

# 如果你浪费了一天，那么你就真的浪费了一天

iOS开发交流群:529052159欢迎大家一起加入探讨问题！

目录视图

摘要视图

RSS 订阅

文章搜索

个人资料



上天眷顾我



访问：139421次

积分：2003

等级：5

排名：第14675名

原创：8篇    转载：313篇

译文：1篇    评论：36条

文章分类

iOS开发 (297)

OC (3)

C语言 (1)

HTML5 (1)

iOS网络编程 (1)

iOS开发的故事 (4)

关于iOS面试 (8)

Android (2)

Java (1)

JavaScript (2)

HTML5 面试集 (1)

JavaScript 面试集 (1)

Java面试集 (1)

iOS开发iOS 10 创建iMessage App (1)

iOS开发iOS-RunLoop (1)

移动信息安全的漏洞和逆向原理

【观点】世界上最好的语言是什么

58到家周俊鹏：webpack PK fis，实现前端工程化我更喜

欢前者

晒知识图谱，享技术荣誉

[置顶] iOS 10 的适配问题

标签：iOS 资源大全   iOS学习资料   iOS 10 适配

2016-09-19 17:10    2009人阅读    评论(1)    收藏    举报

分类：

iOS开发 (296)

目录(?)

[+]

随着iOS10发布的临近,大家的App都需要适配iOS10,下面是我总结的一些关于iOS10适配方面的问题,如果有错误,欢迎指出.

1.系统判断方法失效:

在你的项目中,当需要判断系统版本的话,不要使用下面的方法:

```
#define isiOS10 ([[[[UIDevice currentDevice] systemVersion] substringToIndex:1] intValue]>=10)
```

它会永远返回NO,substringToIndex:1在iOS 10 会被检测成 iOS 1了,应该使用下面的这些方法:

Objective-C 中这样写:

```
#define SYSTEM_VERSION_EQUAL_TO(v) ([[[UIDevice currentDevice] systemVersion] compare:v options:NSNumericSearch] == NSOrderedSame)
#define SYSTEM_VERSION_GREATER_THAN(v) ([[[UIDevice currentDevice] systemVersion] compare:v options:NSNumericSearch] == NSOrderedDescending)
#define SYSTEM_VERSION_GREATER_THAN_OR_EQUAL_TO(v) ([[[UIDevice currentDevice] systemVersion] compare:v options:NSNumericSearch] != NSOrderedAscending)
#define SYSTEM_VERSION_LESS_THAN(v) ([[[UIDevice currentDevice] systemVersion] compare:v options:NSNumericSearch] == NSOrderedAscending)
#define SYSTEM_VERSION_LESS_THAN_OR_EQUAL_TO(v) ([[[UIDevice currentDevice] systemVersion] compare:v options:NSNumericSearch] != NSOrderedDescending)
```

或者使用:

## 文章存档

2016年11月 (2)  
 2016年10月 (33)  
 2016年09月 (33)  
 2016年08月 (40)  
 2016年07月 (42)

展开

## 阅读排行

iOS10 iMessage 带来了1 (5201)  
 iOS应用支持IPv6, 就那 (3676)  
 iOS申请邓白氏总结 (3503)  
 30分钟搞定iOS自定义相 (2800)  
 快速集成iOS基于RTMP的 (2760)  
 CornerStone的使用 (2592)  
 iOS - Masonry 使用中的 (2471)  
 iOS 二维码扫描 (你想要 (2426)  
 iOS中使用RSA加密 (2304)  
 iOS 资源大全 (2127)

## 评论排行

iOS10 iMessage 带来了1 (6)  
 iOS 开发中你是否遇到这 (4)  
 深入研究Block用weakSe (2)  
 30分钟搞定iOS自定义相 (2)  
 TableView优化之高度缓 (2)  
 iOS开发 - 清除缓存 (2)  
 关于使用SDWebImage (2)  
 iOS百度推送的基本使用 (1)  
 iOS CoreSpotlightSearch (1)  
 iOS使用Charts框架绘制 (1)

## 推荐文章

\* 程序员10月书讯, 评论得书  
 \* Android中Xposed框架篇--修改系统位置信息实现自身隐藏功能  
 \* Chromium插件 (Plugin) 模块 (Module) 加载过程分析  
 \* Android TV开发总结--构建一个TV app的直播节目实例  
 \* 架构设计: 系统存储--MySQL简单主从方案及暴露的问题

