

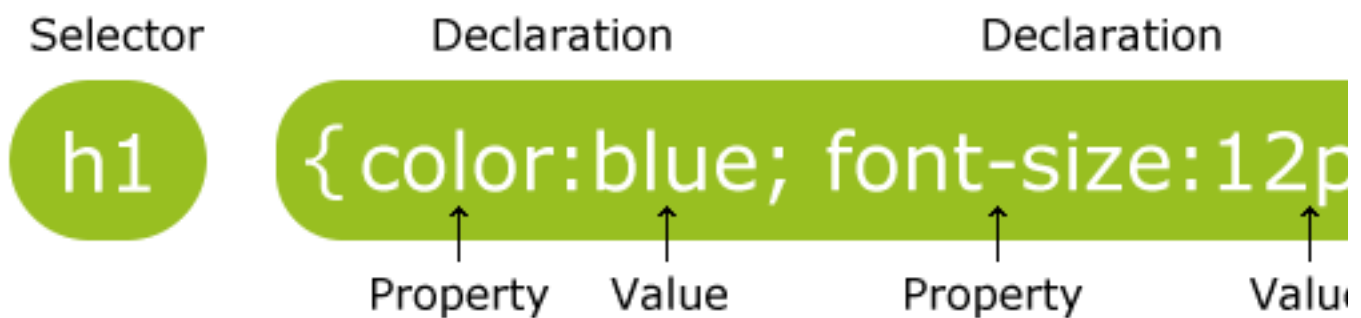
Cú pháp CSS

CSS được tạo thành từ các Style Rule. Các rule này được thông dịch bởi trình duyệt và sau đó được áp dụng tới các phần tử tương ứng trong tài liệu của bạn. Style Rule bao gồm ba phần:

- **Selector: (hiểu nôm na là bộ chọn phần tử)** Một selector là một thẻ HTML mà tại đó một bạn áp dụng một style cho nó. Đó có thể là bất kỳ một thẻ HTML nào, như thẻ div, thẻ span, thẻ p, ...
- **Property: (thuộc tính)** Là property của một kiểu thuộc tính của thẻ HTML. Nói một cách đơn giản thì tất cả các thuộc tính trong HTML được chuyển đổi thành các CSS property. Đó có thể là *color*, *border*....
- **Value:** Là các giá trị được gán cho các property. Ví dụ, *color* có thể có các giá trị là *red* *#F1F1F1* etc.

Dưới đây là cú pháp CSS chung:

```
selector { property: value }
```



Trong ví dụ trên, selector trỏ tới phần tử HTML là h1. Đây là phần tử mà mình muốn tạo style cho nó.

Khối khai báo Declaration có thể bao gồm một hoặc nhiều khối và được đặt phân biệt nhau bởi dấu chấm phẩy.

Mỗi khối khai báo này bao gồm một tên của CSS property (trong ví dụ là color và font-size) và một giá trị (trong ví dụ là blue và 12px) được gán cho property đó.

Bạn cần chú ý một điều rằng, một khai báo CSS luôn luôn kết thúc với một dấu chấm phẩy và phần khai báo này sẽ được bao quanh bởi cặp dấu ngoặc ôm.

Trong ví dụ tiếp theo, tất cả các phần tử

sẽ được căn chỉnh theo lề trái và có màu là màu đỏ:

```
p { color: red; text-align: left; }
```

Các loại Selector trong CSS (hoặc CSS Selector) Element Selector trong CSS

Hiểu nôm na là bộ chọn phần tử. Element Selector này sẽ chọn các phần tử dựa trên tên của chúng. Như trong ví dụ trên, chúng ta chọn phần tử p (hay là thẻ p – trong HTML, mỗi thẻ là một phần tử) và áp dụng style cho phần tử này. Có nhiều cách chọn phần tử khác nhau, và tùy theo mục đích thì bạn có thể sử dụng Type Selector, Universal Selector, hoặc Descendant Selector.

```
p { text-align: center; color: red; }
```

Type Selector trong CSS

Cũng giống như trong ví dụ trên, ví dụ sau sẽ áp dụng màu cho thẻ h1.

```
h1 { color: #36CFFF; }
```

