# Angular课程安排

一、什么是Angular，核心特性（理解，面试会说即可）

二、什么是MVC,MVVM,MVP等设计思想（理解，面试会说即可）

三、Angular的常用指令：ng-init、ng-repeat、ng-if、ng-show、ng-bind等等（重点）

四、三目运算、隔行改变、class、style、ng-include（会用）

五、过滤器和综合练习（重点）

六、原生JS和Angular互通的controller

七、定时器、延时器

八、watch监视

九、ng-src、ng-href、ng-copy、ng-change等等指令和属性

十、controller传递broadcast（广播），解决闪烁问题

十一、$http初级

十二、ng-disabled、ng-options（下拉用）

十三、创建指令

十四、添加指令

十五、指令创建面板

十六、依赖注入

十七、配置路由

十八、路由加事件

十九、多页面跳转

二十、angular-ui-router.min使用

二十一、复习前期讲的JS不互通求和

二十二、创建服务

二十三、服务类型

二十四、ajax复习和小型项目

最后做一APP项目做为Angular的收发工作

# Angular介绍

AngularJS 诞生于2009年，由Misko Hevery 等人创建，后为Google所收购。是一款优秀的前端JS框架，已经被用于Google的多款产品当中。AngularJS有着诸多特性，最为核心的是：MVC、模块化、自动化双向数据绑定、语义化标签、依赖注入等等。

AngularJS是一款由Google公司开发维护的前端MVC框架，其克服了[HTML](http://baike.baidu.com/view/692.htm)在构建应用上的诸多不足，从而降低了开发成本提升了开发效率。

## 特点

AngularJS与我们之前学习的jQuery是有一定的区别的，jQuery更准确来说只一个类库（类库指的是一系列函数的集合）以DOM做为驱动（核心），而AngularJS则一个框架（诸多类库的集合）以数据和逻辑做为驱动（核心）。

框架对开发的流程和模式做了约束，开发者遵照约束进行开发，更注重的实际的业务逻辑。

AngularJS有着诸多特性，最为核心的是：模块化、双向数据绑定、语义化标签、依赖注入等。

与之类似的框架还有BackBone、Vue、React等。

## 下载

1. 通过[AngularJS](https://angularjs.org/)官网下载，不过由于国内特殊的国情，需要翻墙才能访问。
2. https://angularjs.org/（官网经常打不开）
3. http://www.angularjs.cn/update（中文社区）

4、通过npm下载，npm install angular

5、通过bower下载，bower install angular

bower是什么？

## MVC

MVC是一种开发模式，由模型（Model）、视图（View）、控制器（Controller）3部分构成，采用这种开发模式为合理组织代码提供了方便、降低了代码间的耦合度、功能结构清晰可见。

模型（Model）一般用来处理数据（读取/设置），一般指操作数据库。

视图（View）一般用来展示数据，比如通过HTML展示。

控制器（Controller）一般用做连接模型和视图的桥梁。



通过ThinkPHP来演示后端MVC的执行流程，其重点在于理解。

MVC更多应用在后端开发程序里，后被引入到前端开发中，由于受到前端技术的限制便有了一些细节的调整，进而出现了很多MVC的衍生版（子集）如MVVM、MVW、MVP、MV\*等。

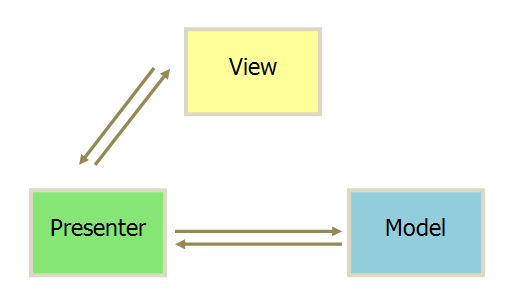
## MVP

MVP 模式将 Controller 改名为 Presenter，同时改变了通信方向

1. 各部分之间的通信，都是双向的。

2. View 与 Model 不发生联系，都通过 Presenter 传递。

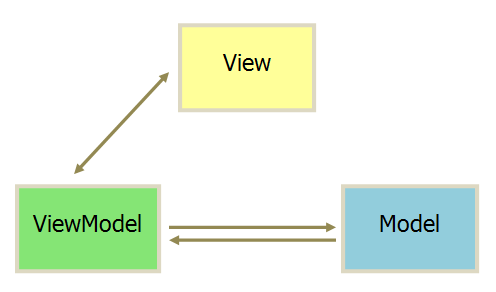
3. View 非常薄，不部署任何业务逻辑，称为"被动视图"（Passive View），即没有任何主动性，而 Presenter非常厚，所有逻辑都部署在那里。



## MVVM

ModelView-ViewModel

MVVM 模式将 Presenter 改名为 ViewModel，基本上与 MVP 模式完全一致。

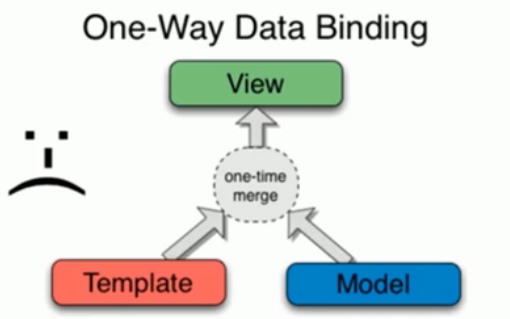


唯一的区别是，它采用双向绑定（data-binding）：View的变动，自动反映在 ViewModel，反之亦然。Angular 和 Vue 都采用这种模式。

## 单向绑定

单向数据绑定是指将模型（Model）数据，按着写好的视图（View）模板生成HTML标签，然后追加到DOM中显示，如之前所学的artTemplate 模板引擎的工作方式。

如下图所示，只能模型（Model）数据向视图（View）传递。



## 双向绑定

双向绑定则可以实现模型（Model）数据和视图（View）模板的双向传递，如下图所示。

