

# 兼容iOS 10 资料整理笔记

2016-09-17 判若两人、 CocoaChina

▲点击上方“CocoaChina”关注即可免费学习iOS开发

原文链接: <http://www.jianshu.com/p/0cc7aad638d9>

## 1.Notification(通知)

自从Notification被引入之后, 苹果就不断的更新优化, 但这些更新优化只是小打小闹, 直至现在iOS 10开始真正的进行大改重构, 这让开发者也体会到UserNotifications的易用, 功能也变得非常强大。

- iOS 9 以前的通知

- 1.在调用方法时, 有些方法让人很难区分, 容易写错方法, 这让开发者有时候很苦恼。
- 2.应用在运行时和非运行时捕获通知的路径还不一致。
- 3.应用在前台时, 是无法直接显示远程通知, 还需要进一步处理。
- 4.已经发出的通知是不能更新的, 内容发出时是不能改变的, 并且只有简单文本展示方式, 扩展性根本不是很好。

- iOS 10 开始的通知

- 1.所有相关通知被统一到UserNotifications.framework框架中。
- 2.增加了撤销、更新、中途还可以修改通知的内容。
- 3.通知不再是简单的文本了, 可以加入视频、图片, 自定义通知的展示等等。
- 4.iOS 10相对之前的通知来说更加好用易于管理, 并且进行了大规模优化, 对于开发者来说是一件好事。
- 5.iOS 10开始对于权限问题进行了优化, 申请权限就比较简单了(本地与远程通知集成在一个方法中)。

- iOS 10 通知学习相关资料:

UserNotifications: 苹果官方文档 - 苹果官方视频1 - 苹果官方视频2 - 苹果官方视频3  
活久见的重构 - iOS 10 UserNotifications 框架解析

## WWDC2016 Session笔记 - iOS 10 推送Notification新特性

### 2.ATS的问题

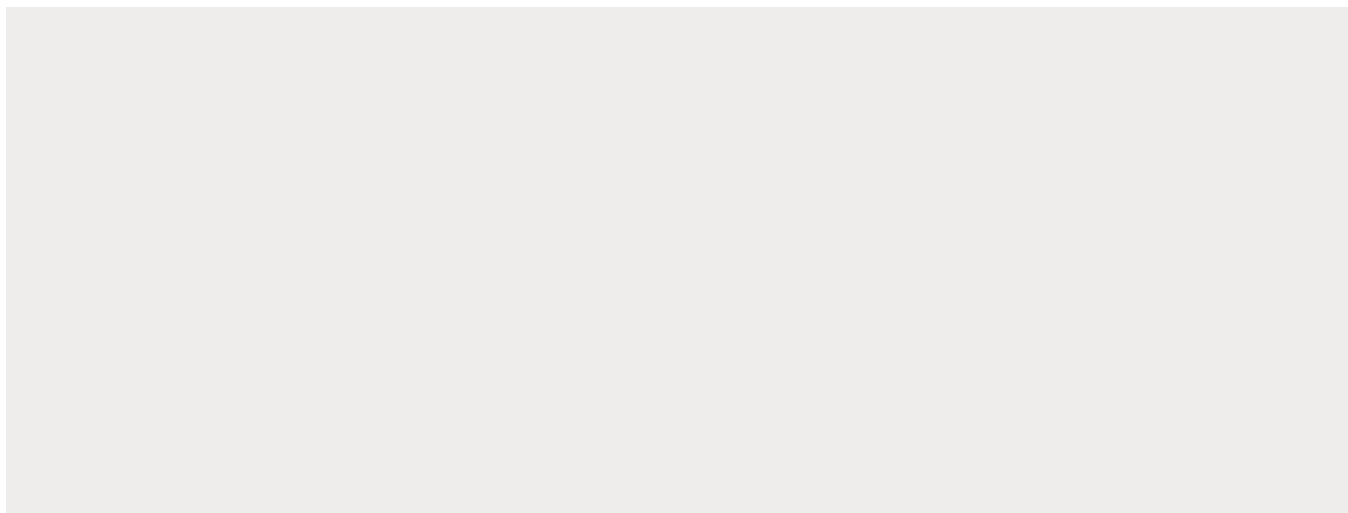
iOS 9中默认非HTTPS的网络是被禁止的，当然我们也可以把NSAllowsArbitraryLoads设置为YES禁用ATS。不过iOS 10从2017年1月1日起苹果不允许我们通过这个方法跳过ATS，也就是说强制我们用HTTPS，如果不这样的话提交App可能会被拒绝。但是我们可以通过NSExceptionDomains来针对特定的域名开放HTTP可以容易通过审核。

