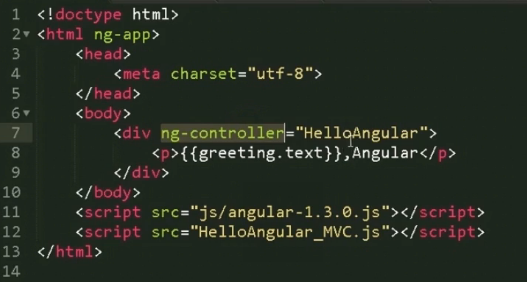
# 第一章 快速上手

## 四个新特性

### 1 MVC

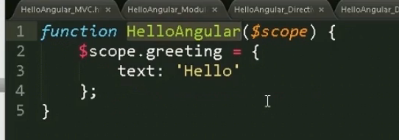


ng-controller是控制器（C-CONTROLLER）

<p>标签就是视图，（V-VIEW）

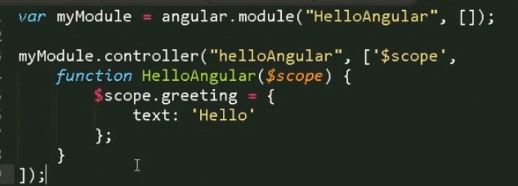
{{greeting.text就是module}} 双花括号代表取值

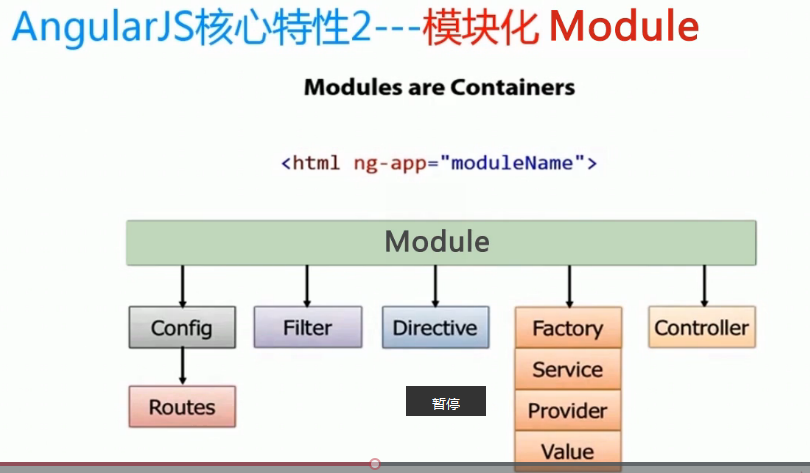
整个的输出是“Hello，Angular“，Hello是来自HelloAngular.js里面给text的赋值

