

Level 1

App Anatomy





In this level we'll cover

01 Starting an iPhone project

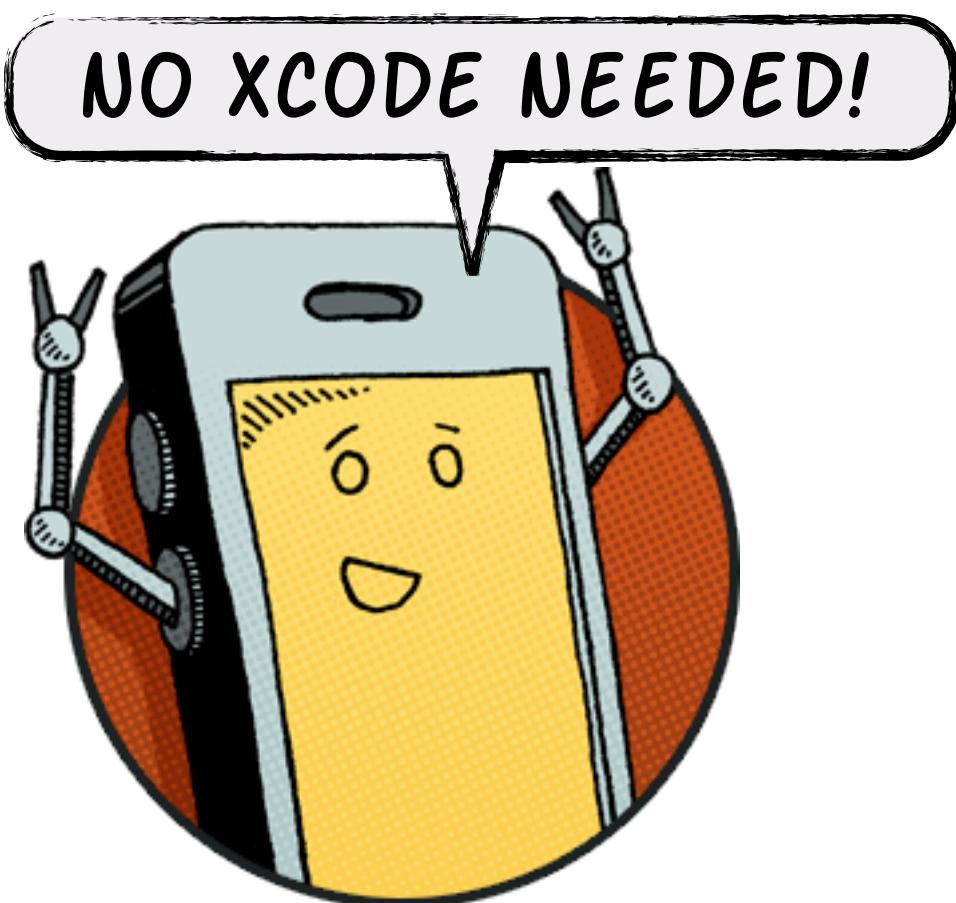
02 The Application Delegate

03 Your First iPhone App

04 How to Draw

05 Refactoring Our Code

06 Touch



MR. HIGGIE



Your First iOS Application

Set a background color

Click to change the transparency



Choose a template for your new project

The screenshot shows the 'Choose a template for your new project' screen. On the left, there's a sidebar with sections for iOS (Application, Framework & Library, Other) and OS X (Application, Framework & Library, Application Plug-in, System Plug-in, Other). The 'Application' section under iOS is highlighted. Below the sidebar, several project templates are listed with icons: Master-Detail Application (grid icon), OpenGL Game (game controller icon), Page-Based Application (document icon), Single View Application (square icon with a '1'), Tabbed Application (stacked square icon), Utility Application (wrench and screwdriver icon), and Empty Application (dashed square icon). A large orange arrow points upwards from the bottom of the 'Single View Application' section towards the 'Next' button. The 'Single View Application' section contains a sub-section with its own icon and text: '1 Single View Application'. Below this, a detailed description reads: 'This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a storyboard or nib file that contains the view.'

iOS

Application

Framework & Library

Other

OS X

Application

Framework & Library

Application Plug-in

System Plug-in

Other

Master-Detail Application

OpenGL Game

Page-Based Application

Single View Application

Tabbed Application

Utility Application

Empty Application

1 Single View Application

This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a storyboard or nib file that contains the view.

Cancel

Previous

Next

Choose options for your new project:



Product Name

Organization Name

Company Identifier

Bundle Identifier

Class Prefix

Devices

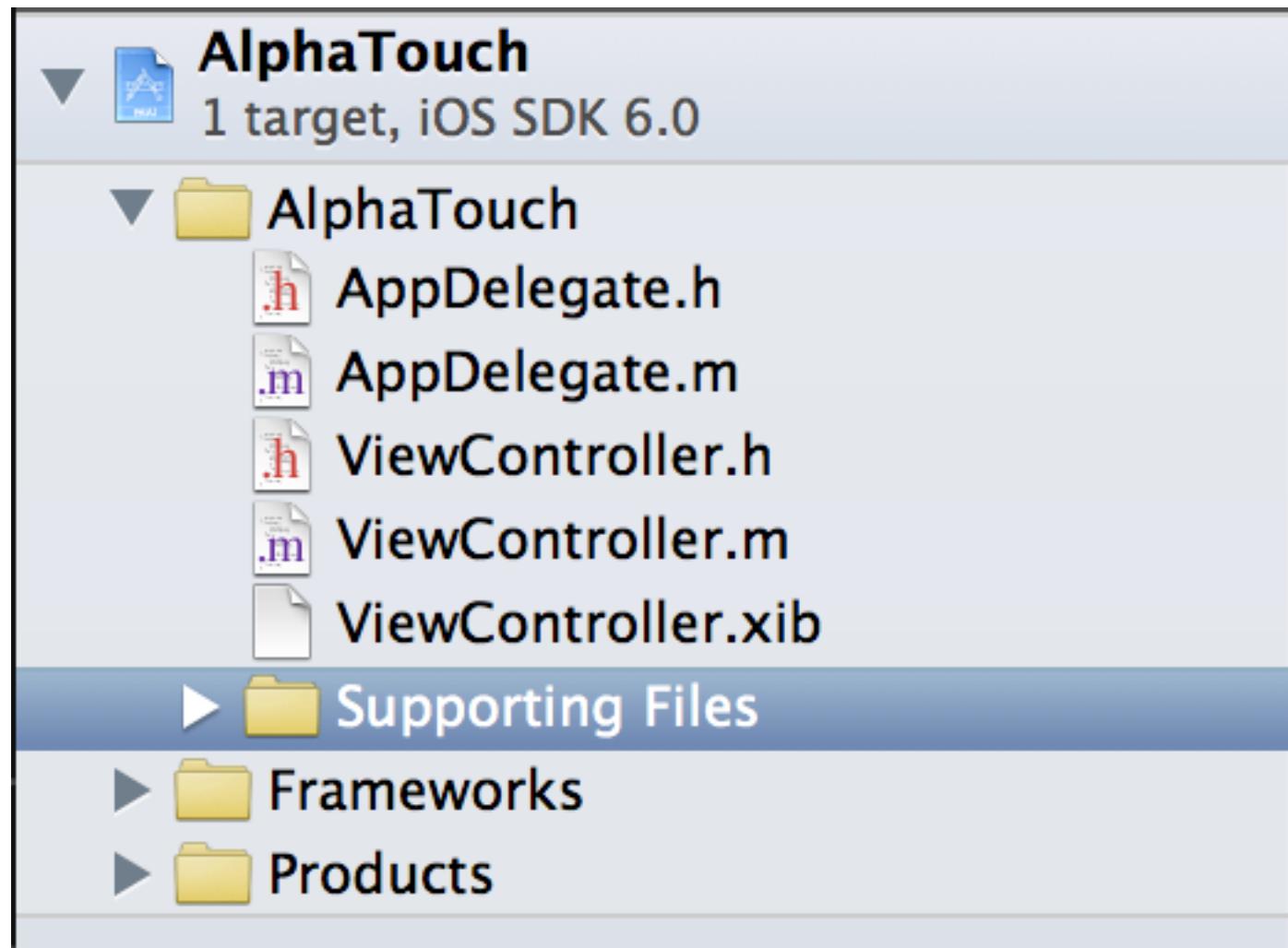
Use Storyboards

Use Automatic Reference Counting

Include Unit Tests

Try
iOS

Creates these files for you





AppDelegate.m

**Defines methods that will be called at different points
in your applications lifecycle.**

`application:didFinishLaunchingWithOptions:`

`applicationWillResignActive:`

`applicationDidEnterBackground:`

`applicationWillEnterForeground:`

`applicationDidBecomeActive:`

`applicationWillTerminate:`

We only care about this



application:didFinishLaunchingWithOptions:

Gets called when application is launched

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions



This is small, let's go two lines

- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions

Method Name

Parameter type

Parameter names



application:didFinishLaunchingWithOptions:

Gets called when application is launched

```
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    // your code here  
    return YES;  
}
```

