



第一讲 AngularJS概述

前端 JS 框架知多少？



React



Vue.js

MVVM Made Simple



gulp.js

The streaming build system



教学目标

1. AngularJS 概述
2. AngularJS 的使用
3. AngularJS 核心特性

1

AngularJS概述

1

AngularJS简介

AngularJS是什么？

—— 是一款优秀的前端JS框架。

诞生于2009年，由Misko Hevery 等人创建，**AngularJS是为了克服HTML在构建应用上的不足而设计的。**后为Google所收购。已经被用于Google的多款产品当中。



①

AngularJS简介

解决静态网页技术在构建动态应用上的不足

类库 - 类库是一些函数的集合，它能帮助你写WEB应用。起主导作用的是你的代码，由你来决定何时使用类库。类库有：jQuery等

框架 - 框架是一种特殊的、已经实现了的WEB应用，你只需要对它填充具体的业务逻辑。这里框架是起主导作用的，由它来根据具体的应用逻辑来调用你的代码。框架有：knockout、sproutcore等

①

AngularJS简介

AngularJS 是一个 **JavaScript 框架**。

- ◆ 是一个以 JavaScript 编写的库
- ◆ 是以一个 JavaScript 文件形式发布的
- ◆ 可通过 script 标签添加到网页中

②

AngularJS的优点

1. AngularJS是一个单页面应用（ SPA : Single Page Applications ）
框架。
2. 代码更少更简洁，大部分代码在业务逻辑或核心功能上。
3. 易于编写和理解应用。
4. 通过CSS和HTML来编写静态页，应用模板完全是HTML
编写，方便页面设计师参与。
5. 单元测试非常简单。



AngularJS的优势在哪？

解耦
视图、逻辑解耦，具有更大的灵活性。

功能完整
提供完整的解决方案。内置丰富的服务。

模块化
引入依赖注入技术管理各模块。

组件化
通过指令方式可以灵活封装html组件。



利于测试
因为模块化管理，可以对模块进行块单元测试。

③

AngularJS理念

Angular的理念是：组建视图(UI) 又要写软件逻辑时，声明式的代码会比命令式的代码好得多，尽管命令式的代码非常适合用来表述业务逻辑。

- ◆ 将DOM操作和应用逻辑**解耦**，大大改善代码的可调性。
- ◆ 将测试和开发同等看待，测试的难度在很大程度上取决于代码的结构。
- ◆ 将客户端和服务端解耦，使两边并行开发，并且使两边代码都能实现重用。