HelloAngular.js内容，controller赋值给了HelloAngular.js，即这个函数充当了控制器

### 2 模块化

Html代码没有变化。Js代码如下：

myModule定义了一个模块，然后调用了controller方法，定义控制器的名称是helloAngular，$scope是依赖注入。



每一个单页面只能有一个ng-app 一个模块。Ng-app相当于java的main函数。

### 3 指令系统

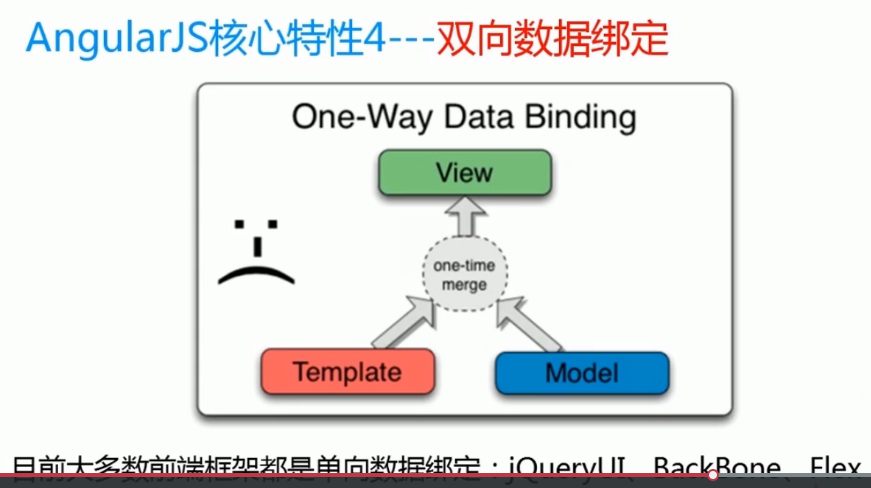
### 

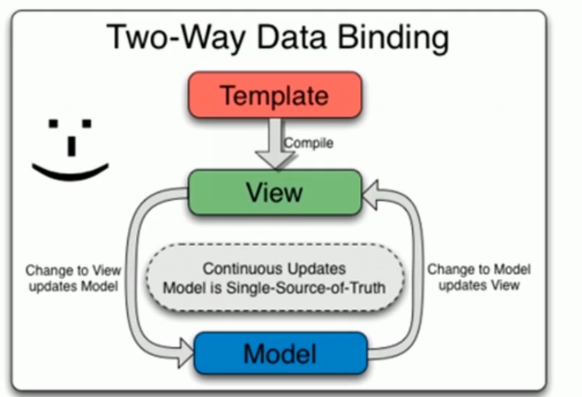


看js代码，一切都是从模块开始，先创建myModule的模块，调用directive方法，两个参数，第一个是指令的名称，后面是function。

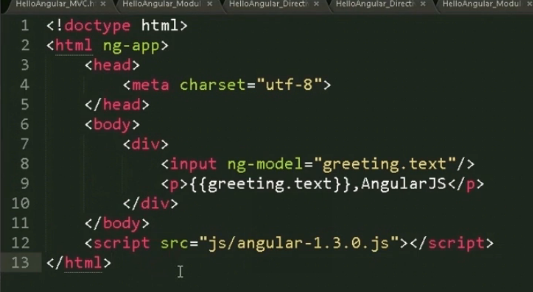
Template是模版，最后输出到html里面。这样html里面的<hello>这个自定义的标签，浏览器就认识了。

