1. 判断一个view 是否已经加到window，判断self.window是否为空

- (void)didMoveToWindow

{

if (nil != self.window) {

[self shine];

}

}

*查询一些类的私有属性 可用运行时一些代码：*

*比如查询 UITextField的一些私有属性*

*unsigned int count = 0;*

*Ivar \*ivars = class\_copyIvarList([UITextField class], &count);*

*for (int index = 0;index < count -1;index ++) {*

*Ivar ivar = ivars[index];*

*NSLog(@"%s" ,ivar\_getName(ivar));*

*};*

1. [iOS](http://lib.csdn.net/base/ios" \o "iOS知识库" \t "_blank)应用的界面方向可以通过以下几个地方来控制：   
   1. 项目的info.plist文件。其中的Supported interface orientations字段规定了初始状态下的App允许的界面方向。这是初始设置，后面可以通过AppDelegate动态修改。   
   2. AppDelegate对象的application:supportedInterfaceOrientationsForWindow:委托方法，能够在设备每次旋转的时候调用，其返回值规定了App允许的界面方向。该委托返回的值会覆盖info.plist的设置。   
   3. 每次旋转的时候，窗口的RootViewController对象，或者当前模态显示的ViewController对象，其supportedInterfaceOrientations方法返回的值，会与AppDelegate的application:supportedInterfaceOrientationsForWindow:或者info.plist文件的Supported interface orientations字段取交集，该结果决定了界面方向。如果交集为空，即没有公共交集，则报异常。   
   4. 每次旋转的时候，窗口的RootViewController对象，或者当前模态显示的ViewController对象，如果实现了shouldAutorotate方法并返回YES，则会参考其supportedInterfaceOrientations与AppDelegate或者info.plist的交集。如果返回NO，则不调用supportedInterfaceOrientations。   
   5. 我们可能会按照条件动态决定shouldAutorotate方法返回的值，以控制界面是否随设备旋转。如果在设备旋转后，动态修改了条件，则界面不会随设备旋转。如果需要让界面随设备旋转，则需要调用UIViewController类的attemptRotationToDeviceOrientation方法，这样会重新刷新界面方向。

2.返回nsstring中某个索引所在的unicode字符

[self.attributedString.string characterAtIndex:i]])

//取出属性字符串的string 调用characterAtIndex：方法得到某个位置的unicode字符

3.判断字符串第0个位置是不是空格

BOOL iswhitespace = [[NSCharacterSet whitespaceAndNewlineCharacterSet] characterIsMember:[@" dsafsdaf" characterAtIndex:0]];

4. 去除文字多余的空格

NSString \*str = @" this is a test . ";

NSCharacterSet \*whitespaces = [NSCharacterSet whitespaceCharacterSet]; NSPredicate \*noEmptyStrings = [NSPredicate predicateWithFormat:@"SELF != ''"];

NSArray \*parts = [str componentsSeparatedByCharactersInSet:whitespaces];

NSArray \*filteredArray = [parts filteredArrayUsingPredicate:noEmptyStrings];

str = [filteredArray componentsJoinedByString:@" "];

NSLog(@"%@",str);

// **this is a test .**

5.从UIColor获取当前颜色透明度

CGFloat currentAlpha = CGColorGetAlpha([(UIColor \*)value CGColor]);

Label传入一个小于1的数值返回一个和原来颜色一样，但是透明度改变的UIColor

[self.textColor colorWithAlphaComponent:percentage];

6.获取当前的时间，返回CFTimeInterval 小数（有可能是毫秒）

CFTimeInterval now = CACurrentMediaTime();

7. 返回一张带圆角的模糊化的图片（uiimage的方法）

参考yy。。kit（优酷ios）

什么是 dSYM 文件

Xcode编译项目后，我们会看到一个同名的 dSYM 文件，dSYM 是保存 16 进制函数地址映射信息的中转文件，我们调试的 symbols 都会包含在这个文件中，并且每次编译项目的时候都会生成一个新的 dSYM 文件，位于 /Users/<用户名>/Library/Developer/Xcode/Archives 目录下，对于每一个发布版本我们都很有必要保存对应的 Archives 文件 ( AUTOMATICALLY SAVE THE DSYM FILES 这篇文章介绍了通过脚本每次编译后都自动保存 dSYM 文件)。

