# Angular 组件

# 组件基础

## 什么是组件

组件（Component）是构成Angular应用的基础和核心。通俗来说，组件用来包装特定的功能，应用程序的有序运行依赖于组件之间的协同工作。以我们汽车而言：轮胎、发动机、变速箱就都是组件。细化下来就是金属零件。我们的Angular项目就是一颗组件树！

## 模块的概念

组件化（把我们的项目以组件的形式构建）-> 模块化。在Node.js中，模块就是一个文件，引入一个文件是简单import、require(‘filePaht’)。在我们的前端领域衍生出了很多的模块化的规范。我们有了很多类似于 AMD、CMD、ES6等等。从项目的组织（资源目录）而言我们通常能见到两种。

1. 按资源划分
2. 按模块划分

资源逐渐按按模块划分就渐渐形成了我们组件化的方式。（优点：高内聚、低耦合）

因为各个框架的模块化实现不同，所以我们出现很多的模块的工具，Grunt、Gulp、Webpack，这些工具之间的模块化又各不相同。

## Web Component标准

W3C为了统一组件化的标准方式。通过标准化的非入侵方式封装组件，每个组件都包含自己的HTML、JS、CSS代码，并且不会对页面上其他的组件产生影响。Angular起初以Web Component标准为蓝本进行设计的。在Angular中引入了视图包装（ViewEncapsulation）的概念，允许通过设置ViewEncapsulation.Native选项来使用原生的 Shadow DOM。

## 组件创建

命令： ng g c xxx（组件名称）；带分组那么就是 ng g c group/xxx。

1. 从@angular/core 中引入Component装饰器。
2. 建立一个普通的类，并用@Component装饰它。
3. 在@Component中设置selector 自定义标签和template模板。

## 元数据

不仅仅是templateUrl 、selector这几种，还有很多感兴趣自己了解。

Angular中的每一句代码都是可复用的！

# 组件交互

## 基础用法

Angular提供了输入（@Input）和输出（@Output）语法来处理组件数据的流入流出。组件间数据交互的方式还有路由参数、共享服务。

打个比方游戏机：

它不关心谁在操作，只关心按了什么键。同样，它也不关心要把我们操作结果显示给谁看，只管显示在屏幕上。 总结：黑盒模型。在Angular而我们的组件就是要设计成黑盒模型。

（在Angular中快速引入包 alt + enter）

@Input 父组件传递数据到子组件的方式。（单向传递）写在子组件中

@Output 子组件传递数据到父组件。重点通过EventEmitter（发射器）从子组件发射数据到父组件中。通过EventEmitter. Emit() 执行发射。写在子组件中

## 非父子关系组件传递数据

如果是通过@Input、@Output来进行的话是不鼓励的，推荐使用共享服务。把两个没关系的组件找他们共同的“爸爸”，来当中间人。

## 通过局部变量实现数据交互

模板表达式的方式： “#” +自定义名称，就可以代表它所被申明所在的组件或者DOM元素的实例。

功能：能够让父组件调用（读）到子组件的参数或者方法。

## 通过@viewChild注入的方式

特点：能让父组件去读写我们子组件中的方法或者参数。

使用：

1. 位置是写在父组件中。
2. 方式：

@viewChild(OrderComponent) order: OrderComponent;