参考学习文章如下：

关于 iOS 10 中 ATS 的问题

### 3.iOS 10 隐私权限设置

iOS 10 开始对隐私权限更加严格，如果你不设置就会直接崩溃，现在很多遇到崩溃问题了，一般解决办法都是在info.plist文件添加对应的Key-Value就可以了。

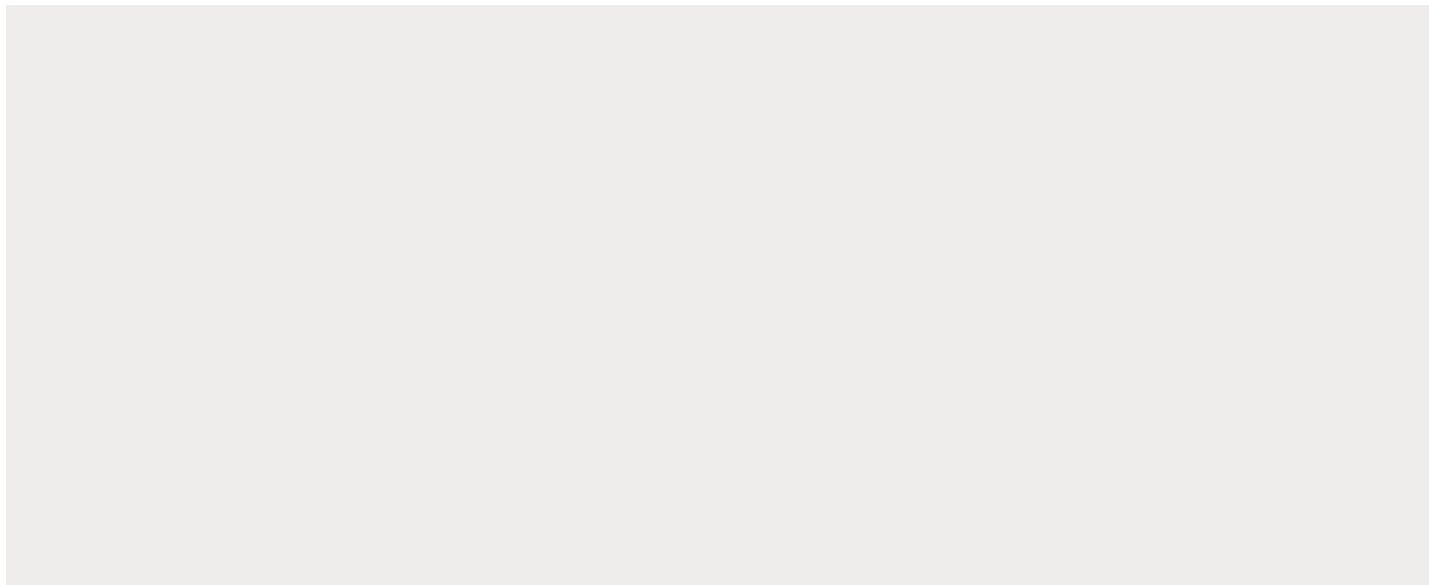