耦合是指两个或两个以上的体系或两种运动形式间通过相互作用而彼此影响以至联合起来的现象。**解耦**就是用数学方法将两种运动分离开来处理问题。



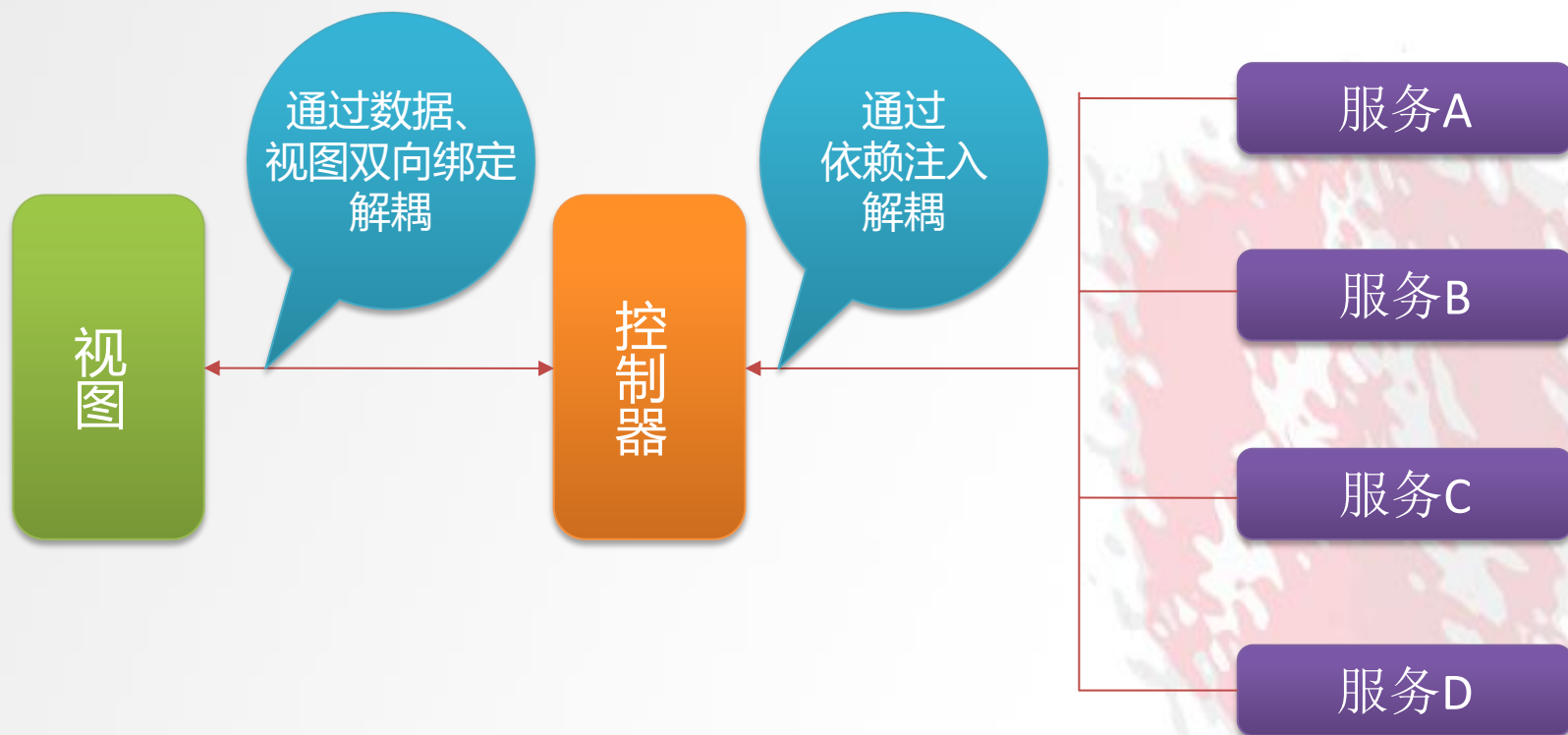
AngularJs的核心思想

- 将视图与业务逻辑解耦。在AngularJS中通过数据视图双向绑定实现。这将提高代码的可测试性。





- 遵循MVC模式开发，鼓励视图、数据、逻辑组件间**松耦合**。



2

AngularJS的使用

<https://angularjs.org/>



The screenshot shows the AngularJS homepage. At the top is a dark navigation bar with the AngularJS logo, links for Home, Learn, Develop, and Discuss, and a search bar. The main content area features the large AngularJS logo with 'by Google' underneath. Below the logo is the tagline 'HTML enhanced for web apps!'. There are two blue buttons: 'Download AngularJS 1' with a download icon and the Angular logo, and 'Try the new Angular 2' with the Angular logo. At the bottom are two buttons: 'View on GitHub' with the GitHub icon and 'Design Docs & Notes' with a folder icon.

ANGULARJS

Home Learn Develop Discuss Search

ANGULARJS
by Google

HTML enhanced for web apps!

Download AngularJS 1

Try the new Angular 2

View on GitHub

Design Docs & Notes

1

AngularJS的使用

AngularJS的使用

```
<!doctype html>  
<html ng-app>  
<body>  
<script src="angular.js"><script>  
</body>  
</html>
```

demo:Listing 01.html



HTML中的AngularJS

demo:Listing 02.html

```
<html ng-app="exampleApp" >
<body ng-controller= "ctrl1">
    <input type= "text" ng-model= "user">
        {{user}}
</body>
</html>
```

- **ng-app** : AngularJS程序入口，对该标签内的元素进行初始化。
- **ng-controller** : 在当前元素范围内绑定指定的控制器(controller)。
- **ng-model** : 指定当前元素与数据模型（\$scope）中的属性绑定，如果数据模型中没有此属性，会自动新建一个。