To create log messages

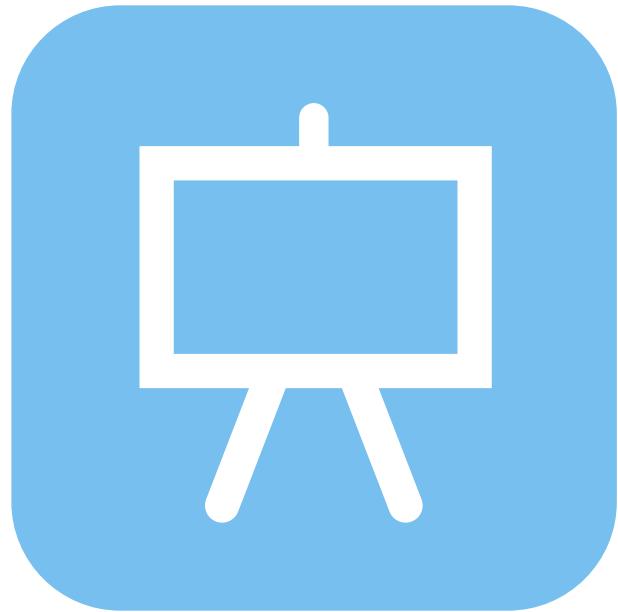
```
NSLog(@"%@", @"Hello World");
```

String

What does an artist need to paint?



Canvas



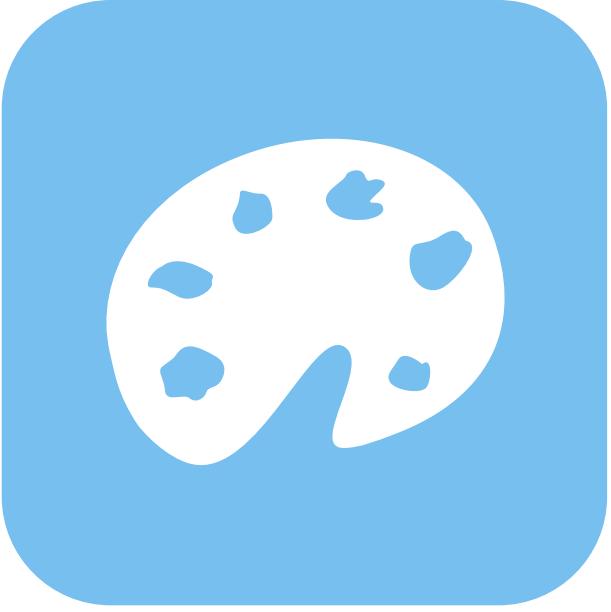
Window

Paint Brush



ViewController

Paint



View

What does an iOS app need to draw something?



Find the dimensions of the screen

AppDelegate.m

```
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
  
    UIScreen *screen = [UIScreen mainScreen]; Get the main screen  
  
    CGRect viewRect = [screen bounds]; Get the bounds of screen  
  
    NSLog(@"Screen is %f tall and %f wide",  
          viewRect.size.height, viewRect.size.width);  
  
    return YES;  
}
```

2012-10-03 21:11:59.012 AlphaTouch[44703:c07] Screen is 480.000000 tall and 320.000000 wide

DO WE REALLY NEED THE SCREEN VARIABLE?





Find the dimensions of the screen

AppDelegate.m

```
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    CGRect viewRect = [[UIScreen mainScreen] bounds];  
  
    NSLog(@"Screen is %f tall and %f wide",  
          viewRect.size.height, viewRect.size.width);  
  
    return YES;  
}
```

2012-10-03 21:11:59.012 AlphaTouch[44703:c07] Screen is 480.000000 tall and 320.000000 wide

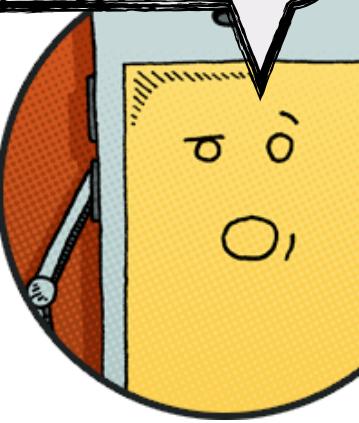


Create our UIWindow (canvas)

AppDelegate.m

```
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    CGRect viewRect = [[UIScreen mainScreen] bounds];  
  
    UIWindow *window = [[UIWindow alloc] initWithFrame:viewRect];  
  
    NSLog(@"Screen is %f tall and %f wide",  
          viewRect.size.height, viewRect.size.width);  
  
    return YES;  
}
```

THIS IS ALREADY A PROPERTY!



Allocate memory for a UI Window
and initialize object with frame size to the bounds of the main screen



Create our UIWindow (canvas)

AppDelegate.h

```
@property (strong, nonatomic) UIWindow *window; ~instance variable
```

AppDelegate.m

```
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    CGRect viewRect = [[UIScreen mainScreen] bounds];  
  
    self.window = [[UIWindow alloc] initWithFrame:viewRect];  
  
    NSLog(@"Screen is %f tall and %f wide",  
          viewRect.size.height, viewRect.size.width);  
  
    return YES;  
}
```

Tell the UIWindow it's the Key Window & Visible



AppDelegate.m

```
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    CGRect viewRect = [[UIScreen mainScreen] bounds];  
    self.window = [[UIWindow alloc] initWithFrame:viewRect];  
    [self.window makeKeyAndVisible];  
    NSLog(@"Screen is %f tall and %f wide",  
          viewRect.size.height, viewRect.size.width);  
    return YES;  
}
```

2012-10-03 21:50:31.019 AlphaTouch[45090:c07] Application windows are expected to have a root view controller at the end of application launch



Try
iOS



Create the ViewController

AppDelegate.m

```
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    CGRect viewRect = [[UIScreen mainScreen] bounds];  
    self.window = [[UIWindow alloc] initWithFrame:viewRect];  
  
    UIViewController *colorTouchVC = [[UIViewController alloc] init];  
    self.window.rootViewController = colorTouchVC;  
  
    [self.window makeKeyAndVisible];  
    NSLog(@"Screen is %f tall and %f wide",  
          viewRect.size.height, viewRect.size.width);  
    return YES;  
}
```

This ViewController gets control of the window



Create the View

AppDelegate.m

```
CGRect viewRect = [[UIScreen mainScreen] bounds];
self.window = [[UIWindow alloc] initWithFrame:viewRect];

UIViewController *colorTouchVC = [[UIViewController alloc] init];

UIView *colorView = [[UIView alloc] initWithFrame:viewRect];
colorTouchVC.view = colorView;

self.window.rootViewController = colorTouchVC;
[self.window makeKeyAndVisible];
NSLog(@"Screen is %f tall and %f wide",
      viewRect.size.height, viewRect.size.width);
```

Create a view the size of the whole screen



Set the background color

AppDelegate.m

```
CGRect viewRect = [[UIScreen mainScreen] bounds];
self.window = [[UIWindow alloc] initWithFrame:viewRect];

UIViewController *colorTouchVC = [[UIViewController alloc] init];

UIView *colorView = [[UIView alloc] initWithFrame:viewRect];
colorView.backgroundColor = [UIColor yellowColor];
colorTouchVC.view = colorView;

self.window.rootViewController = colorTouchVC;
[self.window makeKeyAndVisible];
NSLog(@"Screen is %f tall and %f wide",
      viewRect.size.height, viewRect.size.width);
```



