iOS有三种基本的界面布局的方法

分别是手写UI，xib和storyboard。手写UI是最早进行UI界面布局的方法，优点是灵活自由，缺点是需要写大段的代码进行布局。xib也是比较早出现的UI布局的方式，优点是不需要手写代码，但是每个界面对应一个xib，管理起来复杂。而storyboard则是在iOS5以后出现的，是苹果官方主推的一个代替xib的策略，不仅能将xib汇总统一管理，还可以描述各种场景之间的过渡，缺点是多人协作开发时容易产生冲突。

UIViewController的布局过程

VC的生命周期的部分过程viewDidLoad -> viewWillAppear -> updateViewConstraints -> viewWillLayoutSubviews -> viewDidLayoutSubviews -> viewDidAppear -> viewWillDisAppear -> updateViewConstraints -> viewDidDisAppear

UIViewController在有个updateViewConstraints 方法，这个方法实际是self.view 被设置了setNeedsUpdateConstraints（第一次展示的时候），必然会调用这个方法（ 第一次可以理解为为self.view增加了各种约束）。而这个方法的默认实现是调用子view的updateConstraints方法，这样就自上而下的完成了布局。

对应updateConstraints -> layoutSubViews -> drawRect

注意:

1.不要忘记调用父类的方法，避免有时候出现一些莫名的问题。

2.在view的layoutSubViews或者ViewController的viewDidLayoutSubviews方法里后可以拿到view的实际frame，所以当我们真的需要frame的时候需要在这个时间点以后才能拿到。

3.下面我们可以解释是为什么viewDidLoad里通过setFrame的方式改过原先在storyboard里拖动的约束代码无效了。因为updateViewConstraints在viewDidLoad后执行，会覆盖掉之前的设置的frame，所以无效。

## 补充总结

#### 以下情况下会调用layoutSubviews：

* init初始化不会触发layoutSubviews，但是是用initWithFrame 进行初始化时，当rect的值不为CGRectZero时,也会触发。
* addSubview会触发layoutSubviews
* 设置view的Frame会触发layoutSubviews，当然前提是frame的值设置前后发生了变化
* 滚动一个UIScrollView会触发layoutSubviews
* 旋转Screen会触发父UIView上的layoutSubviews事件
* 改变一个UIView大小的时候也会触发父UIView上的layoutSubviews事件

#### 刷新子对象布局

-layoutSubviews方法：这个方法，默认没有做任何事情，需要子类进行重写  
 -setNeedsLayout方法： 标记为需要重新布局，异步调用layoutIfNeeded刷新布局，不立即刷新，但layoutSubviews一定会被调用  
 -layoutIfNeeded方法：如果，有需要刷新的标记，立即调用layoutSubviews进行布局（如果没有标记，不会调用layoutSubviews）如果要立即刷新，要先调用[view setNeedsLayout]，把标记设为需要布局，然后马上调用[view layoutIfNeeded]，实现布局在视图第一次显示之前，标记总是“需要刷新”的，可以直接调用[view layoutIfNeeded]

#### 重绘

-drawRect:(CGRect)rect方法：重写此方法，执行重绘任务

-setNeedsDisplay方法：标记为需要重绘，异步调用drawRect

-setNeedsDisplayInRect:(CGRect)invalidRect方法：标记为需要局部重绘

sizeToFit会自动调用sizeThatFits方法；

sizeToFit不应该在子类中被重写，应该重写sizeThatFits

sizeThatFits传入的参数是receiver当前的size，返回一个适合的sizesizeToFit可以被手动直接调用sizeToFit和sizeThatFits方法都没有递归，对subviews也不负责，只负责自己

layoutSubviews对subviews重新布局

layoutSubviews方法调用先于drawRectsetNeedsLayout**在receiver标上一个需要被重新布局的标记，在系统runloop的下一个周期自动调用layoutSubviews**

layoutIfNeeded方法如其名，UIKit会判断该receiver是否需要layout，根据Apple官方文档,layoutIfNeeded方法应该是这样的layoutIfNeeded遍历的不是superview链，应该是subviews链drawRect是对receiver的重绘，能获得context。

约束方式：

1.autorising

2.VFL（可视化格式语言）

3.autolayout（xib或storyboard，masonry）

当一个视图控制器被创建，并在屏幕上显示的时候。 代码的执行顺序

1、alloc 创建对象，分配空间

2、init (initWithNibName) 初始化对象，初始化数据

3、loadView 从nib载入视图 ，通常这一步不需要去干涉。除非你没有使用xib文件创建视图

4、viewDidLoad 载入完成，可以进行自定义数据以及动态创建其他控件

5、viewWillAppear 视图将出现在屏幕之前，马上这个视图就会被展现在屏幕上了

6、viewDidAppear 视图已在屏幕上渲染完成

当一个视图被移除屏幕并且销毁的时候的执行顺序，这个顺序差不多和上面的相反

1、viewWillDisappear 视图将被从屏幕上移除之前执行

2、viewDidDisappear 视图已经被从屏幕上移除，用户看不到这个视图了

3、dealloc 视图被销毁，此处需要对你在init和viewDidLoad中创建的对象进行释放