3

AngularJS核心特性

序

AngularJS 核心特征

- ◆ MVC
- ◆ 模块化
- ◆ 指令系统
- ◆ 双向数据绑定
- ◆ 依赖注入

①

MVC

MVC核心：**Model(模型)**，**View(UI)**，**Controller(控制器)**

Model(模型)：数据模型层（例：数据库的表模型）

View(UI)：视图层，Web页面中就是HTML

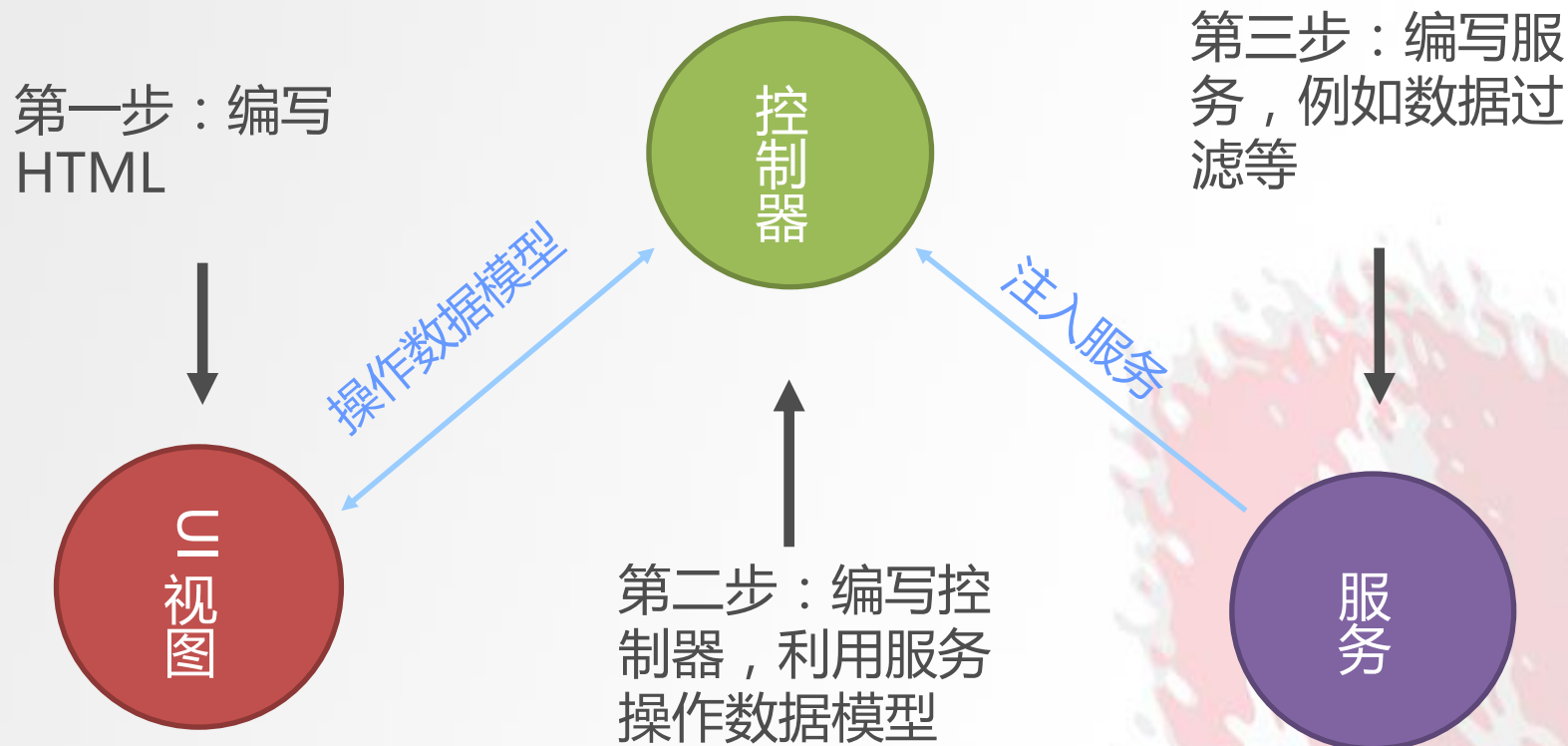
Controller(控制器)：业务逻辑和控制逻辑，处理/加工Model

它们的工作模型应该是：Controller=>Model=>View





来看看使用AngularJs怎么做



几乎没有DOM操作，更专注于业务逻辑！

AngularJS程序示例

```
<html><head>
<title>第一个AngularJS程序</title>
<script src="angular.js"></script>
<script>
```

