

# SWIFTY VIEW CONTROLLER PRESENTERS

JESSE SQUIRES

[JESSESQUIRES.COM](http://JESSESQUIRES.COM) • [@JESSE\\_SQUIRES](https://twitter.com/JESSE_SQUIRES) • [GITHUB/JESSESQUIRES](https://github.com/JESSESQUIRES)

@swiflybrief

# VIEW CONTROLLERS

# EVERWHERE

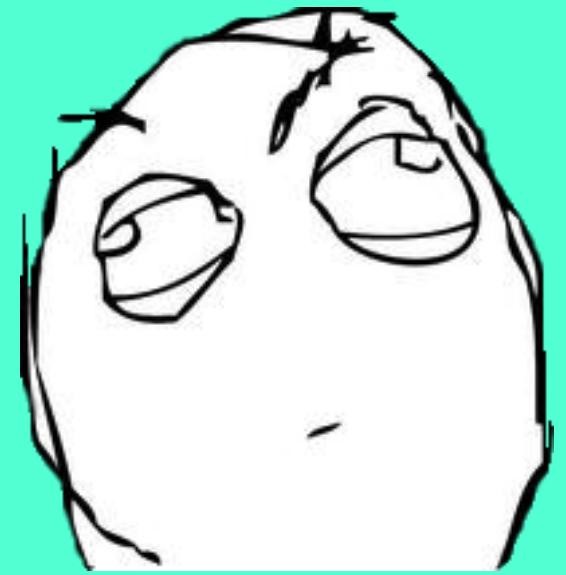
# PRESENTATION CONTROLLERS

# EVERYWHERE.

# PRESENTATION

VS.

# TRANSITION



What's the difference?

