AngularJS 简介（Introduction）

AngularJS是一个开发动态Web应用的框架。它让你可以使用HTML作为模板语言并且可以通过扩展的HTML语法来使应用组件更加清晰和简洁。它的创新之处在于，通过数据绑定和依赖注入减少了大量代码，而这些都在浏览器端通过JavaScript实现，能够和任何服务器端技术完美结合。

Angular是为了扩展HTML在构建应用时本应具备的能力而设计的。对于静态文档，HTML是一门很好的声明式的语言，但对于构建动态WEB应用，它无能为力。所以，构建动态WEB应用往往需要一些技巧才能让浏览器配合我们的工作。

通常，我们通过以下手段来解决动态应用和静态文档之间不匹配的问题：

- **类库** - 一些在开发WEB应用时非常有用的函数的集合。你的代码起主导作用，并且决定何时调用类库的方法。例如： `jQuery` 等。
- **框架** - 一种WEB应用的特殊实现，你的代码只需要填充一些具体信息。框架起主导作用，并且决定何时调用你的代码。例如： `knockout` , `ember` 等。

Angular另辟蹊径，它尝试去扩展HTML的结构来弥合以文档为中心的HTML与实际Web应用所需要的HTML之间的鸿沟。Angular通过指令（directive）扩展HTML的语法。例如：

- 通过 `{{}}` 进行数据绑定。
- 使用DOM控制结构来进行迭代或隐藏DOM片段。
- 支持表单和表单验证。
- 将逻辑代码关联到DOM元素上。
- 将一组HTML做成可重用的组件。

一个完整的前端解决方案

在构建WEB应用的前端时，Angular提供的不是一个部分解决方案，而是一个完整的解决方案。它能够处理所有你写过的混杂了DOM和AJAX的代码，并能够将它们组织的结构良好。这使得Angular在决定应该怎样构建一个CRUD应用时显得甚至有些“偏执（opinionated）”，但是尽管它“偏执”，它也尝试确保使用它构建的应用能够灵活的适应变化。下面是Angular的一些出众之处：

- 构建一个CRUD应用时可能用到的所有技术：数据绑定、基本模板指令、表单验证、路由、深度链接、组件重用、依赖注入。
- 可测试性：单元测试、端到端测试、模拟对象（mocks）、测试工具。
- 拥有一定目录结构和测试脚本的种子应用。

Angular的可爱之处

Angular通过给开发者呈现更高层次的抽象来简化应用的开发。和其他的抽象一样，它也以损失灵活性为代价。换句话说，Angular并不是适合任何应用的开发，Angular考虑的是构建CRUD应用。幸运的是，绝大多数WEB应用都是CRUD应用。为了理解Angular适用哪些场合，知道它不适合哪些场合是很有帮助的。

对于像游戏和有图形界面的编辑器之类的应用，会进行频繁且复杂的DOM操作，和CRUD应用不同。因此，可能不适合用Angular来构建。在这种场景下，使用更低抽象层次的类库可能会更好，例如： `jQuery` 。

Angular之道

Angular是建立在这样的信念之上的：即声明式的代码用在构建用户界面和组装软件组件时更好，而命令式的代码更擅长展现业务逻辑。

- 将应用逻辑与DOM操作解耦，会大大提高代码的可测试性。
- 平等看待应用的测试和开发，测试的难度很大程度上取决于代码的结构。
- 将前端与服务器端解耦，这样使得前端的开发和服务器端的开发可以齐头并进，实现两边代码的重用。
- 框架在整个应用的开发流程中指导开发者：从用户界面设计到实现业务逻辑，再到测试。
- 化繁为简，化整为零总是好的。

Angular将把你从下面的苦海中解脱出来：

- **使用回调：**回调会降低代码的可读性，是代码变得零散。移除像回调之类的常见代码是件好事，大幅移除因为JavaScript这门语言的不足而使你不得不写的代码，从而让应用显得更加清晰。
- **以编程的方式操作HTML DOM：**操作HTML DOM是AJAX应用中很基础的一部分，但它不灵活并且容易出错。通过声明式的语句，描述UI该怎样随着状态的改变而变化，能让你从低级的DOM操作中解脱出来。绝大多数Angular的应用开发中，开发者都不需要自己去写低级的操作DOM的代码，尽管如果你非要这样的话，也是可以的。
- **在用户界面中读写数据：**AJAX应用中的绝大多数操作都是CRUD操作。一个典型的流程是从服务器端取到数据组装成内部对象，然后写到HTML的表单中，在用户对表单进行修改之后，进行表单验证，显示表单验证错误信息，然后将数据重新组装成内部对象，再发给服务器。在这个流程中有很多重复的代码要写，而Angular消除了在这个流程中几乎所有的重复代码，使得代码看起来只是在描述所有的执行流程，而不是所有的实现细节。
- **在开始前写大量的初始化代码：**一般需要写很多的基础性的代码才能完成一个基本的AJAX的Hello World应用。在Angular的应用中，你可以通过一些服务来初始化应用，这些服务都是以类似于Guice (http://code.google.com/p/google-guice/)的方式进行依赖注入的。这会让你很快进入功能开发。另外，你还能完全控制自动化测试的初始化过程。