## 最新评论

关于使用SDWebImage内存爆涨! Krystal\_lei: 没有用啊 怎么回事  
 30分钟搞定iOS自定义相机 dong456tian: 感谢楼主的分享 十分好用  
 iOS的Cookie存取看我绝对够!! Regret\_V: 要是能把代码放到代码的格子里就好了, 这么看实在难受  
 iOS Cell嵌套UIWebView (内附U wuxinios: 你好, 我也正在做一个功能, 就在在滚动webview的时候, 会和外面的tableview的滚动有冲...  
 深入研究Block用weakSelf、strong

```
if ([[NSProcessInfo processInfo] isOperatingSystemAtLeastVersion:
(NSOperatingSystemVersion){.majorVersion = 9, .minorVersion = 1,
.patchVersion = 0}]) { NSLog(@"Hello from > iOS 9.1");}
if ([[NSProcessInfo processInfo] isOperatingSystemAtLeastVersion:
(NSOperatingSystemVersion){9,3,0}]) { NSLog(@"Hello from > iOS 9.3");}
```

或者使用:

```
if (NSFoundationVersionNumber > NSFoundationVersionNumber_iOS_9_0) {
// do stuff for iOS 9 and newer} else { // do stuff for older versions
than iOS 9}
```

有时候会缺少一些常量,NSFoundationVersionNumber是在NSObjCRuntime.h中定义的,作为Xcode7.3.1的一部分,我们设定常熟范围从iPhone OS 2到#define NSFoundationVersionNumber\_iOS\_8\_4 1144.17,在iOS 10(Xcode 8)中,苹果补充了缺少的数字,设置有未来的版本.

```
#define NSFoundationVersionNumber_iOS_9_0 1240.1
#define NSFoundationVersionNumber_iOS_9_1 1241.14
#define NSFoundationVersionNumber_iOS_9_2 1242.12
#define NSFoundationVersionNumber_iOS_9_3 1242.12
#define NSFoundationVersionNumber_iOS_9_4 1280.25
#define NSFoundationVersionNumber_iOS_9_x_Max 1299
```

Swift中这样写:

```
if
NSProcessInfo().isOperatingSystemAtLeastVersion(NSOperatingSystemVersion(majorVersion: 10, minorVersion: 0, patchVersion: 0)) {
// 代码块
}
```

或者使用

```
if #available(iOS 10.0, *) {
// 代码块
} else {
// 代码块
}
```

## 2.隐私数据访问问题:

你的项目中访问了隐私数据,比如:相机,相册,联系人等,在Xcode8中打开编译的话,系统会crash,控制台会输出下面这样的日志:

```
2016-09-05 19:08:06.061758 demo[5505:340803] [access] This app has crashed because it attempted to access privacy-sensitive data without a usage description. The app's Info.plist must contain an NSPhotoLibraryUsageDescription key with a string value explaining to the user how the app uses this data.
```

Snip20160905\_1.png

这是因为iOS对用户的安全和隐私的增强,在申请很多私有权限的时候都需要添加描述,但是,在使用Xcode 8之前的Xcode还是使用系统的权限通知框.

要想解决这个问题,只需要在info.plist添加NSContactsUsageDescription的key, value自己随意填写就可以,这里列举出对应的key(Source Code模式下):

Carousel1024: 我自己写了下 确实会引起循环引用 vc无法调用 dealloc 但是instruments检测不出来...

深入研究Block用weakSelf、strong  
Carousel1024: 博主所列的例子3中的结论我有些疑问我感觉这里有问题 我理解是这样的  
vc→student→block...

iOS人脸识别、自定义相机、图像  
MyOnlySunShine: 楼主啊 请问那个  
AVCaptureMovieFileOutput 和  
AVCaptureVideo...

iOS 10 的适配问题  
chaochso: xcode8 模拟器  
framework报错 真机正常

iOS10 iMessage 带来了你需要的  
duoyu5555: @github\_30943901:  
大神, 原地址能发个吗?

