一个模块提供的服务，怎样被其他模块使用

在模块中导出另一个模块时什么用法，（扩展功能？像typescript那样）

个级别的服务是怎么回事？也需要使用import的方式？那也可以在作用域以外使用import的方式来引入服务啊？

答案：由Angular DI框架自己控制，因为Angular会查找类的构造函数中参数的类型，以及其元数据，这样Angular就可以区分某个服务是属于哪个级别的了，

因此：虽然使用服务就得import服务类，但是所使用的服务实例还是依据服务所属级别，所以，这里针对的是服务实例的级别

Rxjs？

Observable？

对于typescript，如果参数传多了，方法调用会成功吗？

点击一个链接，打开两个路由

为什么在AppModule中导入了CommonModule，其他模块中也可以使用\*ngIf等指令？

答案：1、在特性模块中导入CommonModule可以使他运行在任何目标浏览器中，那些跨平台的作者更喜欢这种方式！，而这里的特性模块不用导入CommonModule是因为，在AppModule中导入了BrowserModule,且，他们运行在浏览器中

如果我两次导入了同一个模块会怎么样！

答案：没有任何问题，Angular只会在首次遇到时加载一次模块，例如：如果模块'B'导入模块'A'、模块'C'导入模块'B'，模块'D'导入 [C, B, A]，那么'D'会触发模块'C'的加载，'C'会触发'B'的加载，而'B'会加载'A'。 当 Angular 在'D'中想要获取'B'和'A'时，这两个模块已经被缓存过了，可以立即使用。但是。不允许模块间的循环依赖

（不过，如果你在 AppModule 的 @NgModule() 元数据中配置了全应用级的提供商，它就会覆盖通过 @Injectable() 配置的那一个），那如果我配置了多个providedIn: ‘root’，他们会相互覆盖吗？

注入器树是和组件树同时存在的还是，需要显示创建(通过providers数组)

答案：看来是同时存在的，一个组件的创建，就会自动建立一个组件注入器

使用RouterModule.forRoot以及RouterModule.forChild来配置多组路由时，Angular时怎样将这些路由信息合并成单一的值？

答案：ROUTES令牌！！（具体还不清楚）

值提供商，非类依赖，工厂提供商？

注入组件的DOM元素？[https://www.angular.cn/guide/dependency-injection-in-action#inject-the-components-dom-element](https://www.angular.cn/guide/dependency-injection-in-action" \l "inject-the-components-dom-element)

