

目录

前言	1.1
iOS开发概览	1.2
ObjC开发	1.3
ObjC语言	1.3.1
约定	1.3.1.1
类	1.3.1.2
宏定义	1.3.1.3
函数	1.3.1.4
字符串	1.3.1.5
枚举	1.3.1.6
url	1.3.1.7
日志	1.3.1.8
数组	1.3.1.9
UI界面	1.3.2
Storyboard	1.3.2.1
view	1.3.2.2
文本框	1.3.2.3
滚动视图	1.3.2.4
事件	1.3.2.5
启动页	1.3.2.6
Swift开发	1.4
相关工具	1.5
包管理器	1.5.1
CocoaPods	1.5.1.1
Carthage	1.5.1.2
其他	1.6
和H5页面交互	1.6.1
附录	1.7
iOS开发资料	1.7.1
参考资料	1.7.2

iOS开发心得

- 最新版本: v0.7
- 更新时间: 20221104

简介

介绍Apple苹果的iOS方面的开发心得。包括ObjC开发相关的，包括ObjC语言相关的和UI界面类的；ObjC语言的包含约定、类、宏定义、函数、字符串、枚举、url、日志、数组；UI界面包含Storyboard、view、文本框、滚动视图、事件、启动页；以及Swift开发相关、相关工具和其他；工具包括包管理器的CocoaPods、Carthage。最后附上iOS开发相关资料。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

HonKit源码

- [crifan/ios_dev_summary: iOS开发心得](#)

如何使用此HonKit源码去生成发布为电子书

详见：[crifan/honkit_template: demo how to use crifan honkit template and demo](#)

在线浏览

- [iOS开发心得 book.crifan.org](#)
- [iOS开发心得 crifan.github.io](#)

离线下载阅读

- [iOS开发心得 PDF](#)
- [iOS开发心得 ePUB](#)
- [iOS开发心得 Mobi](#)

版权和用途说明

此电子书教程的全部内容，如无特别说明，均为本人原创。其中部分内容参考自网络，均已备注了出处。如发现有侵权，请通过邮箱联系我 admin 艾特 crifan.com，我会尽快删除。谢谢合作。

各种技术类教程，仅作为学习和研究使用。请勿用于任何非法用途。如有非法用途，均与本人无关。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 crifan 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

更多其他电子书

本人 crifan 还写了其他 150+ 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme: Crifan的电子书的使用说明](#)

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-11-04 15:06:13

iOS开发概览

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-10-31 17:20:57

ObjC开发

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-10-31 17:20:57

ObjC语言

TODO:

- crifan自己的iOS的ObjC的 常用代码段
 - 【记录】把iOS中通用函数整理成单独的库: CrifanLibiOS.m和CrifanLibiOS.h
- 【已解决】iOS中ObjC中如何打印输出C中的结构体变量struct stat
- 【已解决】iOS代码报错: objc Class is implemented in both app and dylib One of the two will be used Which one is undefined
- 【已解决】iOS的app中用stat打开超过2G的大文件
- 【已解决】普通被测iOS的app的ShowSysInfo的反越狱插件hook不生效

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 09:38:32

约定

iOS中的ObjC中，会看到很多类名是类似于：

最前面是2个大写的字母 + 其他命名

举例：

- NSWindow
- CAAnimation
- NSWindowController
- NSManagedObjectContext

其中的 `NS`、`CA` 等等，都是有对应约定的含义的：

前缀	(所属) 框架
NS	Foundation (OS X and iOS) and Application Kit (OS X)
UI	UIKit (iOS)
AB	Address Book
CA	Core Animation
CI	Core Image

-》以后再看到类似的类名，就能推断所属的框架=库是哪个了。

crifan.org，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2022-11-02 22:06:10

类class

TODO:

- 【已解决】iOS的ObjC的类NSURL的函数中获取到实例本身
- MRC ARC 相关
 - 【已解决】iOS的ObjC中数组NSArray类变量是否需要主动free释放内存
- 【已解决】iOS中Objective-C中NSMutableArray初始化是new还是alloc加init