iOS10 iMessage 带来了你需要的  
猿人在北京: 楼主, 下次把图拉下来上传, 简书直接复制过来图片不能用的吧

```
<!-- 相册 -->
<key>NSPhotoLibraryUsageDescription</key>
<string>App需要您的同意,才能访问相册</string>

<!-- 相机 -->
<key>NSCameraUsageDescription</key>
<string>App需要您的同意,才能访问相机</string>

<!-- 麦克风 -->
<key>NSMicrophoneUsageDescription</key>
<string>App需要您的同意,才能访问麦克风</string>

<!-- 位置 -->
<key>NSLocationUsageDescription</key>
<string>App需要您的同意,才能访问位置</string>

<!-- 在使用期间访问位置 -->
<key>NSLocationWhenInUseUsageDescription</key>
<string>App需要您的同意,才能在使用期间访问位置</string>

<!-- 始终访问位置 -->
<key>NSLocationAlwaysUsageDescription</key>
<string>App需要您的同意,才能始终访问位置</string>

<!-- 日历 -->
<key>NSCalendarsUsageDescription</key>
<string>App需要您的同意,才能访问日历</string>

<!-- 提醒事项 -->
<key>NSRemindersUsageDescription</key>
<string>App需要您的同意,才能访问提醒事项</string>

<!-- 运动与健身 -->
<key>NSMotionUsageDescription</key> <string>App需要您的同意,才能访问运动与健身
</string>

<!-- 健康更新 -->
<key>NSHealthUpdateUsageDescription</key>
<string>App需要您的同意,才能访问健康更新 </string>

<!-- 健康分享 -->
<key>NSHealthShareUsageDescription</key>
<string>App需要您的同意,才能访问健康分享</string>

<!-- 蓝牙 -->
<key>NSBluetoothPeripheralUsageDescription</key>
<string>App需要您的同意,才能访问蓝牙</string>

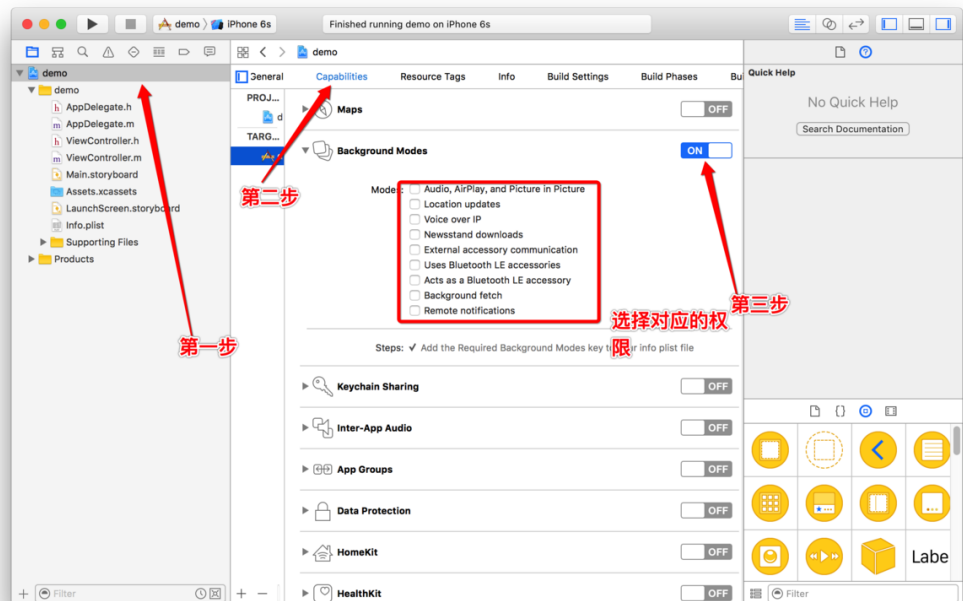
<!-- 媒体资料库 -->
<key>NSAppleMusicUsageDescription</key>
<string>App需要您的同意,才能访问媒体资料库</string>
```

如果不起作用,可以请求后台权限,类似于这样:

```
<key>UIBackgroundModes</key>
<array>
<!-- 在这里写上你在后台模式下要使用权限对应的key -->
<string>location</string>
...
</array>
```