上图的访问相册的Key添加错了，正确的是Privacy - Photo Library Usage Description。

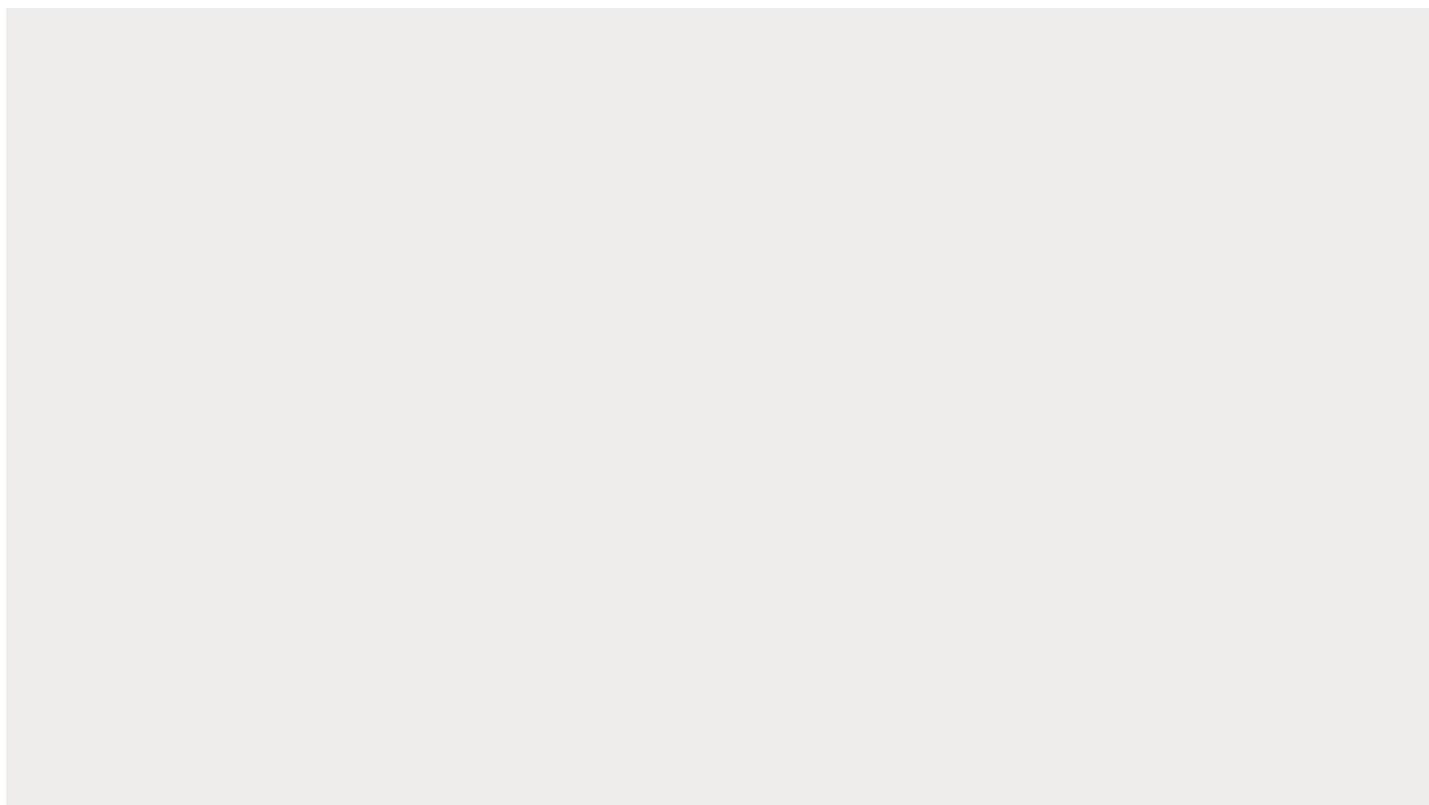
以上Value值，圈出的红线部分的文字是展示给用户看的，必须添加。目前解决办法基本都一样，参考学习文章如下：