Look up the UIColor class to change the color



Lets Review

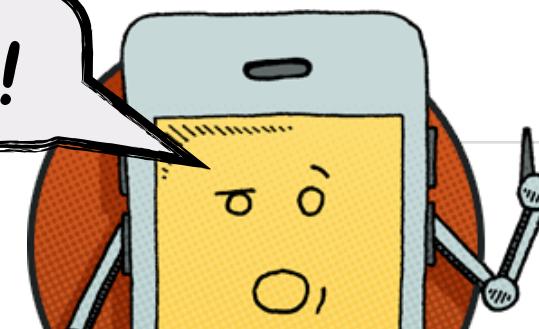
AppDelegate.m

```
CGRect viewRect = [[UIScreen mainScreen] bounds];
self.window = [[UIWindow alloc] initWithFrame:viewRect];

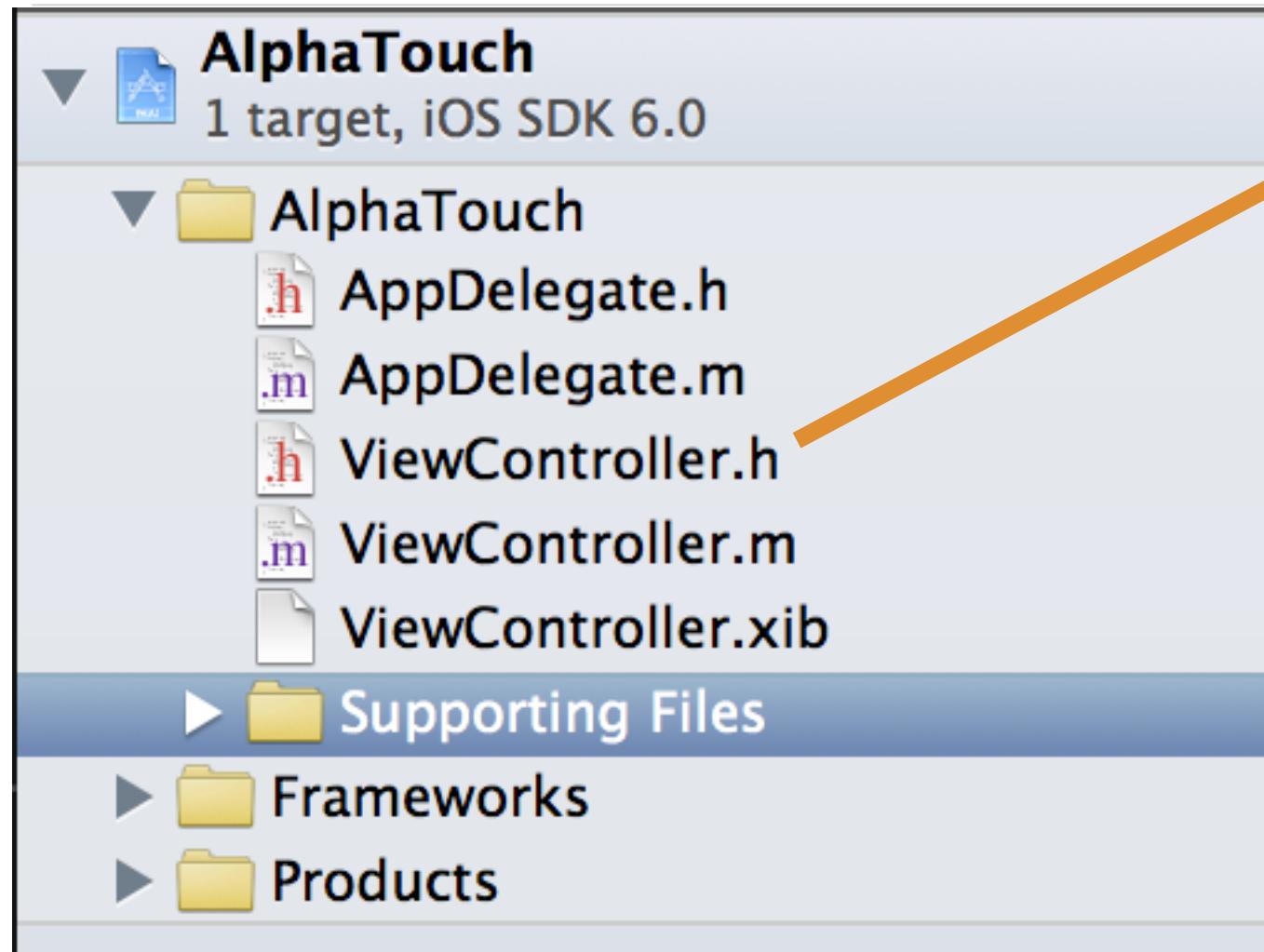
UIView *colorView = [[UIView alloc] initWithFrame:viewRect];
colorView.backgroundColor = [UIColor yellowColor];
colorTouchVC.view = colorView;

self.window.rootViewController = colorTouchVC;
[self.window makeKeyAndVisible];
NSLog(@"Screen is %f tall and %f wide",
      viewRect.size.height, viewRect.size.width);
```

THIS CODE NEEDS TO BE REFACTORED!



We already had a ViewController



ViewController.h

```
#import <UIKit/UIKit.h>
```

```
@interface ViewController : UIViewController
```

```
@end
```

Subclass name

↑

Parent Class

OFTEN WE SAY SUPERCLASS
INSTEAD OF PARENT CLASS





Need to reference the ViewController

ViewController.h

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

@end
```



AppDelegate.h

```
@class ViewController; Forward Declaration
@property (strong, nonatomic) ViewController *viewController;
```



AppDelegate.m

```
#import "ViewController.h" Includes the class
```



Need to reference the ViewController

AppDelegate.m

old code

 CGRect viewRect = [[UIScreen mainScreen] bounds];
self.window = [[UIWindow alloc] initWithFrame:viewRect];



 UIViewController *viewController = [[UIViewController alloc] init];

 UIView *view = [[UIView alloc] initWithFrame:viewRect];
view.backgroundColor = [UIColor yellowColor];

viewController.view = view;

self.window.rootViewController = viewController;

...



Need to reference the ViewController

AppDelegate.m

```
CGRect viewRect = [[UIScreen mainScreen] bounds];
self.window = [[UIWindow alloc] initWithFrame:viewRect];

self.viewController = [[ViewController alloc] init];

UIView *view = [[UIView alloc] initWithFrame:viewRect];
view.backgroundColor = [UIColor yellowColor];
self.viewController.view = view;
self.window.rootViewController = self.viewController;

...
```



Our app delegate shouldn't know about view!

AppDelegate.m

```
CGRect viewRect = [[UIScreen mainScreen] bounds];
self.window = [[UIWindow alloc] initWithFrame:viewRect];  
  
self.viewController = [[ViewController alloc] init];  
  
UIView *view = [[UIView alloc] initWithFrame:viewRect];
view.backgroundColor = [UIColor yellowColor];
self.viewController.view = view;
self.window.rootViewController = self.viewController;  
...
```



Need to move this into ViewController.m



Getting familiar with ViewController

ViewController.m

```
- (void)viewDidLoad After loadView, typically where labels/buttons go
{
    [super viewDidLoad];
    self.view.backgroundColor = [UIColor yellowColor];
}

- (void)loadView Called first time view property accessed
{
    CGRect viewRect = [[UIScreen mainScreen] bounds];
    UIView *view = [[UIView alloc] initWithFrame:viewRect];
    self.view = view;
}
```



With the View removed from the ViewController



AppDelegate.m

```
CGRect viewRect = [[UIScreen mainScreen] bounds];  
self.window = [[UIWindow alloc] initWithFrame:viewRect];
```

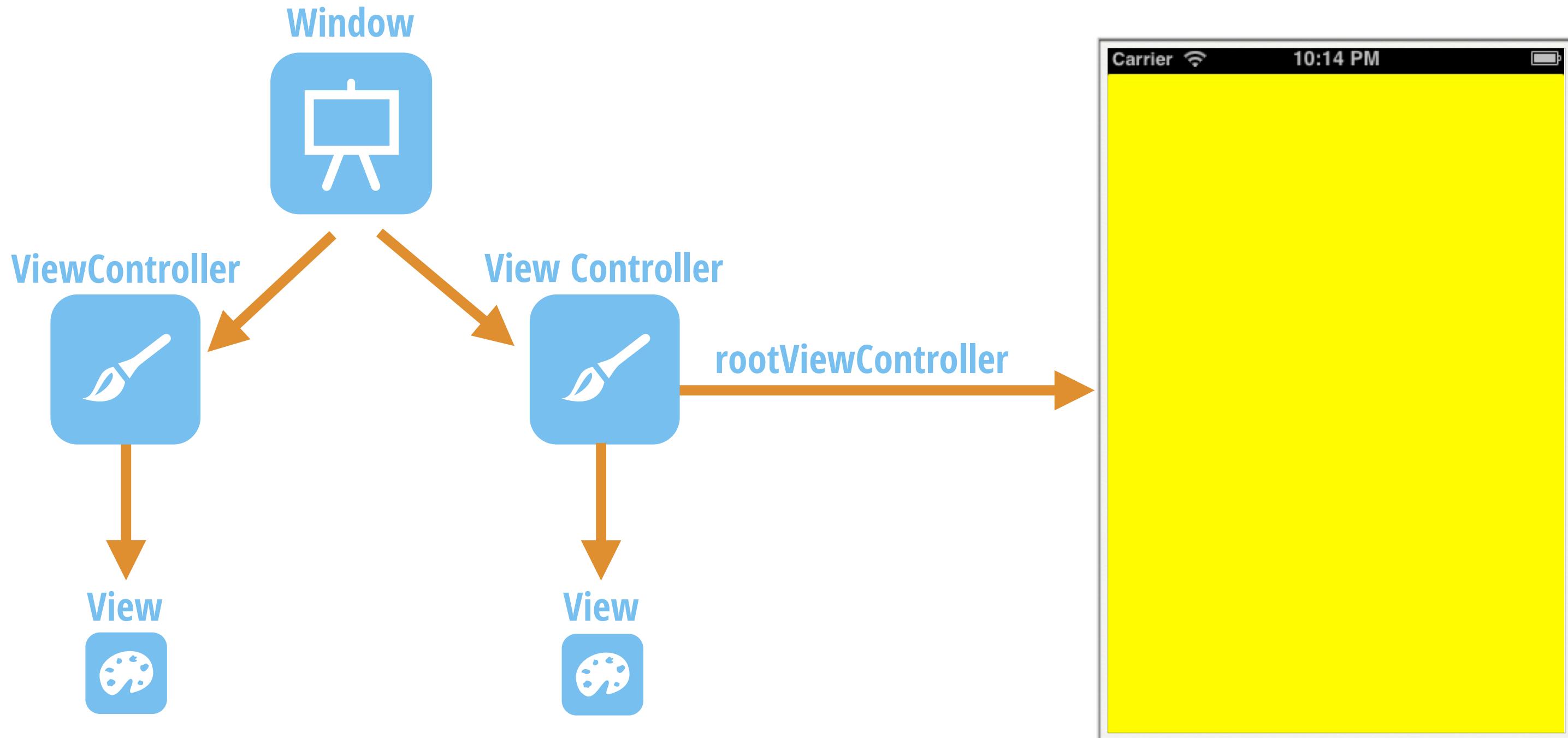
```
self.viewController = [[ViewController alloc] init];  
self.window.rootViewController = self.viewController;  
[self.window makeKeyAndVisible];
```



MUCH BETTER!
IT'S NOT GOOD TO HAVE
VIEWS IN YOUR APPDELEGATE.



App Anatomy



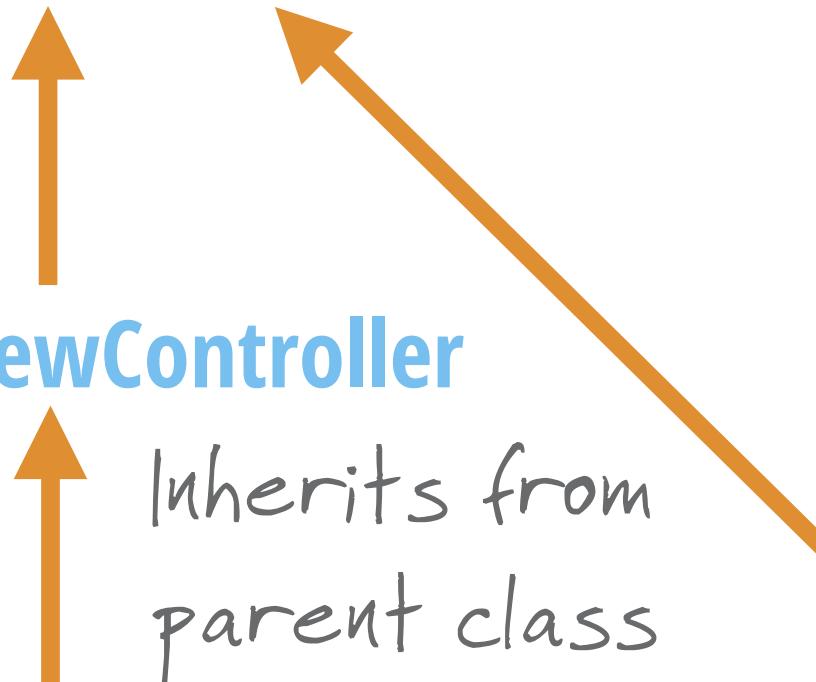


Next objective

Print log message whenever touched

