1. 简述你对UIView、UIWindow和CALayer的理解  
   UIView继承于UIResponder, UIResponder继承于NSObject,UIView可以响应用户事件。CALayer继承于NSObject，所以CALayer不能响应事件。  
   UIView构建界面,UIView侧重于对内容的管理，CALayer侧重于对内容的绘制。  
   UIView是用来显示内容的，可以处理用户事件；CALayer是用来绘制内容的，对内容进行动画处理，依赖与UIView来进行显示，不能处理用户事件。
2. 写一个完整的代理，包括声明，实现  
   略
3. 分析json、xml的区别？json、xml解析方式的底层是如何处理的？  
   json底层原理遍历字符串中的字符，最终根据各市规定的特助字符，比如{}，[],:号等进行区分，{}是字典，[]表示的时数组，：号是字典的键和值的分水岭，最总是将json数据转化为字典。Xml两种解析方式，DOM和SAX,DOM需要读入整个XML文档（文档驱动），SAX是事件驱动的，并不需要读入整个文档，文档的读入过程也就是SAX的解析过程。
4. ViewController 的 didReceiveMemoryWarning 是在什么时候被调用的？默认的操作是什么?  
   didReceiveMemoryWarning在出现内存警告的时候执行该方法，在该方法里面释放掉暂时没使用的可重用的对象。

if([self isViewLoaded] == YES && self.view.window == nil)

{

self.view = nil;

}

1. 面向对象的三大特征，并作简单的介绍  
   封装、继承、多态。封装：是把客观事物封装成抽象的类，隐藏内部的实现，对外部提供接口。继承：可以使用现有类的所有功能，并且在无需重新编写原来的类的情况下对这些功能进行扩展。多态：不同的对象以自己的方式响应相同的的消息的能力叫做多态，或者说父类指针指向子类对象<如UITableView的，cellForRow方法，返回值类型是UITbaleViewCell，但是你返回的cell可以是你自定义的cell,在比如多个类里面都有同一个方法>
2. 重写一个NSStrng类型的，retain方式声明name属性的setter和getter方法
3. 简述NotificationCenter、KVC、KVO、Delegate？并说明它们之间的区别？  
   Notification：观察者模式，controller向defaultNotificationCenter添加自己的 notification，其他类注册这个notification就可以收到通知，这些类可以在收到通知时做自己的操作（多观察者默认随机顺序发通知给 观察者们，而且每个观察者都要等当前的某个观察者的操作做完才能轮到他来操作，可以用NotificationQueue的方式安排观察者的反应顺序，也 可以在添加观察者中设定反映时间，取消观察需要在viewDidUnload 跟dealloc中都要注销）。

KVC键值编码，可以直接通过字符串的名字（key）来间接访问属性的机制，而不是通过调用getter和setter方法访问。  
KVO：观测指定对象的属性，当指定对象的属性更改之后会通知相应的观察者。  
delegate:一对一，delegate遵循某个协议并实现协议声明的方法。

1. What is lazy loading?  
   在使用的时候才去初始化，比如UITableViewCell的imageView属性，懒加载对象的创建是在getter方法里面进行创建的。
2. 什么是Protocol？什么是代理？写一个委托的interface？委托的property声明用什么属性？为什么？  
   协议提供了一组方法，但是并不负责实现，如果一个类遵循了某个协议，并且实现了协议里面的方法，那么我们称这个类就是遵循了某个协议的代理。属性的声明使用assign，防止出现循环引用的问题。
3. 分别描述类别（categories）和延展（extensions）是什么？以及两者的区别？继承和类别在实现中有何区别？为什么Category只能为对象添加方法，却不能添加成员变量？  
   category类目：在不知道源码的情况下为一个类扩展方法，extension：为一个类声明私有方法和变量。  
   继承是创建了一个新的类，而类别只是对类的一个扩展，还是之前的类。  
   类目的作用就是为已知的类添加方法。