### 4 双向数据绑定

单项数据绑定的图。Model和template一起，通过数据绑定机制，生成html标签，然后插入到view文档流。缺点是，html一旦生成 就没法变了。

AngualrJS是双向的。

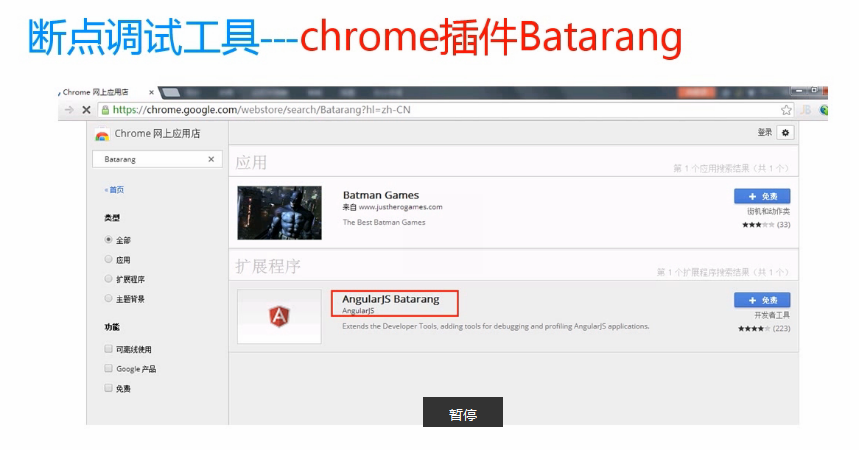
当视图的内容发生变化，模型也会变化。例如 表单的变化，用户的输入。



Model是greeting.text

下一行p标签双花括号取值。P标签上的内容会根据input里面的变化而变化

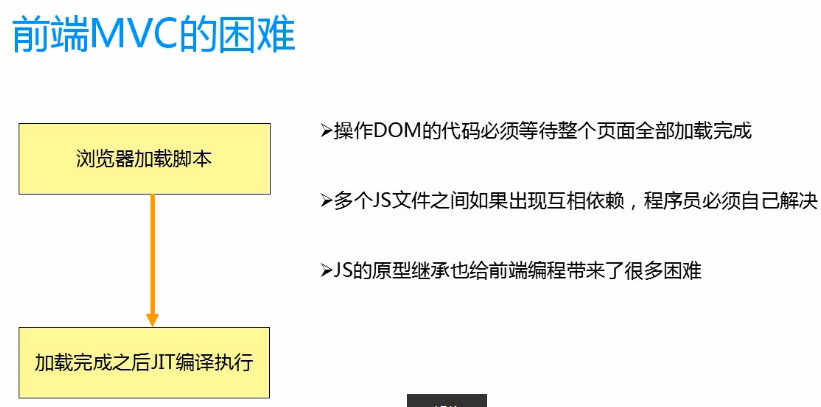
## 调试编辑代码相关

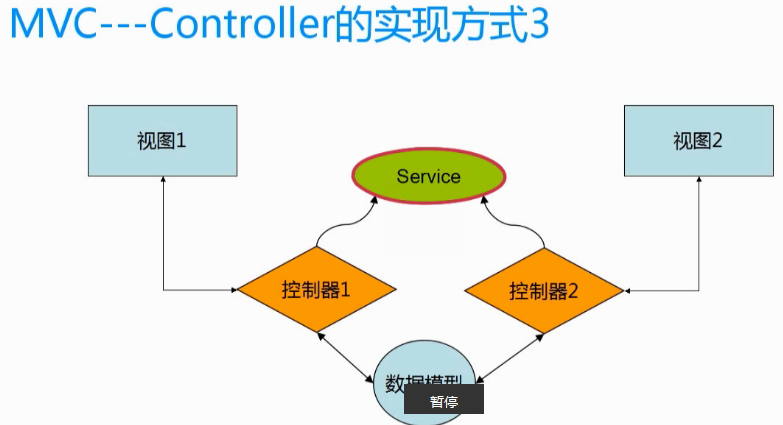




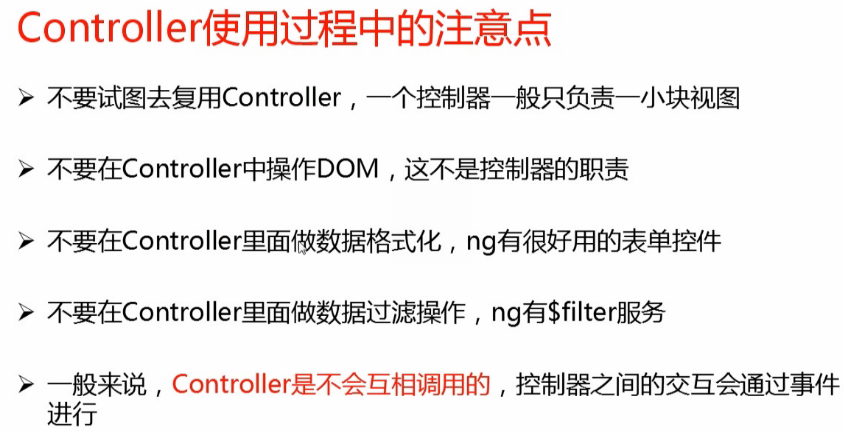
# 第二章 基本概念和用法

## 2.1 MVC



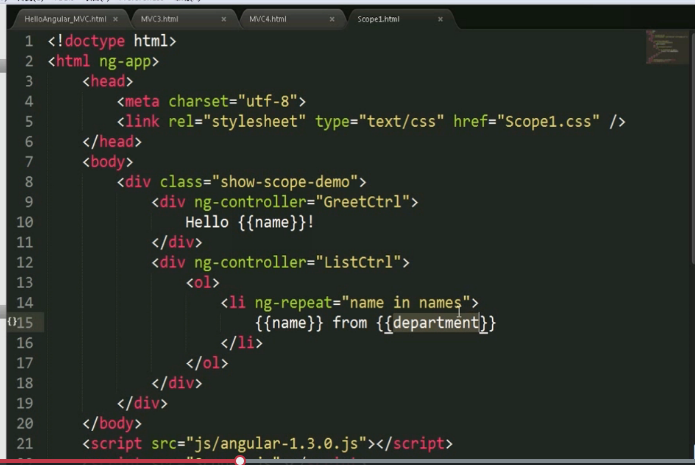