或者在Xcode里选中当前的target,选择Capabilities,找到Background Modes,打开它,在里面选择对应权限



后台模式的操作.png

### 3.UIColor的问题

官方文档中说:大多数core开头的图形框架和AVFoundation都提高了对扩展像素和宽色域的支持.通过图形堆栈扩展这种方式比以往支持广色域的显示设备更加容易.现在对UIKit可以在sRGB的色彩空间下工作,性能更好,也可以在更广泛的色域来搭配sRGB颜色.如果你的项目是通过低级别的api自己实现图形处理的,建议使用sRGB,也就是说在项目中使用了RGB转化颜色的建议转换为使用sRGB,在UIColor类中新增了两个api:

```
- (UIColor *)initWithDisplayP3Red:(CGFloat)displayP3Red green:
(CGFloat)green blue:(CGFloat)blue alpha:(CGFloat)alpha
NS_AVAILABLE_IOS(10_0);
+ (UIColor *)colorWithDisplayP3Red:(CGFloat)displayP3Red green:
(CGFloat)green blue:(CGFloat)blue alpha:(CGFloat)alpha
NS_AVAILABLE_IOS(10_0);
```

### 4.真彩色的显示

真彩色的显示会根据光感应器来自动的调节达到特定环境下显示与性能的平衡效果,如果需要这个功能的话,可以在info.plist里配置(在Source Code模式下):

```
<key>UIWhitePointAdaptivityStyle</key>
```

它有五种取值,分别是:

```
<string>UIWhitePointAdaptivityStyleStandard</string> // 标准模式
<string>UIWhitePointAdaptivityStyleReading</string> // 阅读模式
<string>UIWhitePointAdaptivityStylePhoto</string> // 图片模式
<string>UIWhitePointAdaptivityStyleVideo</string> // 视频模式
<string>UIWhitePointAdaptivityStyleStandard</string> // 游戏模式
```

也就是说如果你的项目是阅读类的,就选择UIWhitePointAdaptivityStyleReading这个模式,

五种模式的显示效果是从上往下递减,也就是说如果你的项目是图片处理类的,你选择的是阅读模式,给选择太好的效果会影响性能.

### 5.ATS的问题

- 1.在iOS 9的时候,默认非HTTPS的网络是被禁止的,我们可以在info.plist文件中添加NSAppTransportSecurity字典,将NSAllowsArbitraryLoads设置为YES来禁用ATS;
- 2.从2017年1月1日起,,所有新提交的 app 默认不允许使用NSAllowsArbitraryLoads来绕过ATS的限制,默认情况下你的 app 可以访问加密足够强的(TLS V1.2以上)HTTPS内容;
- 3.可以选择使用NSExceptionDomains设置白名单的方式对特定的域名开放HTTP内容来通过审核,比如说你的应用集成了第三方的登录分享SDK,可以通过这种方式来做,下面以新浪SDK作为示范(Source Code 模式下):

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSExceptionDomains</key>
  <dict>
    <key>sina.cn</key>
    <dict>
      <key>NSThirdPartyExceptionMinimumTLSVersion</key>
      <string>TLSv1.0</string>
      <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
      <false/>
      <key>NSIncludesSubdomains</key>
      <true/>
    </dict>
    <key>weibo.cn</key>
    <dict>
      <key>NSThirdPartyExceptionMinimumTLSVersion</key>
      <string>TLSv1.0</string>
      <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
      <false/>
      <key>NSIncludesSubdomains</key>
      <true/>
    </dict>
    <key>weibo.com</key>
    <dict>
      <key>NSThirdPartyExceptionMinimumTLSVersion</key>
      <string>TLSv1.0</string>
      <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
      <false/>
      <key>NSIncludesSubdomains</key>
      <true/>
    </dict>
    <key>sinaimg.cn</key>
    <dict>
      <key>NSThirdPartyExceptionMinimumTLSVersion</key>
      <string>TLSv1.0</string>
      <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
      <false/>
      <key>NSIncludesSubdomains</key>
```

