代码注释提高程序可读性

1. 概述：在团队开发过程中，难免会有这种抱怨：某某某的代码，看晕我了，一个页面代码那么多，改个东西比登天还难。不知道各位有没有思考过，这是为什么呢？有什么办法能够提高程序可读性呢？当然，代码的编写规范是很重要的，但这里我要介绍的是另一个很重要的事，做好了就能大大提高程序的可读性，那就是——注释

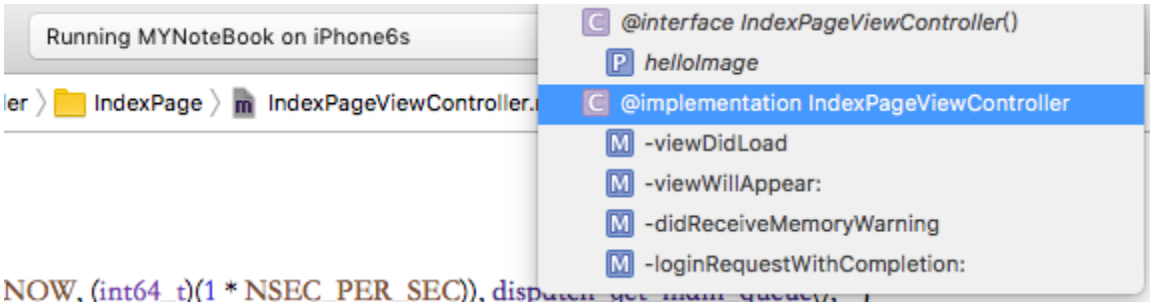
2. 先讲下关于方法的注释

   举个例子，比如你写了一个登录请求的方法，然后你注释，一想到注释，你可能立马会想到，在方法前一行敲个//登录请求，嗯，它就长这样：

```
//发送登录请求
- (void)loginRequestWithCompletion:(void(^)(BOOL success))completion{

}
```

   看着，很好，好像清晰明了？但是，是否考虑过，当.m 里有好几十个方法的时候，怎么快速定位指定的方法？通过点击 xcode 顶部导航的.m 文件，可以看到这个实现文件中所有的方法，如下：



   找到了我们刚写的方法，就在最后一项。那么，有什么办法能够让这里显示的非常直观呢？当然有，我们的目标就是让这里列出的方法模块化、方法名可视化！

   (1) 首先，我们使用#pragma mark - 模块名来进行模块功能注释，比如：

```
#pragma mark - 请求模块

//发送登录请求
- (void)loginRequestWithCompletion:(void(^)(BOOL success))completion{

}
```

   我们再看看现在这个类的方法显示，如下：

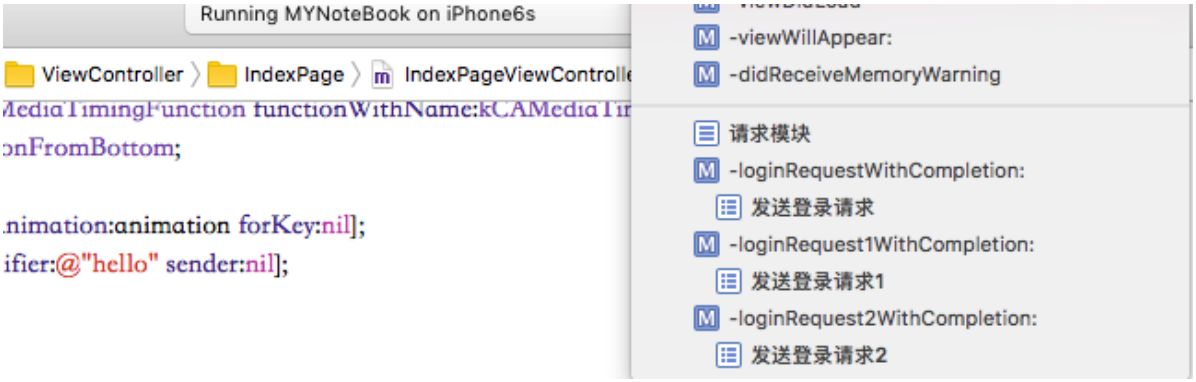
(2) ok，我们看到了请求模块，还有个分割线是吧，没错，这个分割线往下就是我们的请求模块了。现在还有个问题，方法的注释还是没有出来，接下来，我们就来让它显示出来，这里我建议使用 TODO: 方法名来注释，我把它定义为，这个方法将做什么事情。写法如下：

```
#pragma mark - 请求模块

- (void)loginRequestWithCompletion:(void(^)(BOOL success))completion{
  //TODO: 发送登录请求

}
```