把控制器1，2之中通用的东西抽出来，形成service，让控制器去调用它。

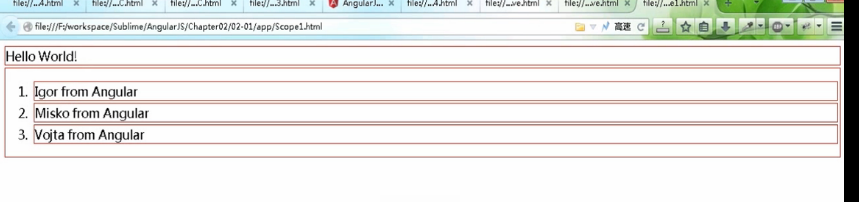


第二点 在directive里面封装func来操作DOM

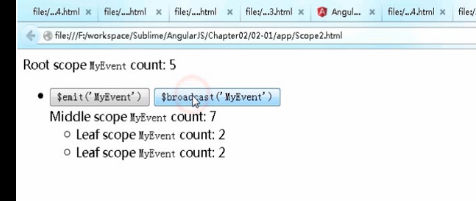
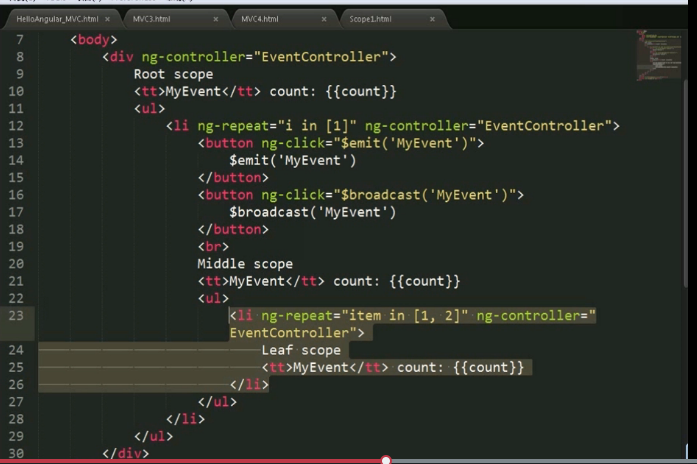
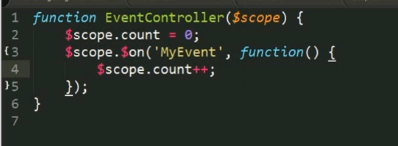
controller要做的就是 绑定特定的视图，操纵视图中所以需要的model







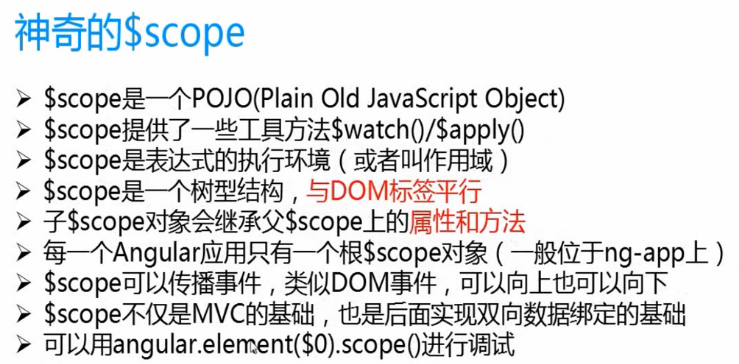
如果在本层找不到，就会向上一层一层的找，department是在rootScope里面，是根作用域，所以即使ListCtroller没有定义department，依然可以获得它的值。