dSYM 文件有什么作用：

当我们软件 release 模式打包或上线后，不会像我们在 Xcode 中那样直观的看到用崩溃的错误，这个时候我们就需要分析 crash report 文件了，iOS 设备中会有日志文件保存我们每个应用出错的函数内存地址，通过 Xcode 的 Organizer 可以将 iOS 设备中的 DeviceLog 导出成 crash 文件，这个时候我们就可以通过出错的函数地址去查询 dSYM 文件中程序对应的函数名和文件名。大前提是我们需要有软件版本对应的 dSYM 文件，这也是为什么我们很有必要保存每个发布版本的 Archives 文件了。

如何将文件一一对应：

每一个 xx.app 和 xx.app.dSYM 文件都有对应的 UUID，crash 文件也有自己的 UUID，只要这三个文件的 UUID 一致，我们就可以通过他们解析出正确的错误函数信息了。

1.查看 xx.app 文件的 UUID，terminal 中输入命令 ：

dwarfdump --uuid xx.app/xx (xx代表你的项目名)

2.查看 xx.app.dSYM 文件的 UUID ，在 terminal 中输入命令：

dwarfdump --uuid xx.app.dSYM

1. crash 文件内第一行 Incident Identifier 就是该 crash 文件的 UUID。

手动裁剪图像，任意粘贴，到想要的位置，实现手动p图

制作常见的电视剧场景，替换人物

类似于秒拍的应用 视频美化处理

[https://github.com/xujingzhou/VideoBeautify](https://github.com/xujingzhou/VideoBeautify" \t "_blank)

关于ios 图形处理的文章，还有其他相关的链接

<http://www.cocoachina.com/ios/20140925/9756.html>

学习opencv的英文网站

http://www.raywenderlich.com/59602/make-augmented-reality-target-shooter-game-opencv-part-1

iOS图像处理第2部分：核心图形,核心图像,GPUImage

<http://www.cocoachina.com/ios/20140925/9756.html>

ffmpeg在iOS的使用 – iFrameExtractor

ios 编译ffmpeg

<http://www.mamicode.com/info-detail-476094.html>

视频直博方案hls rtmp等

<http://www.tuicool.com/articles/MFbeUn>

FFMPEG视音频编解码零基础学习方法

<http://blog.csdn.net/leixiaohua1020/article/details/15811977/>

ios硬解码流媒体播放

http://download.csdn.net/detail/u013346305/9016225

ios ui的一些知识点 （一些不常见方法）

<http://blog.csdn.net/zhangao0086/article/details/45622875>

在本地调试远程通知

<http://blog.csdn.net/zhangao0086/article/details/45448785>

可用来伪造HTTP响应，这样不依赖服务器端，iOS的开发人员就可以测试网络服务了。<http://blog.shiqichan.com/using-afnetworking-sdwebimage-and-ohhttpstubs/>

iOS 异步图片加载优化与常用开源库分析

<https://segmentfault.com/a/1190000002776279>

ios 全景图片展示

<https://code.google.com/archive/p/panoramagl/>

开源的360度视频播放器

https://github.com/hanton/HTY360Player

webview 离线缓存

<http://bbs.csdn.net/topics/390831054>

<http://www.keakon.net/2011/08/14/为UIWebView实现离线浏览>

<https://github.com/asialee/CustomURLCache>

Oc中并发编程的相关api和面临的挑战

http://www.cocoachina.com/ios/20130718/6626.html

IOS富文本组件的实现 DTCoreText源码解析 （亮点：HTML+CSS转换成NSAttributeString）

<http://blog.cnbang.net/tech/2630/>

ios中GIF的制作的简单版本记录

http://www.2cto.com/kf/201312/268731.html

iOS开发中实现显示gif图片的方法

http://www.jb51.net/article/72773.htm

加载gif动态图的三种方式

<http://www.bkjia.com/IOSjc/996621.html>

尝试手写一个更好用的performSelector/msgSend(详细修改版)

