**Explain angular standalone vs module with example?**

Angular Standalone Components provide an alternative to the traditional NgModule-based structure for building Angular applications. Here's a breakdown of their differences with examples:

NgModule:

* **Purpose:** Organizes related components, directives, pipes, and services into a cohesive unit.
* **Structure:**
  + Requires a module declaration (@NgModule) with imports, declarations, exports, and providers.
  + Components, directives, and pipes are declared within the module.
* **Example:**

TypeScript

import { NgModule } from '@angular/core';  
import { CommonModule } from '@angular/common';  
import { MyComponent } from './my.component';  
  
@NgModule({  
 declarations: [MyComponent],  
 imports: [CommonModule],  
 exports: [MyComponent]  
})  
export class MyModule { }

**Standalone Component:**

* **Purpose:** Enables building components independently, without the need for a surrounding NgModule.
* **Structure:**
  + Uses the standalone: true flag in the component decorator (@Component).
  + Imports dependencies directly within the component.
* **Example:**

TypeScript

import { Component } from '@angular/core';  
import { CommonModule } from '@angular/common';  
  
@Component({  
 selector: 'app-my-standalone',  
 templateUrl: './my-standalone.component.html',  
 standalone: true,  
 imports: [CommonModule]  
})  
export class MyStandaloneComponent { }

**When to Use:**

* **NgModule:**
* Large applications with complex organization.
* Sharing modules across multiple projects.
* Grouping related functionalities.
* **Standalone Component:**
* Smaller, focused components.
* Building reusable libraries.
* Reducing boilerplate code.
* Improving tree-shaking and bundle size.

Key Differences:

* **Dependency Management:**

NgModule imports other modules, while standalone components import dependencies directly.

* **Declaration:**

Components are declared within NgModules, but standalone components are self-contained.

* **Bootstrapping:**

An NgModule is bootstrapped to start the application, whereas standalone components can be bootstrapped directly.

Example of Migration:

**NgModule-based.**

TypeScript

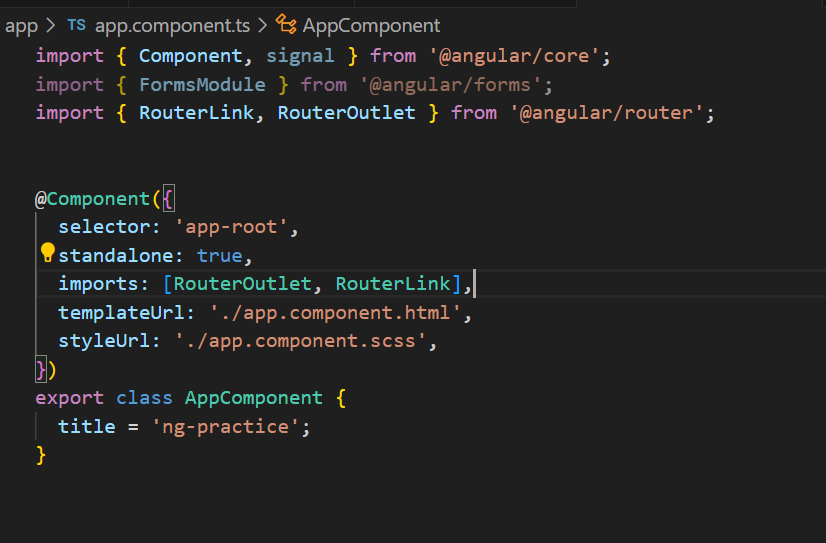
import { NgModule } from '@angular/core';  
import { BrowserModule } from '@angular/platform-browser';  
import { AppComponent } from './app.component';  
  
@NgModule({  
 declarations: [AppComponent],  
 imports: [BrowserModule],  
 bootstrap: [AppComponent]  
})  
export class AppModule { }



**standalone.**

TypeScript

import { bootstrapApplication } from '@angular/platform-browser';  
import { AppComponent } from './app.component';  
  
bootstrapApplication(AppComponent);

****