# 项目创建与开发

## 1、创建项目

进入Windows命令行方式，新建项目。例如，“张三”的姓名缩写是“zs”，则键入下面的命令（将zs改为你自己的姓名缩写）：

ionic start zs-course-demos

选择框架和Tabs操作与之前测试环境时相同。

项目创建完毕后，退出命令行方式。

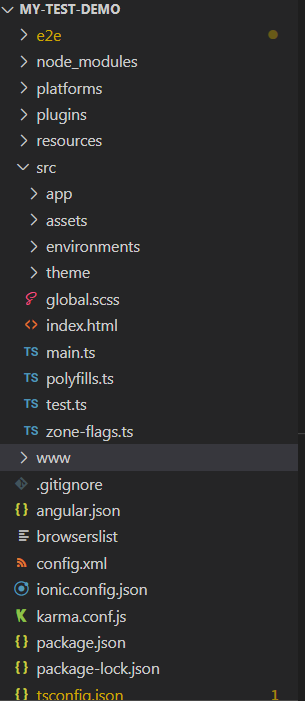
## 2、在VSCode里打开项目

鼠标右击上一步创建的项目文件夹，选择【Open with Code】。例如，如果创建的项目名是zs-course-demos，那么，鼠标右击该文件夹就会在VSCode中打开此项目。

或者先打开VSCode，然后使用File菜单【打开文件夹】找到项目所在的文件夹。

注意：“项目文件夹”是指创建的项目名称，不要进入到项目下的子文件夹内。

例如：项目my-test-demo的组织结构如下所示。核心程序都在src文件夹内。



**e2e**：ionic5.x中端对端测试文件  
**node\_modules** ：ionic5.x中项目所需要的依赖包

**platforms**：生成android或者ios安装包需要的资源---(cordova platform add android后会生成)

**plugins**：插件文件夹，里面放置各种cordova安装的插件  
**resources**：android/ios 资源（更换图标和启动动画）  
**src**：ionic5.x中开发工作目录，页面、样式、脚本和图片都放在这个目录下  
**www**：ionic5.x中静态文件，ionic build --prod 生成的单页面静态资源文件

