介绍：

2016年9月14号Angular宣布发布了Angular2的版本.

并且angular2提供的angular-cli来帮助我们快速搭建angular2应用环境。

但是angular-cli目前一直处于测试阶段，版本迭代比较快，陆陆续续发布了各种迭代的alpha和beta版本。并最终使用了@angula/cli来代替原来的angular-cli。当然angular2的内部功能也在不断地变化，这也导致了目前网络上angular2的教程鱼龙混杂，语法不一。很坑爹！！！有木有！！！

目前本人使用的是@angular/cli的最新版：@angular/cli@1.0.0-rc.4

# Angular2介绍

首先，Angular2跟Angular1.x版本没有多大关系，虽然2沿用了版本1的一些概念，像directive, model, service, router, dependency injection等， 但是很多用法和原理都已经不是那么一回事了。Angular2有如下一些特性：

## 模块化

* 在Angular2开发中，所有的系统功能都是模块化的，你只需要导入需要的部分。用Angular2开发的应用也是模块化的，一个模块、一个功能、甚至页面上的一个标签都是组件。
* 模块化带来的最大的好处就是按需加载，这就大大提高了APP的启动时的性能，也使得需要下载的代码比较小，因为不需要的模块可以不用导入。按照Angular官方的说法，使用Angular2的hello world的实例创建的App，在做完代码压缩合并以后,下载的代码在50K以内。
* 模块化也给开发带来方便，使得协作开发、代码合并等都非常方便。

## 使用TypeScript

* TypeScript是Javascript的超集，TypeScript的代码会被编译成Javascript的代码再执行。所以，如果有任何的语法或其他错误，就可以在编译的时候及时发现。而这个编译的过程可以通过工具自动执行，这样就可以在你写完某一段代码的时候就发现其中的错误。
* TypeScript的静态类型的，所以，我们就可以通过一些工具来实现像JAVA这样的强类型语言才会有自动补全、重构、代码查找等功能。虽然说像Webstorm这样的工具对于javascript也能提供自动代码补全之类的功能，但是他是通过检索代码里面的所有关键字，然后匹配关键字去补全，而不是通过语言自身的特性。
* TypeScript具有面向对象语言的一些特性，如抽象和继承、命名空间、包等。以抽象和继承为例，虽然用js也可以实现类和类的继承关系，但是使用TypeScript我们可以直接使用interface, implements这样的语法来实现，这使得代码结构更清晰、更易读。

