**BEM và cách đặt tên Class cho element**

Trong bài này chúng ta sẽ cùng nhau tìm hiểu

* Think – Build – Architect Mindset
* Think (nghĩ về layout trước)
* Build (Đặt tên classes)
* Architect ( Tổ chức folder, sử dụng Sass)

## Think – Build – Architect Mindset

việc code CSS nên được làm thỏa mãn các tiêu chí: **Rõ ràng (Clean), Chia nhỏ thành các phần (Modular), dễ tái sử dụng  (Reusable),**và **dễ mở rộng (Ready for growth)**.

Để làm được điều đó ta cần 3 bước:

* **Bước 1: THINK**suy nghĩ về cách tổ chức code, các layout hiển thị lên webpage và thường sẽ sử dụng wireframe.
* **Bước 2:** **BUILD**: Xây dựng code HTML dựa trên wireframe và đặt tên class 1 cách có cấu trúc.
* **Bước 3: ARCHITECT**: Chia nhỏ các phần CSS thành các file và folder 1 các logic

## Think about the layout

* Website sẽ xây dựng dựa trên các **component** nhỏ (Component - Driven design).
* Những **component** gắn kết với nhau dựa vào layout của webpage.
* Các **component** này được xây dựng độc lập và có thể tái sử dụng ở các page khác nhau trong cùng 1 project hoặc các project khác nhau.

## Build( Đặt tên class)

Ở đây chúng ta làm quen với **BEM** (**B**lock **E**lement **M**odifier)

* **Block**: tên của component, có nghĩa mô tả tổng thể của component đó
* **Element**: nó dùng để đặt tên các phần từ bổ trợ cho component (Block)
* **Modifier**: Các dạng khác biến thể của 1 block hoặc element
* .block{}
* .block\_\_element{}
* .block\_\_element--modifier{}

Chúng ta sẽ hiểu rõ hơn qua ví dụ sau:

<**div** class="card">

<**h1** class="card\_\_heading--sub">Hello Everyone</**h1**>

<**h3** class="card\_\_heading--main">Nice to meet you</**h3**>

<**p** class="card\_\_content">Lorem ipsum dolor sit amet consectetur adipisicing elit. Repudiandae, sed!</**p**>

</**div**>

Trong đó, chúng ta thấy:

* **Block** (card) là mô tả toàn bộ component.
* **Element** (heading, content) chỉ các bộ phận của component này.
* **Modifier** (sub, main) chỉ ra có 2 loại element (heading) gồm heading chính và heading phụ

Trong Sass sẽ hỗ trợ chúng ta rất nhiều trong việc sử dụng **BEM** ( chúng ta sẽ tìm hiểu ở những bài học sau)

Vậy qua các phần trên, chúng ta đã thấy được lợi ích của việc sử dụng **BEM** là giúp tổ chức tốt code CSS, dễ bảo trì, tái sử dụng, tránh tình trạng code chồng chéo lên nhau. Ví dụ như cùng là heading nhưng card heading và header heading khác nhau, …

## Architect (Tổ chức folders, Sass)

Sass sẽ hỗ trợ chúng ta trong việc chia code CSS thành các folder và các file chứa các component nhỏ.

Nôm na của mô hình tổ chức chúng ta sẽ chia làm **7 folders**và 1 file **main.sass**chính import tất cả các files trong các folder để compile thành CSS

7 folder ở đây chia thành từng mảng:

1. **Base/** (chứa các setting ban đầu)
2. **Components/** ( chứa các bộ phận nhỏ dễ tái sử dụng như button, card, …)
3. **Layout/**( chứa footer, navbar,…)
4. **Pages/** (chứa tên các page nếu project web của bạn có nhiều page khác nhau vd: home, login, …)
5. **Abstracts/** (chứa animation, các function, mixin khi sử dụng sass sẽ được đề cập kĩ hơn khi chúng ta làm quen với sass)
6. **Themes/** (các mẫu có sẵn)
7. **Vendors/**(code CSS bên ngoài như bootstrap, …)