<http://www.cocoachina.com/ios/20160111/14927.html>

ios 关于ipv6审核的一些问题

<http://www.2cto.com/kf/201605/512206.html>

IPv6迁移过程当中踩的坑

<http://www.cocoachina.com/ios/20160630/16890.html>

iOS应用支持IPV6，就那点事儿

http://blog.csdn.net/sunnyboy9/article/details/51503052

(其中的一个网络诊断工具：

本地网络环境的监测（本机IP＋本地网关＋本地DNS＋域名解析）；

通过TCP Connect监测到域名的连通性；

通过Ping 监测到目标主机的连通耗时；

通过traceRoute监测设备到目标主机中间每一个路由器节点的ICMP耗时；

https://github.com/Lede-Inc/LDNetDiagnoService\_IOS.git

)

适配IPv6遇到的坑,终极解决方案

http://blog.csdn.net/u010069091/article/details/51915179



ios视频直播

<https://github.com/runner365/LiveVideoCoreSDK>

做一款仿映客的直播App？看我就够了

<http://www.jianshu.com/p/5b1341e97757>

文章重点摘要：

8.0以上IOS还提供现成的 Video ToolBox框架，可以对摄像头和流媒体数据结构进行处理

采集端

1.目前比较知名的有VideoCore

目前国内很多知名的推流框架都是对VideoCore的二次开发。这个框架主要使用C++写的，支持RTMP推流，但对于iOS开发者来说有点晦涩难懂（精通C++的除外）。想开源和免费的可以选择现在的几个知名项目VideoCore + GPUImage+基于GPU的美颜滤镜 ,播放用IJKPlayer自己修改。

2.国内比较火的LiveVideoCoreSDK

框架提供IOS苹果手机的RTMP推流填写RTMP服务地址，直接就可以进行推流，SDK下载后简单的工程配置后能直接运行，实现了美颜直播和滤镜功能，基于OpenGL，前后摄像头随时切换，提供RTMP连接状态的回调。

这个框架是国内比较早的一款推流框架有不少在使用这个SDK，功能非常齐全，作者也比较牛，用来学习推流采集相关内容非常好，但是集成到工程中有些困难（对于我来说）。总的来说这是一款非常厉害的推流SDK，几乎全部使用C++写的，编译效率非常好，如果有实力的话推荐使用这个框架来做自己项目的推流端。

3.可读性比较好的推流LFLiveKit

框架支持RTMP（Real Time Messaging Protocol ）：实时消息传输协议，Adobe公司的。

HlS (HTTP Live Streaming) ：苹果自家的动态码率自适应技术。主要用于PC和Apple终端的音视频服务。包括一个m3u(8)的索引文件，TS媒体分片文件和key加密串文件。

推荐这个框架第一是因为它主要使用OC写的，剩下的用C语言写的，框架文件十分清晰，这对不精通C++的初学者提供了很大的便利，并且拓展性非常强，支持动态切换码率功能，支持美颜功能。

4.美颜功能

美颜的话一般都是使用的GPUImage基于OpenGl开发，纯OC语言，这个框架十分强大，可以做出各种不同滤镜，可拓展性高。如果对美颜没有具体思路可以直接用BeautifyFace，可以加入到项目中，很方便的实现美颜效果。

播放端用的针对RTMP优化过的ijkplayer（），ijkplayer是基于FFmpeg的跨平台播放器，这个开源项目已经被多个 App 使用，其中映客、美拍和斗鱼使用了 ijkplayer（5700+️） 。在本文的末未提供了，已经打包好的ijkplayer，直接拖入项目就可以使用。省去了编译的过程（编译十分麻烦，并且容易出错）。播放端调用ijkplayer参考VoideSun的简仿映客所写，如有关于ijkplayer的问题，可以请教这位大牛。

VoiderSun: http://pan.baidu.com/s/1eRJAcim

https://github.com/GrayJIAXU/520Linkee :

本项目比较完备的实现了作为一个直播App的基本功能，比如本地视频流采集、播放、美颜、礼物、点赞出心等。

关于直播优化：

其实最难的难点是提高首播时间、服务质量即Qos（Quality of Service，服务质量），如何在丢包率20%的情况下还能保障稳定、流畅的直播体验，需要考虑以下方案：

1.为加快首播时间，收流服务器主动推送 GOP :（Group of Pictures:策略影响编码质量)所谓GOP，意思是画面组，一个GOP就是一组连续的画面至边缘节点，边缘节点缓存 GOP，播放端则可以快速加载，减少回源延迟

