数据绑定:双括号{{}}

Angular 应用是模块化的，并且 Angular 有自己的模块系统，它被称为 Angular 模块或 NgModules。

看src/app/app.module.ts

[NgModule](https://angular.cn/api/core/NgModule)是一个装饰器函数，它接收一个用来描述模块属性的元数据对象。其中最重要的属性是：

* declarations - 声明本模块中拥有的*视图类*。Angular 有三种视图类：[组件](https://angular.cn/guide/architecture#components)、[指令](https://angular.cn/guide/architecture#directives)和[管道](https://angular.cn/guide/pipes)。
* exports - declarations 的子集，可用于其它模块的组件[模板](https://angular.cn/guide/architecture#templates)。
* imports - *本*模块声明的组件模板需要的类所在的其它模块。
* providers - [服务](https://angular.cn/guide/architecture#services)的创建者，并加入到全局服务列表中，可用于应用任何部分。
* bootstrap - 指定应用的主视图（称为*根组件*），它是所有其它视图的宿主。只有*根模块*才能设置bootstrap属性。

我们通过引导根模块来启动应用。 在开发期间，你通常在一个main.ts文件中引导AppModule，就像这样： 看main.ts

NgModule（一个带@[NgModule](https://angular.cn/api/core/NgModule)装饰器的类）是 Angular 的基础特性之一。

JavaScript 也有自己的模块系统，用来管理一组 JavaScript 对象。 它与 Angular 的模块系统完全不同且完全无关。

JavaScript 中，每个*文件*是一个模块，文件中定义的所有对象都从属于那个模块。 通过export关键字，模块可以把它的某些对象声明为公共的。 其它 JavaScript 模块可以使用*import 语句*来访问这些公共对象。

content\_copyimport { [NgModule](https://angular.cn/api/core/NgModule) } from '@angular/core';

import { AppComponent } from './app.component';

content\_copyexport class AppModule { }



Angular 提供了一组 JavaScript 模块。可以把它们看做库模块。

每个 Angular 库的名字都带有@angular前缀。

You install them with the npm package manager and import parts of them with JavaScript import statements. 用 npm 包管理工具安装它们，用 JavaScript 的import语句导入其中某些部件。

[Learn more about the JavaScript module system on the web.](http://exploringjs.com/es6/ch_modules.html) [学习更多关于 JavaScript 模块的知识。](http://exploringjs.com/es6/ch_modules.html)

这两个模块化系统是互补的，我们在写程序时都会用到。