**angular.json**angular配置文件

**config.xml**: 打包成app的配置文件

**ionic.config.json、ionic.starter.json**：ionic配置文件

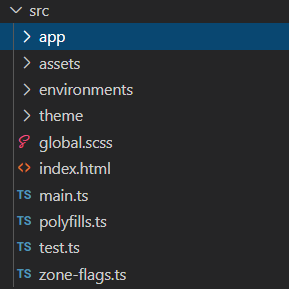
**karma.conf.js**：测试相关的配置文件

**package.json**: 配置项目的元数据和管理项目所需要的依赖

**tsconfig.json**: TypeScript项目的根目录，指定用来编译这个项目的根文件和编译选项

**tslint.json**：格式化和校验typescript

---------Src目录-----------------------------



**app**：ionic5.x中应用根目录 （组件、页面、服务、模块...）

**assets**：ionic5.x中资源目录（静态文件（图片，js框架...）

**theme**：ionic5.x中主题文件，里面有一个scss文件，设置主题信息。

**global.scss**：ionic5.x中全局css文件

**index.html**：ionic5.x中 index入口文件 **main.ts**：ionic5.x中 主入口文件

**polyfills.ts**: 这个文件包含Angular需要的填充，并在应用程序之前加载

**test.ts**：测试相关的配置文件

## 3、在VSCode里面使用终端进行操作

使用VSCode里面，打开查看菜单【终端】打开程序终端。

（1）如果希望观察项目在Android和iOS手机上运行的效果，还需要将Android和iOS平台添加到项目中。

在VSCode的【终端】窗口中依次键入下面的命令：

ionic cordova platform add android

ionic cordova platform add ios

要确保这两条命令分别执行成功。

（2）要观察项目在默认浏览器（IE、Chrome、Firefox等）中的运行效果，只需要在VSCode的【终端】窗口中键入下面的命令：

ionic serve

此时就可以看到该项目运行的效果了。要终止运行，只需要在【终端】窗口中按Ctrl+C，然后关闭浏览器即可。

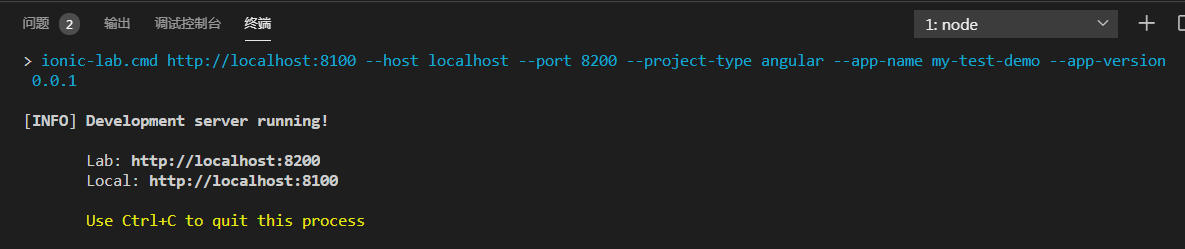
（3）除了在桌面浏览器中运行本项目以外，利用Chrome浏览器提供的仿真器，还可以直接看到该项目在移动设备上的运行效果。

在VSCode的【终端】窗口中键入下面的命令：

ionic serve --lab

此时就可以在Chrome仿真器中

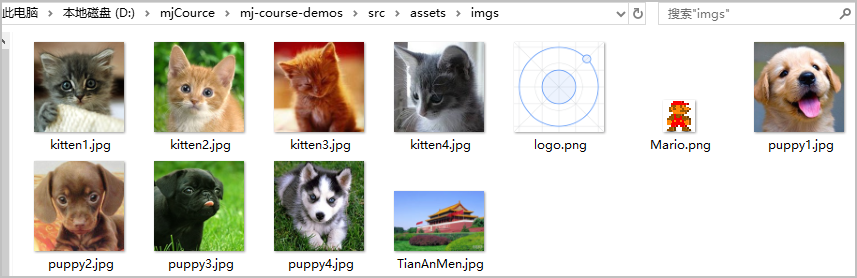
分别观察项目在Android和ios手机上的运行效果了。



## 4、进行项目开发

#### 4.1 添加图片等资源

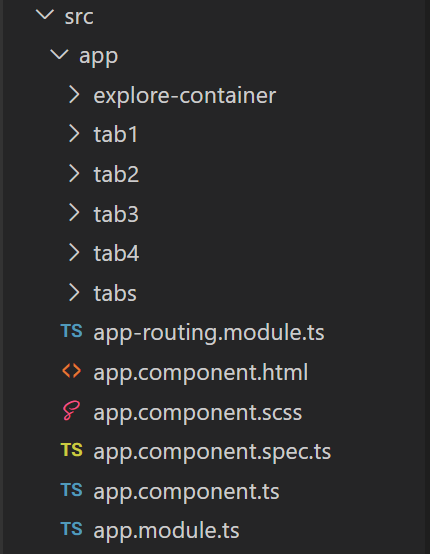
将需要的图片资源复制到项目中的src/assets/imgs文件夹下。使用的图片（一般是jpg或者png文件）可以自己找，或者用其他图片代替。



注意，图片文件、音频文件、视频文件等都应该保存在assets文件夹下。

#### 4.2 观察Src/app下面默认生成的页面并添加tab4

默认情况下，app下面生成了1个tabs文件和tab1，tab2和tab3文件夹。App下面还有一个app-\*开头的一系列文件。每个文件夹都包含很多文件，我们常用到的有：\*-routing.module.ts路由文件，\*.html文件,\*.page.ts脚本文件,\*.module.ts文件。



我们约定:

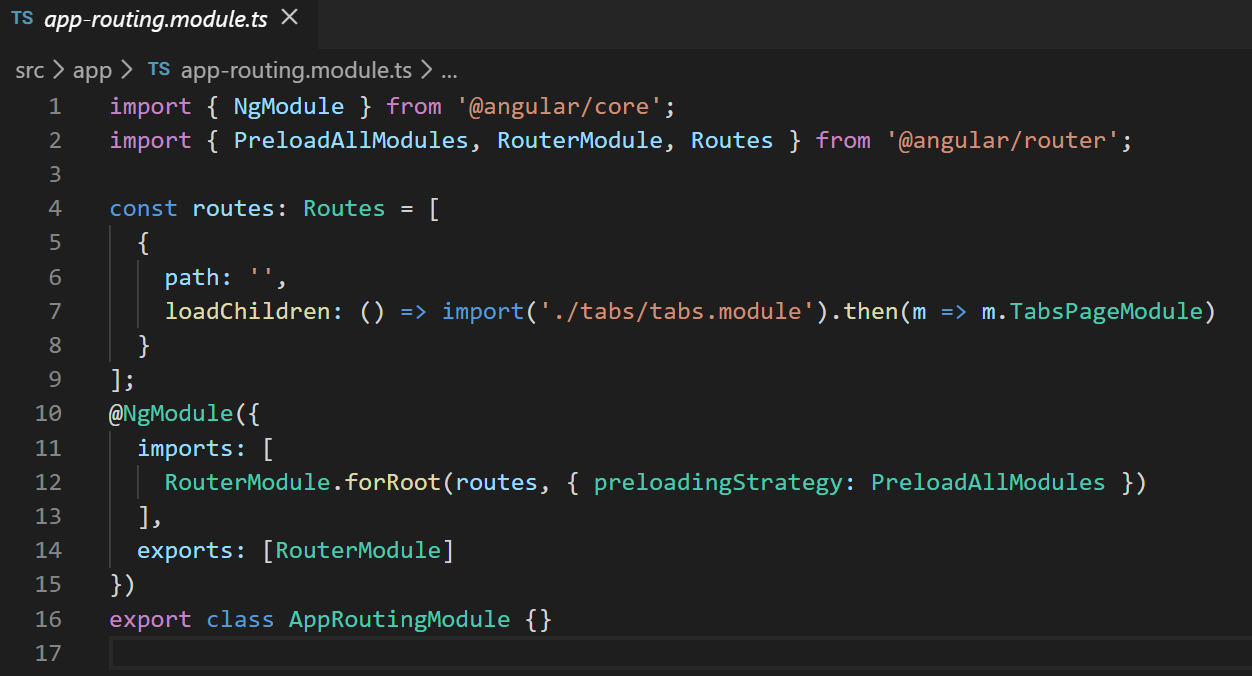
* tab1是Typescript的导航页（即指导书上的a-ts-demos）
* tab2是Angular的导航页（即指导书上的b-ts-demos）
* tab3是Inoic的导航页（即指导书上的c-ts-demos）

这里我们创建一个tab4让其代表我们实例源程序中百度地图应用那个选项(即指导书上的d-ts-demos)。步骤如下：

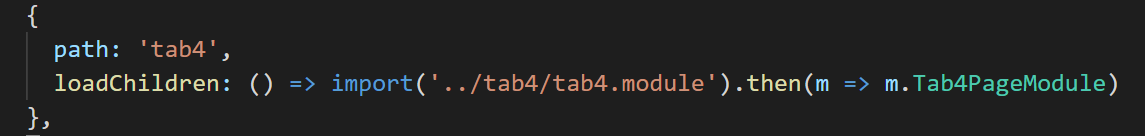
a. 添加页面。在VSCode的【终端】中键入下面的命令：

ionic g page tab4

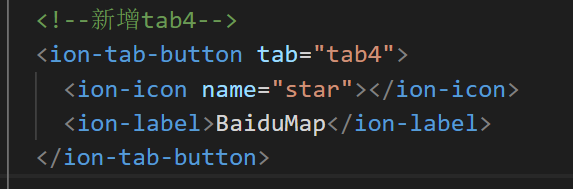
b. 删掉app.routing.module.ts中生成的有关tab4的路由信息。此时app.routing.module.ts文件内容如下：