兼容iOS 10：配置获取隐私数据权限声明

### 4.Xcode 8 运行一堆没用的logs解决办法

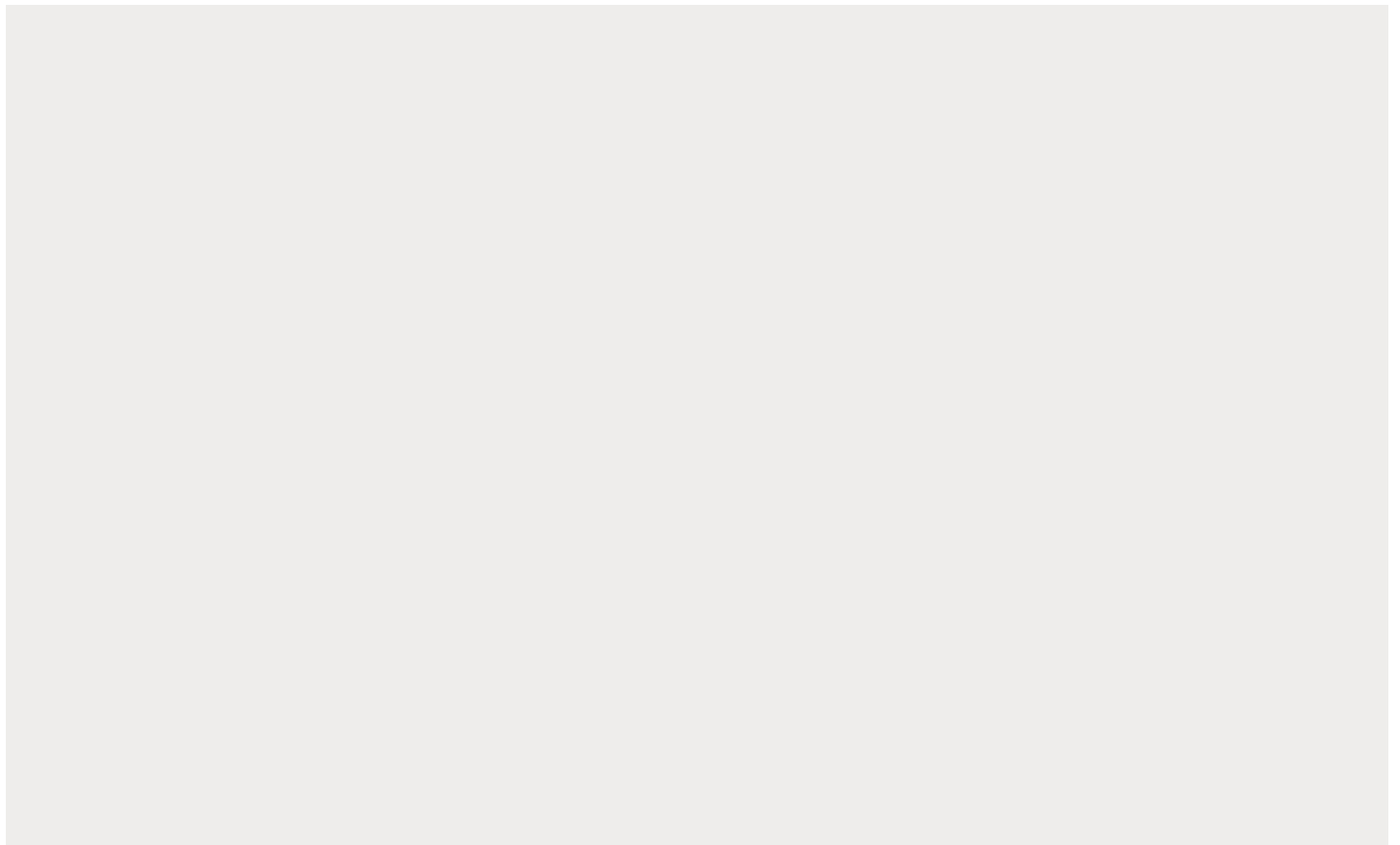


上图我们看到，自己新建的一个工程啥也没干就打印一堆烂七八糟的东西，我觉得这个应该是Xcode 8的问题，具体也没细研究，解决办法是设置OS\_ACTIVITY\_MODE : disable如下图：



相关问题连接：  
[stackoverflow问答](#)

## 5.iOS 10 UIStatusBar方法过期:



在我们开发中有可能用到UIStatusBar一些属性，在iOS 10 中这些方法已经过期了，如果你的项目中有用的话就得需要适配。上面的图片也能发现，如果在iOS 10中你需要使用preferredStatusBar比如这样：

```
//iOS 10 - (UIStatusBarStyle)preferredStatusBarStyle {    return
UIStatusBarStyleDefault;
}
```

## 6.iOS 10 UICollectionView 性能优化

随着开发者对UICollectionView的信赖，项目中用的地方也比较多，但是还是存在一些问题，比如有时会卡顿、加载慢等。所以iOS 10 对UICollectionView进一步的优化，因为叙述起来比较复杂耗费时间，在这里只提供学习参考文章如下：

WWDC2016 Session笔记 - iOS 10 UICollectionView新特性

## 7.iOS 10 UIColor 新增方法

以下是官方文档的说明：

Most graphics frameworks throughout the system, including Core Graphics, Core Image, Metal, and AVFoundation, have substantially improved support for extended-range pixel

formats and wide-gamut color spaces. By extending this behavior throughout the entire graphics stack, it is easier than ever to support devices with a wide color display. In addition, UIKit standardizes on working in a new extended sRGB color space, making it easy to mix sRGB colors with colors in other, wider color gamuts without a significant performance penalty.

