Syntax

- \* là 1 chỉ thị cấu trúc

- \*ngFor: cấu trúc vòng lặp for

- \* ngIF: cấu trúc vòng lặp if nếu có

- interpolation {{}} cú pháp nội suy để lấy data từ property's **phép nội suy** "tiêm" giá trị vào html, vì vậy khi bạn nói value="{{ hello }}"Angular đang chèn biến của bạn vào giữa các dấu ngoặc.

- Prototype binding []: Khi bạn nói [value]="hello"Angular đang lấy thuộc tính giá trị của phần tử và đặt biến của bạn làm giá trị của thuộc tính đó.

- Event binding: **liên kết sự kiện** cho phép bạn sử dụng các sự kiện như nhấp chuột để kích hoạt các chức năng. các ràng buộc này sử dụng dấu ngoặc đơn chẳng hạn (click)="myFunction($event)". điều này sẽ gọi phương thức myFunction () được xác định trong tệp .ts của bạn. dấu ngoặc đơn quanh '(click)' liên kết hàm với sự kiện dom. $eventlà một từ khóa chuyển đối tượng sự kiện cho hàm. bạn cũng có thể truyền một chuỗi với các dấu ngoặc đơn hoặc thậm chí là một biến có nội suy.

Component

- Component define areas of responsibility in your UI that let you reuse these sets of UI functionality. You have already built one product with list component

- Component is comprised of three things:

+ A component class: handle data and functionality. In the previous section, the product data and the share() method were defined for you in the component class

+ **An HTML template:** which determines what is presented to the user

+ **Component-specific styles**

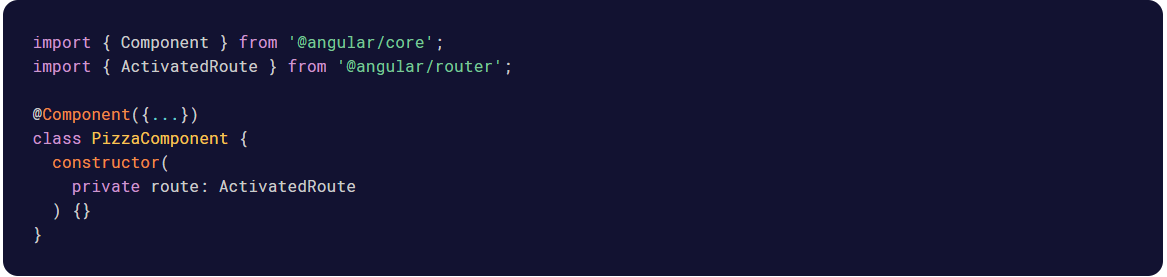
**-** An Angular application is composed of a tree of components, in which each Angular component has a specific purpose and responsibility.

- Currently, our app has three components

MyBankAccount

1. Contructor

- Là nơi kết nối các component khác lại với nhau chúng ta có thể sử dụng các thành phần của component đó thông qua THIS.ROUTE.



- Ví dụ ở đây nhúng ActivateRoute đến class và chúng ta có thể sử dụng các thành phần của component đó thông qua THIS.ROUTE.

1. Lifecycle hooks in angular
   1. ngOnInit

- Khởi tạo các giá trị cho component

- Thực hiện khởi tạo những thứ bên trong component sau khi Angular đã khởi tạo những thuộc tính liên kết dữ liệu. Đây là nơi để lấy dữ liệu cho template từ một back-end service

* 1. ngOnDestroy

- Thu hồi trước khi Angular hủy bỏ directive/component. Hủy đăng ký observables và tháo gỡ các sự kiện để tránh tốn bộ nhớ

* 1. ngOnChanges

- Phản hồi lại sau khi angular cài đặt một thuộc tính dữ liệu ràng buộc. Phương thức nhận một object changes của hiện tại và giá trị trước đó

1. Structural directive (ngIf, ngFor, ngSwitch)
   1. NgIf

- Chỉ thị ngIf cho phép chúng ta thêm hoặc loại bỏ một element ra khỏi trang, chúng ta gán giá trị cho chỉ thị này là một biểu thức nào đó có trả về giá trị true hoặc false, nếu biểu thức trả về true thì element sẽ hiện ra, ngược lại thì không.

- VD: <customer \*ngIf="isActive"></customer>

Trong đoan code trên, element <customer> sẽ được hiển thị nếu isActive trả về true, isActive có thể là một thuộc tính/biến nào đó hoặc một phương thức…v.v

* 1. NgFor

- Đây là chỉ thị lặp, có tác dụng lặp qua một danh sách các phần tử, khi chúng ta có một danh sách các phần tử, muốn hiển thị chúng lên trang web thì chúng ta lặp qua danh sách đó và hiển thị các phần tử theo một khuôn mẫu giống nhau

* 1. ngSwitch

- Chỉ thị ngSwitch cũng tương tự như câu lệnh switch trong Javascript vậy, chỉ thị này có tác dụng hiển thị một element trong một danh sách các element, dựa vào một điều kiện cho trước.