## 补充

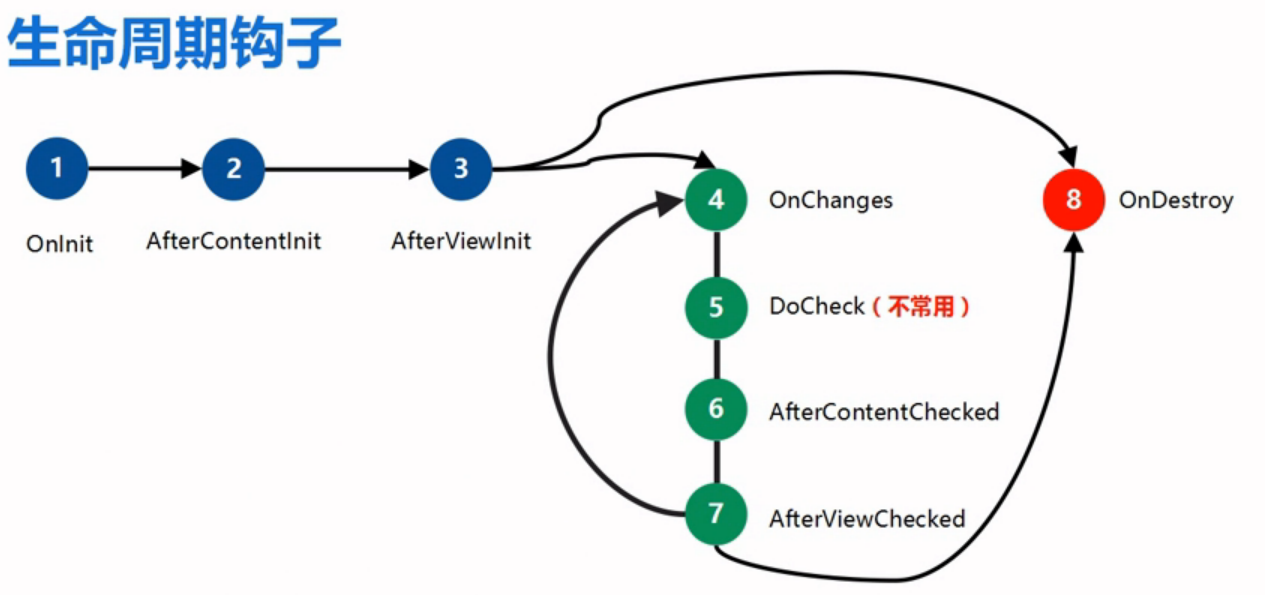
1、[(ngModel)]报错解决办法： ngModel属于FormsModule，在app.module.ts中import这个FormsModule。

2、@Input和@Output 可以设置别名。(‘’)

3、@Input 有钩子函数。Set get

# 组件生命周期

从组件创建开始Angular的变更检测机制就会监控组件。 组件的生命周期由Angular内部管理，从组件的创建、渲染，到数据变动事件的触发，再到组件从DOM中移除，Angular都提供了一些列的钩子（接口）。这些钩子可以让开发者很方便地在这些事件触发时，执行相应的回调函数。



上图中，蓝色的钩子以及红色的是只执行一次的，绿色钩子是会重复执行的。

整个组件生命周期分为三个阶段：

1. 初始化阶段。
2. 变更检测阶段。
3. 销毁阶段。

初始化阶段结束后会让组件显示在用户面前，钩子的执行顺序为：

Constructor -> ngOnChanges（可选当有@Input发生时） -> ngOnInit -> ngDoCheck -> ngAfterContentInit -> ngAfterContentChecked -> ngAfterViewInit -> ngAfterViewChecked

具体的钩子含义：



钩子函数实际上就是接口，而从纯技术的角度上来说，接口对于Javascript和Typescript都是可选的。Javascript语言本身没有接口。 Angular在运行时候看不到Typescript 接口。