创建类的典型逻辑

Object Creation Method Names Must Follow Conventions

创建一个类的几种方法:

举例:

- NSMutableArray
 - alloc + init
`NSMutableArray array = [[NSMutableArray alloc] init];`
 - new
`NSMutableArray array = [NSMutableArray new];`
 - class factory methods
`NSMutableArray array = [NSMutableArray array];`

类的声明=Class Declaration

不论是正向的: 普通的iOS的代码

还是逆向的: iOS的Logos的hook tweak代码,

写法都是一样的, 类似的:

```
// Class Declaration
@interface ClassName : SuperClass
- (ReturnType) PropertyOrFunction;
@end
```

举例:

(1) YouTube的Logos的tweak代码

```
// Class Declaration
@interface YTWatchController : NSObject
- (id)activeVideoID;
@end

hook YTWatchController

- (_Bool)isPlaybackVideoPlayingAd{
    _Bool origIsPlaybackVideoPlayingAd = orig;
    NSLog(@"origIsPlaybackVideoPlayingAd=%d", origIsPlaybackVideoPlayingAd);
```

```

if (origIsPlaybackVideoPlayingAd) {
    NSLog(@"Playing Ads, activeVideoID=%@", [self activeVideoID]);
}

return origIsPlaybackVideoPlayingAd;
}

```

(2) iOS的普通的类的代码

```

@interface CUICatalog : NSObject

@property(nonatomic,readonly) bool isVectorBased;

-(id)initWithURL:(NSURL *)URL error:(NSError **)error;
-(id)initWithName:(NSString *)n fromBundle:(NSBundle *)b;
-(id)allKeys;
-(id)allImageNames;
-(CUINamedImage *)imageWithName:(NSString *)n scaleFactor:(CGFloat)s;
-(CUINamedImage *)imageWithName:(NSString *)n scaleFactor:(CGFloat)s deviceIdiom:(int)idiom;
-(NSArray *)imagesWithName:(NSString *)n;

@end

void exportCarFileAtPath(NSString * carPath, NSString * outputDirectoryPath)
{
    ...
    CUIThemeFacet facet = [CUIThemeFacet themeWithContentsOfURL:[NSURL fileURLWithPath:carPath] error:&error];
    CUICatalog catalog = [[CUICatalog alloc] init];

    /* Override CUICatalog to point to a file rather than a bundle */
    [catalog setValue:facet forKey:@"_storageRef"];
}

```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 21:11:06

宏定义

TODO:

- 【已解决】iOS中宏定义__IPHONE_NA

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 09:41:31

函数

TODO:

- 【已解决】iOS中ObjC的多参数函数的定义和调用
 - 【已解决】iOS中C代码报错：Implicitly declaring library function malloc with type void *unsigned long
 - 【已解决】iOS中如何把ObjC提取出通用函数到独立.c文件中供别处调用
-

