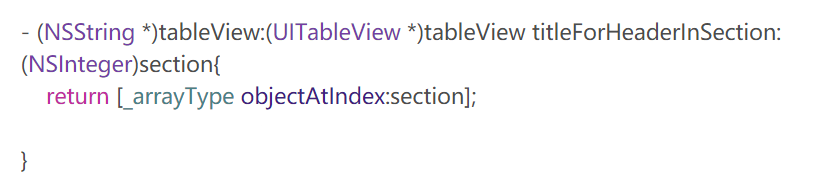
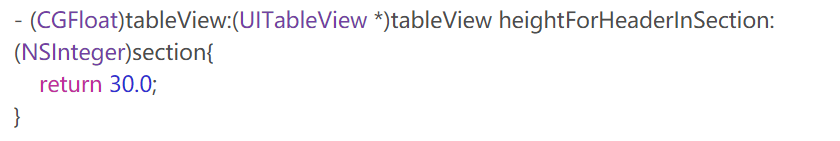
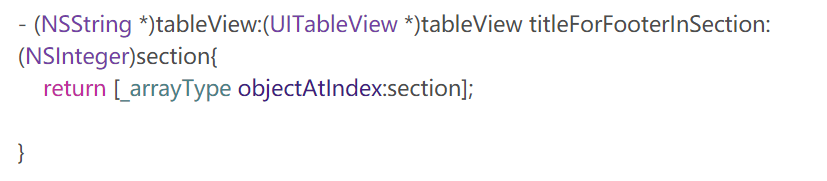
1. 按顺序打印出 App、ViewController 生命周期的各个事件
   1. App的生命周期
      1. 进程启动但还没有进入保存状态：applicationwillFinishLaunchingWithOptions
      2. 启动基本完成，程序准备开始运行：applicationdidFinishLaunchingWithOptions
      3. 程序将要进入非活动状态，在此期间，应用程序不接收事件：applicationWillResignActive
      4. 应用程序进入活动状态：applicationDidBecomeActive
      5. 程序被推送到后台，在后台继续运行：applicationDidEnterBackground
      6. 程序从后台重新调回前台：applicationWillEnterForeground
      7. 程序将要退出，需要保存一些数据并且做一些清理工作：applicationWillTerminate
      8. 程序载入后：applicationDidFinishLaunching
   2. ViewController的生命周期
      1. 首先判断是否由xib或者storyboard创建
      2. 如果是的话，调用awakeFromNib函数，此方法是当xib或者storyboard加载完之后调用，此时nib文件中已经连线好的对象已经完成初始化，frame位置已固定。之后再去调用loadView函数。
      3. 如果不是的话，直接调用loadView函数，这是用于创建Controller的，每次访问controller的view且view为nil，loadView方法就会被调用。（loadView在ViewController的生命周期中一般情况只会被调用一次。）
      4. 在视图控制器将视图层次结构加载到内存之后调用viewDidLoad，无论视图层次结构是从nib文件中加载还是从loadView方法里用代码实现都会执行此方法。如果是在代码中创建的视图加载器，该函数将会在loadView方法后被调用；如果是从nib视图页面输出，他将会在视图设置好后后被调用。
      5. 之后调用viewWillAppear，通知视图控制器其视图即将添加到视图层次结构中。
      6. 调用viewWillLayoutSubViews，通知视图控制器其视图将要布局它的子视图。
      7. 调用viewDidLayoutSubViews，通知视图控制器其视图已将其子视图布局好。
      8. 调用viewDidApper，通知视图控制器其视图已经添加到视图层次结构中。
      9. 调用viewWillDisappear，通知视图控制器其视图将要从视图层次结构中删除。
      10. 调用viewDidDisappear，通知视图控制器其视图已经从视图层次结构中删除。
      11. 当应用收到内存警告时，调用didReceiveMemoryWarning
2. 五种常用的UI控件
   1. UISlider: 滑块控件，可以让用户在一定范围内选择值的大小
   2. UIPageControl:提供了一系列小圆点，它们展示单个页面并让用户导航到后面或者前面的页面。
   3. UISegmentedControl：提供了一行大小相等的按钮。
   4. UIButton：该实例提供了屏幕按钮
   5. UITextField：输入文字的控件。
3. 三个 UITableViewDelegate 声明的方法
   1. 

作用：设置UITableView每个分组的Header的Title

* 1. 

设置UITableView分组Header的高

* 1. 

设置UITableView每个分组的Footer的Title