

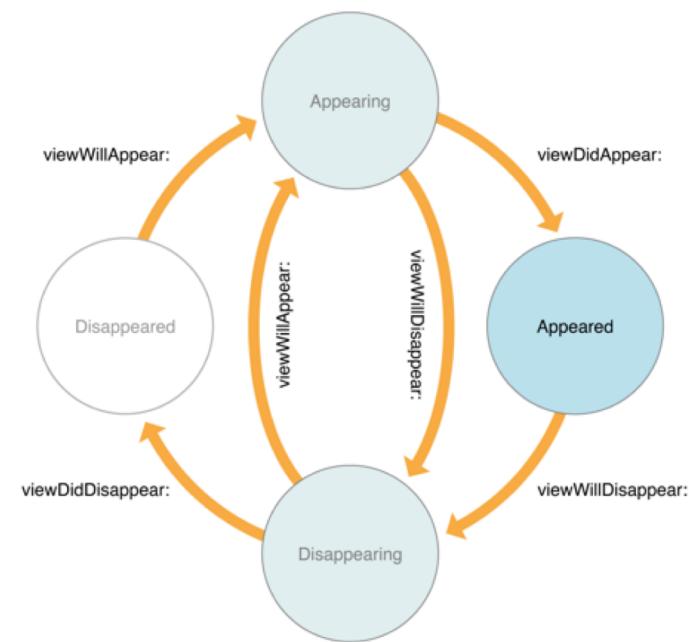


# iOS Life Cycle

## ADDITIONAL READING MATERIALS

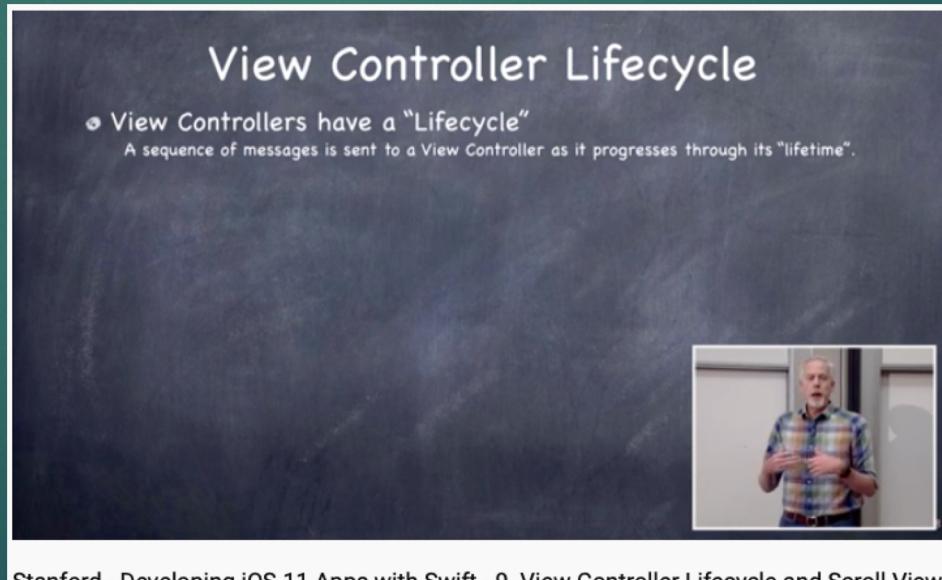
- ▶ When the visibility of its views changes, a view controller automatically calls its own methods so that subclasses can respond to the change. Use a method like `viewWillAppear(_:)` to prepare your views to appear onscreen, and use the `viewWillDisappear(_:)` to save changes or other state information. Use other methods to make appropriate changes.

Figure 1 Valid State Transitions



# ViewController Life Cycle

- ▶ Stanford - Developing iOS 11 Apps with Swift - 9. View Controller Lifecycle and Scroll View
- ▶ <https://youtu.be/B281mrPUGjg>



# Super?

```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view.
}

override func viewWillAppear(_ animated: Bool) {
    super.viewWillAppear(false)
}

override func viewWillDisappear(_ animated: Bool) {
    super.viewWillAppear(false)
}
```

- ▶ Always include `super` to ensure that any code in the super class gets called before executing any additional code.

For the override functions, you must call the super implementation to give parent classes the opportunity to perform any additional initialization that they require. Although the default implementation of this method does nothing, many UIKit classes provide non-empty implementations.

## Apple's Document

This method is called before the receiver's view is about to be added to a view hierarchy and before any animations are configured for showing the view. You can override this method to perform custom tasks associated with displaying the view. For example, you might use this method to change the orientation or style of the status bar to coordinate with the orientation or style of the view being presented. **If you override this method, you must call super at some point in your implementation.**

# Android vs iOS Life Cycle