C. 在tabs-routing.module.ts中Routes里面的children里面添加：



D。在tabs.page.html中的ion-tab-bar标记里面添加：



在终端运行ionic serve命令观察添加之后的效果。

#### 4.3 实现每个模块的导航页

**由于每个模块的导航页使用方法类似，本节以tab1为例进行演示讲解。**

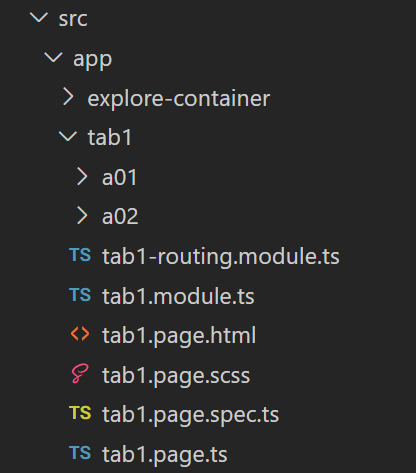
1. **新建TypeScript页面里要导航到的子页面**

此处创建2个页作为代表，在终端运行:

ionic g page a01

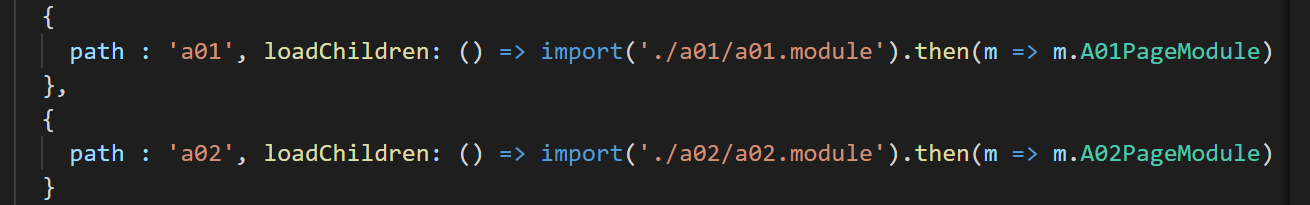
ionic g page a02

然后将创建好的a01和a02拖到tab1文件夹的下面。弄好后的效果如下所示：



**（2）修改tab1-routing.moudle.ts**

在该文件的Routes内增加路由信息：



**（3）修改tab1.page.html**

将其内容改为下面的内容：

**<ion-header [translucent]="true">**

**<ion-toolbar>**

**<ion-title>TypeScript </ion-title>**

**</ion-toolbar>**

**</ion-header>**

**<ion-content [fullscreen]="true" padding>**

**<ion-list>**

**<ion-item>**

**<button (click)="showPage1()">a01-hello