* [13.1 面试-C语言-二叉树](http://192.168.88.8:8090/iosroot/play.php?id=88e78b66160zb76b33b3177z1dc1bbc6347z387d9e7f153)  次数(7)   26分钟 二分查找
* [13.2 面试-C语言-冒泡与链表](http://192.168.88.8:8090/iosroot/play.php?id=2c2caf67172z3eab1b63189z4a8f9130256z7600311d165)  次数(1)   1.3小时 快速排序
* [13.3 面试-C语言-双向链表](http://192.168.88.8:8090/iosroot/play.php?id=50bd2bed45zb59c10d962z94764719130z373bd31438)  次数(0)   28分钟 NSArray
* [13.4 面试-C语言-哈希](http://192.168.88.8:8090/iosroot/play.php?id=8b1cee1e121zc93d4be0138z335245b9207z2cf40bc6114)  次数(4)   20分钟 NSDictionary
* [13.5 面试-iOS代理-如何做相册](http://192.168.88.8:8090/iosroot/play.php?id=72b9d4d589z9600a62b106zda376234324z78b9afd282)  次数(4)   1.2小时
* [13.6 面试-iOS面试-ARC介绍](http://192.168.88.8:8090/iosroot/play.php?id=e23943f6159z766bec35176zf0894d60393z984cae36152)  次数(3)   49分钟
* [13.7 面试-iOS面试-UITableView优化](http://192.168.88.8:8090/iosroot/play.php?id=911f25e399z9d07f669116z473e636e332z2f543d3c92)  次数(2)   1.1小时

[13.8 面试-OC面试](http://192.168.88.8:8090/iosroot/play.php?id=6a6f760e110z47397305127zb54fe9ed342z48828705103)

* 1. 简单
  2. [4.1 OC-5.1-内存管理1-黄金法则-HD](http://192.168.88.8:8090/iosroot/play.php?id=5b4de2a8178zb504e7bc195z677cfbf9334z4186ea2c171)  次数(24)   1小时
  3. [4.2 OC-5.2-内存管理2-retain点语法-HD](http://192.168.88.8:8090/iosroot/play.php?id=194b0210165zaf12342e182z56bfe2cd322z5789ceb4158)  次数(17)   8分钟
  4. [4.3 OC-5.4-内存管理4-AutoreleasePool-HD](http://192.168.88.8:8090/iosroot/play.php?id=6e03279f90zf28a7f5a107zb3a9aca0249z19135c6d83)  次数(10)   53分钟
  5. 困难
  6. [4.4 OC-5.3-内存管理3-MyArray-HD](http://192.168.88.8:8090/iosroot/play.php?id=4732c9e9203z772f827b220z38f20c45361z6a6a04f0196)  次数(11)   37分钟
  7. [4.5 OC-5.5-内存管理5-浅拷贝详述-HD](http://192.168.88.8:8090/iosroot/play.php?id=5617b258128z16afe747145z191fe11a288z458b4617121)  次数(11)   24分钟
  8. [4.6 OC-5.6-内存管理6-深拷贝详述-HD](http://192.168.88.8:8090/iosroot/play.php?id=7def0860119z4e3c69db136z1311f636280z99b9b6d2112)  次数(8)   18分钟
  9. [4.7 OC-5.7-内存管理7-自己创建AutoreleasePool](http://192.168.88.8:8090/iosroot/play.php?id=101a2fd641zcc1e77d658z4a41c81f212zc20b51c634)  次数(10)   45分钟
* [5.5 OC-6.1-设计模式-单例设计模式-HD](http://192.168.88.8:8090/iosroot/play.php?id=abb4b82687zd85b2bb4104z5e491566249z795acb9280)  次数(45)   13分钟
* [5.6 OC-6.2-设计模式-单例各种写法-HD](http://192.168.88.8:8090/iosroot/play.php?id=c082048b131zca945894148z853e86ef294za3d3cf40124)  次数(18)   9分钟
* [5.7 OC-6.3-设计模式-通知中心-HD](http://192.168.88.8:8090/iosroot/play.php?id=4de39e90176z93767984193z01f80ca4340z9f98f531169)  次数(12)   16分钟
* [5.8 OC-6.5-设计模式-KVO-HD](http://192.168.88.8:8090/iosroot/play.php?id=d7583dbb134zbd6ebad6151z54ab7773300zcd753547127)  次数(14)   22分钟