类型（type），类，值，在typescript中到底有啥区别，假设有一个string类型，一个Test类，和一个‘test’字符串，为啥下面写法是正确的

Const aaa = {

Key: Test

}（为啥对象的属性的值可以为类名？）

这样写也是正确的 type cd = Test，const fa：cd = new Test（）；

而这样写就是错误的！！

Const aaa = {

Key：string

}

Const fa = new cd（）；

模块与模块之间怎么联系？只能通过路由吗？

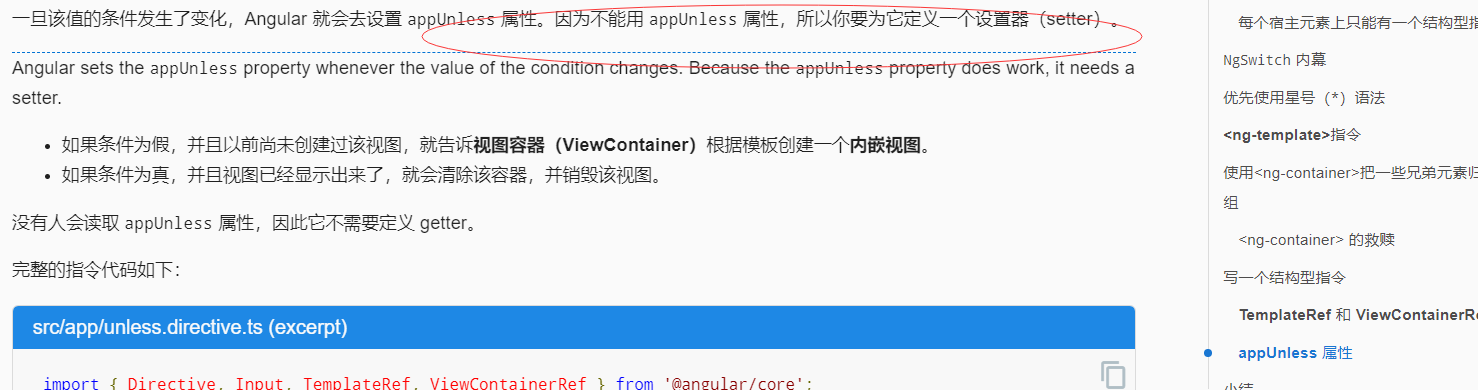
使用setter拦截输入属性值得变化时咋回事？

为什么Angular在查找属性绑定的时候，有优先查找是否有对应的指令

答案：因为组件是指令的一种，所以指令也有属性绑定，而有一种指令叫属性型指令，他可以使用属性绑定的额形式，同时干两件事：1、将该指令应用到宿主元素上，2、属性绑定同名的指令类中的一个属性！！！

属性型指令控制其他指令、组件、元素的外观或者行为，这个行为咋理解，有什么实例吗？

属性型指令：[ngSwitch],就是一个很好的例子，因为他控制着其他两个结构型指令的行为！\*ngSwitchCase、\*ngSwitchDefault



为什么？

答案：难道是因为Angular在解开\*语法糖的时候，该指令被移到了<ng-template>中，那么在本指令中就访问不到了？

在为模板驱动表单添加自定义的验证器的时，在将自己注册为NG\_VALIDATORS提供商的服务时，为什么使用的是useExisting来提供服务实例？

答案：因为这是指令，也就是说：Angular会在模板中识别出来，并实例化它，并赋值它的输入属性，这时，已经存在一个该指令的实例，并且输入属性已被赋值！！，然后（Angular帮我们做了很多事情，其中一样应该就是在某处使用NG\_VALIDATORS提供商！！）这样的话，如果使用useExisting，就可以直接使用已经存在的指令实例提供服务，如果使用useClass，那么会新建一个实例，这时的实例的输入属性是没有被赋值的！！

那组件类中的private修饰符有什么用？就算是private，组件的模板还是可以直接访问他！！

答案：难道是，在实例之间其作用？

NgModule、指令、组件的注入器有啥不同

答案：吊炸天，是否记得有这样一句话：注入器本质上不属于组件本身，而是属于组件所依附的DOM元素，指令也一样，所以，组件和指令共享所依附的DOM元素的注入器！！！

（而ngModule？？？）ngModule级的注入器和组件级的注入器并不是一回事，可以说：ngModule级的注入器时旗下所有组件级注入器的父级，因此，当你在惰性加载的模块中提供服务，而你在旗下的组件中使用服务，并且使用@SkipSelf，它找到的服务实例还是该模块提供的那个！！！！

1、ngModule不会为其imports数组中的每个都module新建一个实例！！也就是说：如果有多个模块导入了同一模板，也只会有一个实例（注意：这里只是普通的模板包含，而不是惰性加载的模板），并注入到自己的注入器中，

2、而对于exports数组中的东西，新建一个实例，而不是共享实例

3、ngModule不会将其declarations数组中的可声明对象的实例注入到其自己的注入器中

4、ngModule注入器中可能有哪些东西？有自己ngModule的实例，但是没有其惰性加载模块的实例

5、组件注入器中可能有哪些东西？，组件注入器中至少包括本组件自身的实例、但不包括其子组件的实例，（很容易理解，因为一个组件可能有很多子组件，但只可能有一个父组件）、组件级别的服务实例、指令实例（同属于一个元素的指令）

6、看来，ngModule不仅会将自己注入自己的注入器，还会将该实例注入到惰性模块的注入器中,以及惰性模块的组件注入器中！！！而且好像都是同一实例

7、加载惰性模块，如果服务是module级的，那么久会生成多个实例！！，而如果不是使用惰性加载模块，而是普通的模块的包含，那么这时，不管哪个module级的服务。它都是应用级别的服务！！！（如果将普通的模板包含看成是组合成一个模板的形式，是不是好理解一些）