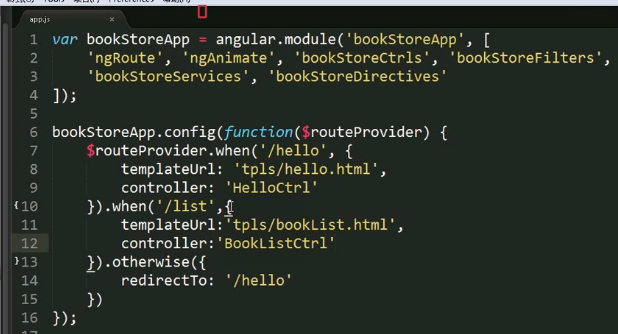
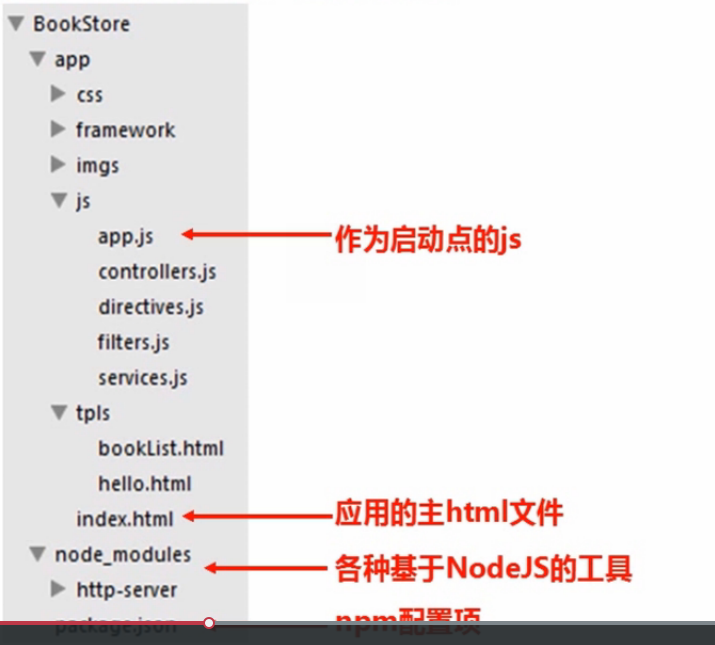
上述代码证明了scope是一个树形结构，和DOM平行的。子层标签可以继承复层标签的属性和方法

Angular的MVC是借助$scope实现的

Scope可以被赋值方法和属性



## 2.2路由 模块 依赖注入



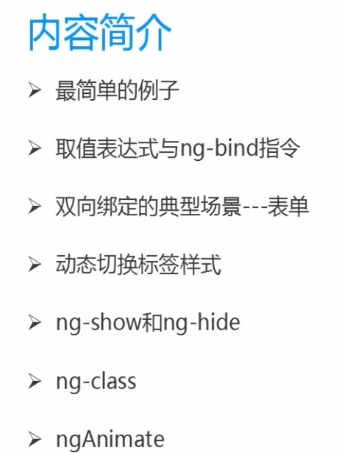
App.js

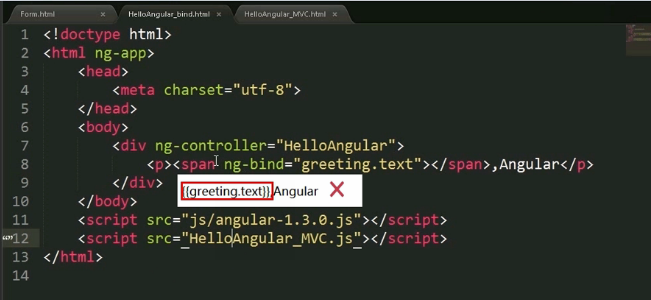




Control.js

## 2.3 双向数据绑定

（数据模型到视图，视图到数据模型，双向的绑定）



不再用双花括号的方式来取值，而是使用ng-bind。原因是当用双花括号的时候，如果网速比较差，有可能在html端页面，显示出｛｛greeting.text｝｝而不是取值后的“hello”。