng hoàn toàn)

Tìm Hiểu Về Box Model

Mô hình hộp bao gồm tập hợp các thuộc tính xác định các phần của phần tử chiếm không gian trên trang web. Mô hình hộp bao gồm kích thước của vùng nội dung (width và height), phần đệm của phần tử(padding), đường viền của phần tử(border) và phần bên ngoài với phần tử khác(margin)

1. width and height: chiều rộng và chiều cao của vùng nội dung
2. padding: Khoảng cách giữa vùng nội dung và đường viền
3. border: độ dày, kiểu dáng, màu sắc của đường viền hộp
4. margin: Khoảng cách giữa đường viền và cạnh ngoài của phần tử



Height and Width

Nội dung của phần tử có 2 chiều: chiều rộng và chiều cao, thuộc tính `height`, `width` có thể được sử dụng để sửa đổi kích thước mặc định của box

```
p {  
  height: 80px;  
  width: 240px;  
}
```

border

border là đường bao quanh một phần tử, giống như một khung bao quanh một bức tranh, border có thể được thiết lập với **width**, **style**, **color**

1. width: Độ dày của đường viền (có thể sử dụng px hoặc các từ khóa thin, medium, thick)
2. style: Thiết kế đường viền, trình duyệt cung cấp các loại style khác nhau
3. color: màu sắc đường viền

```
p {  
  border: 3px solid coral;  
}
```

padding

Khoảng cách giữa nội dung của hộp và đường viền của hộp được gọi là phần đệm. Bạn có thể tưởng tượng padding giống như không gian giữa bức tranh và khung bao quanh nó

```
p.content-header {  
  border: 3px solid coral;  
  padding: 10px;  
}
```

Đoạn code trên đặt khoảng cách 10px giữa nội dung của văn bản và đường viền trên cả 4 hướng: top, right, bottom, left. Thuộc tính padding thường được sử dụng để mở rộng background color và làm cho nội dung bớt chật chội hơn

Nếu bạn muốn nói rõ về padding ở mỗi hướng khác nhau, bạn có thể sử dụng padding nối với các hướng bằng kí tự -

1. padding-top
2. padding-right
3. padding-bottom
4. padding-left

Mỗi thuộc tính chỉ ảnh hưởng đến phần đệm ở mỗi bên nội dung của hộp, giúp bạn linh hoạt hơn trong việc tùy chỉnh

```
p.content-header {  
  border: 3px solid fuchsia;  
  padding-bottom: 10px;  
}
```

Viết tắt padding

1. 4 Values

Cả 4 giá trị khác nhau, với cách viết này giúp bạn xác định padding trên 1 dòng theo chiều kim đồng hồ bắt đầu từ top right bottom left

```
p.content-header {  
  padding: 6px 11px 4px 9px;  
}
```

2. 3 Values

Nếu giá trị trái-phải giống nhau ta có thể viết lại như sau

```
p.content-header {  
  padding: 5px 10px 20px;  
}
```

3. 2 Values

Nếu top, bottom bằng nhau và left, right bằng nhau, ta có thể viết lại như sau

```
p.content-header {  
  padding: 5px 10px;  
}
```

4. 1 Values

4 giá trị như nhau ta có thể viết lại

```
p.content-header {  
  padding: 10px;  
}
```

margin

Chính là khoảng trống bên ngoài hộp. Phân biệt rõ margin và padding, cách sử dụng margin tương tự padding

Thay Đổi Box Model

Tại sao thay đổi Box Model

Xem xét ví dụ sau:

```
h1 {  
  border: 1px solid black;  
  height: 200px;  
  width: 300px;  
  padding: 10px;  
}
```

Kích thước box thực tế sẽ bao gồm padding và border bởi vậy: width thực tế = width + padding x 2 + (border width)x2 = 300 + 20 + 2 = 322px height thực tế = height + padding x 2 + (border width)x2 = 200 + 20 + 2 = 222px

Điều này sẽ gây khó khăn cho căn chỉnh layout, lấy ví dụ bạn set width = 50% và xếp 2 phần tử trên 1 hàng, nếu bạn thêm padding hoặc border thì sẽ không đủ không gian cho 2 phần tử nên mô hình hộp thay thế được ra đời

Box Model: Content-Box

Mặc định box-sizing: content-box nghĩa là width được xác định dựa vào nội dung(content box)

Box Model: Border-Box

Chúng ta có thể đặt lại toàn bộ mô hình hộp bằng cách xác định value border-box

```
* {  
  box-sizing: border-box;  
}
```

Code phía trên đặt lại box model đến border-box cho tất cả các phần tử HTML. Mô hình hộp này tránh được các vấn đề xảy ra với mô hình hộp cũ.

Trong mô hình hộp này chiều cao và chiều rộng sẽ cố định, độ dày đường viền và phần đệm sẽ bao gồm bên trong hộp, hiểu đơn giản thì kích thước width, height bạn set sẽ kéo dài đến 2 cạnh border

```
* {  
  box-sizing: border-box;  
}  
  
h1 {  
  border: 1px dashed #4f768e;  
  height: 150px;  
  width: 200px;  
  padding: 20px;
```

```
}
```



Trả Lời Câu Hỏi Khi Nào Sử Dụng Class, ID

ID:

Trên 1 page chỉ có 1 id với value nhất định Trên một element cũng chỉ có 1 ID với value nhất định

class:

Trên 1 page có nhiều class cùng value Trên 1 element có nhiều class value cách nhau space

ID dùng cho các trường hợp form focus, click điều hướng cùng page hoặc các phần ngữ nghĩa như header, main, footer class giúp tái sử dụng code bằng cách chia nhỏ css theo class chịu trách nhiệm

Trả lời câu hỏi khi nào sử dụng display: inline-block