Universal Selector trong CSS

Nếu bạn muốn áp dụng một style nào đó cho tất cả các thẻ HTML trong tài liệu, bạn có thể sử dụng cú pháp giống như dưới đây.

```
* { color: #000000; }
```

Descendant Selector trong CSS

Giả sử bạn chỉ muốn áp dụng style này cho một phần tử mà là con của phần tử khác (phần tử con là phần tử mà nằm bên trong phần tử khác). Ví dụ sau sẽ áp dụng style cho phần tử vietjack chỉ khi nào nó nằm bên trong thẻ <div>:

```
ul em { color: #000000; }
```

Class Selector trong CSS

Tương tự như ID Selector, bạn cũng có thể áp dụng các Style Rule cho các phần tử dựa trên thuộc tính class của phần tử đó.

Để chọn các phần tử với một class cụ thể, bạn viết một dấu chấm (.) ở trước tên class của phần tử đó.

Ví dụ dưới đây sẽ định nghĩa Style Rule cho tất cả phần tử HTML với class= "center":

```
.center { text-align: center; color: red; }
```

Với trường hợp có nhiều phần tử với cùng một class, nếu bạn chỉ muốn áp dụng style rule cho một phần tử nào đó, bạn viết tên phần tử đó trước cú pháp trên, như sau:

```
div.center { text-align: center; color: red; }
```

Các phần tử HTML có thể tham chiếu tới nhiều class. Do đó, bạn có thể áp dụng nhiều Style Rule cho một phần tử (chẳng hạn div) với các class = "center" (như trên) và class = "black" (để xác định màu đen chẳng hạn), như sau:

```
<div class="center black"> DIV này sẽ được tạo style bởi hai class là center và black. </div>
```

ID Selector trong CSS

ID Selector sử dụng thuộc tính id của một phần tử HTML để chọn một phần tử cụ thể. Trong một webpage, mỗi phần tử nên có một id duy nhất, vì thế ID Selector được sử dụng để lựa chọn một phần tử duy nhất.

```
#black { color: #000000; }
```

Để lựa chọn một phần tử với một id cụ thể, bạn viết một ký tự # (đọc là ký tự hash) ở trước id của phần tử đó.

```
h1#black { color: #000000; }
```

Ví dụ dưới đây sẽ áp dụng style rule cho phần tử có id= "para1";

```
#para1 { text-align: center; color: red; }
```

Bạn cũng có thể xác định cụ thể hơn như trong ví dụ sau:

```
div#para1 { text-align: center; color: red; }
```

Style Rule này sẽ chỉ áp dụng cho các phần tử div có id là para1.

Ngoài ra, nếu bạn muốn áp dụng style rule cho một phần tử mà là con của (hay nằm trong) phần tử khác, bạn có thể sử dụng:

```
#para1 div { text-align: center; color: red; }
```

Style Rule này sẽ áp dụng cho phần tử div là phần tử con của phần tử có id là para1.

Child Selector trong CSS

Phần trên bạn đã tìm hiểu về Descendant Selector. Ngoài ra, còn có một kiểu Selector nữa mà khá giống với kiểu này. Bạn theo dõi ví dụ:

```
body > p { color: #000000; }
```

Style Rule này sẽ được áp dụng cho tất cả các phần tử p mà trực tiếp là con của phần tử body. Các phần tử p khác nằm bên trong các phần tử như div hoặc td sẽ không bị tác động.

Attribute Selector trong CSS

Bạn cũng có thể áp dụng style cho các phần tử HTML có các thuộc tính cụ thể nào đó. Ví dụ sau sẽ áp dụng style rule cho tất cả phần tử input nào mà có thuộc tính type= "text".

```
input[type = "text"]{ color: #000000; }
```

Phương thức này đem lại lợi thế: phần tử <input type = "submit" /> sẽ không bị tác động bởi style rule trên, và color sẽ chỉ được áp dụng cho trường text.

