

摘录

2016年6月14日 星期二 下午7:32

唐巧技术博客

1、iOS处理图片

1、iOS支持APNG、GIF，但不能播放，保存方式：

[ALAssetsLibrary writeImageDataToSavedPhotosAlbum:metadata:completionBlock]

UIImageWriteToSavedPhotosAlbum() 写相册，图像会被强制转码为PNG

2、保存UIImage方式：

UIImageJPEGRepresentation(0.9) (最佳)

UIImagePNGRepresentation() (其次)

NSKeyedArchiver调用UIImagePNGRepresentation进行序列化 (耗时)

3、UIImage缓存：

通过 imageNamed 创建 UIImage 时，系统实际上只是在 Bundle 内查找到文件名，然后把这 UIImage 里返回，并没有进行实际的文件读取和解码。当 UIImage 第一次显示到屏幕上时，其法才会被调用，同时解码结果会保存到一个全局缓存去。图片解码后，App第一次退到后台和时，该图片的缓存才会被清空，否则一直存在。imageWithData也会缓存，但如果图片被释放码数据也会被立刻释放。

4、避免缓存：

1、把图片用 CGContextDrawImage() 绘制到画布上，然后把画布的数据取出来当作图片

2、手动调用 CGImageSourceCreateWithData()来创建图片，关掉缓存，每次都需显示解码

5、图片的3中编码方式：

baseline (逐行扫描)：JPEG、PNG、GIF 默认保存方式

interlaced (隔行扫描)：PNG、GIF

progressive (渐进式)：JPEG

这个文件名放到
其内部的解码方
收到内存警告
，其内部的解

progressive (渐进式) : JPEG

2、RAC

[学习资料汇总](#)

将一个对象的一个属性和一个signal绑定，signal没产生一个value (id类型) 都会自动执行

```
RAC(TARGET, [KEYPATH, [NIL_VALUE]])
```

```
RAC(self.outLabel,text,@"nil时默认值")=self.inputField.rac_textSign
```

观察target的keyPath属性，相当于KVO，产生一个RACSignal，text和name绑定实现联动

```
RAC(self.outLabel,text) = RACObserve(self.model,name);
```

在block内部管理对self的引用，@是@autoreleasepool{}前面的那个@，成对出现

@weakify(Obj); 在外部定义一个weakSelf

@strongify(Obj); 在内部定义一个strongSelf指向weakSelf

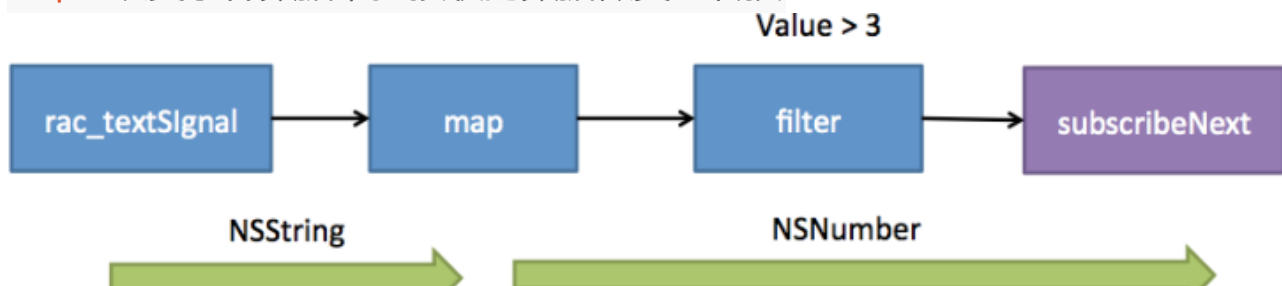
RACSignal的每个操作都会返回一个RACsignal -> 链式编程

RACSignal -> subscriber ==> next,error,completed

filter : 实现过滤、验证逻辑



map : 改变事件数据，把接收的数据做类型转换



combine : 多个信号可以聚合成一个新的信号 (同时满足条件)

分割 : 信号可以有多个subscriber订阅者，作为很多后续步骤的信号源

then方法会等待**completed**事件的发送 (会跳过**error**)，然后再订阅由**then** block返回的

flattenMap返回一个新信号

deliverOn回主线程更新UI

al;

]

的信号

observeOn: 回主线程更新UI

subscribeOn: RACScheduler在后台线程中下载图片

throttle：前一个**next**事件在指定的时间段内没有被接收到才会发送**next**事件，用于停顿

RACDisposable：中断订阅某信号；订阅完成后清理回收资源

当订阅者subscriber接收到error或completed中的一个时，不会再接收任何事件了

