为了更好的阅读团队成员所写的代码，特约定下面的代码规范，有不合适和欠缺的地方希望积极提出修正补充，或者后期代码实践有一些好的代码心得都可以写在本文档中，希望本文档起到抛砖引玉的作用。

 本代码规范主要参考了下面两篇文章，因此，在阅读本指南之前，请先阅读下面的文档

 [Apple’s Cocoa Coding Guidelines](http://developer.apple.com/documentation/Cocoa/Conceptual/CodingGuidelines/index.html)

 [Google’s Open Source C++ Style Guide](http://codinn.com/projects/google-cpp-styleguide/)

1，空格 vs制表符 ：只使用空格，且一次缩进两个空格，我们使用空格缩进。不要在代码中使用制表符。你应该将编辑器设置成自动将制表符替换成空格。

2，尽量让你的代码行宽保持在 80 列之内。

3，方法声明和定义

 - / +和返回类型之间须使用一个空格，星号和参数类型之间也必须有一个空格，参数列表中只有参数之间可以有空格。至于方法体的大括号是另起一行还是加在最后参数列表的后面，建议加在最后参数列表的后面。

方法应该像这样：



如果一行有非常多的参数，更好的方式是将每个参数单独拆成一行。如果使用多行，将每个参数前的冒号对齐。



当第一个关键字比其它的短时，保证下一行至少有 4 个空格的缩进。这样可以使关键字垂直对齐，而不是使用冒号对齐：



4，方法调用

 方法调用应尽量保持与方法声明的格式一致。当格式的风格有多种选择时，新的代码要与已有代码保持一致。

5，命名

总体感觉，命名是件非常头疼的事情

Objective-C 使用驼峰命名法

类名首字母大写，方法首字母小写，方法中的参数首字母小写，同时尽量让方法的命名读起来像一句话，能够传达出方法的意思。

6，注释

虽然写起来很痛苦，但注释是保证代码可读性的关键。下面的规则给出了你应该什么时候、在哪进行注释。记住：尽管注释很重要，但最好的代码应该自成文档。与其给类型及变量起一个晦涩难懂的名字，再为它写注释，不如直接起一个有意义的名字。

建议使用“#pragma mark”，方便阅读代码

项目框架

原则：尽量使用MVC模型，模块之间松耦合，进来新出现MVVM模型,新增ViewModel层，用于将Viewcontroller层的业务逻辑和业务逻辑等剥离出来放到ViewModel层，减轻viewController层的负担。同时工程目录最好有相应的物理文件。

1. 工程目录文件夹首字母大写。
2. 工程目录下面一般有下面几个目录，Models,Views,Controllers,Resources分别用于放置实体对象，视图模型，控制器，资源文件（现在图片管理一般使用imageAsset,该文件夹可以存放音视频资源以及一些不能放在imageAsset中的图片，另外建议该文件下的目录最好根据模块再分类，还有图片的命名也需要统一下：建议图片命名采用下面的规则：模块名\_XXX\_XXX,图片名称全部小写，命名最好可以体现该图片的用途）