Mục Lục Báo Cáo SASS

[**I.** **Giới thiệu** 2](#_Toc21655844)

[**1.** **SASS là gì?** 2](#_Toc21655845)

[**Tại sao chúng ta nên sử dụng SASS?** 2](#_Toc21655846)

[**II.** **Tiến hành cài đặt SASS và demo ví dụ mẫu** 2](#_Toc21655847)

[**1.Cài Đặt** 2](#_Toc21655848)

[**2.Tạo demo mẫu** 3](#_Toc21655849)

[**III.** **Một số tính năng của SASS** 7](#_Toc21655850)

[**1.Sử dụng biến** 7](#_Toc21655851)

[**2.Tính năng xếp chồng trong SASS** 8](#_Toc21655852)

[**3.Tính năng sử dụng lại code** 8](#_Toc21655853)

1. **Giới thiệu**
2. **SASS là gì?**

* SASS là một CSS prepocessor
* Phải chảy qua một lệnh chuyển đổi thì SASS hoặc SCSS mới chuyển sang CSS
* Có 2 loại đuồi mà SASS hỗ trợ là:sass và scss

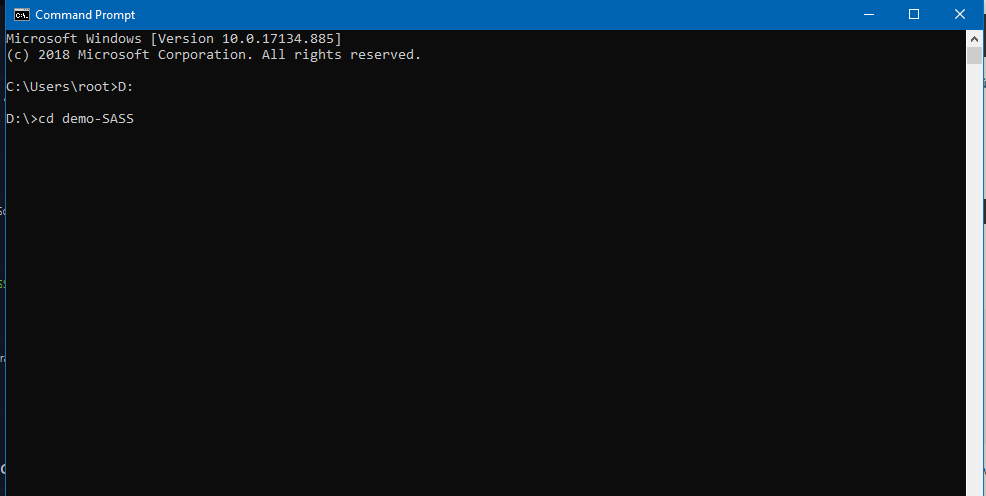
## **Tại sao chúng ta nên sử dụng SASS?**

Nó có thể được sử dụng để lập trình và nhập vào css, tự động nén các tập tin để có thể tiết kiệm dụng lượng một cách tốt nhất

1. **Tiến hành cài đặt SASS và demo ví dụ mẫu**

## **1.Cài Đặt**

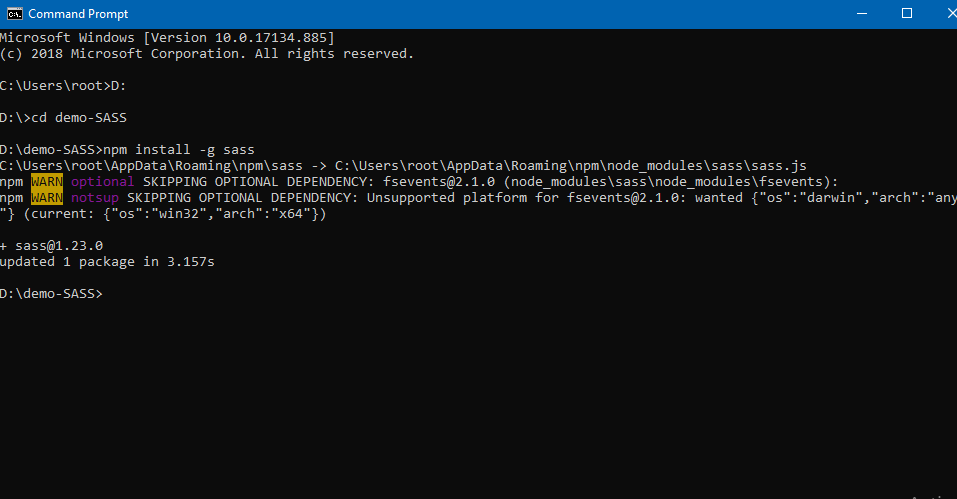
Từ giao diện màn hình Command Prompt(cmd) ta gọi câu lệnh để cài đặt SASS, đảm bảo rằng bạn **cd** đến đúng project của mình.