2.GOP丢帧，为解决延时，为什么会有延时，网络抖动、网络拥塞导致的数据发送不出去，丢完之后所有的时间戳都要修改，切记，要不客户端就会卡一个 GOP的时间，是由于 PTS（Presentation Time Stamp，PTS主要用于度量解码后的视频帧什么时候被显示出来） 和 DTS 的原因，或者播放器修正 DTS 和 PTS 也行（推流端丢GOD更复杂，丢 p 帧之前的 i 帧会花屏）

3.纯音频丢帧，要解决音视频不同步的问题，要让视频的 delta增量到你丢掉音频的delta之后，再发音频，要不就会音视频不同步

4.源站主备切换和断线重连

5.根据TCP拥塞窗口做智能调度，当拥塞窗口过大说明节点服务质量不佳，需要切换节点和故障排查

6.增加上行、下行带宽探测接口，当带宽不满足时降低视频质量，即降低码率

7.定时获取最优的推流、拉流链路IP，尽可能保证提供最好的服务

8.监控必须要，监控各个节点的Qos状态，来做整个平台的资源配置优化和调度

9.如果产品从推流端、CDN、播放器都是自家的，保障 Qos 优势非常大

10.当直播量非常大时，要加入集群管理和调度，保障 Qos

11.播放端通过增加延时来减少网络抖动，通过快播来减少延时。（出自知乎宋少东）。

Mac搭建nginx+rtmp服务器

<http://www.jianshu.com/p/02222073b3f1>

rtmp协议学习总结

<http://blog.chinaunix.net/uid-17102734-id-3986995.html>

录屏：

<http://www.cocoachina.com/ios/20160317/15716.html>

<http://www.cnblogs.com/huangzizhu/p/5073389.html>

mvvm：

1.简单把网络请求与数据模型封装在一个viewmodel里面

<http://www.cocoachina.com/ios/20160421/15919.html>

2.通过ViewModel规范TableView界面开发

代码地址：<https://github.com/youzan/SigmaTableViewModel>

比较好的利用了mvvm封装各种分组tablelview，但是没有对自定义cell提供便利的接口

<http://www.cocoachina.com/ios/20160428/16002.html>

3.对自定义cell提供便利的接口，也是基于mvvm思想

代码地址：<https://github.com/chentoo/AITableView>

http://www.cocoachina.com/ios/20160216/15232.html

4．比较好的mvvm结合tableview的案例：（作者的github首页还收集了很多关于项目架构的文章，滴滴，新浪微博，蘑菇街）

<https://github.com/lovemo/MVVMFramework>

蘑菇街架构代码：

[https://github.com/mogujie/MGJRouter.git](https://github.com/mogujie/MGJRouter.git" \t "_blank)

iOS组件化思路－大神博客研读和思考

<http://www.jianshu.com/p/afb9b52143d4>

版本号规范：

<http://semver.org/>

Fastlane为iOS带来持续部署：http://www.infoq.com/cn/news/2015/01/fastlane-ios-continuous-deploy