```

        <true/>
    </dict>
    <key>sinajs.cn</key>
    <dict>
        <key>NSThirdPartyExceptionMinimumTLSVersion</key>
        <string>TLSv1.0</string>
        <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
        <false/>
        <key>NSIncludesSubdomains</key>
        <true/>
    </dict>
    <key>sina.com.cn</key>
    <dict>
        <key>NSThirdPartyExceptionMinimumTLSVersion</key>
        <string>TLSv1.0</string>
        <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
        <false/>
        <key>NSIncludesSubdomains</key>
        <true/>
    </dict>
</dict>
</dict>

```

4.在iOS 10 中info.plist文件新加入了NSAllowsArbitraryLoadsInWebContent键,允许任意web页面加载,同时苹果会用 ATS 来保护你的app;

5.安全传输不再支持SSLv3, 建议尽快停用SHA1和3DES算法;

#### 6.UIStatusBar的问题:

在iOS10中,如果还使用以前设置UIStatusBar类型或者控制隐藏还是显示的方法,会报警告,方法过期,如下图:

```

[[UIApplication sharedApplication] setStatusBarStyle:UIStatusBarStyleDefault animated:YES];

```

⚠️ 'setStatusBarStyle:animated:' is deprecated: first deprecated in iOS 9.0 - Use -[UIViewController preferredStatusBarStyle]

UIStatusBar的警告.png

上面方法到 iOS 10 不能使用了,要想修改UIStatusBar的样式或者状态使用下图所示的属性或方法:

```

@property(nonatomic, readonly) UIStatusBarStyle
preferredStatusBarStyle NS_AVAILABLE_IOS(7_0) __TVOS_PROHIBITED; //
Defaults to UIStatusBarStyleDefault
@property(nonatomic, readonly) BOOL prefersStatusBarHidden
NS_AVAILABLE_IOS(7_0) __TVOS_PROHIBITED; // Defaults to NO
- (UIStatusBarStyle)preferredStatusBarStyle NS_AVAILABLE_IOS(7_0)
__TVOS_PROHIBITED; // Defaults to UIStatusBarStyleDefault
- (BOOL)prefersStatusBarHidden NS_AVAILABLE_IOS(7_0)
__TVOS_PROHIBITED; // Defaults to NO
// Override to return the type of animation that should be used for
status bar changes for this view controller. This currently only
affects changes to prefersStatusBarHidden.
- (UIStatusBarAnimation)preferredStatusBarUpdateAnimation
NS_AVAILABLE_IOS(7_0) __TVOS_PROHIBITED; // Defaults to

```

`UIStatusBarAnimationFade`

## 7.UITextField

在iOS 10 中,UITextField新增了textContentType字段,是UITextContentType类型,它是一个枚举,作用是可以指定输入框的类型,以便系统可以分析出用户的语义.是电话类型就建议一些电话,是地址类型就建议一些地址.可以在#import <UIKit/UITextInputTraits.h>文件中,查看textContentType字段,有以下可以选择的类型:

```

UIKIT_EXTERN UITextContentType const UITextContentTypeName
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeNamePrefix
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeGivenName
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeMiddleName
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeFamilyName
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeNameSuffix
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeNickname
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeJobTitle
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeOrganizationName
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeLocation
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const
UITextContentTypeFullStreetAddress      NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const
UITextContentTypeStreetAddressLine1     NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const
UITextContentTypeStreetAddressLine2     NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeAddressCity
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeAddressState
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const
UITextContentTypeAddressCityAndState    NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeSublocality
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeCountryName
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypePostalCode
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeTelephoneNumber
NS_AVAILABLE_IOS(10_0);
UIKIT_EXTERN UITextContentType const UITextContentTypeEmailAddress

```



```
NS_AVAILABLE_IOS(10_0);

UIKit_EXTERN UITextContentType const UITextContentTypeURL

NS_AVAILABLE_IOS(10_0);

UIKit_EXTERN UITextContentType const UITextContentTypeCreditCardNumber

NS_AVAILABLE_IOS(10_0);
```

## 8.UserNotifications(用户通知)

iOS 10 中将通知相关的 API 都统一了,在此基础上很多用户定义的通知,并且可以捕捉到各个通知状态的回调.以前通知的概念是:大家想接受的提前做好准备,然后一下全两分发,没收到也不管了,也不关心发送者,现在的用户通知做成了类似于网络请求,先发一个request得到response的流程,还封装了error,可以在各个状态的方法中做一些额外的操作,并且能获得一些字段,比如发送者之类的.这个功能的头文件是:#import <UserNotifications/UserNotifications.h>

主要有以下文件:

```
#import <UserNotifications/NSString+UserNotifications.h>
#import <UserNotifications/UNError.h>
#import <UserNotifications/UNNotification.h>
#import <UserNotifications/UNNotificationAction.h>
#import <UserNotifications/UNNotificationAttachment.h>
#import <UserNotifications/UNNotificationCategory.h>
#import <UserNotifications/UNNotificationContent.h>
#import <UserNotifications/UNNotificationRequest.h>
#import <UserNotifications/UNNotificationResponse.h>
#import <UserNotifications/UNNotificationSettings.h>
#import <UserNotifications/UNNotificationSound.h>
#import <UserNotifications/UNNotificationTrigger.h>
#import <UserNotifications/UNUserNotificationCenter.h>
#import <UserNotifications/UNNotificationServiceExtension.h>
```

## 9.UICollectionViewCell的的优化

在iOS 10 之前,UICollectionView上面如果有大量cell,当用户活动很快的时候,整个UICollectionView的卡顿会很明显,为什么会造成这样的问题,这里涉及到了iOS 系统的重用机制,当cell准备加载进屏幕的时候,整个cell都已经加载完成,等待在屏幕外面了,也就是整整一行cell都已经加载完毕,这就是造成卡顿的主要原因,专业术语叫做:掉帧.要想让用户感觉不到卡顿,我们的app必须帧率达到60帧/秒,也就是说每帧16毫秒要刷新一次.

iOS 10 之前UICollectionViewCell的生命周期是这样的:

- 1.用户滑动屏幕,屏幕外有一个cell准备加载进来,把cell从reusr队列拿出来,然后调用prepareForReuse方法,在这个方法里面,可以重置cell的状态,加载新的数据;
- 2.继续滑动,就会调用cellForItemAtIndexPath方法,在这个方法里面给cell赋值模型,然后返回给系统;
- 3.当cell马上去进屏幕的时候,就会调用willDisplayCell方法,在这个方法里面我们还可以修改cell,为进入屏幕做最后的准备工作;
- 4.执行完willDisplayCell方法后,cell就进去屏幕了.当cell完全离开屏幕以后,会调用didEndDisplayingCell方法.

iOS 10 UICollectionViewCell的生命周期是这样的:

- 1.用户滑动屏幕,屏幕外有一个cell准备加载进来,把cell从reusr队列拿出来,然后调用prepareForReuse方法,在这里当cell还没有进去屏幕的时候,就已经提前调用这个方法了,对比之前的区别是之前是cell的上边缘马上去进屏幕的时候就会调用该方法,而iOS 10 提前到cell还在屏幕外面的时候就调用;
- 2.在cellForItemAtIndexPath中创建cell, 填充数据, 刷新状态等操作,相比于之前也提前了;



- 3.用户继续滑动的话,当cell马上就需要显示的时候我们再调用willDisplayCell方法,原则就是:何时需要显示,何时再去调用willDisplayCell方法;
- 4.当cell完全离开屏幕以后,会调用didEndDisplayingCell方法,跟之前一样,cell会进入重用队列.在iOS 10 之前,cell只能从重用队列里面取出,再走一遍生命周期,并调用cellForItemAtIndexPath创建或者生成一个cell.在iOS 10 中,系统会cell保存一段时间,也就是说当用户把cell滑出屏幕以后,如果又滑动回来,cell不用再走一遍生命周期了,只需要调用willDisplayCell方法就可以重新出现在屏幕中了.在iOS 10 中,系统是一个一个加载cell的,二以前是一行一行加载的,这样就可以提升很多性能;

#### iOS 10 新增加的Pre-Fetching预加载

这个是为了降低UICollectionViewCell在加载的时候所花费的时间,在 iOS 10 中,除了数据源协议和代理协议外,新增了一个UICollectionViewDataSourcePrefetching协议,这个协议里面定义了两个方法:

```
- (void)collectionView:(UICollectionView *)collectionView
prefetchItemsAtIndexPaths:(NSArray<NSIndexPath *> *)indexPaths
NS_AVAILABLE_IOS(10_0);