```
NSLog(@"Started touching the screen");
```

UIResponder Interface for responding to events



Inherits from
parent class

Gives us access to touch events

- touchesBegan:withEvent:
- touchesMoved:withEvent:
- touchesEnded:withEvent:
- touchesCancelled:withEvent:



Print log message whenever touched

ViewController.m

```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
    NSLog(@"Started touching the screen");
}
```



Level 2

Subviews





In this level we'll cover

01 Creating a Button

02 Creating a Label

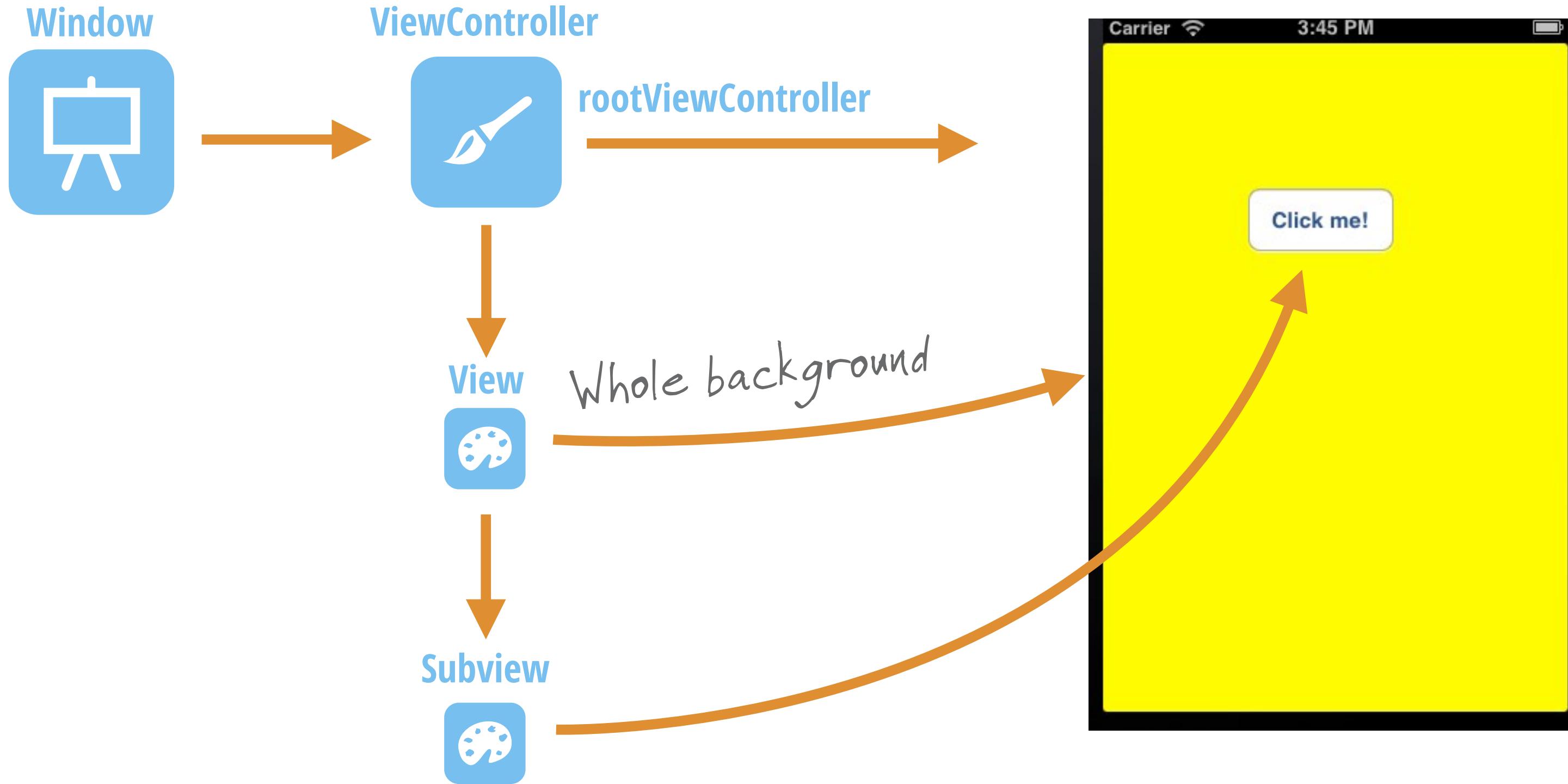
03 Making the Button Work

04 Creating Multiple Buttons

05 Refactoring Our Code



Remember this?



Subviews



Create a click me button

ViewController.m

```
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    self.view.backgroundColor = [UIColor yellowColor];
```



```
UIButton *firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
```

Create a new UIButton of type UIButtonTypeRoundedRect

