TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC



**Khoa Công Nghệ Thông Tin**

****

**Chuyên đề phát triển web 1** | HKI – [2020 – 2021]

PRJ – REPORT

**BEM**

**Nhóm M:**

* **Ngô Đình Trí–** 17211TT4446
* **Nguyễn Y Trinh–** 17211TT1738
* **Doãn Hồng Thăng –** 18211TT4557

# BEM

## **Giới thiệu BEM**

* BEM (Block Element Modifier) là chuẩn quy ước đặt tên trong code Front-end. Giúp code Front-end rõ ràng và dễ quản lý hơn, thuận tiện cho việc mở rộng.
* Vì sao nên dùng BEM?
  + Đặt tên theo chuẩn code giúp làm việc nhóm dễ dàng hơn. Người khác đọc code có thể hiểu được ý nghĩa đoạn code, tránh tình trạng mỗi người có một cách đặt tên riêng.
  + Cấu trúc BEM đơn giản, dễ hiểu và vững chắc.
  + Hỗ trợ cho việc code SASS và LESS.
  + Tái sử dụng.
  + Modules: Các class khác nhau của mỗi block sẽ không ảnh hưởng nhau, tránh tình trạng CSS bị ghi đè.

Ví dụ: ta có 1 block là btn. Ta viết thêm 2 class tô màu cho nó là btn—blue và btn—red, khi muốn tạo nút màu xanh ta viết class=”btn btn—red” còn khi tạo nút màu xành ta viết class=”btn btn—blue”.

## **Quy ước đặt tên**

.block {} */\* Block \*/*

.block\_\_element {} */\* Element \*/*

.block--modifier {} */\* Modifier \*/*

* **Block**: tên component, mô tả tổng quát component đó
* **Element**: dùng đặt tên phần tử con thuộc component (Block)
* **Modifier**: các biến khác của 1 block hoặc element

**Ví dụ**:

<div *class*="intro">

  <div *class*="intro\_\_title--big"></div>

  <div *class*="intro\_\_title--small"></div>

  <div *class*="intro\_\_content"></div>

</div>

## **Biến thể của BEM**

<a *class*="btn btn--small btn--red"></a>

Ta sẽ SCSS như sau:

.btn{

    ...

}

.btn--small

{

    ...

}

.btn--red{

    ...

}

Ta có thể thấy vấn đề ở đây là mỗi khi thêm thuộc tính mới chèn lên những thuộc tính cũ ta lại viết lại btn—"modifier” như thế này?

Cho nên ta sẽ biến tấu lại như sau class="block -modifier":

<a *class*="btn -primary -small -red"></a>

Và style SCSS ta sẽ viết:

.btn{

    ...

    &.-small{

        ...

    }

    &.-red{

        ...

    }

}

Qua đó ta thấy biến thể của BEM cung cấp sự linh hoạt và đơn giản code để cấu hình cho bất kì module nhất định. Thích hợp cho các module có nhiều sửa đổi.