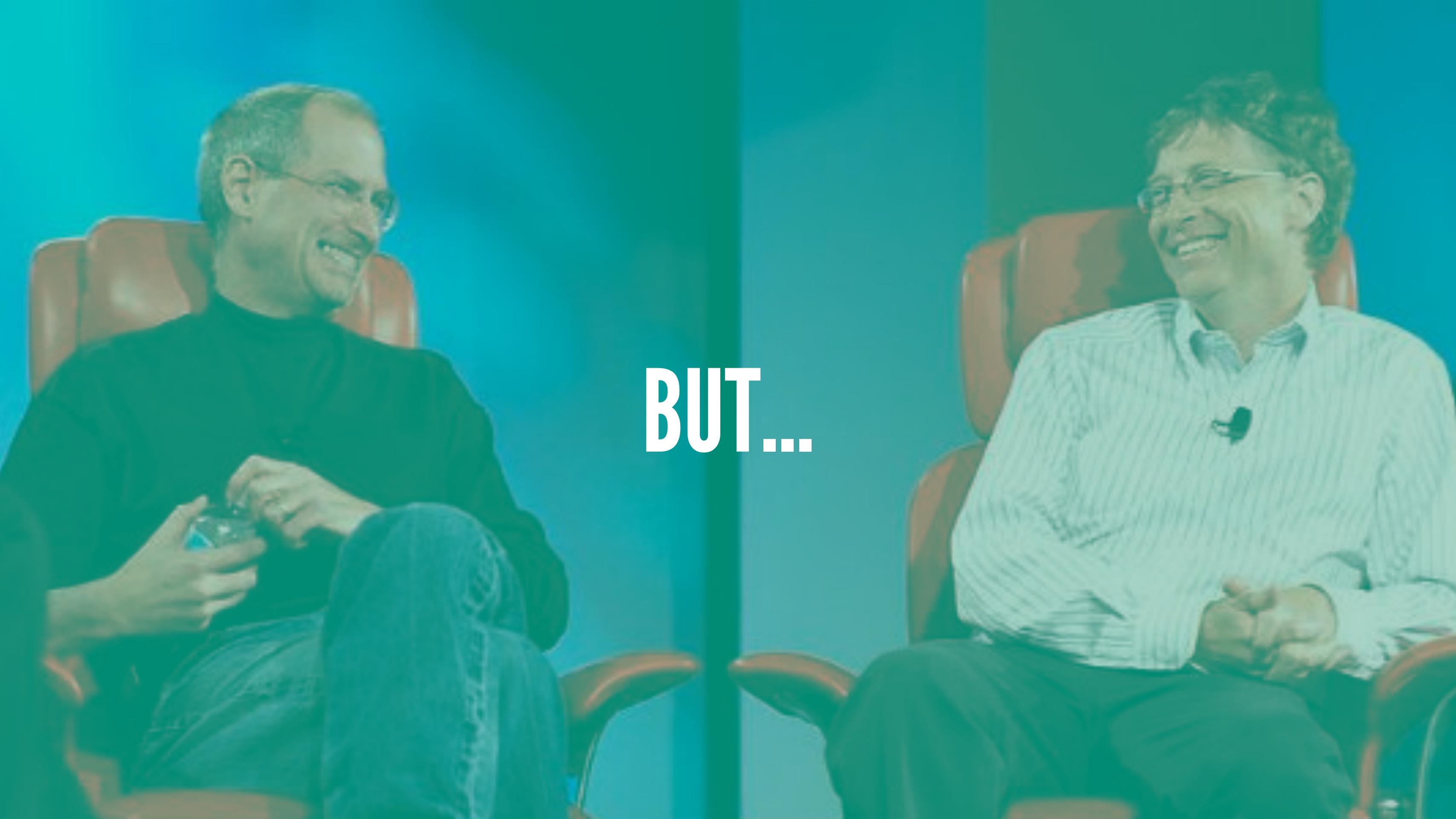
# TRANSITION

the moving  
(or animating)  
from one view controller  
to another view controller

# PRESENTATION

the ending visual state  
(or final appearance)  
of a view controller

after the transition

A photograph of two men sitting in brown leather armchairs, laughing heartily. The man on the left is wearing a dark green t-shirt and jeans, holding a small blue object. The man on the right is wearing a light green striped shirt and glasses. They are positioned on either side of a vertical white line.

BUT...

# TRANSITION VS. PRESENTATION

Concepts are very interrelated.

Presentation API hooks into the transitioning API.

Slightly coupled in this way, but otherwise nicely decoupled.

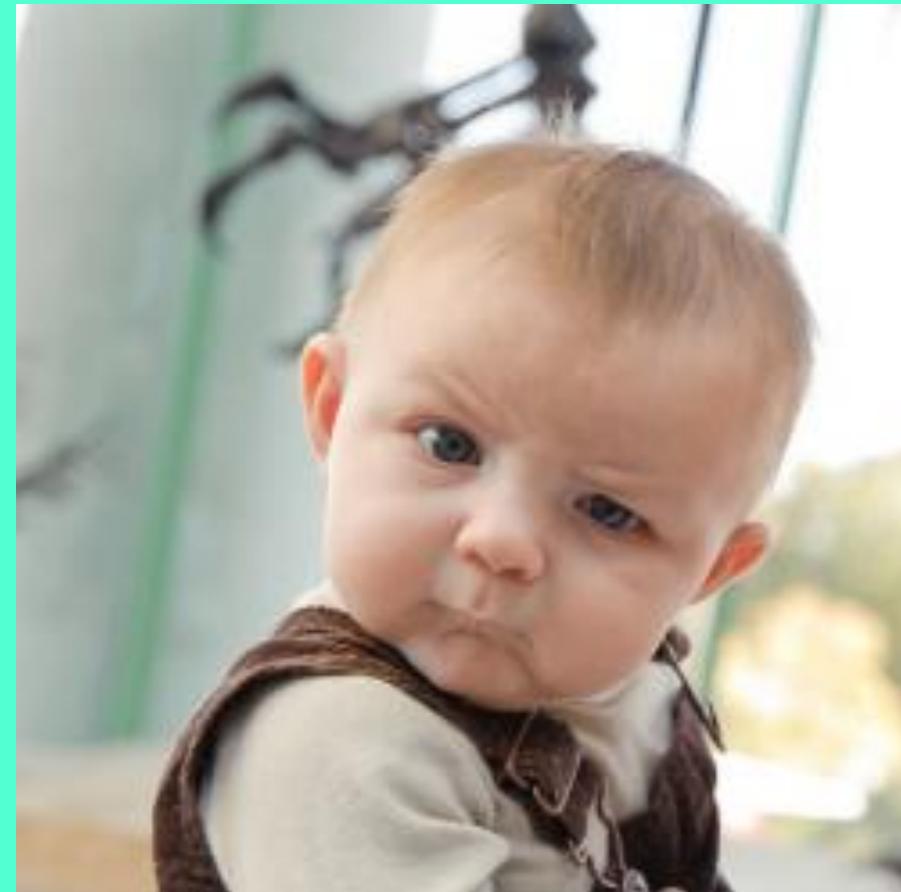
# PRESENTATION

UIPresentationController

iOS 8+

# UIPresentationController

**WAT?**



# UIPresentationController

The APIs are very subtle and usually implicit.