world</button>**

**</ion-item>**

**<ion-item>**

**<button (click)="showPage2()">a02-数据类型、数组、元组</button>**

**</ion-item>**

**</ion-list>**

**</ion-content>**

注意：Button的内容在指导书a-ts-demos.ts中MyPage类型pages里面定义的有，此处直接用title就可以了。

**（4）修改tab1.page.ts**

将其改为下面的内容：

**import { Component } from '@angular/core';**

**import { NavController } from '@ionic/angular';**

**@Component({**

**selector: 'app-tab1',**

**templateUrl: 'tab1.page.html',**

**styleUrls: ['tab1.page.scss']**

**})**

**export class Tab1Page {**

**constructor(private nav: NavController) { }**

**showPage1() {**

**this.nav.navigateForward('/tabs/tab1/a01');**

**}**

**showPage2() {**

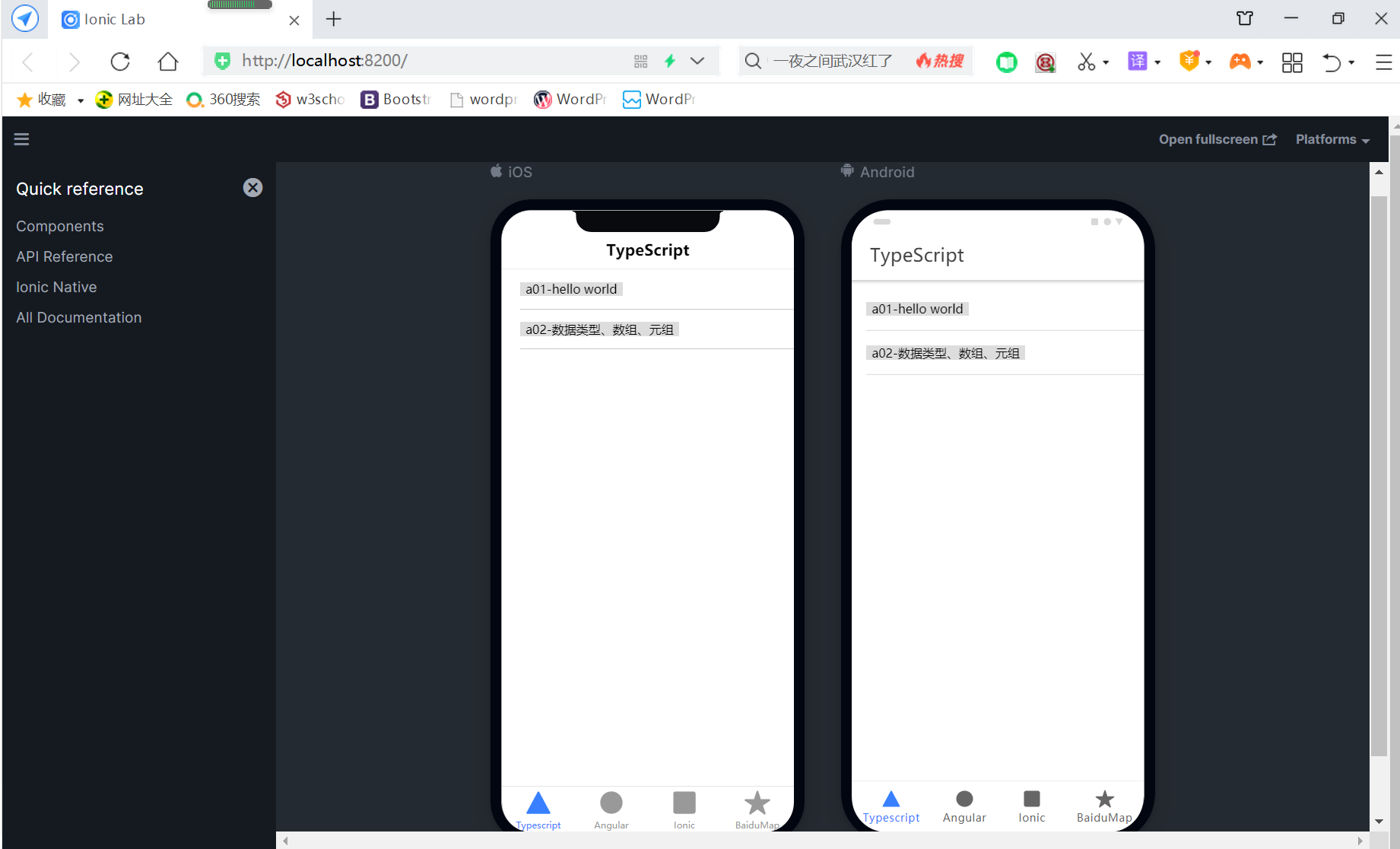
**this.nav.navigateForward('/tabs/tab1/a02');**