# Hình 1:Mở file nguồn chứa project

Tiếp theo chúng ta tiến hành chạy lệnh để cài đặt SASS

**Dòng lệnh cài đặt SASS: npm install –g sass**

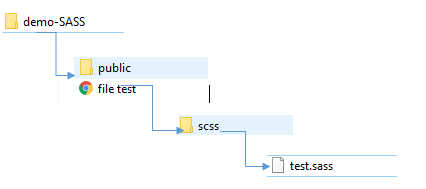


# Hình 2:Cài đặt SASS thành công

## **2.Tạo demo mẫu**

Tiếp theo chúng ta tạo một thu mục có cấu trúc như sau:

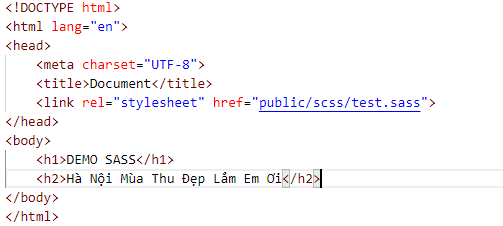
Vd:



Hình 3:Cấu trúc của demo mẫu

ở đây chúng ta tạo file tên là file test.html để chạy demo và tạo một test.sass ở chúng ta không tạo file css mà chúng ta chit tạo file sass khi nào coe hoàn thành thì chúng ta sẽ sử dụng dòng lệnh để chuyển từ file sass sang file css.

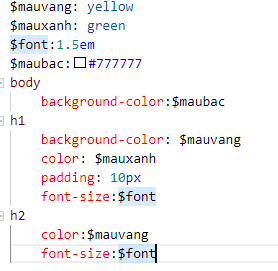
File test.html



# Hình 4:code trong HTML

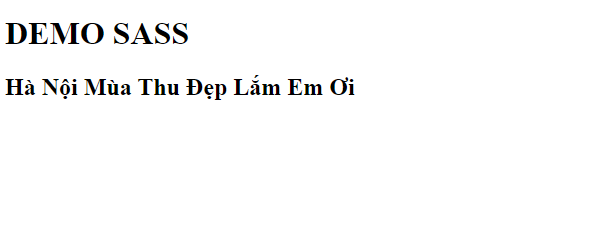
**🡺Chú ý:bạn nào sử dụng Visua Studio code thì nhớ Install thư viện SASS vào trong project**

Test.sass



# Hình 5:code trong file sass

ở đây mình tạo một file sass rất đơn giản để cho mọi người dễ hiểu, file sass cho phép chúng ta đặc một biến chung khi nào cần sử dụng thì gọi đến biến đó và sử dụng việc sự dụng này rất dễ dàng cho việc xóa và sửa



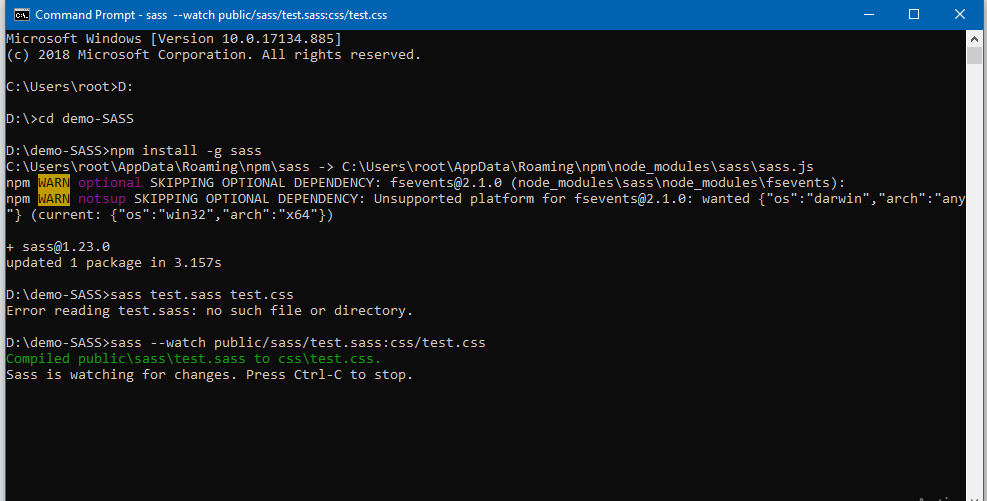
# Hình 6:Giao diện chính khi chưa có css

Tiếp theo chúng ta tiến dùng dòng lệnh gõ vào Command Prompt(cmd) để tiến hành chuyển file sass sang file css

**Dòng lệnh chuyển file sass thành flie css:**

🡺sass --watch public/sass/test.sass:css/test.css

**Chú ý:Bạn nhớ gõ đúng đường dẫn các thư mục trong project của mình đã tạo**



### Hình 7:Chuyển file thành công

Sau khi bạn chuyển file thành công thì trong thư mục css đã chuyển sẽ có 2 file

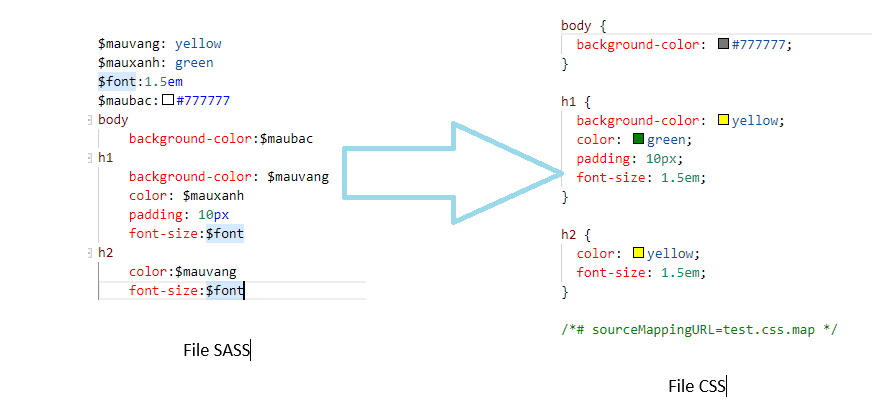
1.file của sass được chuyển thành css

2.file map của hệ thống



# Hình 8:Hai file chính trong CSS

Sau khi chuyển thì toàn bộ những gì chúng ta viết từ file sass sẽ được chuyển sang file css.



# Hình 10:File sass chuyển thành file CSS

Và tiếp theo chúng ta sẽ liên kết CSS để chạy demo của chung ta

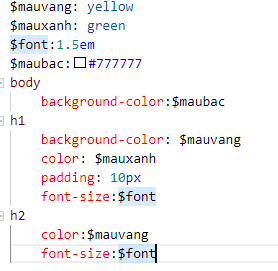


# Hình 11:demo sau khi liên kết CSS

1. **Một số tính năng của SASS**

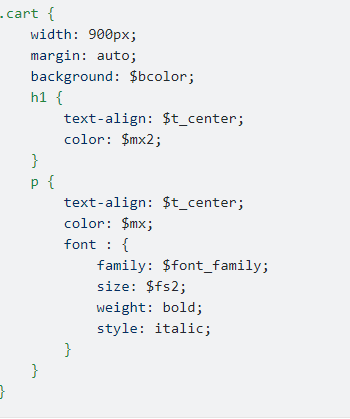
## **1.Sử dụng biến**

SASS sử dụng biến để lưu lại các giá trị, thay thế hàng loạt các thuộc tính của website



# Hình 12:cách sử dụng biến

## **2.Tính năng xếp chồng trong SASS**



# Hình 13:Tính năng xếp chồng

## **3.Tính năng sử dụng lại code**

Sử dụng lại nguyên cả đoạn code

Sử dụng code có tham số(tương tự như function trong lập trình)

Vd demo nhỏ cho dễ hiểu:

ở đây mình có 2 đoạn div



# Hình 14:Hai đoạn Div mẫu

Ở đây mình muốn 2 hình ảnh đều căn trái giống nhau, như vậy ở đây chúng ta sẽ phải có 2 đoạn code css cho 2 img, như vậy sẽ dài code.

Trong SASS có khái niệm là @mixin, @mixin các bạn có thể hiểu nó tương tự như 1 hàm trong lập trình vậy.

Đơn giản bạn chỉ cần đặt tên cho mixin và ở dưới image bạn gán @include tên\_mixin là được



# Hình 15:@mixin để tạo code,@include để sử dụng lại code

Mục Lục Báo cáo LESS

[**Tìm hiểu về LESS** 12](#_Toc23938467)

[**1.** **LESS là gì?** 12](#_Toc23938468)

[**2.** **Cài đặt LESS Preprocessor** 12](#_Toc23938469)

[**3.** **Khai báo biến(variables) trong LESS** 18](#_Toc23938470)

[**4.** **LESS Scope** 18](#_Toc23938471)

[**5.** **LESS – Escaping** 19](#_Toc23938472)

[**6.** **LESS Parametric Mixins** 19](#_Toc23938473)

[**7.** **LESS Mixins** 20](#_Toc23938474)

[**8.** **LESS Operations** 21](#_Toc23938475)

[**9.** **Less Funtions** 21](#_Toc23938476)

[**10.** **LESS Nested Rules** 22](#_Toc23938477)

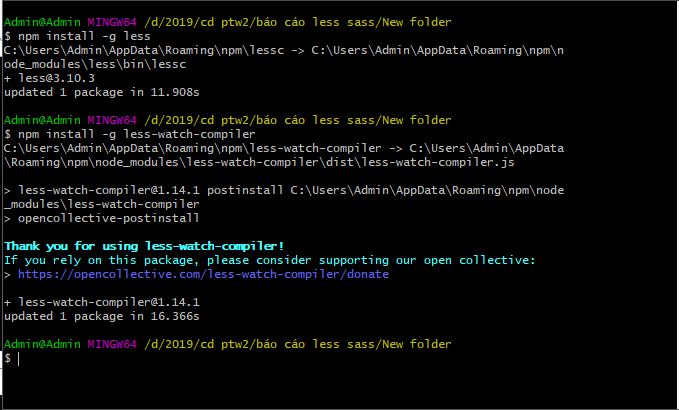
# **Tìm hiểu về LESS**

1. **LESS là gì?**

* LESS là một CSS preprocessor
* LESS giúp ngắn gọn và đơn giản hóa các đoạn mã CSS bằng cách thêm vào mã CSS các thành phần mới như : biến, mixins, toán tử và hàm.
* Nhờ đó mà việc quản lý các mã CSS cũng trở lên dễ dàng hơn rất nhiều.

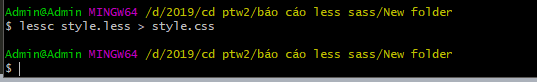
1. **Cài đặt LESS Preprocessor**

* Để cài đặt và sử dụng LESS trước tiên ta cần tải file “less.js” tại trang chủ của less là <http://lesscss.org/>.
* Hoặc bạn có thể dùng câu lệnh ‘npm’ để gõ Git Bash.
* Bước 1: Gõ câu lệnh npm: $ npm install -g less
* Bước 2: Tiếp theo gõ dòng lệnh để cài đặt compiler:$ npm install -g less-watch-compiler



Hình 1

* Bước 3: Tiếp đó gõ lệnh lessc styles.less > styles.css để biên dịch less sang css (Tới đây thì mới chỉ biên dịch được file .less sang .css. Nếu muốn css tự động thay đổi khi bạn code bên file less thì cần thực hiện thêm các bước dưới đây)



Hình 2

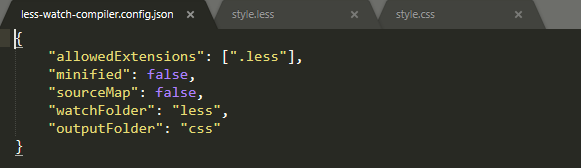
\*Nó sẽ không hiện lên thông báo khi gõ thành công. Vì vậy, cần vào file để kiểm tra.

* Bước 4: Tạo thư mục less và css di chuyển file .less vào thư mục less và file .css vào thư mục css
* Bước 5: Gõ dòng lệnh: less-watch-compiler less css . Để biên dịch file .less trong thư mục less sang css trong thư mục css.



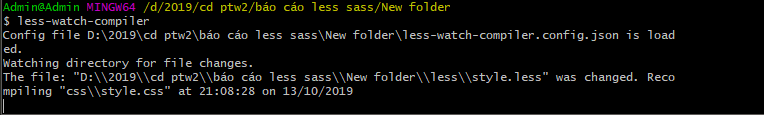
Hình 3

* Bước 6: Tạo file less-watch-compiler.config.json. và thêm đoạn code bên dưới vào.



Hình 4

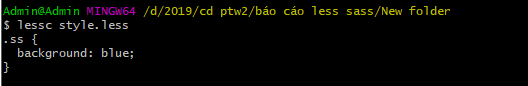
* Bước 7: Gõ câu lệnh less-watch-compiler để css tự động được thay đổi khi code bên file less



Hình 5

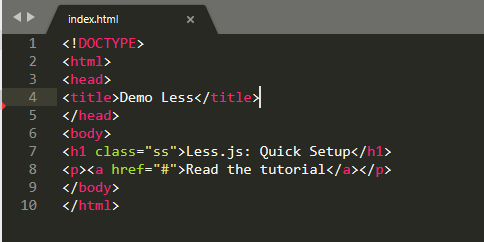
Nếu nó hiển thị dòng thông báo bên dưới thì bạn đã cài đặt thành công. Nhớ đừng tắt cmd khi code nhé (Vì tắt cmd khi code file .less thì file .css sẽ không được cập nhật).

* Gõ lessc styles.less để xem thuộc tính của file less trên cmd.



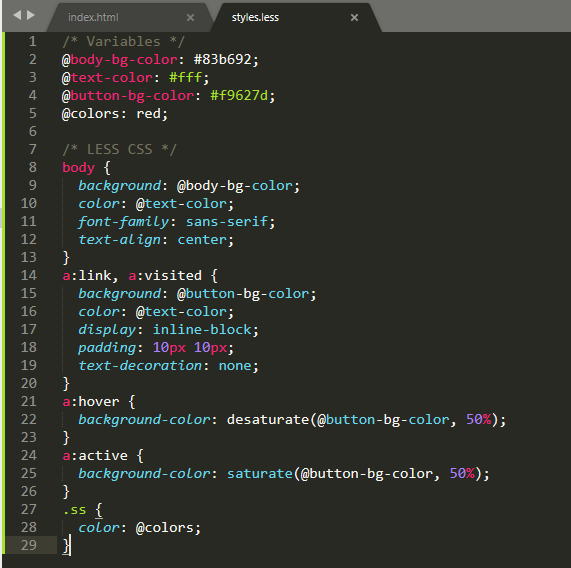
Hình 6

* Tiếp đó ta tạo file index.html và gõ đoạn văn bản để test LESS



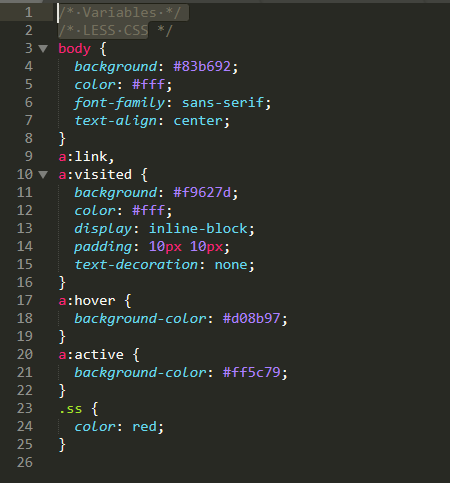
Hình 7

* Tiếp theo ta tạo file styles.less và thực hiện đoạn code để test file.



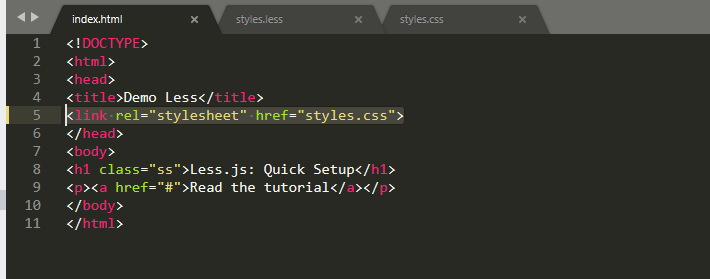
Hình 8

* File .css sẽ được tự động cập nhật khi bạn code bên file .less



Hình 9

* Cuối cùng bạn chỉ cần thêm đường dẫn đến file styles.css để test nó.



Hình 10

* Và đây là kết quả

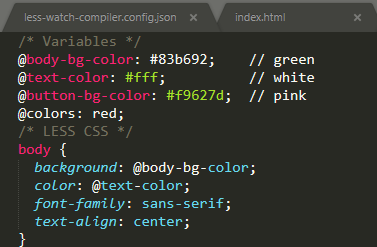


Hình 11

1. **Khai báo biến(variables) trong LESS**

Vì Less hoạt động như một ngôn ngữ lập trình, nó có khai báo biến, toán tử,… Nên không giống như CSS, khi muốn sử dụng Less trước tiên ta phải khai báo biến và khai báo giá trị cho biến.

Ví dụ:



Hình 12

1. **LESS Scope**

Khai báo biến @var: red; màu dùng dùng cho .ss{} và biến @var: green; nằm trong .vv a{} thì màu của nó sẽ là màu của biến nằm trong nó.

Ví dụ:

|  |  |
| --- | --- |
| LESS | CSS |
|  |  |
| @var: red;  .ss{  .vv a{  @var: green;  color: @sh;  }  color: @var;  } | .ss {  color: red;  }  .ss .vv a {  color: green;  } |

1. **LESS – Escaping**

Nó xây dựng các bộ chọn một cách linh hoạt và sử dụng thuộc tính hoặc giá trị biến làm chuỗi tùy ý.

Ví dụ:

|  |  |
| --- | --- |
| LESS | CSS |
| .ss {  background: ~"red";  } | .ss {  background: red;  } |

1. **LESS Parametric Mixins**

Chúng ta có thể đặt tham số khi sử dụng Less.

Ví dụ:

|  |  |
| --- | --- |
| LESS | CSS |
| @backsl: red;  @radius: 5px;  .ss {  background: @backsl;  border-radius: @radius;  } | .ss {  background: red;  border-radius: 5px;  } |

1. **LESS Mixins**

Trong LESS, có thể thêm một nhóm các thuộc tính của một 'ruleset' này vào trong 'ruleset' khác.

Ví dụ:

|  |  |
| --- | --- |
| LESS | CSS |
| @backsl: red;  @radius: 5px;  .ss {  background: @backsl;  border-radius: @radius;  }  .vv {  .ss  } | .ss {  background: red;  border-radius: 5px;  }  .vv {  background: red;  border-radius: 5px;  } |

1. **LESS Operations**

LESS Operations hỗ trợ cho một số thao tác số học như cộng (+), trừ (-), nhân (\*) và chia (/) và chúng có thể hoạt động trên bất kỳ số, màu hoặc biến nào. Các hoạt động tiết kiệm rất nhiều thời gian khi bạn đang sử dụng các biến và bạn cảm thấy như đang làm việc trên toán học đơn giản.

Ví dụ:

|  |  |
| --- | --- |
| LESS | CSS |
| @backsl: red;  @radius: 5px;  .ss {  background: @backsl;  border-radius: @radius \* 2;  }  .vv {  .ss  } | .ss {  background: red;  border-radius: 10px;  }  .vv {  background: red;  border-radius: 10px; |

1. **Less Funtions**

Thao tác với các giá trị và sử dụng các hàm được xác định trước để thao tác các khía cạnh của các thành phần HTML trong định kiểu.

|  |  |
| --- | --- |
| LESS | CSS |
| @backsl: red;  @radius: 5px;  @width:1.0;  .ss {  background: @backsl;  border-radius: @radius \* 2;  width: percentage(@width);  } | .ss {  background: red;  border-radius: 10px;  width: 100%;  } |

1. **LESS Nested Rules**

Nó là một nhóm các thuộc tính CSS cho phép sử dụng các thuộc tính của một lớp vào một lớp khác và bao gồm tên lớp làm thuộc tính của nó. Trong LESS, bạn có thể khai báo Mixin theo cách tương tự như kiểu CSS bằng cách sử dụng bộ chọn lớp hoặc id. Nó có thể lưu trữ nhiều giá trị và có thể được sử dụng lại trong mã bất cứ khi nào cần thiết.

|  |  |
| --- | --- |
| LESS | CSS |
| .container {  h1 {  font-size: 25px;  color:#E45456;  }  p {  font-size: 25px;  color:#3C7949;  }  .myclass {  h1 {  font-size: 25px;  color:#E45456;  }  p {  font-size: 25px;  color:#3C7949;  }  }  } | .container h1 {  font-size: 25px;  color: #E45456;  }  .container p {  font-size: 25px;  color: #3C7949;  }  .container .myclass h1 {  font-size: 25px;  color: #E45456;  }  .container .myclass p {  font-size: 25px;  color: #3C7949;  } |