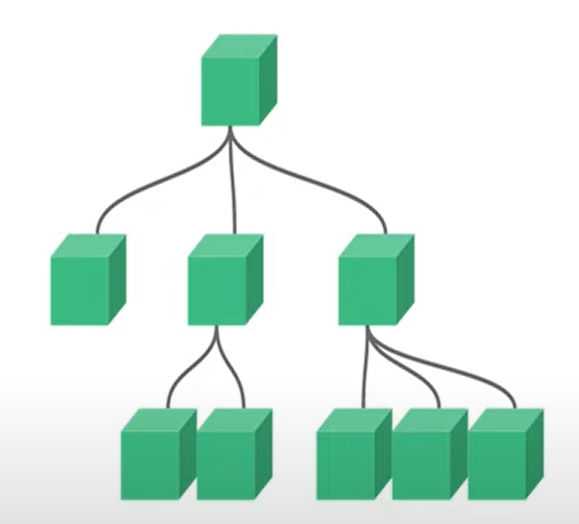
Báo cáo ngày 17/4/2014

Kiến thức hôm nay:

* Cách cài và chạy Angular.
* Template trong Angular:

+ 

+ 3 file quan trọng trong angular: 1. html

2. style

3. component (app.component.ts)

* Component trong Angular:
* Tạo các component quan trọng.
* Hiểu về standalone(là một tính năng được giới thiệu từ Angular 14 (và cải tiến ở các phiên bản sau) nhằm đơn giản hóa việc phát triển ứng dụng bằng cách cho phép các component, directive, và pipe hoạt động độc lập mà không cần phải khai báo trong một NgModule)
* Text, Properties, Attribute, Even và cách dung
* Binding trong angular( có 2 loại ):
* One-Way Binding:

1. Dữ liệu chỉ chảy theo **một hướng**: từ model sang view hoặc từ view sang model.
2. Thích hợp khi chỉ cần hiển thị dữ liệu hoặc xử lý sự kiện đơn giản.
3. Nhẹ hơn, vì chỉ theo dõi thay đổi một chiều.
4. Không cần import module cụ thể.
5. Dễ kiểm soát hơn, vì dữ liệu chỉ thay đổi ở một phía.

* Two-Way Binding:

1. Dữ liệu chảy **hai hướng**: thay đổi ở model cập nhật view và ngược lại.
2. Thích hợp cho form, nơi người dùng nhập liệu và dữ liệu cần đồng bộ ngay lập tức.
3. Nặng hơn, vì phải theo dõi cả thay đổi từ view và model.
4. Cần import FormsModule trong NgModule để sử dụng ngModel.
5. Phức tạp hơn, dễ gây lỗi nếu không quản lý tốt (ví dụ: vòng lặp cập nhật).

* Pipe là một tính năng mạnh mẽ dùng để **chuyển đổi dữ liệu** (transform data) trước khi hiển thị trên giao diện (view). Pipe giúp định dạng, lọc hoặc xử lý dữ liệu một cách dễ dàng và tái sử dụng mà không cần thay đổi dữ liệu gốc. Pipe thường được sử dụng trong template với cú pháp |.
* Thường chia làm 2 loại
  + 1. Pure Pipe:

1. Kiểu mặc định trong Angular
2. Chỉ chạy khi giá trị đầu vào (hoặc tham số) thay đổi (theo tham chiếu).
3. Tối ưu hiệu suất vì không chạy lại nếu dữ liệu không thay đổi.
4. Ví dụ: Các built-in Pipe như uppercase, date là Pure Pipe.
   * 1. Impure Pipe:
5. Chạy lại mỗi khi có bất kỳ thay đổi nào trong component, kể cả khi đầu vào không thay đổi.
6. Hữu ích khi xử lý dữ liệu thay đổi thường xuyên, như mảng hoặc đối tượng phức tạp.
7. Có thể gây ảnh hưởng đến hiệu suất nếu lạm dụng.

Dự kiến ngày mai:

* Cách truyền và nhận dữ liệu qua lại giữa những component.
* Vòng đời của component