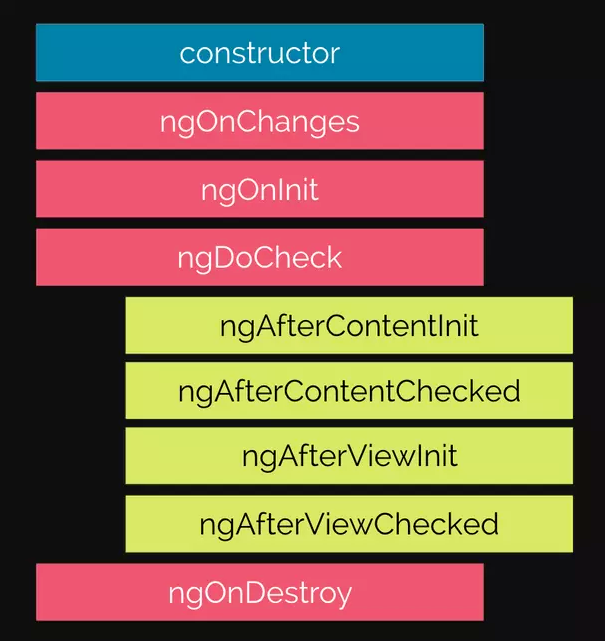
**LIFECYCLE HOOK**

**- Trình tự vòng đời:**



**\*ngOnit**

Trong Angular, "ngOnit" là một lifecycle hook (kết nối vòng đời) được sử dụng trong các component. Được gọi sau khi Angular đã tạo và khởi tạo các thành phần và dữ liệu của component. Nguyên tắc hoạt động của "ngOnit" như sau:

1. Khi một component được tạo và hiển thị trong ứng dụng Angular, hàm "ngOnInit" được gọi tự động.

2. Trong hàm "ngOnInit", bạn có thể thực hiện các tác vụ khởi tạo và xử lý dữ liệu ban đầu cho component, như lấy dữ liệu từ API, gán giá trị mặc định cho các biến, đăng ký các sự kiện, khởi tạo các service, và các tác vụ khác liên quan đến khởi tạo.

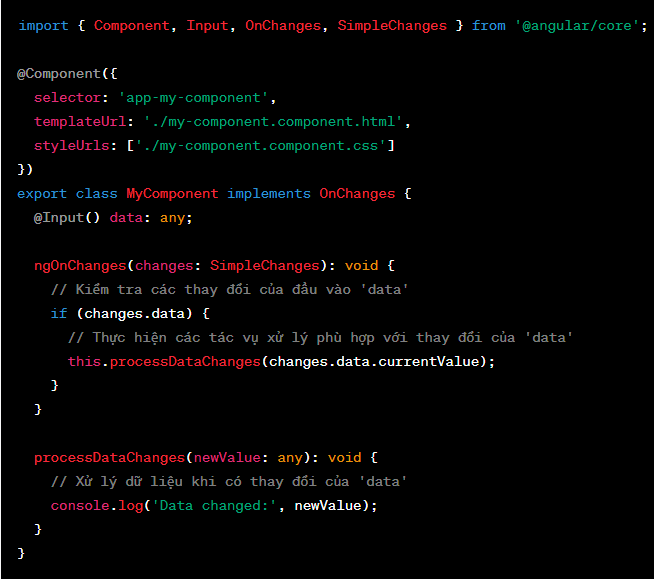
3. "ngOnInit" chỉ được gọi một lần duy nhất trong vòng đời của component, sau khi component đã được khởi tạo và trước khi nó được hiển thị trên giao diện người dùng.

4. "ngOnInit" trong Angular là một lifecycle hook được sử dụng để thực hiện các tác vụ khởi tạo và xử lý dữ liệu ban đầu trong một component.

**\*ngChanges**

"ngOnChanges" là một lifecycle hook khác trong Angular. Nó được sử dụng để xử lý các thay đổi đầu vào (input) của một component. Khi giá trị của một đầu vào của component thay đổi, "ngOnChanges" được gọi để thông báo về sự thay đổi này và cho phép bạn thực hiện các tác vụ xử lý phù hợp.

Nguyên tắc hoạt động của "ngOnChanges" như sau:

1. Khi giá trị của một đầu vào (input) của component thay đổi, Angular sẽ gọi hàm "ngOnChanges" của component đó.
2. Trong hàm "ngOnChanges", bạn có thể truy cập vào đối tượng "SimpleChanges" để kiểm tra các thay đổi của các đầu vào (input). "SimpleChanges" chứa thông tin về các thay đổi, bao gồm giá trị trước và sau khi thay đổi.
3. Bạn có thể thực hiện các tác vụ xử lý phù hợp với sự thay đổi của đầu vào, chẳng hạn như cập nhật dữ liệu, gọi API để lấy dữ liệu mới, hoặc thực hiện các tính toán phụ thuộc vào giá trị đầu vào.
4. "ngOnChanges" trong Angular là một lifecycle hook được sử dụng để xử lý các thay đổi đầu vào (input) của một component.