github：<https://github.com/fastlane/fastlane>

使用 fastlane 实现 iOS 持续集成

http://www.cocoachina.com/ios/20150916/13433.html

5. iOS应用架构谈 开篇

http://casatwy.com/iosying-yong-jia-gou-tan-kai-pian.html

6. 围观神仙打架，反革命工程师《iOS应用架构谈 组件化方案》和蘑菇街Limboy的《蘑菇街 App 的组件化之路》的阅读指导

http://reviewcode.cn/article.html?reviewId=20

7.网易电商的app项目架构：LDBus

<https://github.com/Lede-Inc/LDBusBundle_IOS>

# 8.iOS组件化实践方案－LDBusMediator炼就

http://blog.csdn.net/daiyelang/article/details/51648036

9.一些常用的节省开发时间的分类

https://github.com/randomprocess/SUIUtils



10.cocoapods：

参考文章链接：

http://www.jianshu.com/p/32d9cfb91471

自我总结集成的过程：

1. 查看源：

gem sources -l

2. 删除默认的源（速度慢）

gem sources --remove <http://xxxx.com>

3. 添加淘宝的源：

gem sources -a <https://ruby.taobao.org/>

4. 更新源：

sudo gem update --system

5. 安装cocoapods

由于最新的cocoapods版本安装之后 pod setup会很慢，所以我们先安装 0.39版本，等pod setup成功之后，

再升级cocoapods为最新版本

sudo gem install -n /usr/local/bin cocoapods -v 0.39 （os10.11系统用这个）

sudo gem install cocoapods -v 0.39 （os10.11以前用这个）

pod setup的时候在这个~/.cocoapods目录下载仓库文件 会耗费较长时间，可以在pod setup的时候进入这个目录查看

进度，预计150M左右

0.39版本 pod setup 依然很慢的话：

5.1.删除国外的镜像源：

pod repo remove master

5.2.添加下面的镜像源coding 上有每日更新的，建议使用这个

pod repo add master <https://git.coding.net/CocoaPods/Specs.git>

5.3.然后再次执行 pod setup

更新仓库：

pod repo update

6.pod setup成功之后可以搜索一个第三方试试 pod search AFNetworking

7.使用cocoapods 集成第三方框架：

7.1：cd 切换到xocde项目目录

7.2：创建配置文件：vim Podfile

7.3 vim编辑器 写入以下内容：

platform :ios, '8.0'

#use\_frameworks!个别需要用到它，比如reactiveCocoa

target 'MyApp' do

pod 'AFNetworking', '~> 2.6'

pod 'ORStackView', '~> 3.0'

pod 'SwiftyJSON', '~> 2.3'

end

第二种写法：

platform :ios, '8.0'

#use\_frameworks!个别需要用到它，比如reactiveCocoa

def pods

pod 'AFNetworking', '~> 2.6'

pod 'ORStackView', '~> 3.0'

pod 'SwiftyJSON', '~> 2.3'

end

target 'MyApp' do

pods

end

深入理解 CocoaPods

<https://www.objccn.io/issue-6-4/>

11.混编ObjectiveC++

http://www.jianshu.com/p/1a7d09788ca2

12 . 漫谈iOS AOP编程之路

<http://www.jianshu.com/p/addd4eac54ed>

Aspects AOP 库，封装了 Runtime ， Method Swizzling 这些黑色技巧

13. iOS 启动连续闪退保护方案

<http://wereadteam.github.io/2016/05/23/GYBootingProtection/>

<https://github.com/liuslevis/GYBootingProtection>

14. iOS内存泄漏自动检测工具PLeakSniffer

<http://www.cocoachina.com/ios/20160706/16951.html>

MLeaksFinder：精准 iOS 内存泄露检测工具

http://wereadteam.github.io/2016/02/22/MLeaksFinder/

15：源码解析之–YYAsyncLayer异步绘制

<http://ios.jobbole.com/86878/>

16. iOS初探+load和+initialize

<http://blog.csdn.net/caryaliu/article/details/49284111>

17. 微信读书 iOS 性能优化总结

<http://wereadteam.github.io/2016/05/03/WeRead-Performance/#more>

# 18. iOS 保持界面流畅的技巧

# http://blog.ibireme.com/2015/11/12/smooth\_user\_interfaces\_for\_ios/

19.一行代码处理 cell自动计算高度

<http://blog.sunnyxx.com/2015/05/17/cell-height-calculation/>

<https://github.com/forkingdog/UITableView-FDTemplateLayoutCell>

20.手把手教你利用Jenkins持续集成iOS项目

<http://www.jianshu.com/p/41ecb06ae95f>

21. 携程App网络服务通道治理和性能优化

<http://www.jianshu.com/p/3c1941acbd79>

22. xcode插件开发:

<https://onevcat.com/2013/02/xcode-plugin/>

http://www.tuicool.com/articles/3Q7NFjR

23. ios10 https的问题，

<https://onevcat.com/2016/06/ios-10-ats/>

24为视图添加丝滑的水波纹

<http://www.jianshu.com/p/0990d1d8d9c4>

github地址:

<https://github.com/WelkinXie/WXWaveView>

25. 测试心率算法

<https://github.com/chroman/HeartBeats>

进化型是直接面对摄像头即可测出心率和呼吸频率的[What's My Heart Rate](http://link.zhihu.com/?target=http%3A//facion.net/WhatsMyHeartRate/" \t "_blank)，准确度还惊人的高。