You often use them without knowing it.

Typically overlooked.

Previously, relevant on iPad only.

# UNDERSTANDING THE PRESENTATION API

# RESPONSIBILITIES

- ▶ Positioning
- ▶ Content and "Chrome"
  - ▶ Adaption
  - ▶ Animation

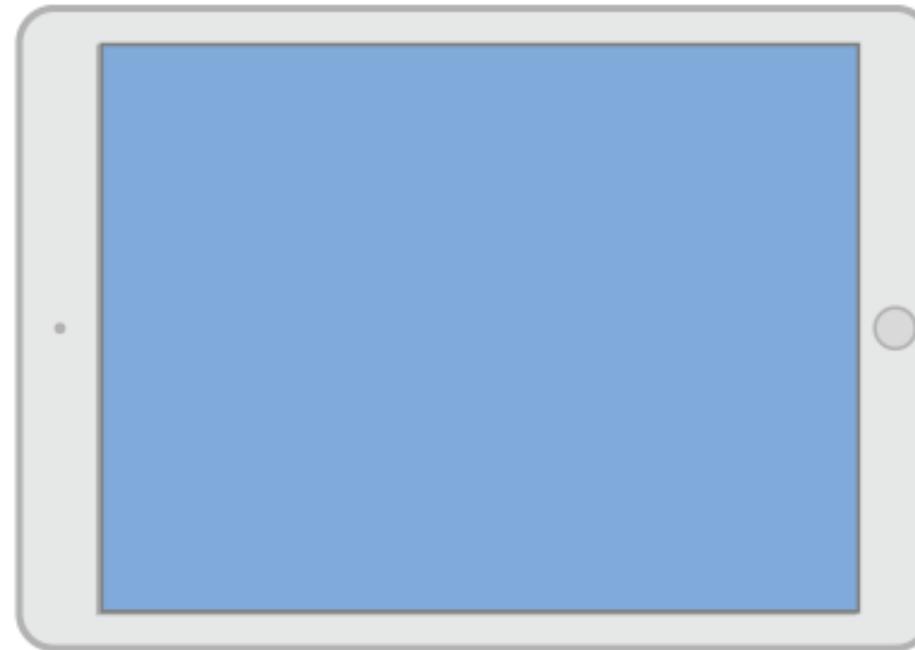
Intended to be reusable

# 3 PHASES

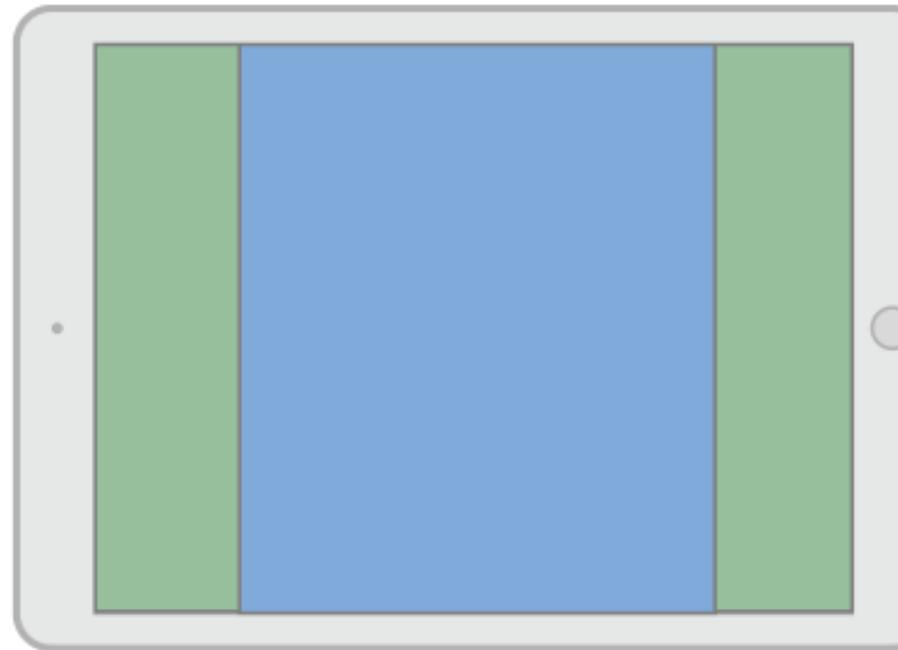
1. Presentation (moving onscreen)\*
2. Management (size class changes, device rotation)
3. Dismissal (moving offscreen)\*

