

iOS 面试题 by JiWei

1.搞清楚touch事件的传递(事件的响应链)

事件的响应（responder chain）

只有继承了UIResponder的类才能响应touch事件，从上图的响应者链可以看出，优先是最上层的view响应事件，如果该view有视图控制器的话会是下一个响应者，否则就是该view的父视图，这样至上而下传递事件。直到单例UIWindow对象，最后是单例UIApplication对象以终止，UIApplication的下一个响应者是nil，已结束整个响应循环。事件在传递过程中视图可以决定是否需要对该事件进行响应。

事件分发（Event Delivery）

第一响应者（First responder）指的是当前接受触摸的响应者对象（通常是一个UIView对象），即表示当前该对象正在与用户交互，它是响应者链的开端。整个响应者链和事件分发的使命都是找出第一响应者。

UIWindow对象以消息的形式将事件发送给第一响应者，使其有机会首先处理事件。如果第一响应者没有进行处理，系统就将事件（通过消息）传递给响应者链中的下一个响应者，看看它是否可以进行处理。

iOS系统检测到手指触摸(Touch)操作时会将其打包成一个UIEvent对象，并放入当前活动Application的事件队列，单例的UIApplication会从事件队列中取出触摸事件并传递给单例的UIWindow来处理，UIWindow对象首先会使用hitTest:withEvent:方法寻找此次Touch操作初始点所在的视图(View)，即需要将触摸事件传递给其处理的视图，这个过程称之为hit-test view。

UIWindow实例对象会首先在它的内容视图上调用hitTest:withEvent:，此方法会在其视图层级结构中的每个视图上调用pointInside:withEvent:（该方法用来判断点击事件发生的位置是否处于当前视图范围内，以确定用户是不是点击了当前视图），如果pointInside:withEvent:返回YES，则继续逐级调用，直到找到touch操作发生的位置，这个视图也就是要找的hit-test view。hitTest:withEvent:方法的处理流程如下：首先调用当前视图的pointInside:withEvent:方法判断触摸点是否在当前视图内；若返回NO,则hitTest:withEvent:返回nil; 若返回YES,则向当前视图的所有子视图(subviews)发送hitTest:withEvent:消息，所有子视图的遍历顺序是从最顶层视图一直到最底层视图，即从subviews数组的末尾向前遍历，直到有子视图返回非空对象或者全部子视图遍历完毕；若第一次有子视图返回非空对象，则hitTest:withEvent:方法返回此对象，处理结束；如所有子视图都返回非，则hitTest:withEvent:方法返回自身(self)。

2.frame, bounds, center, alpha,opaque,hidden

这些都是view的一些基本属性。frame是描述该view在其父视图中的一块区域。其坐标系是在其父视图中的坐标。我们在进行view的初始化时会经常使用到frame。bounds也是描述该view的大小，是其在自身的坐标系中的位置大小。center是描述其在父视图的中心位置坐标。我们在进行view的位置改变而不改变view的大小时，会使用center。alpha是用来描述改view的透明度从0到1，0表示的是透明，1表示不透明。alpha支持动画（animation），alpha = 0 与 hidden = YES 效果一样都是看不到view，但是后者相比开销大。在alpha等于0时view接受touch事件，但是hidden则不接受。并且hidden和opaque 不支持动画。alpha并不影响镶嵌在其内部view行为，而hidden会影响。当把view设置为透明背景时，一般把opaque设置为NO，可以减少开销，优化内存。opaque影响图形绘制系统。设置为YES，会优化view的绘制。