<http://facion.net/wmhr>

26.一行代码获取通讯录联系人,并进行A~Z精准排序

<https://github.com/jkpang/PPGetAddressBook>

27. ios 更新app

<http://www.cnblogs.com/jys509/p/5390505.html>

28．一个非常功能非常全面的即时通讯开源项目

<https://github.com/zyprosoft/zychat>

xcode 常用代码块

1.assign

@property (nonatomic,assign) <#id#> <#object#>;

2.copy

@property (copy, nonatomic) <#NSString#> \*<#str#>;

3. strong

@property (nonatomic,strong) <#id#> <#object#>;

4.lazy

- (<#id#> \*)<#object#> {

if (!<#object#>) {

<#object#> = [[<#objectClass#> alloc] init];

}

return <#object#>;

}

5.tableView

- (NSInteger)tableView:(UITableView \*)tableView numberOfRowsInSection:(NSInteger)section {

return <#arrayCount#>;

}

- (UITableViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexPath \*)indexPath{

<#cellClass#> \*cell = [tableView dequeueReusableCellWithIdentifier:<#id#>];

return cell;

}

- (CGFloat)tableView:(UITableView \*)tableView heightForRowAtIndexPath:(NSIndexPath \*)indexPath{

return <#height#>;

}

- (void)tableView:(UITableView \*)tableView didSelectRowAtIndexPath:(NSIndexPath \*)indexPath {

<#viewControllerClass#> \*subVc = [[<#viewControllerClass#> alloc] init];

[self.navigationController pushViewController:subVc animated:YES];

}

5. registerNib

registerNib:[UINibnibWithNibName:NSStringFromClass([<#cellClass#>class])bundle:nil]forCellReuseIdentifier:<#cellID#>

6.creat button

UIButton \*<#btn#> = [UIButton buttonWithType:UIButtonTypeCustom];

<#btn#>.enabled = NO;

<#btn#>.layer.cornerRadius = 5;

<#btn#>.layer.masksToBounds = YES;

<#btn#>.backgroundColor = HXColor(203, 203, 203);

[<#btn#> setTitle:<#title#> forState:UIControlStateNormal];

[<#btn#> setTitleColor:[UIColor whiteColor] forState:UIControlStateNormal];

[<#btn#> setTitleColor:[UIColor lightGrayColor] forState:UIControlStateDisabled];

[<#btn#> addTarget:self action:@selector(<#action#>) forControlEvents:UIControlEventTouchUpInside];

[self addSubview:<#btn#>];

self.<#btn#> = <#btn#>;

7.creat MutableDictionary

NSMutableDictionary \*<#dict#> = [NSMutableDictionary dictionary];

<#dict#>[@""] = @"";

<#dict#>[@""] = @"";

8．Creat label

UILabel \*<#label#> = [[UILabel alloc] initWithFrame:CGRectMake(0, 0, 0, 40)];

<#label#>.backgroundColor = HXColor(4, 4, 4);

<#label#>.text = @"";

<#label#>.textAlignment = NSTextAlignmentCenter;

<#label#>.textColor = [UIColor whiteColor];

<#label#>.font = [UIFont systemFontOfSize:15];

9. Creat AlertView

UIAlertView \*alert = [[UIAlertView alloc] initWithTitle:@"" message:@"" delegate:self cancelButtonTitle:@"" otherButtonTitles:@"", nil];

[alert show];

10. Creat view

UIView \*view = [[UIView alloc] init];

view.frame = CGRectMake(0, 0, 0, 0);

view.backgroundColor = [UIColor redColor];

[self.view addSubview:view];

11.私有Pods封装个推SDK功能(解决方案)

