**Tìm hiểu về Mixins, functions, extends trong SASS**

## Mixins trong Sass

**Mixins**là cách mà Sass dùng để dùng lại 1 đoạn code lặp đi lặp lại nhiều lần.

Cú pháp khai báo mixin như sau:

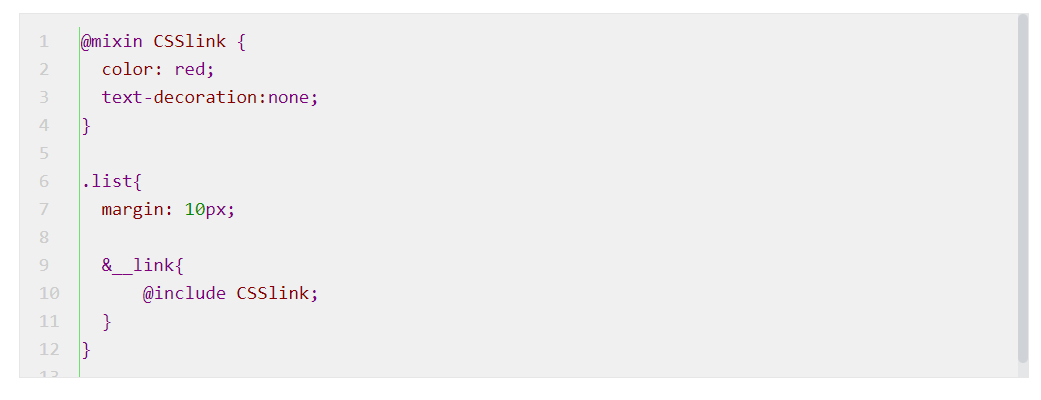
**@mixin tênMixin {**

  ......

**}**

Quay lại ví dụ của bài học trước [VARIABLES VÀ NESTING TRONG SASS](https://www.howkteam.vn/redirect?Id=CDPvobsj8EgqSWOjQulkj7KLp%2fXKC0Hu8VwCLICnWtTdGxSJO%2fvBjGaBeINFHJguJgXooY%2bA8vhgoFKuChRvX30d8K94mJsaUU2VuUfK%2bPp3E2LgOTIU%2fB9yzC6aDAMQDO1q9HcDEtZ9wQXvQlvKAS7adep6tGVlA4l0Zb2vCELrN5hpvTLBdkl5fj3ezCEh).

Ta sử dụng mixin để khai báo như sau:



Để khai báo mixin ta sử dụng **@include** như trên, kết quả hiện ra trên **browser** không có gì thay đổi

**@mixin tênMixin**(**$variables1**, **$variables2**, ...) {  
    
}

Sử dụng **mixins** cũng hoàn toàn có thể kết hợp với sử dụng biến. Trong ví dụ trên ta tiếp tục thay đổi 1 chút:



Sử dụng tham số (biến)**$color,** và **$status**với các giá trị đưa vào là **red** và **none** (chú ý phải có $ trước tên tham số nhé )

Nếu các bạn đã từng học qua về function của các ngôn ngữ lập trình như [**Java**](https://www.howkteam.vn/redirect?Id=CDPvobsj8EgqSWOjQulkj7KLp%2fXKC0Hu8VwCLICnWtTCf0%2b1w1e8nqgbLi99hbzBA3u1VPG44GP7i4yN8IorEHckyOHvhIf4Aj1GO5oJYrM%3d), **Javascript**, [**C#**](https://www.howkteam.vn/redirect?Id=CDPvobsj8EgqSWOjQulkj0l7AcPKsTUVzMX9%2b%2fZNlVc1qcsbpV9Ly9kqZX4r8Hct1SoB7QYmHOYcTaJvkjOVUQ%3d%3d), **C**, [**C++**](https://www.howkteam.vn/redirect?Id=CDPvobsj8EgqSWOjQulkj7KLp%2fXKC0Hu8VwCLICnWtQ7ETwTu7CirW%2bhsSye3P82vv385oh9NdWOW0lm7ZmHcA%3d%3d), .. thì có lẽ phần sử dụng tham số đã rất quen thuộc với các bạn rồi.

## Functions trong Sass

Việc sử dụng **Function**trong Sass dùng để khai báo các giá trị của property với các công thức toán học (thường là chỉ sử dụng cho số đo như **width**, **margin**, **padding**,…)

**@function TênFunction**(**$variable1**, **$variable2**, ...) {

**@return Value**;

}

Tiếp tục quay lại ví dụ ban đầu ta sử dụng Function như sau:



Ở đây ta sử dụng hàm **calMargin**với 2 tham số là **a** và **b** (chú ý là phải có $ trước tên tham số) và có giá trị đầu vào là 2 và 5, giá trị output của nó là **2\*5 = 10**, cuối cùng **10\*1px = 10px**, ta được kết quả hiển thị ra màn hình không có gì thay đổi.

## Extends trong Sass

**Extends**trong Sass cũng tương tự như mixins.

**%tênExtend** {

    .....

}

Ta tiếp tục sử dụng Extend trong ví dụ trên:



Kết qua cho ra không có gì thay đổi so với lúc trước.

Sự khác biết giữa **extends** và **mixins**chính là cách chúng compile sang CSS. Sử dụng extends thì code sẽ được compile 1 cách ngắn gọn hơn và không bị lặp lại, còn mixins thì sẽ copy lặp đi lặp lại đoạn code mà nó chứa.

Sử dụng ví dụ minh họa sau để các bạn có thể thấy được sự khác biệt:

Trong ví dụ trên ta khai báo thêm **pseudo-class ::after** của class**.list**.

* **Trường hợp 1:** Sử dụng **Mixin**
* 

(Sau khi compiled)

Ta có thể thấy là đoạn code trong phần mixin được lặp lại 2 lần khi khai báo **.list\_\_link** và **.list::after**

* **Trường hợp 2:**Sử dụng **Extends**
* 

(Sau khi compiled)

Ta thấy đoạn code trong **extends**chỉ sử dụng 1 lần và khai báo chung cho nhiều class.

Để nói là sử dụng **Extend**hay **Mixin** tốt hơn thì cũng không quan trọng, vì chúng ta chủ yếu quan tâm đến câu trúc file code của chúng ta (Sass) còn việc **compile** sang CSS nó như thế nào thì không cần quá quan tâm. Nên bạn thấy thoải mái khi sử dụng cái nào hơn thì sử dụng cái đó.