模型

```
function HelloController($scope) {
    $scope.helloTo = { title: "AngularJS" };
}
```

```
</script>
```

```
</head>
```

控制器

```
<body >
```

视图

```
<div ng-controller="HelloController" >
  <h2>你好 ! 第一个{{helloTo.title}}程序示例</h2>
</div>
```

```
</body></html>
```

demo:Listing 03.html



当网页加载完毕，AngularJS 自动开启。

- div中定义的ng-controller就是一个控制器，这个控制器被定义为一个函数
- h2标签的内容很明显就是显示层，函数中定义了一个helloTo对象的title属性，而在显示层中通过{{}}获取它的值。
- function HelloCtroller(){}很明显为数据模型层。

页面显示的内容就是数据模型的值

②

模块化

在开发过程中应该尽量避免定义全局变量和全局函数。
因此我们就需要了解angularJS的模块化特性了

一切都是从模块开始的，我们在使用angularJS时，
一定要先想到模块，所有的东西都是建立在模块的基础上，
只有定义了模块才能调用其它方法。

AngularJS程序示例

```
<html><head>
<title>第一个AngularJS程序</title>
<script src="angular.js"></script>
<script>
```

模型

```
angular.module("myapp", [])
.controller("HelloController", function($scope) {
    $scope.helloTo = {title:"AngularJS"};
});
```

控制器

```
</script>
```

```
</head>
```

```
<body ng-app="myapp" >
```

视图

```
<div ng-controller="HelloController" >
```

```
<h2>你好 ! 第一个{{helloTo.title}}程序示例</h2>
```

```
</div>
```

```
</body></html>
```




当网页加载完毕，AngularJS 自动开启。

- **ng-app** : 定义一个 AngularJS 应用程序。AngularJS 程序入口，对该标签内的元素进行初始化。
- **ng-controller** : 在当前元素范围内绑定指定的控制器。
- **ng-model** : 把元素值绑定到应用程序。指定当前元素与数据模型（\$scope）中的属性绑定，如果数据模型中没有此属性，会自动新建一个。

3

依赖注入

AngularJS拥有内建的依赖注入子系统，可以帮助开发人员更容易的开发，理解和测试应用。

依赖注入是一种软件设计模式，该模式分离了客户端依赖本身行为的创建，这使得程序设计变得松耦合，并遵循了依赖反转和单一职责原则。与服务定位器模式形成直接对比的是，它允许客户端了解客户端如何使用该系统找到依赖

4

指令系统

允许自定义指令。

```
<html><head>
<title>第一个AngularJS程序</title>
<script src="angular.js"></script>
<script>
angular.module("myapp", [])
.directive("hello", function() {
    return {
        restrict: 'E',
        template: '<div>Hi everyone!</div>',
        replace: true
    }
});
</script>
</head>
<body ng-app="myapp">
    <hello></hello>
</body>
</html>
```

demo:Listing 05.html

5

双向数据绑定

大部分框架实现的都是单向数据绑定，单向数据绑定的意思是将数据绑定到模板上，并且显示到界面上，这种模式的缺点是显示出来后，当数据更改时，并不能及时的更新到页面上，因此angular提出了双向数据绑定的定义。

```
<div>  
  <input type="text" ng-model="user"/>  
  <p>{{user}}, AngularJS!</p>  
</div>
```

本课小结

1. AngularJS 概述
2. AngularJS 模型
3. AngularJS 应用

TNAKS

主讲：王智娟

QQ: 24132228

Email: wangzhijuan@onest.net