```
firstButton.frame = CGRectMake(100, 100, 100, 44);
```

Located at x = 100pts, y = 100pts, 100pts width, 44pts height

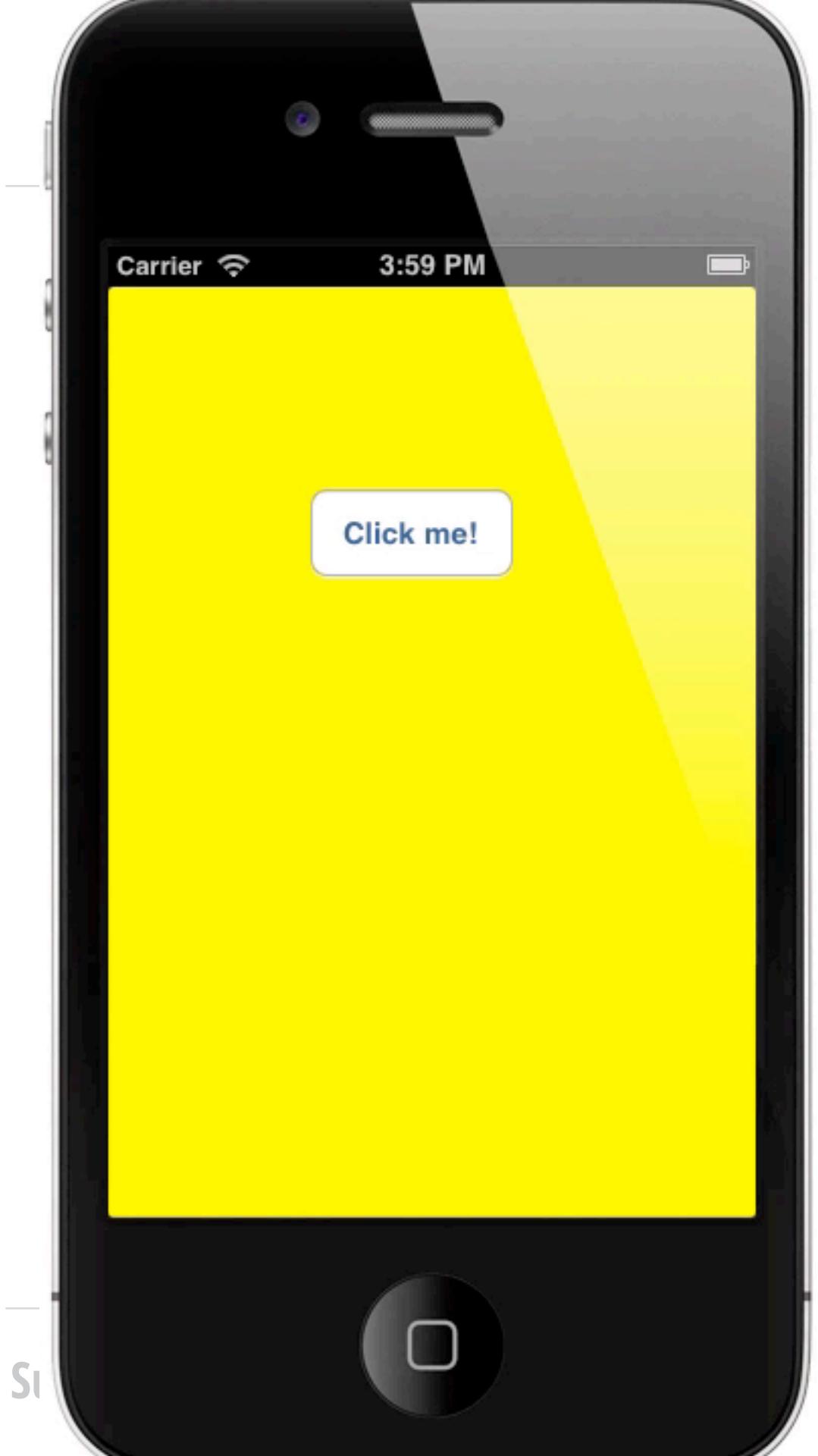
```
[firstButton setTitle:@"Click me!" forState:UIControlStateNormal];
```

Set the button title, what it shows normally

```
[self.view addSubview:firstButton];
```

Add button on top of our view





Yay, a button!



S

Try
iOS



Create a click me button

ViewController.m

```
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    self.view.backgroundColor = [UIColor yellowColor];
```



```
UIButton *firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
```

Create a new UIButton

```
firstButton.frame = CGRectMake(100, 100, 100, 44);
```

Located at x = 100, y = 100, 100 width, 44 height

```
[firstButton setTitle:@"Click me!" forState:UIControlStateNormal];
```

Set the button title, what it shows normally

```
[self.view addSubview:firstButton];
```

}

Add button on top of our view

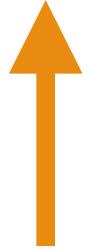


Create a UIButton

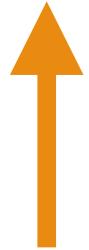
```
UIButton *firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
```



UIResponder



UIView



UIButton

Subviews

UIButtonTypeRoundedRect



UIButtonTypeDetailDisclosure



UIButtonTypeInfoLight



UIButtonTypeInfoDark

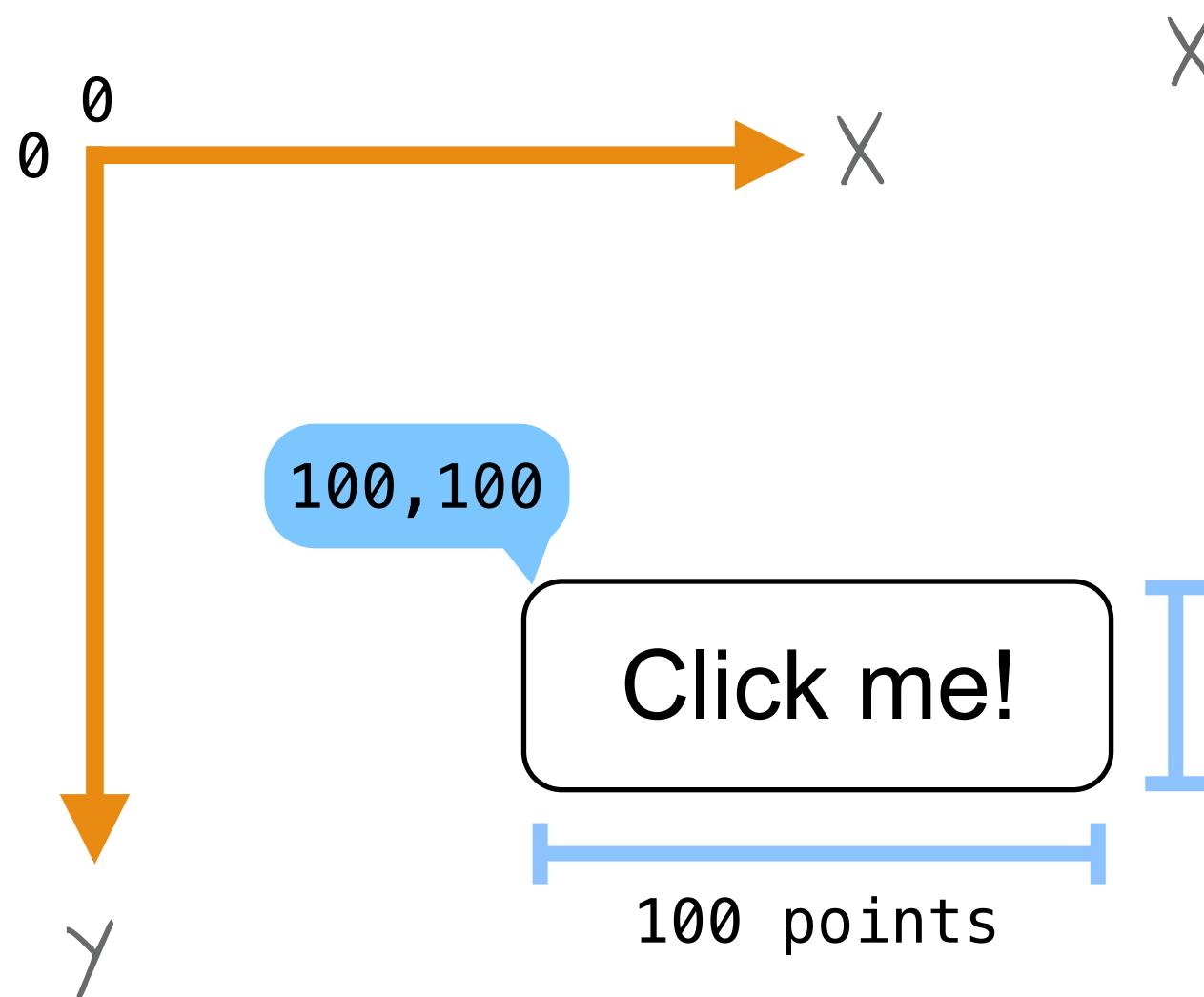


UIButtonTypeContactAdd

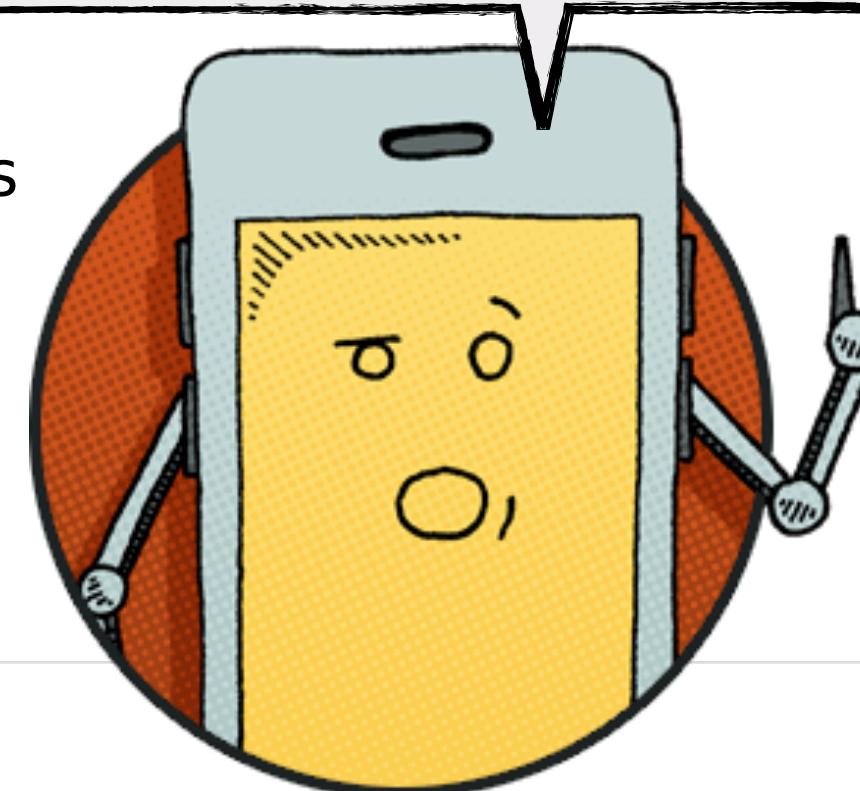


Set the position & size of the button