**}**

**}**

**（5）运行查看效果**

在终端输入: ionic serve --lab 就可以在模拟器中查看运行之后的效果了。点击每个列表项查看跳转效果。



#### 4.4 实现模块内部功能页面

本节以Typescript模块下面的a01为例进行演示。如果事先a01不存在，需要先使用ionic g page a01进行页面的创建。

1. **修改a01.page.html**

**<ion-header>**

**<ion-toolbar>**

**<ion-buttons slot="start">**

**<ion-back-button defaultHref="tabs/tab1" text="返回"></ion-back-button>**

**</ion-buttons>**

**<ion-title>a01</ion-title>**

**</ion-toolbar>**

**</ion-header>**

**<ion-content padding>**

**<div>**

**<button (click)='click1()'>1-1 Hello World</button>**

**<button (click)='click2()'>1-2 var和let的区别</button>**

**<button (click)='click3()'>1-3 var的问题示例</button>**

**<button (click)='click4()'>1-4 let消除var很多bug</button>**

**</div>**

**<br/>**

**<div id='mydiv'></div>**

**</ion-content>**

1. **在app下新建一个providers文件夹，然后添加一个Helloworld.ts文件。**

**【注意：就是指导书上70页的a01.ts】**

**将其内容改为如下：**

**export interface Person {**

**name: string;**

**welcomeInfo: string;**

**}**

**export class Student {**

**message: string;**

**constructor(**

**public name,**

**public welcomeInfo**

**) {**

**this.message = name + '、n' + welcomeInfo;**

**}**

**}**

**export function greeter(person: Person) {**

**return 'Hello, ' + person.name + ' ' + person.welcomeInfo;**

**}**

1. **修改a01.page.ts文件**

**import { Component, OnInit } from '@angular/core';**

**import { NavController } from '@ionic/angular';**

**import { Person, greeter, Student } from 'src/app/providers/Helloworld';**

**@Component({**

**selector: 'app-a01',**

**templateUrl: './a01.page.html',**

**styleUrls: ['./a01.page.scss'],**

**})**

**export class A01Page implements OnInit {**

**constructor(public navCtrl: NavController) { }**

**myDiv: HTMLDivElement;**

**ionViewDidLoad() {**

**this.myDiv = document.getElementById('mydiv') as