Modules;

Angular applications are modular has its own modularity system called NgModules.

Every Angular app has at least one NgModule class.

NgModule is a decorator function that takes a single metadata object whose properties describe the module.

important properties are:

/// Declarations the view classes that belong to this module.

/// Angular has three kinds of view classes: [components](https://angular.io/guide/architecture#components), [directives](https://angular.io/guide/architecture#directives), and [pipes](https://angular.io/guide/pipes).

/// exports   the subset of declarations that should be visible and usable in the component [templates](https://angular.io/guide/architecture#templates) of other modules.

/// Imports other modules whose exported classes are needed by component templates declared in this module.

/// providers - creators of services that this module contributes to the global collection of services; they become accessible in all parts of the app.

/// Bootstrap the main application view, called the root component, that hosts all other app views. Only the root module should set this bootstrap property.

### NgModules;

The NgModule is a class decorated with @ngmodule is a fundamental feature of Angular.

### JavaScript modules;

JavaScript also has its own module system for managing collections of JavaScript objects. It's completely different and unrelated to the NgModule system.

### Angular libraries;

Angular as a collection of JavaScript modules

Each Angular library name begins with the @angular prefix.

Examples= import { [Component](https://angular.io/api/core/Component) } from '@angular/core';

import { [BrowserModule](https://angular.io/api/platform-browser/BrowserModule) } from '@angular/platform-[browser](https://angular.io/api/animations/browser)';

## Components;

## Components control a patch of screen called a view ( app root)

## Templates

You define a component's view with its companion **template**. A template is a form of HTML that tells Angular how to render the component.

## Metadata

Metadata tells Angular how to process a class

## Directives

Angular templates are dynamic. When Angular renders them, it transforms the DOM according to the instructions given by **directives**.

## Services

Service is a broad category encompassing any value, function, or feature that your application needs.

* logging service
* data service
* message bus
* tax calculator
* application configuration

## Dependency injection

Dependency injection is a way to supply a new instance of a class with the fully-formed dependencies it requires.