3, nil,NSNULL,NULL区别

nil是指向obj - c中对象的空指针，是一个对象，在o - c中nil对象调用方法不会引起crash。

Nil是指向obj - c中的类的空指针，表示的是一个空类。

NULL是指向任何类型的空指针（如c / c++中的空指针），在objective - c中是一个数值。

NSNULL用于集合操作，在集合对象中，表示一个空值的集合对象。

4.NSThread,NSOperation,GCD

NSThread,NSOperation,GCD是IOS中使用多线程的三种方式之一。他们各有优缺点。抽象层次是从低到高的，抽象度越高的使用越简单。

NSThread, 缺点：需要自己维护线程的生命周期和线程的同步和互斥，但是这些都需要耗费系统的资源。优点：比其它两个更轻。

NSOperation,优点：不需要自己管理线程的生命周期和线程的同步和互斥等。只是需要关注自己的业务逻辑处理，需要和NSOperationQueue一起使用。

GCD, 是Apple开发的一个多核编程解决方法，优点：比前面两者更高效更强大。

5.autorelease ,ARC 和非ARC

autorelease 自动释放，与之相关联的是一个自动释放池（NSAutoreleasePool）。autorelease的变量会被放入自动释放池中。等到自动释放池释放时（drain）时，自动释放池中的自动释放变量会随之释放。ios系统应用程序在创建是有一个默认的NSAutoreleasePool，程序退出时会被销毁。但是对于每一个RunLoop，系统会隐含创建一个AutoreleasePool，所有的release pool会构成一个栈式结构，每一个RunLoop结束，当前栈顶的pool会被销毁。

ARC，自动应用计数。（iOS 6加入）IOS内存管理是基于变量的应用计数的。这样系统帮你管理变量的release，retain等操作。

非ARC，非自动应用计数。手动管理内存。自己负责系统变量的release，retain等操作。做到谁分配谁释放，及alloc和release像对应。函数返回对象时使用autorelease。

可以使用Xcode将非ARC转化为ARC，ARC和非ARC混编。可在在编译ARC时使用 -fno-objc-arc，-fobjc-arc标签。实际需要看工程是支持还是不支持ARC模式。

6.xib，storyboard，手动书写代码

xib（interface buider），方便对界面进行编辑。可以在窗口上面直接添加各种视图，优点：直接看到界面的效果，操作简单。缺点：不方便对视图进行动态控制，不灵活。

手动编写代码，继承（主要是UIView，UIViewController），优点：可以对视图进行定制，灵活控制方便。缺点：不能马上看到效果，复杂。

storyboard（故事板在ios6加入）。优点：可以看到界面效果，能同时进行多个界面的交互，高效快速。缺点：不能进行进行界面的定制，却笑灵活性。

xib和storyboard主要用于界面中的元素位置固定和清楚里面有哪些元素。但是如果需要动态变化界面还是手动编写代码比较好。一般还是各种方式混合使用。

7.loadView,viewDidLoad,ViewDidLoad,viewWillAppear,viewDidAppear,viewwilld

当view的nib文件为nil时，手动创建界面时调用loadView，当view的nib文件存在时，会在viewDidLoad中实现。但是当你的程序运行期间内存不足时，视图控制器收到didReceiveMemoryWarning时，系统会检查当前的视图控制器的view是否还在使用，如果不在，这个view会被release，再次调用loadView来创建一个新的View。viewDidLoad ,不论是从xib中加载视图，还是从loadview生成视图，都会被调用。但是如果改view在栈中下一次显示是不会被调用。ViewWillAppear，ViewDidAppear会在view每次即将可见和完全显示时都会调用。我们会在ViewWillAppear里面进行一些view显示的准备工作，ViewDidLoadWillAppear 和ViewWillDisAppear时会在view每次消失时都会调用。当系统收到didReceiveMemoryWarning通知时显示内存不足时，会调用ViewDidLoadUnload来清理View中的数据和release后置为nil。

8，copy 和retain区别

retain,相当于指针拷贝。变量的引用计数加一。另外一个指针也指向改地址。

copy，相当于内容拷贝。变量的引用计数加一。但是自己本身计数不变。开辟另外一个地址空间放入相同变量的值进去。

9，手动写setter和getter方法

```
- (void) setOldValue: (NSString*) newValue {  
  
    if (newValue !=oldValue) {  
  
        [oldValue release];  
  
        oldValue = [newValue retain];  
  
    }  
  
}
```

10，NSRunLoop 和NSOperationQueue

NSRunLoop 是所有要监视的输入源和定时源以及要通知的注册观察者的集合.用来处理诸如鼠标，键盘事件等的输入源。每一个线程拥有自己的RunLoop有系统自动创建。你不应该自己去创建，只能获取。一般都不会用NSRunLoop,因为它不是线程安全的。一般都用CFRunLoop，这个是线程安全的，是一种消息处理模式，我们一般不用进行处理。

NSOperationQueue时一个管理NSOperation的队列。我们会把NSOperation放入queue中进行管理。

11,IOS常用的设计模式

单例模式，DeafutCenter,Deafultqueue等

MVC模式，View，model,ViewController。

观察者模式，通知，KVO

工厂模式，

代理模式，delegate

12.内存管理和优化

原则：

- 谁创建，谁释放（类似于“谁污染，谁治理”）。如果你通过alloc、new或copy来创建一个对象，那么你必须调用release或autorelease。换句话说，不是你创建的，就不用你去释放。例如，你在一个函数中alloc生成了一个对象，且这个对象只在这个函数中被使用，那么你必须在这个函数中调用release或autorelease。如果你在一个class的某个方法中alloc一个成员对象，且没有调用autorelease，那么你需要在这个类的dealloc方法中调用release；如果调用了autorelease，那么在dealloc方法中什么都不需要做。 1.2 除了alloc、new或copy之外的方法创建的对象都被声明了autorelease。 1.3 谁retain，谁release。只要你调用了retain，无论这个对象是如何生成的，你都要调用release。有时候你的代码中明明没有retain，可是系统会在默认实现中加入retain。
- 在收到内存didReceiveMemoryWarning的警告时，释放掉一些不再需要的资源，注意编码规范，如一些变量不使用需要及时释放。避免是占用太多的内存空间，有时需要用空间去换取时间，尽量使用一些高效的算法和数据结构节约内存空间。最后使用一些内存检测工具和代码的静态分析查找内存泄漏和分配（instrument，leaks，allocations）。

13，tableview的优化

优化：

1.1 正确的复用cell。

1.2 减少在返回每个cell里面的处理逻辑和处理时间。尽量将数据进行缓存和复用。

1.3，尽量减少处理加载和计算的时间，不阻塞UI线程。

1.4，尽量使用绘制每个cell。

1.5，设置每个cell的opaque属性。

1.6，尽量返回每行固定的height。

1.7，在每个cell减少图形效果。

1.8，分段加载数据。

14，opengl，quartz 2d

上面2种方式是进行图形绘制会使用到的技术。

quartz 2d 是Apple提供的基于Core graphic的绘制基本图形工具库。操作简单方便，能够满足大部分需要。只是适用于2D图形的绘制。

opengl，是一个跨平台的图形开发库。适用于2D和3D图形的绘制。功能强大但是复杂。

15, animation

IOS提供丰富的Core Animation动画满足用户的需要，主要实现方式如下3种：

1.1 commitAnimations方式使用UIView动画

```

UIView Animations 动画:
[UIView beginAnimations:@"animationID" context:nil];
[UIView setAnimationDuration:0.5f];
[UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];
[UIView setAnimationRepeatAutoreverses:NO];
//以下四种效果
/*
[UIView setAnimationTransition:UIViewAnimationTransitionFlipFromLeft forView:self.view cache:YES];//oglFlip, fromLeft
[UIView setAnimationTransition:UIViewAnimationTransitionFlipFromRight forView:self.view cache:YES];//oglFlip, fromRight

[UIView setAnimationTransition:UIViewAnimationTransitionCurlUp forView:self.view cache:YES];
[UIView setAnimationTransition:UIViewAnimationTransitionCurlDown forView:self.view cache:YES];
*/
//你自己的操作
[UIView commitAnimations];

```

1.2、CATransition

```

CATransition *animation = [CATransitionanimation];

animation.duration = 0.5f;

animation.timingFunction = UIViewAnimationCurveEaseInOut;

animation.fillMode = kCAFillModeForwards;

animation.type = kCATransitionMoveIn;

animation.subtype = kCATransitionFromTop;

[self.window.layeraddAnimation:animationforKey:@"animation"];

```

// 自己的操作

1.3、UIView animateWithDuration

方法: +(void)animateWithDuration:(NSTimeInterval)duration animations:(void (^)(void))animations;
+ (void)animateWithDuration:(NSTimeInterval)duration animations:(void (^)(void))animations completion:(void (^)(BOOL finished))completion; //多一个动画结束后可以执行的操作。

```

[UIView animateWithDuration:1.25 animations:^(
CGAffineTransform newTransform = CGAffineTransformMakeScale(1.2, 1.2);
[firstImageView setTransform:newTransform];
[secondImageView setTransform:newTransform];}
completion:^(BOOL finished){
[UIView animateWithDuration:1.2 animations:^(
//自己的操作} completion:^(BOOL finished){ 自己的操作}}];

```

16,定制化view

需要自己继承自cocoa touch提供的丰富的类。如（UIView, UIScrollView, UITableView等等）。需要重载实现drawRect, touch事件, init, initWithFrame等方法。

17.机型和尺寸的适配

Iphone 的主要尺寸是3.5和4英寸。分辨率为：320480,480960（retina）。

IPad 主要尺寸是7.9和9.7英寸。分辨率为：1024768, 20481536（retina）。

18.添加手势的方式（gesture和touches事件）

1.自己重载实现touchMoved, touchBegin, touchEnd, touchCanceled事件。

2.通过UIGestureRecognizer添加AddGestureRecognizer事件。该方式方便添加一些诸如点击，双击，拖动等基本的手势事件。

19.应用程序的生命周期和状态

(参照：<http://blog.csdn.net/totogo2010/article/details/8048652>)

Not running 未运行 程序没启动 **Inactive** 未激活 程序在前台运行，不过没有接收到事件。在没有事件处理情况下程序通常停留在这个状态 **Active** 激活 程序在前台运行而且接收到了事件。这也是前台的一个正常的模式 **Backgroud** 后台 程序在后台而且能执行代码，大多数程序进入这个状态后会在这个状态上停留一会。时间到之后会进入挂起状态(**Suspended**)。有的程序经过特殊的请求后可以长期处于Backgroud状态 **Suspended** 挂起 程序在后台不能执行代码。系统会自动把程序变成这个状态而且不会发出通知。当挂起时，程序还是停留在内存中的，当系统内存低时，系统就把挂起的程序清除掉，为前台程序提供更多的内存。 下图是程序状态变化图：

各个程序运行状态时代理的回调：

```
- (BOOL)application:(UIApplication *)application willFinishLaunchingWithOptions:(NSDictionary *)launchOptions
    告诉代理进程启动但还没进入状态保存
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
    告诉代理启动基本完成程序准备开始运行
- (void)applicationWillResignActive:(UIApplication *)application
    当应用程序将要入非活动状态执行，在此期间，应用程序不接收消息或事件，比如来电话了
- (void)applicationDidBecomeActive:(UIApplication *)application
    当应用程序入活动状态执行，这个刚好跟上面那个方法相反
- (void)applicationDidEnterBackground:(UIApplication *)application
    当程序被推送到后台的时候调用。所以要设置后台继续运行，则在这个函数里面设置即可
- (void)applicationWillEnterForeground:(UIApplication *)application
    当程序从后台将要重新回到前台时候调用，这个刚好跟上面的那个方法相反。
- (void)applicationWillTerminate:(UIApplication *)application
    当程序将要退出是被调用，通常是用来保存数据和一些退出前的清理工作。这个需要要设置UIApplicationExitsOnSuspend的键值。
- (void)applicationDidFinishLaunching:(UIApplication *)application
    当程序载入后执行
    加载应用程序进入前台
```

加载应用程序进入后台

20.block编程

Block 是一种具有匿名功能的内嵌函数块。**Block** 一般是用来表示、简化一小段的程式码，它特别适合用来建立一些同步执行的程式片段、封装一些小型的工作或是用来做为某一个工作完成时的回传呼叫（callback）。格式如下：^(传入参数列){行为主体};

21.常用的开源框架

网络框架：ASIHttpRequest, AFNetworking,cocoaHttpServer等。

进度条：SVProgressHUD,MBProgressHUD,

工具类：SSToolKit等。

分享类：ShareKit等

日志框架：log4j, cocoa lumberJack 等。

等等。

22.通知消息和代理的区别

通知：分为本地和远程通知。接受通知的接受者需要进行注册改通知。这样通知被NSNotificationCenter发送出来后会被注册的接受者所接受。远程通知需要借助苹果的服务器去实现通知的中转。

代理：把某个对象要做的事情委托给别的对象去做。

两者区别：

delegate针对one-to-one关系，用于sender接受到reciever的某个功能反馈值。

notification针对one-to-one/many/none,reciver,用于通知多个object某个事件，sender只是负责把notification发送出去。

23.数据解析（json和XML）

json数据的解析通常借助一些开源的框架如：SBJson，TouchJson,jsonKit,Apple 提供的原生的JSON解析 NSJSON Serialization等。去json数据转化为IOS中常用的字典等。

XML数据的解析。xml分为SAX和DOM两种解析方式。

DOM解析XML时，读入整个XML文档并构建一个驻留内存的树结构（节点树），通过遍历树结构可以检索任意XML节点，读取它的属性和值。而且通常情况下，可以借助XPath，直接查询XML节点。

SAX解析XML，是基于事件通知的模式，一边读取XML文档一边处理，不必等整个文档加载完之后才采取操作，当在读取解析过程中遇到需要处理的对象，会发出通知对其进行处理。

一般在iOS平台下，比较常用的XML解析类库有如下几种：

NSXMLParser，这是一个SAX方式解析XML的类库，默认包含在iOS SDK中，使用也比较简单。

libxml2，是一套默认包含在iOS SDK中的开源类库，它是基于C语言的API，所以使用起来可能不如NSXML方便。这套类库同时支持DOM和SAX解析，libxml2的SAX解析方式还是非常酷的，因为它可以边读取边解析，尤其是在从网上下载一个很大的XML文件，就可以一边下载一边对已经下载好的内容进行解析，极大的提高解析效率。

TBXML，这是一套轻量级的DOM方式的XML解析类库，有很好的性能和低内存占用，不过它不对XML格式进行校验，不支持XPath，并且只支持解析，不支持对XML进行修改。

TouchXML，这也是一套DOM方式的XML解析类库，支持XPath，不支持XML的修改。

KissXML，这是一套基于TouchXML的XML解析类库，和TouchXML相比，支持了XML的修改。

TinyXML，这是一套小巧的基于C语言的DOM方式进行XML解析的类库，支持对XML的读取和修改，不直接支持XPath，需要借助另一个相关的类库TinyXPath才可以支持XPath。

GDataXML，这是一套Google开发的DOM方式XML解析类库，支持读取和修改XML文档，支持XPath方式查询。

24.webservice

Web service是一个平台独立的，松耦合的，自包含的、基于可编程的web的应用程序，可使用开放的XML标准来描述、发布、发现、协调和配置这些应用程序，用于开发分布式的互操作的应用程序。技术支持包含如下：

1.1 xml 和xsd

1.2 Soap

1.3wsdl

1.4 uddi

1.5调用RPC和消息传递

25.开发App的步骤，开发者账号，发布app到appstore

证书分两种：开发者证书、发布者证书。前者开发时使用，后者发布使用

（1） 模拟器调试无需代码签名；真机调试需开发者证书代码签名；发布时需发布证书签名

（2） 代码签名需要：证书+私钥，

（3） 真机调试时要求在设备上安装描述文件（provision profile），该文件包含信息：调试者证书，

授权调试设备清单，应用ID。一个应用对应一个描述文件。

26.类继承，类的扩展（extension），类别（category）

category 可以在不获悉，不改变原来代码的情况下往里面添加新的方法，只能添加，不能删除修改。并且如果类别和原来类中的方法产生名称冲突，则类别将覆盖原来的方法，因为类别具有更高的优先级。类别主要有3个作用：(1)将类的实现分散到多个不同文件或多个不同框架中。(2)创建对私有方法的前向引用。(3)向对象添加非正式协议。继承可以增加，修改或者删除方法，并且可以增加属性。

category和extensions的不同在于后者可以添加属性。另外后者添加的方法是必须要实现的。 extensions可以认为是一个私有的Category。

27.CALayer介绍

一个UIView包含CALayer树，CALayer是一个数据模型。包含了一些用来显示的对象，在UIView的子类中都可以找到层这个组件，层是位于固定的画布上的一个子片，可以被覆盖。层是彼此堆叠在一起的最终产生一个界面。除此之外层可以包含多个层，通过层可以操作位于此层上面的其他内容，例如旋转，动画，翻页等。

28.ios 怎么实现多继承

IOS通过实现protocol委托代理，实现多个接口来实现多继承。

29.app性能测试方式

通过Xcode提供的工具如Instrument，测试CPU，Memory性能。也可以适用一些开源的自动化测试工具：如Frank，KIF等。

30.NSArray可以放基本数据类型不（int，float，nil）怎么放进一个结构体

NSArray 只能存放objective - c对象数据模型，这些基本数据类型需要先转化为NSNumber对象再存放在数组中。

31.objective-c和c，c++混合编写

在 Objective-C++中，可以用C++代码调用方法也可以从Objective-C调用方法。在这两种语言里对象都是指针，可以在任何地方使用。例如，C++类可以使用Objective-C对象的指针作为数据成员，Objective-C类也可以有C++对象指针做实例变量。Xcode需要源文件以".mm"为扩展名，这样才能启动编译器的Objective-C++扩展。

32.常见的语言编码(utf-8,unicode,gb2312,gbk)

常见的语言编码有：

GB2312:简体中文编码，一个汉字占用2字节，在大陆是主要编码方式。

BIG5:繁体中文编码。主要在台湾地区采用。

GBK:支持简体及繁体中文，但对他国非拉丁字母语言还是有问题。

UTF-8:Unicode编码的一种。Unicode用一些基本的保留字符制定了三套编码方式，它们分别UTF-8,UTF-16和UTF-32。在UTF - 8中，字符是以8位序列来编码的，用一个或几个字节来表示一个字符。这种方式的最大好处，是UTF - 8保留了ASCII字符的编码做为它的一部分。UTF-8俗称“万国码”，可以同屏显示多语种，一个汉字占用3字节。为了做到国际化，网页应尽可能采用UTF-8编码。

当然，处理中文时http头也要改成UTF-8编码的-----加上。

语言 字符集 正式名称

英语、西欧语 ASCII，ISO-8859-1 MBCS多字节

简体中文 GB2312 MBCS多字节

繁体中文 BIG5 MBCS多字节

简繁体中文 GBK MBCS多字节

中文、日文及朝鲜语 GB18030 MBCS多字节

各国语言 UNICODE，UCS DBCS宽字节

33.常见的加解密方式(rsa,aes,md5)

常见的加解密方式有：

RSA：基于公钥和私钥的非对程加密算法。适用范围广。

AES：是一种对程加密的流行方式。加密涉及矩阵运算。

MD5:将任意长度的“字节串”变换成一个128bit的大整数，并且它是一个不可逆的字符串变换算法，

34.objective – c语言的优缺点

objc优点：

1) Categories 2) Posing 3) 动态识别 4) 指标计算 5) 弹性讯息传递 6) 不是一个过度复杂的 C 衍生语言 7) Objective-C 与 C++ 可混合编程 缺点: 1) 不支援命名空间 2) 不支持运算符重载

3) 不支持多重继承

4) 使用动态运行时类型，所有的方法都是函数调用，所以很多编译时优化方法都用不到。（如内联函数等），性能低劣。

35，ios应用的调试技巧

1.如遇到crash，分析崩溃日志（symbolicatedrash工具的适用）保留崩溃版本的.dSYM文件

2.在 XCode 中进入断点管理窗口；然后点击右下方的 +，增加新的 Exception Breakpoint。

3.如遇到EXC_BAD_ACCESS，打开Scheme选项选择EditScheme。然后按图勾上Enable Zombie Objects和Malloc Stack那两项。

4.有效的日志管理。NSLog和加入一些开源的日志管理框架。

5.程序断点debug模式。

36，应用程序性能的调优

（转<http://www.open-open.com/lib/view/open1365861753734.html>）

1.用ARC去管理内存（Use ARC to Manage Memory）

2.适当的地方使用reuseIdentifier（Use a reuseIdentifier Where Appropriate）

3.尽可能设置视图为不透明（Set View as Opaque When Possible）

4.避免臃肿的XIBs文件（Avoid Fat XIBs）

5.不要阻塞主进程（Don't Block the Main Thread）

6.调整图像视图中的图像尺寸（Size Images to Image Views）

7.选择正确集合（Choose the Correct Collection）

8.启用Gzip压缩（Enable GZIP Compression）

9.重用和延迟加载视图（Reuse and Lazy Load Views）

10.缓存，缓存，缓存（Cache,Cache,Cache）

11.考虑绘图（Consider Drawing）

12.处理内存警告（Handle Memory Warnings）

13.重用大开销对象（Reuse Expensive Objects）

14.使用精灵表（Use Sprite Sheets ）

15.避免重复处理数据（Avoid Re-Processing Data）

16.选择正确的数据格式（Choose the Right Data Format）

17.适当的设置背景图片（Set Background Images Appropriately）

18.减少你的网络占用（Reduce Your Web Footprint）

19.设置阴影路径（Set the Shadow Path ）

20.你的表格视图Optimize Your Table Views）

21.选择正确的数据存储方式（Choose Correct Data Storage Option）

22.加速启动时间（Speed up Launch Time ）

23.使用自动释放池（Use AutoRelease Pool）

24.缓存图像（Cache Images-Or not）

25.尽可能避免日期格式化器（Avoid Date Formatters Where Possible）

37.UIScrollView 的contentSize、contentOffset和contentInset属性的区别

contentSize表示UIScrollView滚动区域的大小。UIScrollView的frame属性在设置好了以后不会随内容的变化而变化。

contentOffset表示是UIScrollView当前显示区域顶点相对于frame顶点的偏移量，一般用来设置UIScrollView显示的位置。

contentInset表示是scrollview的contentView的顶点相对于scrollview的位置，假设你的contentInset = (0 ,100)，那么你的contentView就是从scrollview的(0 ,100)开始显示。一般都是（0， 0）表示从scrollView的开始显示。

38.iOS6 AutoLayout

AutoLayout是iOS6之后引进的自动布局功能，有点类型有android的相对布局属性。通过勾选AutoLayout设置各种Constraint约束来实现在不同设备和不同方向上的自动布局。autosizing mask也就是“springs and struts”模式。autosizing mask决定了一个view会发生什么当它的superview 改变大小的时候。而autolayout 不仅可以设置superview改变时view所做的变化，还支持当相邻view变化时自己所做的变化。