HTMLDivElement;**

**}**

**ngOnInit() {**

**this.ionViewDidLoad();**

**}**

**// hello world**

**click1() {**

**var user1: Person = { name: '张三', welcomeInfo: '您好！' };**

**var user2: Student = new Student('李四', '欢迎你！');**

**this.myDiv.innerText = greeter(user1) + '\n' + user2.message;**

**}**

**// let和var的区别**

**click2() {**

**let mydiv = this.myDiv;**

**mydiv.innerText = 'var和let语法相同，其实两者的区别并不是在语法上，而是在语义上！\n';**

**for ( var i = 0; i < 5; i++) {**

**mydiv.innerText += 'a' + i + '\n';**

**var age1 = 23;**

**}**

**// 注意：i和age1都是在循环体内声明的，但是循环结束后仍然可用！**

**mydiv.innerText += 'i=' + i + '，age=' + age1 + '\n\n';**

**// 解决var全局污染的办法很简单，只需要将所有var全部都改为let就行了**

**for (let j = 0; j < 5; j++) {**

**mydiv.innerText += 'b' + j + '\n';**

**}**

**}**

**click3() {**

**this.myDiv.innerText = '\n想一想，为什么这段代码执行的结果和预期的不一致？\n\n';**

**var that = this;**

**for ( var i = 0; i < 10; i++) {**

**// 等待200毫秒，然后开始执行function()函数**

**// tslint:disable-next-line: only-arrow-functions**

**setTimeout( function () {**

**that.myDiv.innerText += i + '\n';**

**}, 200);**

**}**

**}**

**// 前面都是为了说明问题，这个才是正确的写法**

**click4() {**

**this.myDiv.innerText = '';**

**for (let i = 0; i < 10; i++) {**

**setTimeout(() => {**

**this.myDiv.innerText += i + '\n';**

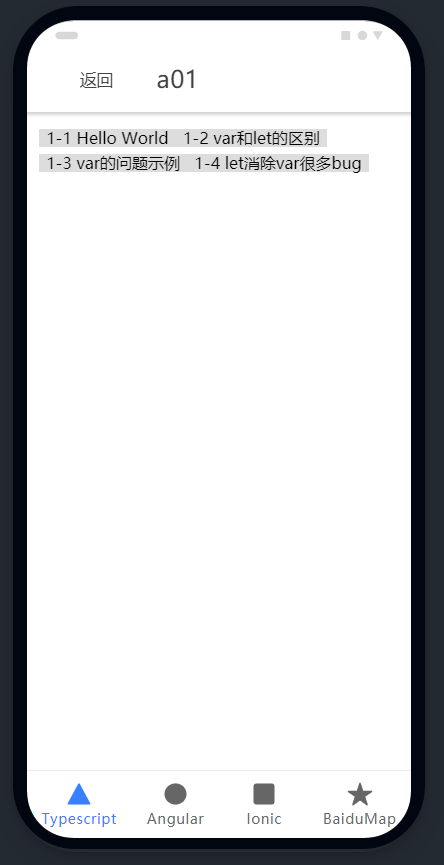
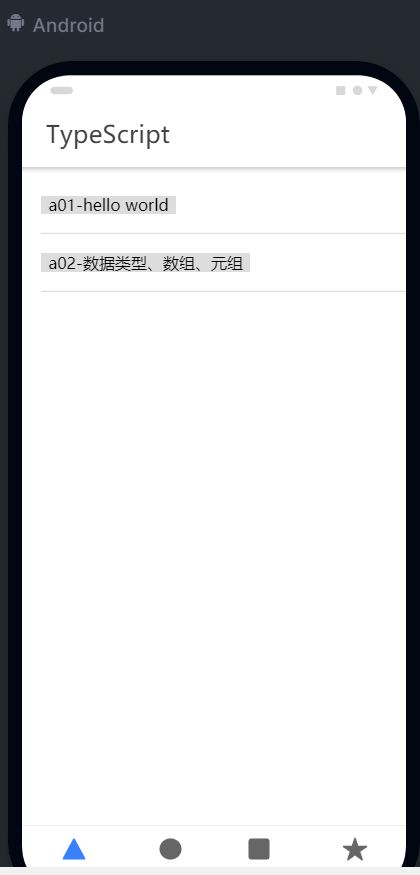
**}, 200);**

**}**

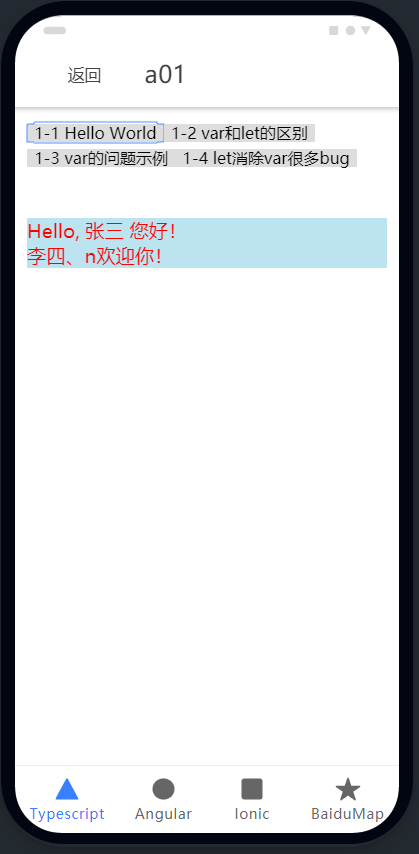
**}**

**}**

1. **运行程序查看效果**



**分别单击不同的例子就可以看到效果，比如点击第一个：**



## 5、实践项目各个模块的具体要求

**基本功能：**

**模块A --TypeScript (Tab1.html)**

**要求实现指导书P69页图3-10中的所有页面上的功能。**

**模块B--Angular（Tab2.html）**

**要求实现指导书P90页图3-20中所有页面上的功能。**

**模块C--Ionic（Tab3.html）**

**要求实现指导书P105页图3-25中的全部功能。**

**模块D--BaiduMap（Tab4.html）**

**要求实现指导书P128页图3-35中的全部功能。**

**扩展功能：可以在此基础上进行界面样式更改或者功能扩展。**