\*Transitioning

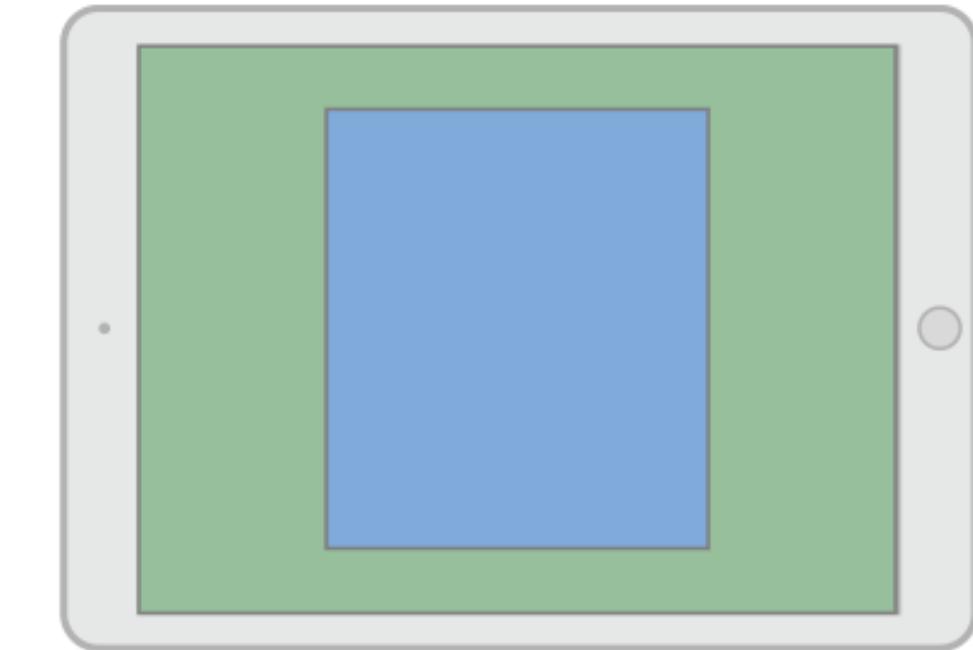
# EXAMPLES



UIModalPresentationFullscreen



UIModalPresentationPageSheet



UIModalPresentationFormSheet

&lt; General

## Reset

[Reset All Settings](#)[Erase All Content and Settings](#)[Reset Network Settings](#)[Reset Keyboard Dictionary](#)[Reset Home Screen Layout](#)[Reset Location & Privacy](#)

This will reset all settings. No data or media will be deleted.

**Reset All Settings****Cancel**

&lt; ⓘ

AA Q B



**A** A A

Swift  
for iO

that  
Objec

C con  
prog

fea  
more

clean slate, backed by the mature and  
much-loved Cocoa and Cocoa Touch  
frameworks, is an opportunity to  
reimagine how software development  
works.

Swift has been years in the making.  
Apple laid the foundation for Swift by  
advancing our existing compiler,  
debugger, and framework infrastructure.

&lt; Notes



Done

Undo Typing

Cancel

Undo

[Cancel](#)[iCloud](#)[Next](#)

example@icloud.com

Required

[Forgot Apple ID or Password?](#)[Create a new Apple ID](#)

An Apple ID is the login you use for just about everything you do with Apple.

[About Apple ID & Privacy...](#)[Cancel](#)[iCloud](#)[Next](#)

example@icloud.com

Required

[Forgot Apple ID or Password?](#)[Create a new Apple ID](#)

An Apple ID is the login you use for just about everything you do with Apple.

[About Apple ID & Privacy...](#)

# UIPresentationController

## Subtle and Implicit

### UIModalPresentationStyle

```
// just set an enum!  
self.modalPresentationStyle = .FullScreen  
self.modalPresentationStyle = .Popover
```



**MAGIC**

# PRESENTING

```
func presentViewController(_, animated: , completion: )  
func pushViewController(_, animated: )  
func showViewController(_, sender: )  
func showDetailViewController(_, sender: )
```

# DISMISSING

```
func dismissViewControllerAnimated(_ completion:())
func popViewControllerAnimated(_ animated:) -> UIViewController?
```

*Special*  
**PRESENTATIONS**

# UIPresentationController

- ▶ UIAlertController
- ▶ UISearchController
- ▶ UIPopoverPresentationController (**explicit!**)

Unified, consistent APIs.

All view controllers with custom presentation controllers.