幸运的接口不是必须的！

## 了解可变和不可变对象。

从内存的角度来分析：

可变：对象 Object。

不可变的： string / number/ Boolean/…

<http://www.cnblogs.com/Ziksang/p/5203551.html>

## OnChange钩子

前提： 输入属性 @Input / @viewChild

1、只有输入的不可变对象，发生改变时（内存消耗的时候）才会触发。

## 变化监测Zone

官方地址：<https://github.com/angular/zone.js>

**出身**：Zones实际上是Dart的一种语言特性

**核心概念**：Zone 是下一个 ECMAScript 规范的建议之一，Angular团队实现了Javascript版本的zone称为zone.js。

**作用**：它是用于拦截和跟踪异步工作的机制，也就是说Zone能够hook到异步任务的执行上下文，并在一些关键节点上重写相应的钩子方法，以此来完成某些操作。

什么情况下才需要变化监测：模型改变的！！！

什么操作（事件）会改变模型：

三类：

1. Events: click、mouseover、keyup… （改变View）
2. Timers: setInterval、 setTimeout
3. XHRs: Ajax(GET、POST…) （数据交互）

总结这些事件源有一个共同的特点，即它们都是**异步操作。**那我们可以这样认为，所有的异步操作都有可能会引起模型的变化。

## ngZone

zoneJS 民间介绍：<http://www.cnblogs.com/whitewolf/p/zone-js.html>

ngZone 民间介绍: <http://blog.csdn.net/qq_34398308/article/details/51546091>

Zone的功能：

1、拦截异步任务调度

2、提供了将数据附加到 zones 的方法

3、为异常处理函数提供正确的上下文

4、拦截阻塞的方法，如 alert、confirm 方法。

ngZone是Zone子类，原理：(Monkey-patched)猴子补丁的方式暴力的将所有的异步事件都包裹了。

我们的Angular项目是有组件树构成的，然后每当我们创建一个组件的时候Zone.fork()克隆一个子zone属于我们父组件中的zone监控中。

默认情况下zone重写了以下方法:

setInterval、clearInterval、setTimeout、clearTimeout

alert、prompt、confirm

requestAnimationFrame、cancelAnimationFrame

addEventListener、removeEventListener

## doCheck

注意：不要轻易的使用，就算使用也要写得足够轻量级。

它会监听所有的可能跟我本身组件相关的事件，所以会不停（无意义）的触发。

Angular中两种变更检测的策略：

1. Default： 会检测所有组件树。（默认）
2. OnPush: 当该组件的输入属性变化时才检测。

## AfterView 钩子

主要是监听 @viewChild 改变时的钩子。

注意：

1、先Init/Checked。都是在视图创建完毕后调用。

2、如果有子组件，会先子后父。

3、钩子里面不能做会导致组件视图更新的操作。（也就是说我们不能直接写改变代码，你需要写在另外的上下文中 setTimeout。）

## AfterContent钩子

监听 <ng-content></ng-content>

# 组件内容嵌入

内容嵌入又称投影（ng-content）是组件的一个高级功能特性，使用组件的内容嵌入特性能很好地扩充组件的功能，方便代码的复用。它和AngularJS 1.x中指令的transclude 属性非常类似。

内容嵌入通常用来创建可复用的组件，典型的例子是模态对话框或导航栏。

具体内容以及使用请参考learnComponent 中的embed 文件夹。

重点API：<ng-content></ng-content>

# 组件的复用示例

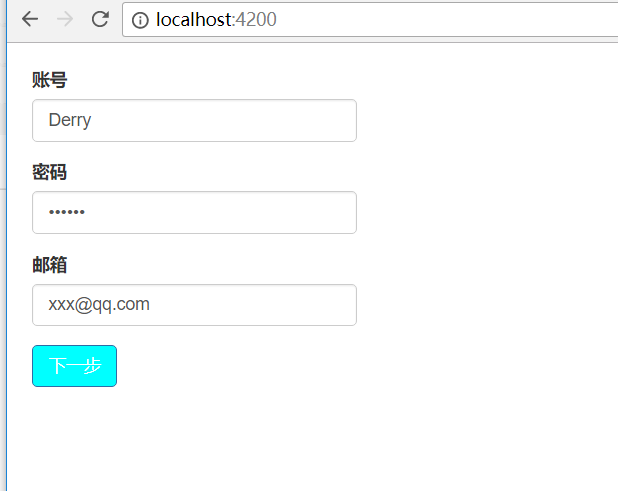
请参见multiplexComponent 项目。

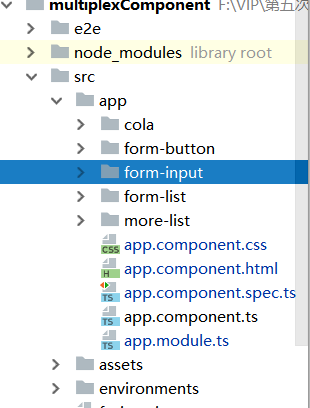
# 作业

## 作业内容

参照老师的multiplexComponent项目，完成自己的一个可复用的项目的实例。

## 示例：





## 格式以及发送方式

请把作业的结果（所有页面的截图，以及工程结构的截图），打成.zip或者.rar的压缩包通过QQ邮箱发给Derry老师（[2714977534@qq.com） 邮件的](mailto:2714977534@qq.com）%20邮件的)主题为作业课程的名称以及时间例如（Angular组件-9.13）。邮件内容：你的所在地以及真实姓名。例如（长沙-Derry）