[5.9 OC-6.6-设计模式-KVC-HD](http://192.168.88.8:8090/iosroot/play.php?id=4cbc583d126zc6c73979143zd6c2e31a293z4031e45b119)

什么是缓存

文件的缓存。一般是缓存网络数据

一句话，缓存的目的是为了第二次访问快速一些。

iOS开发中，常常图片是需要缓存的， 为什么图片是需要缓存的.

<http://xxx.com/a.jpg> hash/md5

Library, Documents, XXX.app Tmp

缓存应该放在Library里面自己建立文件夹

清除缓存是删除某个文件夹里面的资源，比如图片，视频，音乐等

缓存的策略: 凡是超过10天的资源都删掉 策略是开发者根据资源特性来定

ASI也是有缓存的。比如把下载的数据保存一定时间。

内存的缓存.一般是缓存文件数据

UIImage \*img = [UIImage imageWithNamed:@”demo.png”];

UIImage \*img2 = [UIImage imageWithNamed:@”demo.png”];

UIImage \*img3 = [UIImage imageWithNamed:@”demo.png”];

+ (id) imageWithNamed:(NSString \*)f {

NSString \*s = [[NSBundle mainBundle] pathForResource:f withType:nil];

NSData \*d = [NSData dataWithContentsOfFile:s];

UIImage \*img = [UIImage alloc] initWithData:d];

return [img autorelease];

}

static NSMutableArray \*cacheArr = [[NSMutableArray alloc] init];

+ (id) imageWithNamedWithCache:(NSString \*)f {

NSString \*s = [[NSBundle mainBundle] pathForResource:f withType:nil];

NSData \*d = nil;

// 判断缓存中时候已经存在如果存在就读取

forin(NSDictinary \*mydict in cacheArr) {

id d = [mydict objectForKey:s];

if (d) { goto createImage; }

}

d = [NSData dataWithContentsOfFile:s];

NSDictionary \*dict = [NSDictionary dictionaryWithObjectsAndKey:

d, s, nil];

[cacheArr addObject:dict];

createImage:

UIImage \*img = [UIImage alloc] initWithData:d];

return [img autorelease];

}

Three20 / 320

如果界面比较卡怎么办

1. 图片要用异步下载，并且要缓存起来

2. 对于图片的加载到界面上 要复用

为什么要用异步？

异步的主要目的是为了界面流畅

ASI和原生网络下载的优缺点

1. ASI下载小数据的有有点，原生要处理3个函数
2. ASI只能处理标准的HTTP, GET/POST协议. 原生的可以处理任何HTTP协议

如何下载大文件，比如超过10M

最好用原生的去下载，没下载一些就把数据追加到文件的后面。

// asihttprequest有什么缺点1, 没人维护 2, 不支持arc, 3. 很笨重, 每一个asi就是一个线程 asihttprequest : nsoperation,

AFNetworking.

如何断点下载

nsurl \*url = [nsurl urlwithstring:@"http://www.1000phone.com/x.rmvb"];

nsurlrequest \*r = [nsurlrequest requestwithurl:url];

// 请求头 Range 20M-

[r addHeader:@"20M-" forHeader:@"Range"];

nsurlconnnection \*c = [nsurlconnection alloc] initwithrequest:r];

只需要在请求头上加一个Range字段就可以了。

[[UIApplication shareApplication] openURL:@”tel://182143333”];

[[UIApplication shareApplication] openURL:@”sms://182143333”];