ObjC函数调用底层实现是 `objc_msgSend`

ObjC函数调用：

```
[someObj someFunction]
```

对应语法是：

```
[receiver message]
```

底层其实是翻译=转换成：

```
objc_msgSend(receiver, selector)
```

如果有更多参数，则是：

```
objc_msgSend(receiver, selector, arg1, arg2, ...)
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新：2022-11-02 21:05:50

字符串

TODO:

- 【已解决】iOS中ObjC中字符串列表中是否包含某个字符串
- 【已解决】iOS中如何把NSString指针转换为char字符串指针
- 【已解决】XCode中C代码字符串比较警告: Comparison between pointer and integer ('char' and 'char *)'
-

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 09:42:20

枚举

TODO:

- 【已解决】ObjC函数参数引用C中enum枚举变量类型
- 【已解决】iOS的ObjC中函数新增enum类型参数报错: Expected method body

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 09:40:55

url

TODO:

- 【已解决】iOS中NSURL转char*字符串

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-10-31 17:31:20

log 日志

TODO:

- 【已解决】iOS中优化os_log的控制是否输出详细log日志
- 【整理】iOS中的日志相关的库和查看日志的工具
-

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 21:11:25

数组

TODO:

- 【已解决】iOS的ObjC中如何静态初始化和动态创建NSDictionary数组
- 【已解决】iOS中Objective-C如何for循环数组NSArray
- 【已解决】iOS中如何初始化一个空数组并一个个添加数据
- 【规避解决】iOS中Objective-C把dict字典NSDictionary转字符串的二维数组list

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 09:43:06

UI界面

TODO:

- 【未解决】Xcode中用ObjC开发改机参数配置软件
- 【未解决】用XCode开发iOS的app用于配置改机软件参数
- 【记录】iOS的app优化：把多个单独的按钮和列表显示合并为统一的列表显示

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 09:37:03

Storyboard

TODO:

- 【已解决】iOS中如何把Storyboard中的UIView换成UIScrollView

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 09:37:33

view

TODO:

- 【已解决】iOS的app优化：把view变可滚动以扩大当前显示区域
- 【已解决】iOS越狱检测：优化app的UI拆分不同检测方式到不同视图UIViewController

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 09:38:01

文本框

TODO:

- 【已解决】iOS中用ObjC实现显示多行字符串的文本框

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 09:43:31

滚动视图

TODO:

- 【已解决】iOS中UIScrollView无法上下滚动
- 【已解决】iOS中如何设置UIScrollView的contentSize

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 09:37:20

事件

TODO:

- 【已解决】iOS中ObjC中如何在函数点击事件中判断sender的来源是哪个按钮

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 09:36:36

启动页

TODO:

- 【已解决】iOS的app中更换启动主视图主页面
- 【已解决】iOS的app界面启动后黑屏

app启动页的尺寸

- 图片格式: png图片

- 尺寸

- iOS

- 包含

- 640x960
 - 640x1136
 - 750x1334
 - 1242x2280

- 举例



640x960.png



640x1136.png



750x1334.png



1242x2280.png

- Android

- 包含

- 480x800
 - 720x1280
 - 1080x1920
 - 1440x2560

- 注意

- 要确保iOS的启动图片的尺寸是正确的，否则会导致启动时无法显示

Swift开发

iOS SDK的版本检测

TODO:

【已解决】如何在swift中判断iOS版本

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 21:50:47

相关工具

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-10-31 17:20:57

包管理器

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-10-31 17:15:12

CocoaPods

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2022-10-31 17:20:57

Carthage

TODO:

【已解决】Mac中安装Carthage

Carthage包管理器

- 官网
 - [Carthage/Carthage: A simple, decentralized dependency manager for Cocoa](#)

安装

安装Carthage有3种方式

- 可以下载Carthage.pkg去安装
 - 从
 - Releases · Carthage/Carthage
 - <https://github.com/Carthage/Carthage/releases>
 - 下载
- 也可以直接用brew去安装:
 - brew install carthage
- 也可以用MacPorts:
 - sudo port selfupdate
 - sudo port install carthage
- 也可以从源码安装
 - make install

此处用的是:

```
brew install carthage
```

安装后，确认已安装

```
which carthage
/usr/local/bin/carthage
```

查看版本:

```
> ll /usr/local/bin/carthage
lrwxr-xr-x  1 limao  admin   38B  2 14 13:47 /usr/local/bin/carthage -> ../../Cellar/carthage/0.34.0/bin/carthage
```

-> 得知当前安装的版本是: 0.34.0

基本用法

Xcode的项目中，已有配置文件