Sau đây là ví dụ về một số rule áp dụng cho Attribute Selector:

- **p[lang]** : lựa chọn tất cả phần tử p mà có thuộc tính lang.
- **p[lang="fr"]** : lựa chọn tất cả phần tử p mà có thuộc tính lang có giá trị chính xác là fr.
- **p[lang~="fr"]**: lựa chọn tất cả phần tử p mà có thuộc tính lang có giá trị có chứa từ fr.

- **p[lang]="en"** : lựa chọn tất cả phần tử p mà có thuộc tính lang có giá trị chính xác là en hoặc bắt đầu với en.

Multiple Style Rule trong CSS

Bạn có thể định nghĩa nhiều style rule cho cùng một phần tử. Tuy nhiên, thay vì viết nhiều Style Rule riêng biệt để áp dụng cho chỉ một phần tử, thì bạn có thể viết gộp như sau:

```
h1 { color: #36C; font-weight: normal; letter-spacing: .4em;
margin-bottom: 1em; text-transform: lowercase; }
```

Như trên, tất cả các cặp property và value được phân biệt bởi *dấu chấm phẩy* (;). Tùy theo phong cách của mình, bạn có thể viết chúng trên cùng một dòng hay trên nhiều dòng. Nhưng tốt hơn hết, để cho code của bạn dễ đọc và dễ duy trì, bạn nên viết từng cặp trên từng dòng riêng biệt.

Khi lần đầu tiên nhìn vào ví dụ trên, có thể bạn sẽ không hiểu các property này có tác dụng gì. Hãy bình tĩnh, bởi vì chúng ta mới bắt đầu học mà. Nếu bạn muốn biết ngay tác dụng của chúng, bạn có thể tham khảo chi tiết về tất cả property trong CSS tại đây..

Grouping Selector trong CSS

Có một tình huống xảy ra là giả sử bạn muốn áp dụng cùng một style rule cho nhiều phần tử khác nhau. Nếu theo những gì đã học ở trên, bạn phải viết lần lượt từng style rule này cho từng phần tử. Như vậy cũng được, nhưng hãy tưởng tượng nếu số lượng phần tử là lớn và việc lặp đi lặp lại này có thể sẽ tốn khá nhiều thời gian. Có cách nào tốt hơn không. Có.

Bạn hãy nhóm tất cả các phần tử này lại, và sau đó áp dụng style rule chung mà bạn muốn áp dụng cho chúng. Đó cũng chính là ý nghĩa của Grouping Selector trong CSS. Bạn theo dõi ví dụ sau:

```
h1, h2, h3 { color: #36C; font-weight: normal; letter-spacing: .4em;
margin-bottom: 1em; text-transform: lowercase; }
```

Style rule này sẽ áp dụng cho các phần tử h1, h2, và h3. Bạn cần chú ý là giữa các phần tử này phải có dấu phẩy (,) nhé. Đó là cú pháp chung. Sử dụng Grouping Selector sẽ giúp bạn tiết kiệm được thời gian và tối thiểu hóa được lượng code mà bạn cần viết.

Tương tự, nếu bạn muốn áp dụng cho nhiều class, thì bạn có thể sử dụng nhiều Class Selector trong CSS như ví dụ dưới đây:

<http://vietjack.com/>

Trang chia sẻ các bài học online miễn phí

```
#content, #footer, #supplement { position: absolute; left: 510px; width: 200px; }
```

Comment trong CSS

Comment đôi khi đóng vai trò khá quan trọng trong lập trình. Nó cung cấp nhiều thông tin hữu ích cho người đọc, giúp họ (hay chính bạn khi cần xem lại code) hiểu được những gì bạn đang thực hiện.

Comment trong CSS bắt đầu với `/*` và kết thúc với `*/`. Comment cũng có thể trải rộng trên nhiều dòng. Ví dụ:

```
p { color: red; /* vi du ve comment nam tren mot dong */ text-align: center; } /* vi du ve comment nam tren nhieu dong */
```

Ví dụ

```
<!DOCTYPE html> <html> <head> <style> p { color: red; /* Vi du ve comment tren mot dong */ text-align: center; } /* Vi du ve comment tren nhieu dong trong CSS */ </style> </head> <body> <p>Hoc CSS co ban tai VietJack!</p> </body> </html>
```

Kết quả là:

Hoc CSS co ban tai VietJack!