Here are some best practices to adopt as you start working with Wide Color.

- In iOS 10, the UIColor class uses the extended sRGB color space and its initializers no longer clamp raw component values to between 0.0 and 1.0. If your app relies on UIKit to clamp component values (whether you're creating a color or asking a color for its component values), you need to change your app's behavior when you link against iOS 10.
- When performing custom drawing in a UIView on an iPad Pro (9.7 inch), the underlying drawing environment is configured with an extended sRGB color space.
- If your app renders custom image objects, use the new UIGraphicsImageRenderer class to control whether the destination bitmap is created using an extended-range or standard-range format.
- If you are performing your own image processing on wide-gamut devices using a lower level API, such as Core Graphics or Metal, you should use an extended range color space and a pixel format that supports 16-bit floating-point component values. When clamping of color values is necessary, you should do so explicitly.
- Core Graphics, Core Image, and Metal Performance Shaders provide new options for easily converting colors and images between color spaces.

因为之前我们都是用RGB来设置颜色，反正用起来也不是特别多样化，这次新增的方法应该就是一个弥补吧。所以在iOS 10 苹果官方建议我们使用sRGB，因为它性能更好，色彩更丰富。如果你自己为UIColor写了一套分类的话也可尝试替换为sRGB，UIColor类中新增了两个Api如下：

```
+ (UIColor *)colorWithDisplayP3Red:(CGFloat)displayP3Red green:(CGFloat)green
blue:(CGFloat)blue alpha:(CGFloat)alpha NS_AVAILABLE_IOS(10_0);

- (UIColor *)initWithDisplayP3Red:(CGFloat)displayP3Red green:(CGFloat)green
blue:(CGFloat)blue alpha:(CGFloat)alpha NS_AVAILABLE_IOS(10_0);
```

## 8.iOS 10 UITextViewContentType

```
// The textContentType property is to provide the keyboard with extra
information about the semantic intent of the text
document.@property(n nonatomic, copy) UITextViewContentType textContentType
NS_AVAILABLE_IOS(10_0); // default is nil
```

在iOS 10 UITextField添加了textContentType枚举，指示文本输入区域所期望的语义意义。

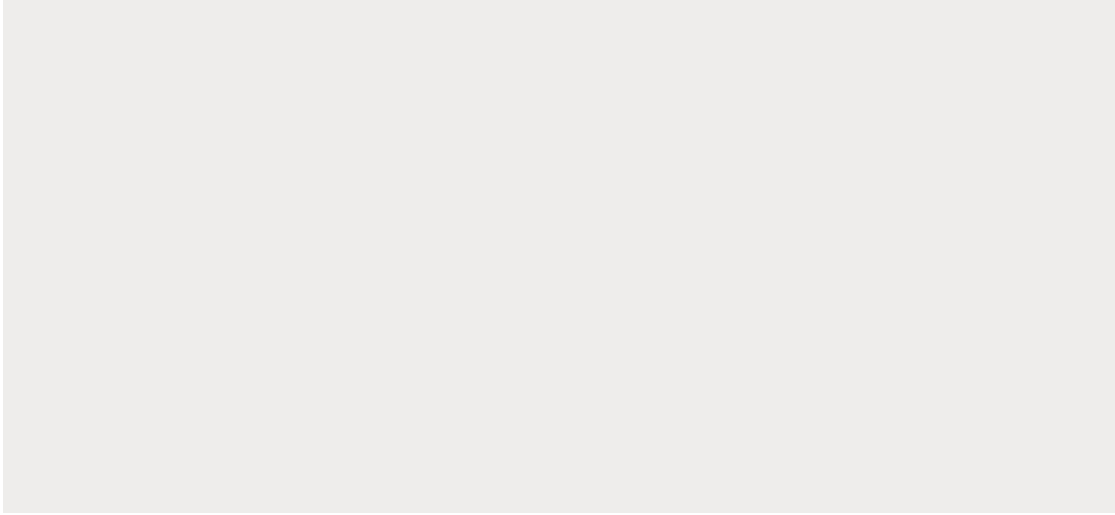
使用此属性可以给键盘和系统信息，关于用户输入的内容的预期的语义意义。例如，您可以指定一个文本字段，用户填写收到一封电子邮件确认uitextcontenttypeemailaddress。当您提供有关您期望用户在文本输入区域中输入的内容的信息时，系统可以在某些情况下自动选择适当的键盘，并提高键盘修正和主动与其他文本输入机会的整合。