```
Cartfile
```

```
github "Alamofire/Alamofire" ~> 4.7.2
```

然后再去运行

```
carthage update
```

即可更新，安装对应的库。

如果要指定平台，则加上参数：

- 只针对Mac平台

```
carthage update --platform macOS
```

- 只针对iOS平台

```
carthage update --platform iOS
```

用法举例

```
carthage bootstrap --platform iOS
```

举例：

```
carthage bootstrap --platform iOS
*** Checking out RoutingHTTPServer at "v1.2.2"
*** Checking out YYCache at "1.1.0"
*** xcodebuild output can be found in /var/folders/gt/5868sbcd1jq4rxvryqhy2_1sz8n0s3/T/carthage-xcodebuild.L619ZF.log
*** Building scheme "RoutingHTTPServer iOS" in RoutingHTTPServer.xcodeproj
*** Building scheme "YYCache iOS" in YYCache.xcodeproj
```

其他

相关教程

关于iOS开发相关的内容，比如iPhone开发等，已整理至另外独立教程：

[苹果相关开发总结 \(crifan.org\)](#)

另外，和当前iOS开发总结的相关教程：

- [移动端APP开发总结 crifan.org](#)

其中也有一部分的，和Android开发对应的，也属于iOS开发的内容，供参考。

iOS开发经验

不要在启动页去检测新版本，可以改在登录页去检测新版本

之前做了个企业版的iOS的App

在启动页去检测新版本，所以需要访问网络

但是遇到一些情况是：

有2个 iPhone 6 Plus 的手机，ios 版本都是 10.2.1 左右

在首次安装后，即使已经去信任该app了

但是很奇怪的是，默认系统是把该app的网络访问权限禁止掉了

导致这两个6P，首次安装后，卡死在启动页，始终进不去，看不到登录页

解决办法是：

把在启动页需要访问网络的操作（比如检测新版本），改到登录页中，即可避免此问题。

但是目前还有疑惑的是：

为何 iPhone 6 Plus + iOS 10.2.1，首次安装的企业版的app，默认会禁止其网络

而之前的 iPhone 5、iPhone 6 等（好像也是 iOS 10.2.1）却没有出现这类问题。

crifan.org，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2022-11-02 22:36:14

和H5页面交互

在折腾混合app开发时，会涉及到：H5页面开发的混合app，和原生端（此处iOS的）app的交互

此处举例说明：

iOS原生app和H5的web页面的交互的定义

某APP（主要通过H5的web页面实现）和原生端（iOS/Android）的接口交互的定义

下面简称为：

- 某APP = H5
- 原生端（iOS/Android） = 原生

下面给出相关的接口定义，以及核心代码实现的示例。

功能描述：H5端调用原生的相机去扫码设备的二维码，原生解析得到设备的二维码后返回给H5

H5调用原生去扫码

- H5端调用 `scanDeviceCode()`

```
if (window.AppHost === undefined) {
    console.log(window.AppHost.scanDeviceCode);

    //Native(iOS/Android) scanDeviceCode();
    window.AppHost.scanDeviceCode();
}
```

- 原生端的具体实现

```
// 扫一扫
- (void)scanDeviceCode {
    dispatch_async(dispatch_get_main_queue(), ^{
        if ([PermissionTool cameraPemission]) { // 判断相机权限
            SubLBXScanViewController *scan_vc = [[SubLBXScanViewController alloc] init];
            scan_vc.resultBlock = ^(NSString *idString){
                // oc转js
                [self scanDeviceCodeCallback:idString];
            };
            [self.webViewController.navigationController pushViewController:scan_vc animated:YES];
        } else {
            [PermissionTool getCameraPemission];
        }
    });
}
```

原生端返回扫码结果给H5：`scanDeviceCodeCallback(scannedString)`

- H5端

需要注册js的接口

```
componentWillMount() {
    console.log("Header componentWillMount");

    window.scanDeviceCodeCallback = this.scanDeviceCodeCallback;
}
```

得到结果后进行后续处理：

```
//Native(iOS/Android): scanDeviceCodeCallback(String s);
scanDeviceCodeCallback(scannedString) {
    console.log(`scanDeviceCodeCallback: scannedString=${scannedString}`);
    let deviceCode = this.parseDeviceCode(scannedString);
    console.log(deviceCode);

    if (deviceCode === ""){
        //manually input, so return empty string
        //so goto input device code page
        route(`${ROUTE_PREFIX.DEVICE_BIND_DEVICE}`);
    } else {
        //scanned out QR code and got device code
        this.checkDeviceCodeBelongToCurCowfarm(deviceCode);