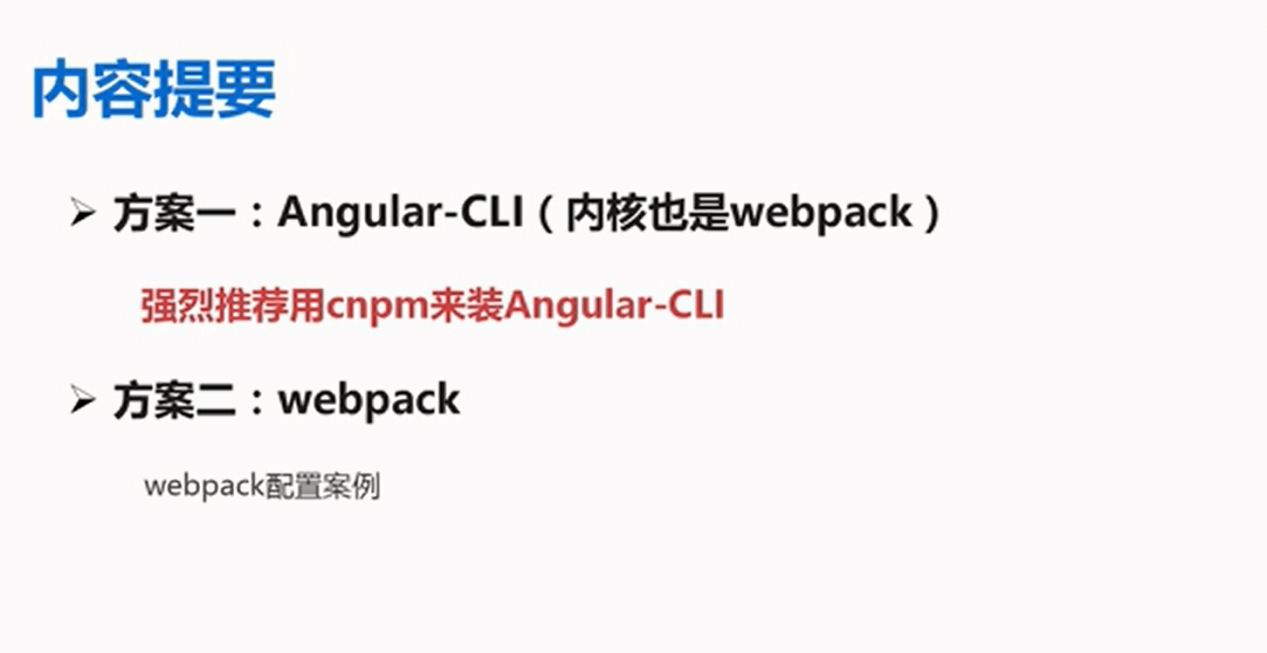
## 更容易

* 好吧，这只是官方的说法，要说容易也要看从什么方面来说。如果学过Angular1，你们可能知道，1非常容易上手，有一点js基础的人，几天就能学会并开始开发。如果再有一点MVC的背景或知识，就更容易了。但是2就不一样，一开始学习成本比较高，首先是因为TypeScript的语法，模块化的开发，对于没有多少开发经验的人来说，理解和上手都有一定困难。但是一旦上手以后，之后的开发和学习就比较容易了。不像Angular1，一开始的学习曲线很平缓，很容易上手，但是真正开始开发一个比较复杂的应用的时候，就会发现有很多问题，需要理解他的背后的原理，像双向绑定、模板、路由等。这时候的学习曲线就会变得很陡。
* Angular2是一个完整的前端开发框架，利用Angular2提供的功能和模块，你可以开发一个完整的应用。但是如果你用React，你还需要整合很多其他的库来实现某些方面的功能，这样就会面临很多选择，Redux, Flux, React-router, axios…
* 前面提到Angular2使用TypeScript，这也给开发带来很多便利，使得代码更易读、容易维护。

更多介绍：

<http://codin.im/2016/09/15/angular2-tutorial/>

安装方案：



方案1：

**安装@angular/cli@latest**

**安装之前：**

确保您的**Node版本 >= 6.9.0**  && **npm 版本更新到最新** && **VPN**。

**安装：**

1, 卸载之前的angular-cli

npm remove -g angular-cli

2, 卸载之前的angular/cli

npm remove -g @angular/cli

3, 清理npm 缓存

npm cache clean

4，安装最新版的angular/cli (最好在搭好梯子的情况下)

npm install -g @angular/cli@latest

安装完成可以使用ng -v 查看版本。

5，创建ng项目（可能比较慢，建议备份一份）

ng new <myProjectName>

6, cd到项目，启动ng项目(在http://localhost:4200下访问)

ng server

有两点需要注意：

==================================

第一点：

有的教程推荐大家使用cnpm安装，本人试过不太好使，可能是我打开的方式不对。。。

1，安装国内的cnpm.(i 就是install的意思)

npm i -g cnpm

2，使用cnpm安装

cnpm i -g @angular/cli@latest

可以试试，但是不推荐。

-----------------------------------------------------------

第二点：

推荐大家使用nrm 来管理npm的镜像。

安装:npm install -g nrm

使用：

1,列出所有可选的源

命令：nrm ls

\* npm ---- https://registry.npmjs.org/

cnpm --- http://r.cnpmjs.org/

taobao - http://registry.npm.taobao.org/

eu ----- http://registry.npmjs.eu/

au ----- http://registry.npmjs.org.au/

sl ----- http://npm.strongloop.com/

nj ----- <https://registry.nodejitsu.com/>

会在当前源前边显示\*号。

2，切换到指定的源

命令： nrm use <registry>

更多查看nrm文档。。。

==================================

一些常用的指令操作：

方案2：

下载webpack项目:

<https://github.com/kunl/Angular-webpack-AOT>

https://github.com/Jean-Xu/angular2-quickstart

引用webpack.config.js和package.json文件(注意修改配置信息)。其实只需要package.json即可。

现在我们来逐个分析一下这些可以导航的方式：

1. <a routerLink='/dashboard/home' routerLinkActive='active'>Home1</a>  
   所有以斜线开头的都必须使用绝对路径来进行导航。也就是说我们会导航到www.app.com/dashboard/home 这个路径，这个路径是绝对的。一般我们不会使用这种方式。如果不使用绝对路径，一定谨记不要以斜线开头。
2. <a routerLink='/dashboard/home' [routerLinkActive]="['active']">Home2</a>  
   可能咋一看这个和第一条是一样的，但是请我们看routerLinkActive，第一条是单独的一个属性，而第二条作为一个属性指令来使用。还有一个重点，如果我们使用属性指令，在给属性指令赋值的时候，**必须在中括号内在加入一个单引号**。这是为什么？因为如果我们不加这个单引号或双引号，那么angular会认为这是一个变量，会当做变量来使用。
3. <a routerLink='../home' routerLinkActive="active">Home 2.2 </a>  
   可以看到这个routerLink使用的是相对路径，可以参考第七条，相对路径，是相对的哪个路径，其实就是当前路由。以当前路由作为起点进行相对导航。举个简单的例子，我们当前的路径为/dashboard/chart,现在我们要导航到home(背景：chart和home都是dashboard下的一个组件，属于平级)，那么我们就要使用两个点号..来表示。一个点号表示相对当前路径，两个点号就是相对父路径。
4. <a [routerLink]="['/dashboard/home']" [routerLinkActive]="['active']">Home3</a>  
   通过指令方式来进行导航，可以参考第一条。绝对路径导航
5. <a [routerLink]="['../home']" [routerLinkActive]="['active']">Home4</a>  
   通过指令方式进行导航，可以参考第一条和第三条，采用的routerLink指令和相对路径方式。
6. <a [routerLink]="['./home']" [routerLinkActive]="['active']">Home5</a>  
   和第五条相同
7. <a [routerLinkActive]="['active']" (click)="onSelect()">Home6</a>  
   通过angular脚本方式进行导航。

this.router.navigate(['../home'],{relativeTo:this.route}); 如果要使用相对导航，必须加上relativeTo属性，并且赋值为route，route是ActivatedRoute的值。  
  
如果采用绝对路径，那么不需要添加这个属性。如果以斜线开头，并且定义了这个relativeTo属性会出现什么的？其实还是绝对路径优先，relativeTo没有起作用。