## 9.iOS 10 字体随着手机系统字体而改变

当我们手机系统字体改变了之后，那我们App的label也会跟着一起变化，这需要我们写很多代码来进一步处理才能实现，但是iOS 10 提供了这样的属性adjustsFontForContentSizeCategory来设置。因为没有真机，具体实际操作还没去实现，如果理解错误帮忙指正。

```
UILabel *myLabel = [UILabel new];    /*
    UIFont 的preferredFontForTextStyle: 意思是指定一个样式，并让字体大小符合用户设定的
    字体大小。
    */
    myLabel.font =[UIFont preferredFontForTextStyle: UIFontTextStyleHeadline];
/*
    Indicates whether the corresponding element should automatically update its
    font when the device's UIContentSizeCategory is changed.
    For this property to take effect, the element's font must be a font vended
    using +preferredFontForTextStyle: or
    +preferredFontForTextStyle:compatibleWithTraitCollection: with a valid
    UIFontTextStyle.
    */
    //是否更新字体的变化
    myLabel.adjustsFontForContentSizeCategory = YES;
```

## 10.iOS 10 UIScrollView新增refreshControl

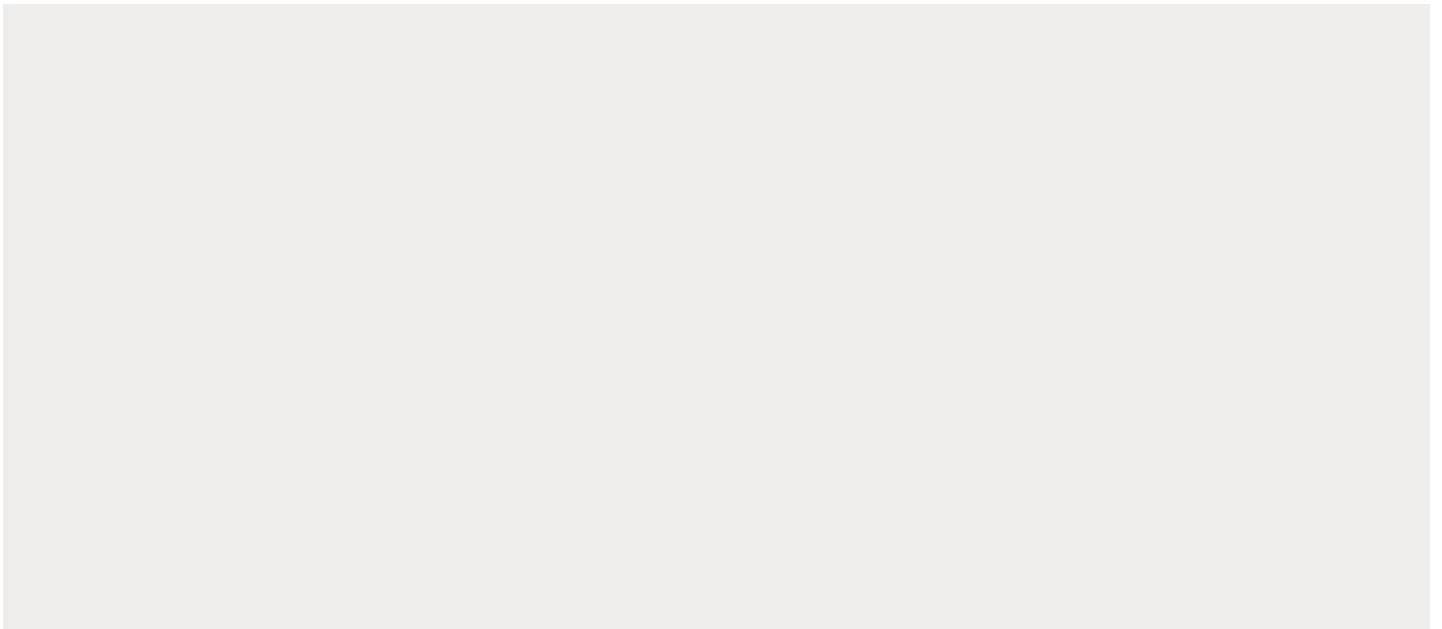


iOS 10 以后只要是继承UIScrollView那么就支持刷新功能：

```
@property (nonatomic, strong, nullable) UIRefreshControl *refreshControl  
NS_AVAILABLE_IOS(10_0) __TVOS_PROHIBITED;
```