<http://www.cocoachina.com/ios/20160717/17077.html>

12.常见状态码

<http://www.cnblogs.com/hjxcode/p/5663830.html>

13. 使用SDWebImage和YYImage下载高分辨率图，导致内存暴增的解决办法

<http://blog.csdn.net/guojiezhi/article/details/52033796>

14. SDWebImage源码解读

http://www.cnblogs.com/polobymulberry/p/5060110.html

14 HTTPS那些事（一）HTTPS原理

<http://www.guokr.com/post/114121/>

<http://www.cnblogs.com/oc-bowen/p/5896041.html>（很好的讲解https）

15.缓存实现 对yycache的最佳应用，网络请求缓存

<https://github.com/321zhangyang/SPHttpWithYYCache>

16.轮播图最佳实现，包括下面一些引入博文，剖析scrollView

<http://www.cocoachina.com/ios/20161209/18326.html>

UIScrollView 实践经验

http://tech.glowing.com/cn/practice-in-uiscrollview/

17.很牛逼的大神 介绍http://yulingtianxia.com/blog/2016/11/28/pbxprojHelper/

pbxprojHelper--Xcode工程文件助手

优化 App 的启动时间

Let's Talk About project.pbxproj

Objective-C 消息发送与转发机制原理（深入到汇编） 等等

18. iOS中的imageIO与image解码

<http://blog.mrriddler.com/2016/07/20/iOS%E4%B8%AD%E7%9A%84imageIO%E4%B8%8Eimage%E8%A7%A3%E7%A0%81/?utm_source=tuicool&utm_medium=referral>

19.iOS可视化动态绘制八种排序过程

http://mp.weixin.qq.com/s/XX1uUmJBXuqzkTgMr\_YDbQ

[https://github.com/lizelu/DataStruct-Swift/tree/master/AllKindsOfSortForiOS](https://github.com/lizelu/DataStruct-Swift/tree/master/AllKindsOfSortForiOS" \t "_blank)

20.JSPatch技术文档

<http://www.jianshu.com/p/0cb81bf23d7a>

21.安卓热更新类似IOS JSPatch

AndFix is a library that offer hot-fix for Android App.

<https://github.com/alibaba/AndFix>

22.ios里面的各种lock

http://www.jianshu.com/p/6c8bf19eb10d

23.动态化方案

腾讯-OCS <https://mp.weixin.qq.com/s/zctwM2Wf8c6_sxT_0yZvXg>

滴滴-DynamicCocoa http://www.cocoachina.com/ios/20161220/18400.html

24. 神经病院Objective-C Runtime出院第三天——如何正确使用Runtime

<http://ios.jobbole.com/89452/?utm_source=blog.jobbole.com&utm_medium=relatedPosts>

iOS动态性(二)可复用而且高度解耦的用户统计埋点实现 http://www.jianshu.com/p/0497afdad36d

25.算法

<http://blog.mrriddler.com/2016/11/14/%E7%AE%97%E6%B3%95%E7%82%BD%E7%83%AD/>

26.点击 Run 之后发生了什么？<http://url.cn/43h0BTR>

iOS 开发中的『库』 <http://url.cn/43h0EHC>

IOS 性能提升总结

<http://codecloud.net/15311.html>

即时通讯下数据粘包、断包处理实例（基于CocoaAsyncSocket）

<http://www.jianshu.com/p/2e16572c9ddc>

iOS第三方库有重复文件的问题解决方案

http://www.jianshu.com/p/122135a1f491

JSPatch被拒之完美解决方案

<http://www.mamicode.com/info-detail-1729784.html>

[在现有工程中实施基于CTMediator的组件化方案](https://casatwy.com/modulization_in_action.html)

https://casatwy.com/modulization\_in\_action.html

分享下clang AST 生成 JS相关的开源项目，其中Emscripten最成熟

Emscripten: An LLVM-to-JavaScript Compiler

git： <https://github.com/kripken/emscripten>

main page： <http://kripken.github.io/emscripten-site/>

clang ported to JS：<https://github.com/kripken/clangor>

c2js-compiler： <https://github.com/emnh/c2js-compiler>

seashell-clang-js： <https://github.com/cs136/seashell-clang-js>

<https://github.com/1e1f/clang-js>

Reporter for jscs that provides clang-style output

git：<https://github.com/cappuccino/jscs-clang-reporter>

<https://github.com/Benvie/node-clang>

解决第三方库冲突

<http://www.cnblogs.com/lizilong/p/5191214.html>

<http://www.jianshu.com/p/c7274dca62f7>

<http://blog.csdn.net/djl4104804/article/details/43099061>

tableView的优化

<http://m.blog.csdn.net/article/details?id=39827405>

<https://github.com/johnil>（实例）

<http://www.jianshu.com/p/af6b095aaaf3>

UIView中hidden、alpha、clear color与opaque的区别

<http://m.blog.csdn.net/article/details?id=50754251>

[iOS 开发：绘制像素到屏幕](https://segmentfault.com/a/1190000000390012)

https://segmentfault.com/a/1190000000390012