**1. SASS**

* Sass là một ngôn ngữ kịch bản tiền xử lý được diễn giải hoặc biên dịch thành Cascading Style Sheets (CSS).
* Một số đặc điểm và khái niệm quan trọng trong Sass:

+ Biến (Variables): Sass cho phép định nghĩa biến để lưu trữ các giá trị như màu sắc, kích thước, font chữ và các giá trị khác

+ Lồng (Nesting): Sass cho phép bạn lồng các quy tắc CSS bên trong nhau, tạo ra cấu trúc rõ ràng và dễ đọc hơn

+ Mixin: Mixin là một khối mã Sass có thể được sử dụng lại trong nhiều nơi

+ Extend/Inheritance: Sass cho phép bạn kế thừa các quy tắc CSS từ một lớp (class) khác

+ Import: Sass cho phép bạn chia mã CSS thành nhiều file và nhập chúng vào các file khác

- Ưu điểm: cú pháp gọn gàng, tính kế thừa cao, tái sử dụng CSS. Cấu trúc CSS rõ ràng do có tính chất class lồng, quản lý code dễ dàng hơn.

- Nhược điểm: kích thước dự án lớn hơn do SASS phải biên dịch thành CSS, tốc độ biên dịch giảm

**2. BEM**

* BEM là viết tắt của "Block, Element, Modifier" và là một phương pháp đặt tên lớp (class) trong CSS.
* Block (Khối): Khối là thành phần độc lập lớn nhất trong cấu trúc. Nó đại diện cho một phần tử trang web độc lập và có thể chứa các phần tử và/hoặc các khối con bên trong. Ví dụ: .header, .menu, .button.
* Element (Phần tử): Phần tử là một thành phần nhỏ hơn nằm trong khối và không tồn tại độc lập. Nó là một phần không thể tách rời của khối và không có ý nghĩa khi được sử dụng độc lập bên ngoài khối. Ví dụ: .header\_\_logo, .menu\_\_item, .button\_\_text.
* Modifier: là các lớp dùng để thay đổi hoặc tùy chỉnh kiểu dáng, trạng thái hoặc hành vi của khối hoặc phần tử. Chúng giúp tái sử dụng kiểu dáng hoặc hành vi của khối hoặc phần tử mà không cần viết lại mã CSS. Ví dụ: .button--primary, .menu\_\_item--active, .header\_\_logo--small.
* Ưu điểm: + Nó giúp tổ chức và quản lý mã CSS một cách hiệu quả, đặc biệt khi làm việc trên các dự án lớn và phức tạp.

+ Tạo ra một cấu trúc rõ ràng, dễ đọc và dễ bảo trì.

- Nhược điểm: + cú pháp có thể phức tạp ban đầu, có thể gây khó hiểu khi đọc code,

+ tạo ra các lớp có tên dài, phức tạp, các tên lớp có thể bị lặp

**3. Boostrap**

* Boostrap cung cấp một bộ các lớp CSS, các thành phần UI, JavaScript và các mẫu thiết kế sẵn để giúp tạo ra giao diện nhanh chóng, responsive và dễ sử dụng.

Boostrap đã cung cấp sẵn các lớp, thành phần UI đã được định nghĩa sẵn như menu, button, form, …

* Ưu điểm: + giúp tiết kiệm thời gian và công sức trong quá trình phát triển web

+ tương tính nhiều trình duyệt khác nhau

+ độ phổ biến lớn, giao diện responsive

* Nhược điểm: + kích thước file lớn, có nhiều thuộc tính, class cần phải tìm hiểu.

+ không gợi ý code khi dùng

**4. Các kỹ thuật làm việc với block**

* Sử dụng CSS Box Model: Bằng cách điều chỉnh các giá trị của các thành phần này, bạn có thể tạo kiểu và điều chỉnh kích thước của các block.
* Định vị bố trí: sử dụng các thuộc tính: position, display, float, clear để kiểm soát vị trí và tạo kiểu cho các block theo hướng ngang
* Responsive Design: sử dụng mediaquery và grid system
* Flexbox, CSS grid
* Sử dụng Pseudo-classes và Pseudo-elements
* Sử dụng Animation và Transition

**5. Các thẻ thuộc block và inline**

- Các thẻ thuộc block : div, p, h1, h2…, h6

- Các thẻ thuộc inline: span, a, strong, em