# UIPopoverPresentationController

**The only custom presentation controller publicly available in UIKit**

```
// Current UIViewController APIs  
self.presentationController  
self.popoverPresentationController
```

# PRIVATE API

```
// Modal styles: .FullScreen, .FormSheet, .PageSheet, etc.  
_UIFullscreenPresentationController  
  
// Alerts  
_UIAlertControllerAlertPresentationController  
  
// ActionSheet on iPhone  
_UIAlertControllerActionSheetCompactPresentationController  
  
// ActionSheet on iPad (popover)  
_UIAlertControllerActionSheetRegularPresentationController
```

# API SUMMARY

- ▶ 4 methods to display a view controller
- ▶ Set styles via an enum value
- ▶ Use popoverPresentationController directly
- ▶ Provide a custom Presentation Controller

The API is easy to misuse.

A photograph of two men sitting in theater-style seats, laughing heartily. The man on the left is wearing a dark green t-shirt and has his arm around the man next to him. The man on the right is wearing a light-colored striped shirt and glasses. They are both looking towards the left of the frame. The background is a solid teal color.

#SORRYNOTSORRY

# HOW TO USE CUSTOM PRESENTATION CONTROLLERS

# CUSTOM PRESENTATION CONTROLLERS

1. Subclass `UIPresentationController`
2. Vend controller via transitioning API



# 1. SUBCLASS: TRANSITIONING AND CHROME

```
func presentationTransitionWillBegin()
```

```
func dismissalTransitionWillBegin()
```

# 1. SUBCLASS: POSITIONING

```
func sizeForChildContentContainer(_, withParentContainerSize:) -> CGSize  
func frameOfPresentedViewInContainerView() -> CGRect
```

# 1. SUBCLASS: ADAPTIVITY

```
func adaptivePresentationStyle() -> UIModalPresentationStyle
```

```
func shouldPresentInFullscreen() -> Bool
```

```
func containerViewWillLayoutSubviews()
```

## 2. VEND CONTROLLER

UIViewControllerAnimatedTransitioningDelegate

Presenting view controller conforms to delegate

Set this delegate on the Presented view controller

```
let vc = MyViewController()  
vc.transitioningDelegate = self
```

## 2. VEND CONTROLLER

UIViewControllerAnimatedTransitioningDelegate

```
func presentationControllerForPresentedViewController(_,
presentingViewController: ,
sourceViewController:) -> UIPresentationController?
```

Demo

BETTER THAN VIEW  
CONTROLLER CONTAINMENT

# Let's build a micro-library

- ▶ Refine this API
- ▶ Make it *Swift*y

```
enum PresentationType {  
    case Modal(NavigationStyle, UIModalPresentationStyle, UIModalTransitionStyle)  
    case Popover(PopoverConfig)  
    case Push  
    case Show  
    case ShowDetail(NavigationStyle)  
    case Custom(UIViewControllerTransitioningDelegate)  
}
```

```
enum NavigationStyle {  
    case None  
    case WithNavigation  
}
```

```
struct PopoverConfig {  
    enum Source {  
        case BarButtonItem(UIBarButtonItem)  
        case View(UIView)  
    }  
  
    let source: Source  
    let arrowDirection: UIPopoverArrowDirection  
    let delegate: UIPopoverPresentationControllerDelegate?  
}
```

```
extension UIViewController {  
  
    // maps PresentationType to UIKit properties and methods  
    func presentViewController(  
        controller: UIViewController,  
        type: PresentationType,  
        animated: Bool = true)  
  
}
```

# USAGE

```
presentViewController(vc, type: .Push)
```

```
let config = PopoverConfig(source: .View(v))
presentViewController(vc, type: .Popover(config))
```

```
let type = .Modal(.WithNavigation, .FullScreen, .CoverVertical)
presentViewController(vc, type: type)
```

```
presentViewController(vc, type: .Custom(self))
```

# *Swifty* API

- ▶ More expressive
- ▶ More concise
- ▶ Single "entry point" to API
  - ▶ More structured  
(less likely to misuse)

## Additional resources:

- ▶ WWDC 2014 "A Look Inside Presentation Controllers"
- ▶ [pspdfkit.com/blog/2015/presentation-controllers/](http://pspdfkit.com/blog/2015/presentation-controllers/)
- ▶ [petersteinberger.com/blog/2015/uipresentationcontroller-popover-detection/](http://petersteinberger.com/blog/2015/uipresentationcontroller-popover-detection/)

*Thank you!*

# QUESTIONS?

JESSESQUIRES.COM • @JESSE\_SQUIRES • GITHUB/JESSESQUIRES