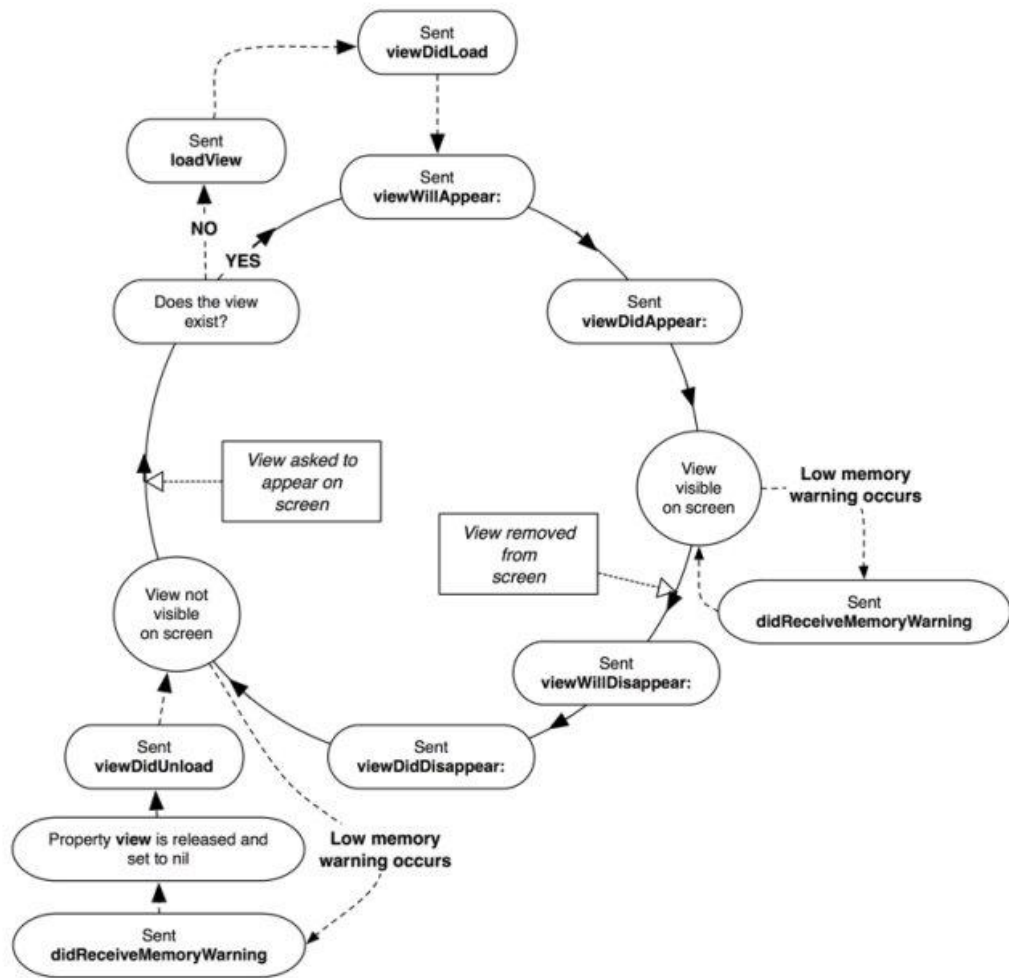


1. UIView 的生命循环周期



loadView: 永远不要主动调用这个函数。view controller 会在 view 的 property 被请求并且当前 view 值为 nil 时调用这个函数。如果你手动创建 view，你应该重载这个函数,且 不要在重载的时候调用[super loadview]。

viewDidLoad: 这个函数的作用主要是让你可以进一步的初始化你的 views。viewDidLoad 通常负责的是 view 及其子 view 被加载进内存之后的数据初始化的工作，即视图的数据部分的初始化

viewDidUnLoad: 这个函数是 viewDidLoad 的对立函数。在程序内存欠缺时，这个函数被 controller 调用，来释放它的 view 以及 view 相关的对象。由于 controller 通常保存着 view 以及相关 object 的引用，所以你必须使用这个函数来放弃这些对象的所有权以便内存回收。但不要释放那些难以重建的数据

viewWillAppear: 视图即将可见时调用。默认情况下不执行任何操作

viewDidAppear: 视图已完全过渡到屏幕上时调用

viewWillDisappear: 视图被驳回时调用，覆盖或以其他方式隐藏。默认情况下不执行任何操作

viewDidDisappear: 视图被驳回后调用，覆盖或以其他方式隐藏。默认情况下不执行任何操作

didReceiveMemoryWarning: 当程序内存过度时，系统会调用该方法

2.关于 UIView 生命周期的建议

- a.尽量不要在 `viewWillAppear` 里面写代码，因为 ios 较低版本没有这个方法
- b.尽量实现 `viewDidLoad`，因为你需要在里面进行内存的释放
- c.除了实现这些方法外，你应该重载 `dealloc` 方法，进行内存释放