        //for debug
        // route(`${ROUTE_PREFIX.DEVICE_BIND_DEVICE}/${deviceCode}`);
    }
}
```

- 原生端的具体实现

```
// 扫一扫结束
- (void)scanDeviceCodeCallback:(NSString *)idString

// 扫一扫结束
- (void)scanDeviceCodeCallback:(NSString *)idString {
    NSString *jsFunctStr = [NSString stringWithFormat:@"scanDeviceCodeCallback('%@')", idString];
    [self.jsContext evaluateScript:jsFunctStr];
}
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新：2022-11-03 18:04:35

附录

下面列出相关参考资料。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-10-31 17:12:46

iOS开发资料

官网文档：

- Programming with Objective-C
 - [About Objective-C \(apple.com\)](#)
 - [Defining Classes \(apple.com\)](#)
 - [Working with Objects \(apple.com\)](#)
 - [Encapsulating Data \(apple.com\)](#)
 - [Customizing Existing Classes \(apple.com\)](#)
 - [Working with Protocols \(apple.com\)](#)
 - [Values and Collections \(apple.com\)](#)
 - [Working with Blocks \(apple.com\)](#)
 - [Dealing with Errors \(apple.com\)](#)
 - [Conventions \(apple.com\)](#)
- Objective-C Runtime
 - [Objective-C Runtime | Apple Developer Documentation](#)
 -
 - [Objective-C Runtime Programming Guide](#)
 - [Introduction \(apple.com\)](#)
 - [Runtime Versions and Platforms \(apple.com\)](#)
 - [Interacting with the Runtime \(apple.com\)](#)
 - [Messaging \(apple.com\)](#)
 - [Dynamic Method Resolution \(apple.com\)](#)
 - [Type Encodings \(apple.com\)](#)
 - [Declared Properties \(apple.com\)](#)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新：2022-10-31 17:24:48

参考资料

- [【已解决】iOS中iPhone6 Plus的启动页面出错不显示 – 在路上](#)
- [Conventions \(apple.com\)](#)
- [Messaging \(apple.com\)](#)
- [Carthage/Carthage: A simple, decentralized dependency manager for Cocoa](#)
- [Using the Inspector · facebookarchive/WebDriverAgent Wiki](#)
- [Swift Tutorial 1: Starting with a Single View Application -> Hello World | Swift for IOS](#)
- [Swift Tutorial 2: Adding a View Controller | Swift for IOS](#)
- [About iOS App Architecture](#)
- [iOS中文文档: UIView | 应用开发笔记](#)
- [UIView Class Reference](#)
- [UIView与 UIWindow, UIResponder那些事儿 - 简书](#)
- [iOS UIView非常有用方法及属性详解 - kingsley_cxz的专栏 - 博客频道 - CSDN.NET](#)
- [UIView、Subview的几个基础知识点-iOS开发\(实例 - CocoaChina 苹果开发中文站 - 最热的iPhone开发社区 最热的苹果开发社区 最热的iPad开发社区](#)
- [View Controller Programming Guide for iOS: Design Tips](#)
- [View Programming Guide for iOS](#)
- [clipsToBounds - UIView Class Reference](#)
- [NSLayoutConstraint Class Reference](#)
- [Gesture Recognizers](#)
- [iOS Human Interface Guidelines: Designing for iOS](#)
- [Conventions](#)
- [View Controller Programming Guide for iOS: The Adaptive Model](#)
- [View Controller Programming Guide for iOS: Defining Your Subclass](#)
- [View Controller Programming Guide for iOS: The Role of View Controllers](#)
- [UIBezierPath Class Reference](#)
- [UIKit User Interface Catalog: About Views](#)
- [About Text Handling in iOS](#)
- [About Drawing and Printing in iOS](#)
- [...](#)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2022-11-02 22:08:11