- (void)collectionView:(UICollectionView *)collectionView
cancelPrefetchingForItemsAtIndexPaths:(NSArray<NSIndexPath *> *)indexPaths
NS_AVAILABLE_IOS(10_0);
```

在CollectionView prefetchItemsAt indexPaths这个方法是异步预加载数据的,当中的indexPaths数组是有序的,就是item接收数据的顺序;

CollectionView cancelPrefetchingForItemsAt indexPaths这个方法是可选的,可以用来处理在滑动中取消或者降低提前加载数据的优先级.

注意:这个协议并不能代替之前读取数据的方法,仅仅是辅助加载数据.

Pre-Fetching预加载对UITableViewCell同样适用.

### 10. UIRefreshControl的使用

在iOS 10 中, UIRefreshControl可以直接在UICollectionView和UITableView中使用,并且脱离了UITableViewController.现在RefreshControl是UIScrollView的一个属性.

使用方法:

```
UIRefreshControl *refreshControl = [[UIRefreshControl alloc] init];
[refreshControl addTarget:self action:@selector(loadData)
forControlEvents:UIControlEventValueChanged];
collectionView.refreshControl = refreshControl;
```

大家遇到问题欢迎向我提出,我会不断完善这个项目.

文 / Eternaldream (简书作者)

原文链接: <http://www.jianshu.com/p/f8151d556930>

著作权归作者所有, 转载请联系作者获得授权, 并标注“简书作者”。

顶 0 踩 0

[上一篇](#) [兼容iOS 10 资料整理笔记](#)[下一篇](#) [深入研究Block用weakSelf、strongSelf、@weakify、@strongify解决循环引用](#)

## 我的同类文章

### iOS开发 (296)

- 2016年末闲谈iOS开发的... 2016-11-02 阅读 33
- 基于iOS 10、realm封装的... 2016-10-31 阅读 15
- iOS流媒体开发之二: 滑动... 2016-10-31 阅读 8
- iOS 10 SceneKit 新特性 - ... 2016-10-31 阅读 20
- iOS 如何实现Aspect Orien... 2016-10-27 阅读 23
- Xcode8带来的新特性和坑 2016-10-27 阅读 235
- 仿UC浏览器首页下拉动画... 2016-11-02 阅读 56
- iOS流媒体开发之三: HLS... 2016-10-31 阅读 12
- iOS流媒体开发之一: 总结... 2016-10-31 阅读 11
- Swift3.0已出坑-适配iOS10... 2016-10-31 阅读 143
- iOS 如何实现Aspect Orien... 2016-10-27 阅读 21

[更多文章](#)

## 参考知识库



### Objective-C知识库

1943 关注 | 1143 收录



### iOS知识库

1666 关注 | 1277 收录



### Swift知识库

3207 关注 | 464 收录



### 算法与数据结构知识库

8935 关注 | 2263 收录

## 猜你在找

[Python项目实训](#)[微信公众平台开发入门](#)[反编译Android应用](#)[Swift视频教程\(第三季\)](#)[JavaScript for Qt Quick\(QML\)](#)[iOS10适配问题](#)[iOS开发 适配Xcode8以及iOS10-权限问题](#)[iOS开发 适配Xcode8以及iOS10-权限问题](#)[IOS 10 适配系列 \\_3\\_ Xcode 8 GM seed](#)[iOS 9适配总结](#)阿里大鱼  
阿里巴巴集团旗下

实时 稳定 简单



广告

## 查看评论

1楼 [chaochso](#) 2016-09-23 10:41发表

xcode8 模拟器framework报错 真机正常

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点, 不代表CSDN网站的观点或立场

## 核心技术类目

全部主题   Hadoop   AWS   移动游戏   Java   Android   iOS   Swift   智能硬件   Docker   OpenStack  
VPN   Spark   ERP   IE10   Eclipse   CRM   JavaScript   数据库   Ubuntu   NFC   WAP   jQuery  
BI   HTML5   Spring   Apache   .NET   API   HTML   SDK   IIS   Fedora   XML   LBS   Unity  
Splashtop   UML   components   Windows Mobile   Rails   QEMU   KDE   Cassandra   CloudStack  
FTC   coremail   OPhone   CouchBase   云计算   iOS6   Rackspace   Web App   SpringSide   Maemo

Compuware	大数据	aptech	Perl	Tornado	Ruby	Hibernate	ThinkPHP	HBase	Pure	Solr
Angular	Cloud Foundry	Redis	Scala	Django	Bootstrap					

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#)   [杂志客服](#)   [微博客服](#)   [webmaster@csdn.net](#)   400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持  
京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved 