



---

# ***SASS***

Khóa học “Building Website With HTML & CSS”

- Trình bày được định nghĩa bộ tiền xử lý CSS (CSS Preprocessor)
- Trình bày được lợi ích của việc sử dụng bộ tiền xử lý CSS và SASS
- Trình bày được các tính năng cơ bản của SASS
- Trình bày được các giai đoạn trong build process
- Áp dụng được các tính năng cơ bản của SASS:
  - Variables
  - Nesting
  - Mixin
  - Extend
  - Partials và Import

---

# Thảo luận

Viết và bảo trì mã CSS có những khó khăn nào?  
Chúng ta có thể giải quyết bằng cách nào?

# SASS là gì?

---



- SASS (Syntactically Awesome Style Sheets) là ngôn ngữ kịch bản và là bộ tiền xử lý CSS (CSS Preprocessor).
- SASS biên dịch mã theo cú pháp cụ thể thành CSS. Giúp nhà phát triển viết CSS nhanh hơn, dễ dàng hơn, thanh lịch hơn.
- SASS bổ sung các tính năng và công cụ mới trên CSS cơ bản.

# CSS Preprocessor là gì?

---



- CSS Preprocessor (Bộ tiền xử lý CSS) là một chương trình cho phép chúng ta tạo CSS từ cú pháp duy nhất của bộ tiền xử lý.
- Có nhiều bộ tiền xử lý CSS khác nhau (như SASS, LESS, Stylus,...), tuy nhiên hầu hết các bộ tiền xử lý CSS sẽ thêm một số tính năng không tồn tại trong CSS thuần túy, chẳng hạn như mixin, nesting selector, bộ chọn kế thừa, v.v.

# Ưu điểm của SASS

---



1. Giúp tạo mã sạch hơn (nhờ có tính năng nesting và variables)
2. Giúp hạn chế mã lặp lại trong CSS
3. Linh hoạt hơn CSS
4. Ổn định hơn CSS
5. Tổ chức code CSS tốt hơn

# Cú pháp SCSS

---



- SCSS (còn gọi là Sassy CSS) là tiêu chuẩn hiện đại của CSS sử dụng dấu ngoặc { } và dấu chấm phẩy ;
- SCSS đã được giới thiệu trong phiên bản thứ 3 của SASS dưới dạng superset của CSS
- SCSS không mở rộng tiêu chuẩn CSS mà chỉ đơn giản là cú pháp CSS tiến bộ hơn
- CSS hợp lệ theo cú pháp SCSS, do đó, nó dễ dàng chuyển đổi qua lại giữa cả hai
- Các tệp SCSS sử dụng phần mở rộng .scss
- Khi sử dụng SCSS, chúng ta vẫn gọi nó là SASS.

# Cú pháp SASS

---



- SASS là một bộ tiền xử lý CSS và có cú pháp cũ hơn sử dụng thụt lề để tổ chức và tách các khối code.
- SASS cung cấp một cách viết CSS súc tích, mở rộng chức năng của CSS.
- Các tệp SASS sử dụng phần mở rộng .sass



# Variables

---



Các biến cung cấp một cách để lưu trữ thông tin hoặc đóng gói các kiểu để bạn có thể sử dụng lại trong suốt file CSS của mình.

Ví dụ:

- Lưu trữ các giá trị **color** hoặc **font-family** bằng tên, sau đó sử dụng lại bất cứ lúc nào trong mã CSS. Nếu cần thay đổi thành mã màu khác thì chỉ cần sửa đổi giá trị biến.

```
$color-primary: #ffff00; // Yellow

body {
  background-color: $color-primary;
}
```

# Nesting

---



- Nesting trong SASS cho phép sắp xếp các phần tử DOM theo cách gọn gàng hơn.
- Không phải viết lại các CSS Selector nhiều lần
- Việc thay đổi style dễ dàng hơn.

```
.navbar {  
    background-color: orangered;  
    padding: 1rem;  
  
    ul {  
        list-style: none;  
    }  
  
    li {  
        text-align: center;  
        margin: 1rem;  
    }  
}
```

# Mixin



Mixins trong SASS cho phép nhóm nhiều dòng CSS lại với nhau và có thể sử dụng lại ở bất cứ đâu.

## Tạo một Mixin trong SASS:

```
@mixin transform {  
  -webkit-transform:  
    rotate(180deg);  
  -ms-transform: rotate(180deg);  
  transform: rotate(180deg);  
}
```

## Sử dụng Mixin

```
.navbar {  
  background-color: orangered;  
  padding: 1rem;  
  ul {  
    list-style: none;  
  }  
  li {  
    text-align: center;  
    margin: 1rem;  
    @include transform;  
  }  
}
```

# Extend & Inheritance



- Sử dụng **@extend** cho phép chia sẻ một tập các thuộc tính CSS selector này sang selector khác.
- Giúp giảm được các mã CSS trùng lặp.

```
%message-shared {  
    border: 1px solid #ccc;  
    padding: 10px;  
    color: #333;  
}
```

```
.message {  
    @extend %message-shared;  
}  
.success {  
    @extend %message-shared;  
    border-color: green;  
}  
.error {  
    @extend %message-shared;  
    border-color: red;  
}  
.warning {  
    @extend %message-shared;  
    border-color: yellow;  
}
```

# Partials & Importing

---



- SASS partials cho phép chia nhỏ các tập tin.
- Có thể mô-đun hóa CSS để tạo mã dễ bảo trì hơn.
- Nếu tập tin SASS quá lớn, chúng ta có thể tạo ra một phần tập tin chỉ chứa mã có liên quan cho mỗi phần. Tên của một phần bắt đầu bằng một dấu gạch dưới \_
- Sau đó, import các tập tin bằng cách sử dụng lệnh **@import**.



---

# Demo

Các tính năng cơ bản của SASS

# Build Process

---



- Build Process là một chuỗi các nhiệm vụ thực hiện tự động sau khi chạy dự án.
- Build Process bao gồm một số giai đoạn sau:
  - Giai đoạn 1: Compiling - Biên dịch sang CSS
  - Giai đoạn 2: Concatenating - Hợp nhất tất cả các tập tin CSS thành một
  - Giai đoạn 3: Prefixing - Sử dụng *autoprefixer* để đảm bảo rằng các tập tin hoạt động trên tất cả các trình duyệt chính
  - Giai đoạn 4: Compressing - Nén code để tối đa hóa hiệu suất



---

# Demo

Thiết lập build process cho dự án web



Qua bài học này, chúng ta đã tìm hiểu những nội dung sau:

- SASS là gì? CSS Preprocessor (Bộ tiền xử lý CSS) là gì?
- Ý nghĩa của việc áp dụng SASS trong dự án web
- Các tính năng cơ bản của SASS:
  - Variables
  - Nesting
  - Mixin
  - Extend
  - Partials và Import
- Build process và các giai đoạn trong build process

---

# Hoạt động tiếp theo

- ❑ Xem các nội dung lý thuyết
- ❑ Làm theo hướng dẫn làm bài tập