```
firstButton.frame = CGRectMake(100, 100, 100, 44);
```

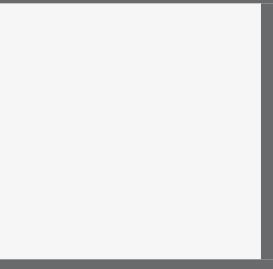
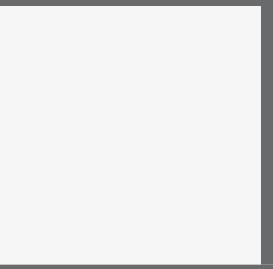
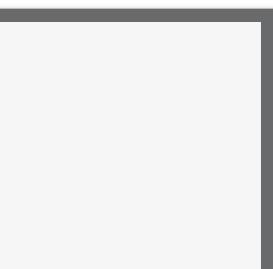
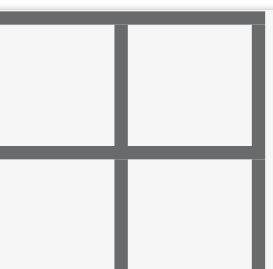


YOU MUST BE WONDERING
WHAT POINTS ARE...





Understanding Points

	Points	Pixels
Non Retina iPhone 1 point = 1 pixel		
iPhone 4 (Retina Screen) 1 point = 4 pixels		
iPhone 5 (Retina Screen) 1 point = 4 pixels		



Set the button title

```
[firstButton setTitle:@"Click me!" forState:UIControlStateNormal];
```

Sets the title shown on button

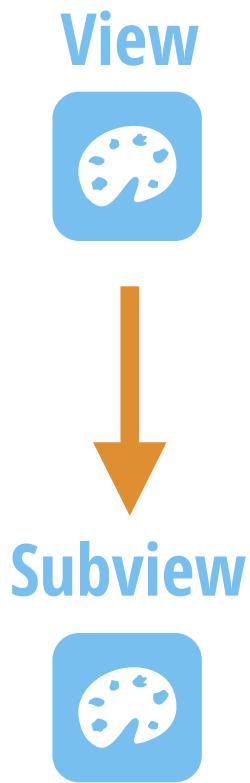
forState:UIControlStateNormal

Sets the title when pressed



Add the Button view as a subview

```
[self.view addSubview:firstButton];
```



Can be multiple subviews

Subviews

Adding a UILabel (also inherits from UIView)



```
UILabel *firstLabel = [[UILabel alloc] initWithFrame:CGRectMake(50, 30, 200, 44)];
```

Create a new label with frame

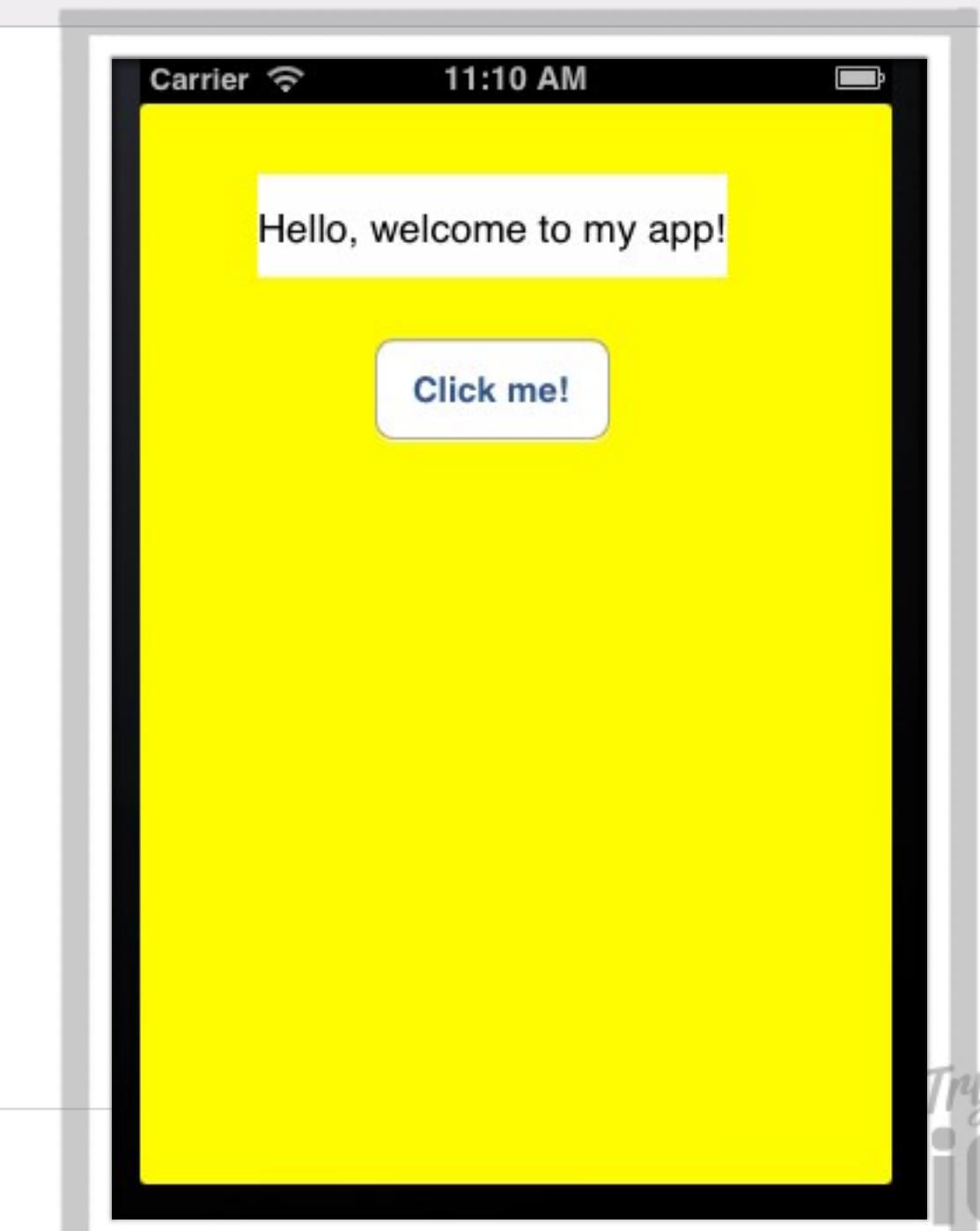
```
firstLabel.text = @"Hello, welcome to my app!";
```

Set the label text

