NgModule为组件集声明了编译的上下文环境，它可以将其组件和一组相关的代码（如服务）关联起来，形成功能单元

组件和模块共同定义了Angular的视图

Angular在每个javascript事件循环中处理所有的数据绑定，从组件树的根部开始，递归处理全部子组件

其实，组件从技术角度来说就是一个指令，但是他的特殊性，angular专门为他定义了一个@Component装饰器，他使用了面向模板的特性扩展了@Directive装饰器

指令包括：结构型指令（添加或移除dom元素）、属性型指令（改变元素外观，看起来像html元素的属性）

当Angular创建组件类的新实例时，它会通过查看该组件类的构造函数，来决定该组件依赖哪些服务或者其他的依赖项

三种级别的服务：

1、根级别（在@Injectable中使用providedIn:’root’属性）：单一共享实例，任何类都可以使用，（很好的一种方式，这样Angular可以通过移除那些从来没有被使用的服务来优化应用）

2、模块级别（在@NgModule中使用providers:[]属性）：同一模块下的所有组件共享同一实例，

3、组件级别（在@Component中使用providers:[]属性）：该组件的每一个实例都会新创建一个服务实例

差值表达式：包括模板表达式、模板语句

模板表达式：双括号之间的东西叫模板表达式（并不包括双括号，在属性绑定中，他是用双引号包裹起来的，出现在等号右侧，注意，在事件绑定中，出现在右侧用双引号包裹起来的叫：模板语句），Angular会对其求值，并转化为字符串，与相邻的字符串字面量连接起来

注意，Angular会将模板表达式转化为属性绑定，并不是简单的赋值操作

被禁用的javascript表达式：<https://www.angular.cn/guide/template-syntax>，赋值、new运算符、自增自减运算符，不支持位运算|和&，拥有新的运算符：（|）、（？.）、（!）

表达式的上下文：三种来源：

1、模板变量（模板输入变量、模板引用变量）

2、指令的上下文变量

3、组件对象实例

优先级依次降低

注意，模板表达式不能引用全局命名空间中的任何东西，比如window、document，也不能调用console.log、Math.max等，他们只能引用表达式上下文中的成员

最佳实践：最好使用幂等的模板表达式，因为他完全没有副作用，幂等表达式总是返回完全相同的东西

常见的模板表达式：属性名或者方法的调用

模板语句：用来相应有绑定目标（如 HTML 元素、组件或指令）触发的事件，有副作用，但是，这是事件处理的关键

注意：模板语句和模板表达式不同，他支持：赋值（=）、表达式链（;和,），但是，new、自增自减、操作并赋值（+=、-=）、位操作符、模板表达式操作符等不支持

语句的上下文：（注意与表达式的上下文的联系与区别）两种来源：

1、组件实例

2、模板自身的上下文（例如：模板的$event对象、模板变量）

优先级依次升高

同表达式上下文一样，他不能引用全局中的任何东西

函数调用、属性赋值是常见的模板语句

注意：数据绑定绑定的是DOM、组件、指令中的Property或事件，而不是html的attribute，这两者有很大的区别

在各种绑定中，出现在等号左边的是目标名（DOM中的某些东西，property或者事件），就是Property，而不是attribute，见<https://www.angular.cn/guide/template-syntax>

HTML+：用封装了 HTML 的组件创建新元素，并把它们当作原生 HTML 元素在模板中使用。也就是指令、组件扩展的html

属性绑定还是差值表达式：当目标需要的是字符串，推荐使用差值表达式，当目标需要的是对象，就只能使用属性绑定

attribute绑定（attr前缀）：因为有些元素并没有相应的property，例如：colspan，这样就需要attribute绑定了

css类绑定（class前缀）：三种形式：

1、覆盖：直接使用[class]=”xx”的形式，

2、添加或者移除某类：[class.xx] = “返回true/false的表达式”，设置单一css类

3、使用NgClass指令，设置多个css类

style样式绑定（style前缀）：两种形式：

1、使用[style.xxx]=””，设置单个样式

2、使用NgStyle指令，设置多个内联样式

事件绑定：有等号左边的目标事件和等号右边的模板语句组成

Angular解析属性绑定和事件绑定的顺序：都是先指令在元素！