# Angular

框架和库的区别：

库 ： 提供一些已有的方法，然后我们调用

框架 ： 提供一种结构或模式，我们按照它提供的这种结构或者模式去书写代码

Angular.module(“模板名”，[])； 创建模块

## 创建控制器

Angular.controller（“控制器名字”，function（$scope））

防止压缩

Angular.controller(“控制器的名字”，[

‘$scope’,

“$route”,

“$http”，

“$routeParams”

function($scope,$route,$http){

$scope.方法名 为页面添加方法数据

$route.updataParams() 可以改变地址的id（不是获取啊）

 前提是控制器与config项链

$routeParams .id 可以获取路由上：后的id



}]

### angular监听

$scope.$watch(“参一监听的内容，也可以是事件”，

“function(new , old){ }”）

New新的值 old 上一个值

## 指令

Ng-modole =’方法名字’ 双向数据绑定

防止闪烁

Ng-bind 只能双标签用

Ng-cloak class= “ng-cloak”

循环 ng-repeat= “item in data (track by $index) ”

一些布尔值的属性

$even $odd $first $last $middle

Ng-class = “样式: true/false”; true表示引用

Ng-show/ng-hide = “” 值也是true

Ng-if = “” 也可以隐藏 在DOM中直接移除

## Angular自定义指令

App-directive(“名字”，[function(){ retune :是个对象

Templata:需要一共一个html字符串

templataUrl：1提供一个地址

2提供一个模板ID

Restrict:”ACEM” 什么样的格式呈现 attribute，class，element， 注释

Replace：true 把模板替换到自定义上

Transclude：true 把自定义标签中的内容替换模板中的内容。

Scope： {  
 // @test是获取dom中test属性的样式（值）; 然后给test,在给模板中写{{test}}  
 //test1 对应模板中的{{test1}} @后的test和dom中的test属性对应  
 //test1:'@test' 对应模板{{test1}}  
 //这的 test和dom中test，模板中的test都一样  
 **test**:"@"//简写方式 对应模板{{test}}

var app = **angular**.module("app",[]);  
//创建自定义指令  
app.**directive**("myDemo",[function () {  
 //必须要有返回值 返回的是一个对象  
 return {  
 //template提供一个html的 字符串  
 //template:"<p>好困啊！-- zzz</p>"  
  
 //templateUrl 的第一种方法 写地址  
 //templateUrl:"./temp.html"  
  
 //templateUrl 的第二种方法 id  
 //重新创建一个script标签 根据id获取它的模板 type属性为 'type/ng-template'  
 **templateUrl**:'tep',  
 //restrict(限制):A- attribute C- class E- element M-'注释'  
 //如果只写一种 就会只显示一种形式 限制其他  
 **restrict**:"ACEM",  
 //取代，代替  
 **replace**:true, //将模板中的标签取代自定义标签  
 //将自定义标签的内容替换模板中有ng-transclude属性的内容，  
 **transclude**:true,  
 **scope**:{  
 // @test是获取dom中test属性的样式（值）; 然后给test,在给模板中写{{test}}  
 //test1 对应模板中的{{test1}} @后的test和dom中的test属性对应  
 //test1:'@test' 对应模板{{test1}}  
 //这的 test和dom中test，模板中的test都一样  
 **test**:"@"//简写方式 对应模板{{test}}  
 },  
  
 link: function (scope, element, attributes) {  
 //这的scope相当于$scope ，作用域整个模板  
 scope.**example** = "我是link里的scope参数";  
 //对应模板的最外层标签  
 element.on("click", function () {  
 alert("duang!!!")  
 });  
 //对应自定义所在的标签里的 属性  
 **console**.log(attributes);  
 }  
 };  
}])

}]

## 过滤器 filter

Currency



<h2>currency</h2>  
<!--在数据后面添加过滤器的名字 ：后面跟参数-->  
<p>{{**money** | currency}}</p>  
<p>{{**money** | currency:'￥'}}</p>  
<h2>date</h2>  
<!--表示、年/月/日-->  
<p>{{**date** | date:'yyyy/mm/dd hh:mm:ss'}}</p>  
  
<h2>limitTo</h2>  
<!--第一个参数表示显示多少字符，第二个参数表示从第几个开始 -->  
<p>{{info | limitTo:8:0}}...</p>  
  
<H2>orderBy</H2>  
<!--遍历数组显示 +。-表示正反方向-->  
<span ng-repeat="**item** in **arr** | orderBy :'+'">{{**item**}}</span>  
  
<h2>json</h2>  
<!--格式化显示json数据 标签必须为<pre> json:3 跟参数的话 是缩进长度 -->  
<pre>{{**obj**| json:4}}</pre>

//在js中使用过滤器  
// 必须添加$filter参数（参1是过滤器的名字）（跟过滤器的内容）  
$scope.**jjss** = $filter("currency")($scope.**money**,"￥");

<ul>  
<!--filter过滤器，跟$filter不一样， 在数据中和filter后面的内容一样的显示（字符串也可以），不一样隐藏-->  
 <li ng-repeat="**item** in data | filter:{**completed**:true}">  
 {{**item**.name}} {{**item**.**completed**}}  
 </li>  
</ul>

## 路由 就是根据描点访问数据

参一：路由的名字 同时也是页面的标签名<haha></haha>

Var app = angular.module(“参一（haha）”，[“ng-Route”])

App.config([ ‘$routeProvider ’, function( $routeProvider){

参1 是路由的地址 如"/:movieType/:page?" “？” 表示没有默认第page页

“：”表示内容是可变的 一般为id值

$routeProvider.when(“参1 ”，{

Template:” ”，

Controller：“”

})

} ]

## 创建服务

// 总js里的[ “” ,””]

Var app = angular.module(“myServise”,[ ])

// 写在控制器里的名字

App.service( “服务名字”，[ function() {

This.name 内容全在this上面

}])