```
[self.view addSubview:firstLabel];
```

Add the label as a subview

WHAT'S WITH THE WHITE
BACKGROUND ON UILabel?

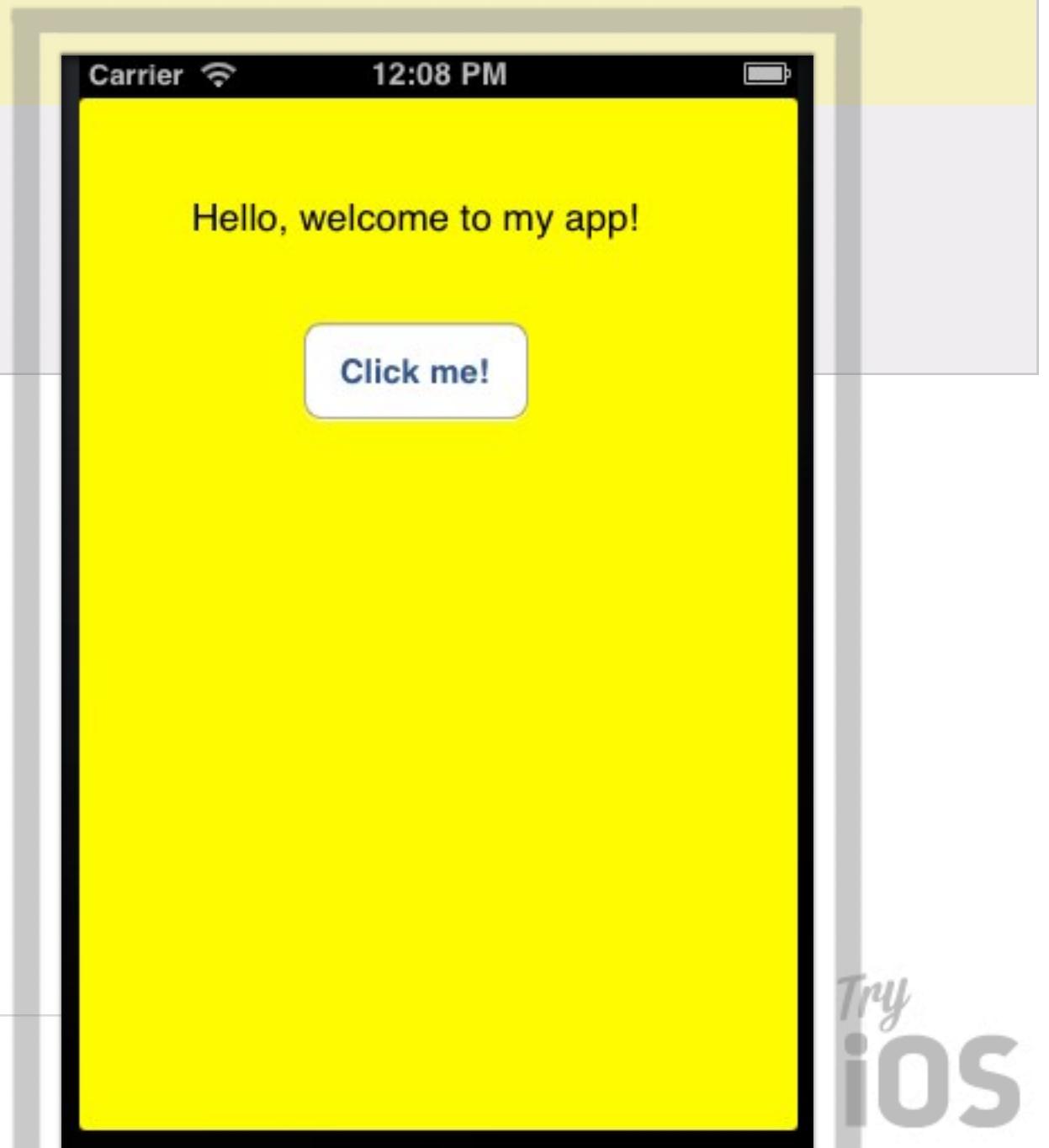


Try
iOS

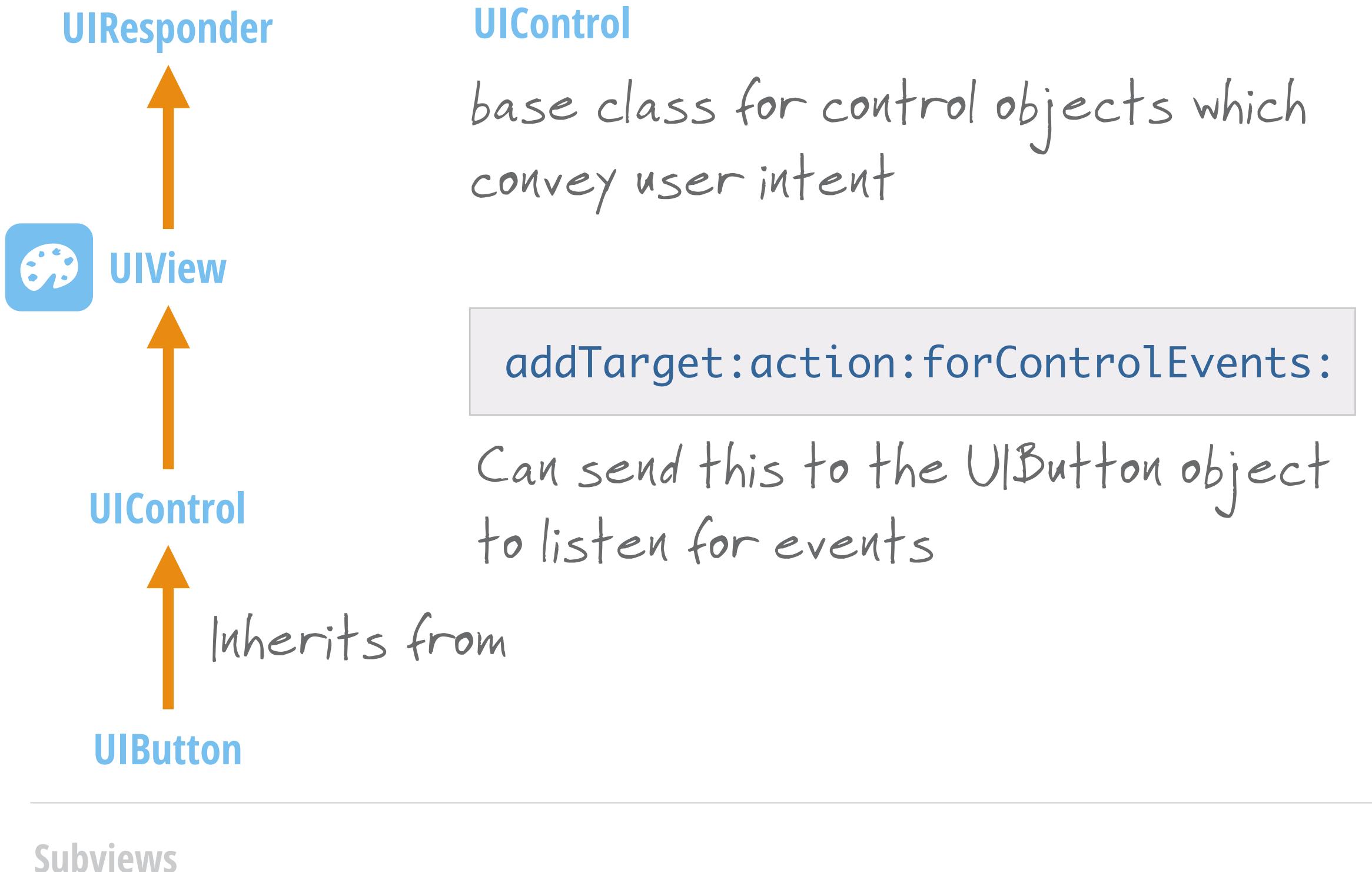
Coloring in the background



```
UILabel *firstLabel = [[UILabel alloc] initWithFrame:CGRectMake(50, 30, 200, 44)];  
  
firstLabel.backgroundColor = [UIColor clearColor];  
  
firstLabel.text = @"Hello, welcome to my app!";  
[self.view addSubview:firstLabel];
```



How do we make the button work?





Adding Control Events

`addTarget:action:forControlEvents:`

```
- (void)addTarget:(id)target           instance method  
          action:(SEL)action  
forControlEvents:(UIControlEvents)controlEvents;
```

target the target object to which the message will be sent

action the message to send to the target

controlEvents the event to listen for



Adding Control Events

ViewController.m

```
- (void)viewDidLoad
{
    ...
    [firstButton addTarget:self
                      action:@selector(buttonPressed:)
            forControlEvents:UIControlEventTouchUpInside];
}

- (void)buttonPressed:(UIButton *)sender
{
    NSLog(@"Button pressed, sender: %@", sender);
    self.view.alpha = ((double)arc4random() / 0x10000000);
}
```



Call `[self buttonPressed:firstButton];` when event triggered

Subviews



Adding Method to Header

ViewController.m