## 11.iOS 10 判断系统版本正确姿势

判断系统版本是我们经常用到的，尤其是现在大家都有可能需要适配iOS 10，那么问题就出现了，如下图：



我们得到了答案是：

```
//值为 1[[[UIDevice currentDevice] systemVersion] substringToIndex:1]
```

```
integerValue]//值为10.000000[[UIDevice currentDevice]  
systemVersion].floatValue,//值为10.0[[UIDevice currentDevice] systemVersion]
```

所以说判断系统方法最好还是用后面的两种方法，哦~我忘记说了[[UIDevice currentDevice] systemVersion].floatValue这个方法也是不靠谱的，好像在8.3版本输出的值是8.2，记不清楚了反正是不靠谱的，所以建议大家用[[UIDevice currentDevice] systemVersion]这个方法！

Swift判断如下：

```
if #available(iOS 10.0, *) {  
    // iOS 10.0  
    print("iOS 10.0");  
} else { }
```

参考文章如下：

iOS 日常工作之常用宏定义大全

## 12.Xcode 8 插件不能用的问题

大家都升级了Xcode 8，但是对于插件依赖的开发者们，一边哭着一边去网上寻找解决办法。那么下面是解决办法：

让你的 Xcode8 继续使用插件

但是看到文章最后的解释，我们知道如果用插件的话，可能安全上会有问题、并且提交审核会被拒绝，所以建议大家还是不要用了，解决办法总是有的，比如在Xcode中添加注释的代码块也是很方便的。

## 13.iOS 10开始项目中有的文字显示不全问题

我用Xcode 8 和Xcode 7.3分别测试了下，如下图：



```

34
35 UILabel *myLabel = [UILabel new];
36 myLabel.frame = CGRectMake(150, 300, 50, 50);
37 myLabel.text = @"测试项目";
38 myLabel.backgroundColor = [UIColor purpleColor];
39 [myLabel sizeToFit];
40 [self.view addSubview:myLabel];
41 NSLog(@"%@",NSStringFromCGSize(myLabel.frame.size));
42
43 }

```

Xcode 8 打印

2016-09-16 19:39:18.173 [7329:6679887] {69.5, 20.5}

Xcode 8

```

35 UILabel *myLabel = [UILabel new];
36 myLabel.frame = CGRectMake(150, 300, 50, 50);
37 myLabel.text = @"测试项目";
38 myLabel.backgroundColor = [UIColor purpleColor];
39 [myLabel sizeToFit];
40 [self.view addSubview:myLabel];
41 NSLog(@"%@",NSStringFromCGSize(myLabel.frame.size));
42
43 }

```

Xcode 7.3 打印

2016-09-16 20:00:31.341 [7983:6705917] {68, 20.5}

Xcode 7

创建一个Label然后让它自适应大小，字体大小都是17最后输出的宽度是不一样的，我们再看一下，下面的数据就知道为什么升级iOS 10 之后App中有的文字显示不全了：

Xxcode 8打印	Xcode 7.3打印
1个文字宽度：17.5	1个文字宽度：17
2个文字宽度：35	2个文字宽度：34
3个文字宽度：52	3个文字宽度：51
4个文字宽度：69.5	4个文字宽度：68
5个文字宽度：87	5个文字宽度：85

6个文字宽度：104	6个文字宽度：102
7个文字宽度：121.5	7个文字宽度：119
8个文字宽度：139	8个文字宽度：136
9个文字宽度：156	9个文字宽度：153
10个文字宽度：173.5	10个文字宽度：170

英文字母会不会也有这种问题，我又通过测试，后来发现英文字母没有问题，只有汉字有问题。目前只有一个一个修改控件解决这个问题，暂时没有其他好办法来解决。

本文只为整理iOS 10 相关资料，也参考部分网上的文章。

iOS 10 苹果官方文档

微信号：CocoaChinabbs



**▲长按二维码“识别”关注即可免费学习 iOS 开发  
月薪十万、出任CEO、赢娶白富美、走上人生巅峰不是梦**

商务合作QQ：2408167315

投稿邮箱：support@cocoachina.com