



Trang icon : font-awesome và material.io

1. **Tạo component:**

B1: Tự tạo component input có @input là 1 icon.

B2: Sử dụng @input để truyền string ( là tên font ) vào component

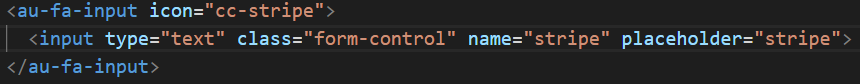


* Input không có dấu [] do truyền vào **string.**

1. **Sử dụng ng-content và style :**

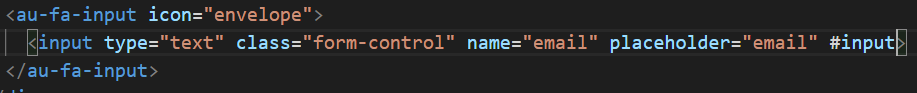
B1 : Nội dung innerHTML của component sẽ là **input** => dùng **ng-content**

* Vì ta cần thêm các attribute từ component cha. VD : form



B2 : áp dụng css cho input tại component dùng selector **:host /deep/ input**

B3 : Xử lý event : **focus, blur, …**



Thêm template ref cho content bên trong component. => dùng **@ContentChild**

* Kiểu dữ liệu là **HTMLInputElement**
* Dùng lifecycle AfterContentInit.. để xử lý.

**@ContentChild(‘ref’)**

**Input: HTMLInputElement;**

B4 ; Mở rộng không cần khai báo **template ref** cho phần content của component: sử dụng **custom directive** và HostListener, HostBinding

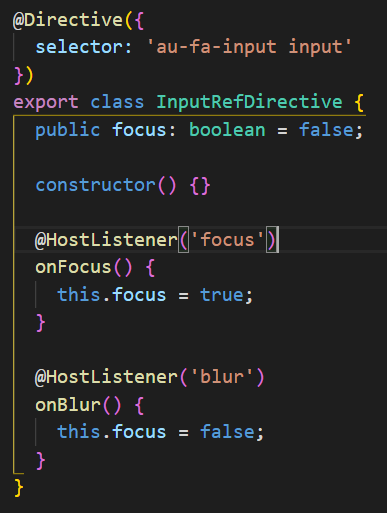
\_ HostListener : là decorator giúp xử lý sự kiện cho các host element trong class directive.

\_ HostBinding : là decorator giúp thiết lập thuộc tính của host element từ class directive.

\_ Tạo directive có tên trùng với component và áp dụng vào cho bất kỳ input input :

Selector : ‘tên-**component** **input’**

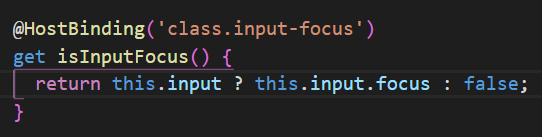
\_ Thay thế kiểu dữ liệu bên trong ContentChild và Input.



B5 : Xử lý nếu có event add thêm các class CSS : tại component custom

@HostBinding(‘class.tên-class’)

get isInputEvent() => return this.input ? this.input.focus : false



* Nếu focus true => binding class tương ứng