这时候，这个类方法概览如下：



看吧，是不是很清晰了，哪个模块，下面哪个方法，是不是都可以快速定位到了？

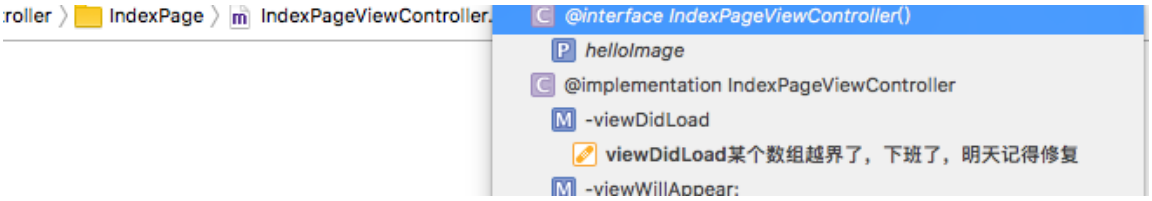
(3) 注释的方法有很多，大家都可以自行百度，这里我再介绍一个很不错的注释：/*FIXME: 说明*/

这个注释意思是，这里需要修复，我们可以用来记录想改但未改的 bug 或做想做却暂时没时间做的事，如下：

```
- (void)viewDidLoad {
  [super viewDidLoad];

  /* FIXME: viewDidLoad某个数组越界了，下班了，明天记得修复*/
  // Do any additional setup after loading the view.
}
```

这时候我们，可以看到一个橙色的标记，创口贴，显示在方法预览界面，一眼就能看到，进行代码修复或完成未完成的工作



(4) 我们来看一下常规的方法注释怎么写，比如你封装了一个类，这个类实现了某个功能，就拿页面加载动画来说把，假如，你写了一个关于 controller 的分类，拓展了加载动画功能，有两个方法供外界调用:开始加载动画、结束加载动画, 如下所示

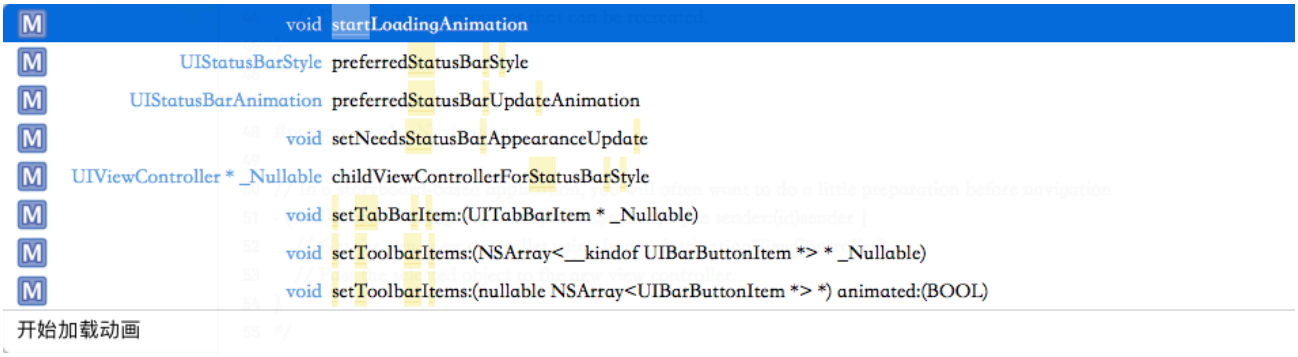
```
- (void)startLoadingAnimation;
- (void)endLoadingAnimation;
```

那 ok，封装好了，是不是要告诉别人怎么用？那我们在这个文件中写下注释呗:

```
/**
  开始加载动画
 */
- (void)startLoadingAnimation;

/**
  结束加载动画
 */
- (void)endLoadingAnimation;
```

这些很明了了把，这种注释的好处是，外界调用的时候，可以显示这个方法的意思，我们来看看



那么，问题来了，以上注释，你是用手一个一个字符敲出来的吗？如果是，那……好吧，介绍个快捷键，option+command+/试试

```
/**
 Description
 */
- (void)startLoadingAnimation;
```

直接帮我们写好了，而且，这个功能是很强大的，以下就它的使用举几个例子，用法都是在方法前按下快捷键就好了。比如在这种方法前：

```
/**
 Description

 @param duration duration description
 */
- (void)startLoadingAnimation:(BOOL)duration;
```

多了个 param 行，这个是让你填输入参数的意思的，你只需填两个空即可，再拓展一下

```
/**
 Description

 @param duration duration description
 @return return value description
 */
- (NSString *)startLoadingAnimation:(BOOL)duration;
```

又多了个返回参数说明，因为我们这个方法需要返回一个 string 类型的值，没错，方法的标准注释包括：语义+输入输出说明

(5) 说明：讲那么多，其实重点就是方法标准的注释这一块了，只是，为什么前面还讲#pragma mark、TODO:等注释方法方法呢，这是因为这种注释方法可以将实现中所有方法进行模块化分类，然后快速定位方法，当然，这只是本人对.m 出现庞大业务逻辑时进行的注释习惯而已，如果你有其他注释习惯，只要别人能认同，那也 ok 了。

3. 看完方法的注释对于属性的注释那就没啥了，同样那个注释快捷键搞定，

```
/**
 是否正在执行加载动画
 */
@property (assign, nonatomic, getter=isAnimating) BOOL animating;
```