```
- (void)buttonPressed:(UIButton *)sender
{
    NSLog(@"Button pressed, sender: %@", sender);
    self.view.alpha = ((double)arc4random() / 0x100000000);
}
```



ViewController.h

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

- (void)buttonPressed:(UIButton *)sender;

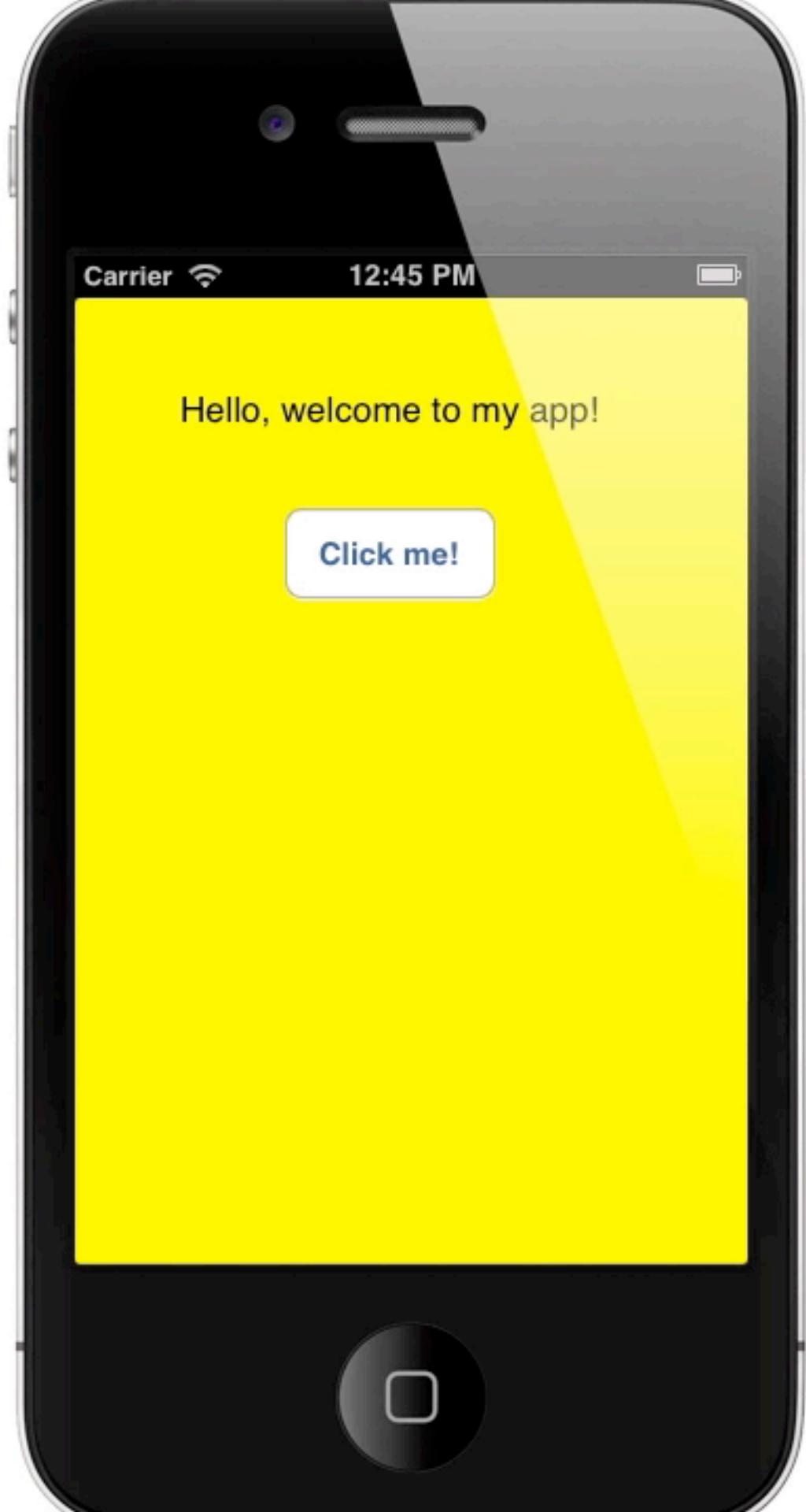
@end
```





Our button works!

**Button pressed, sender: <UIRoundedRectButton:
0x7195870; frame = (100 100; 100 44); opaque = NO;
layer = <CALayer: 0x71959e0>>**





UIControl Events

UIControlEventTouchDown
UIControlEventTouchDownRepeat
UIControlEventTouchDragInside
UIControlEventTouchDragOutside
UIControlEventTouchDragEnter
UIControlEventTouchDragExit
UIControlEventTouchUpInside
UIControlEventTouchUpInside
UIControlEventTouchCancel
UIControlEventValueChanged
UIControlEventEditingDidBegin
UIControlEventEditingChanged
UIControlEventEditingDidEnd
UIControlEventEditingDidEndOnExit
UIControlEventAllTouchEvents
UIControlEventAllEditingEvents
UIControlEventAllEvents



Removing a View

ViewController.m

```
- (void)buttonPressed:(UIButton *)sender
{
    NSLog(@"Button pressed, sender: %@", sender);
    self.view.alpha = ((double)arc4random() / 0x10000000);
    [sender removeFromSuperview];
}
```



A Second Button



ViewController.m



```
UIButton *firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
firstButton.frame = CGRectMake(100, 100, 100, 44);
[firstButton setTitle:@"Make 50%" forState:UIControlStateNormal];
[self.view addSubview:firstButton];
```

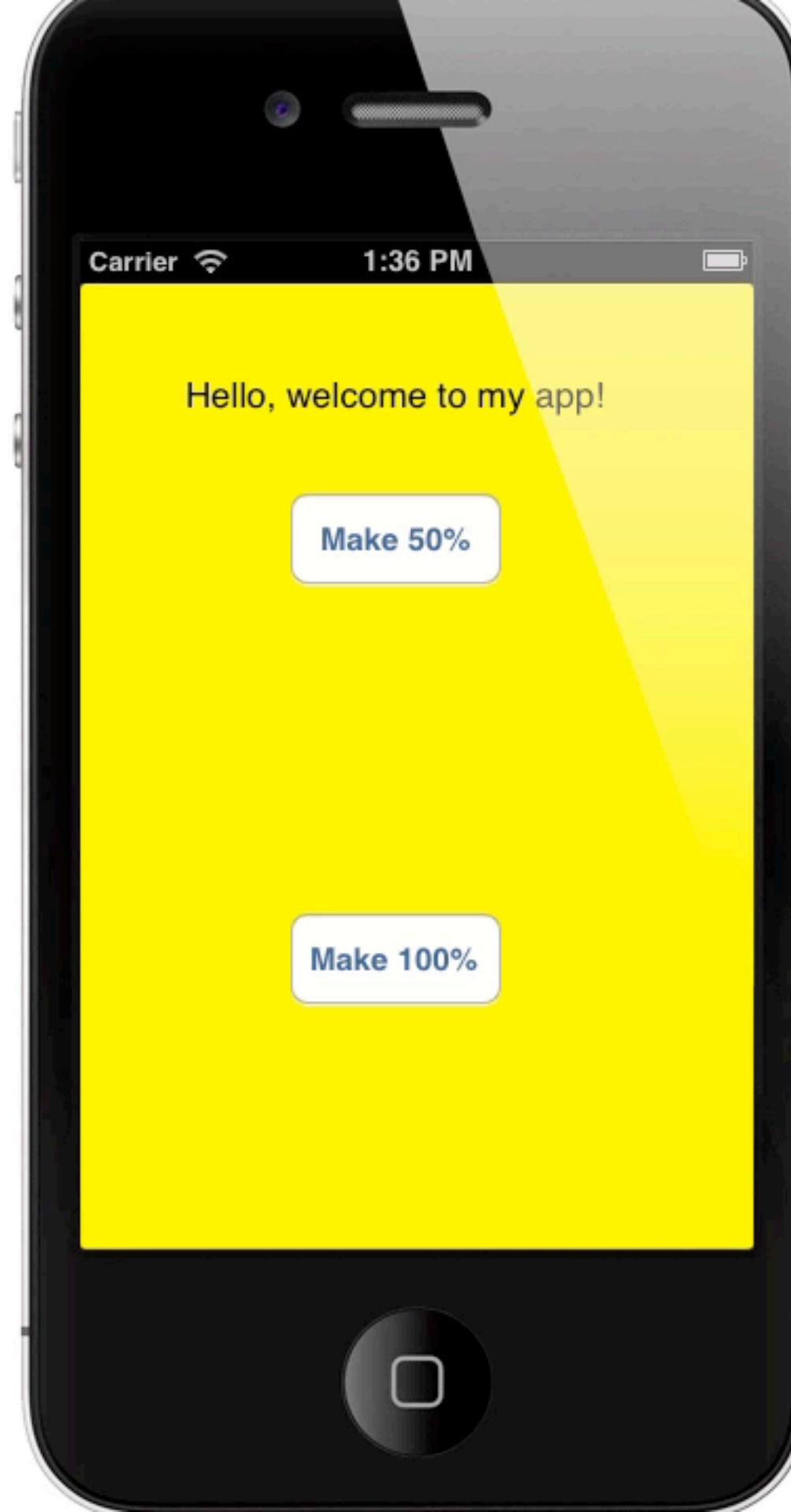
```
[firstButton addTarget:self
                  action:@selector(buttonPressed:)
            forControlEvents:UIControlEventTouchUpInside];
```



```
UIButton *secondButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
secondButton.frame = CGRectMake(100, 300, 100, 44);
[secondButton setTitle:@"Make 100%" forState:UIControlStateNormal];
[self.view addSubview:secondButton];
```

```
[secondButton addTarget:self
                  action:@selector(buttonPressed:)
            forControlEvents:UIControlEventTouchUpInside];
```

Now with two Buttons



BUT THEY BOTH DO
THE SAME THING!





Completing alpha Functionality

ViewController.m

```
- (void)buttonPressed:(UIButton *)sender
{
    NSLog(@"Button pressed, sender: %@", sender);

    if (<is this the 50% button?>) {
        self.view.alpha = .5;
    } else {
        self.view.alpha = 1;
    }
}
```



Checking for the right button

Documentation for UIButton

titleLabel

A view that displays the value of the `currentTitle` property for a button. (read-only)

```
@property(nonatomic, readonly, retain) UILabel *titleLabel
```

Documentation for UILabel

text

The text displayed by the label.

```
@property(nonatomic, copy) NSString *text
```



Documentation for NSString

isEqualToString:

Returns a Boolean value that indicates whether a given string is equal to the receiver using a literal Unicode-based comparison.

```
- (BOOL)isEqualToString:(NSString *)aString
```

```
if ([sender.titleLabel.text isEqualToString:@"Make 50%"]) {
```

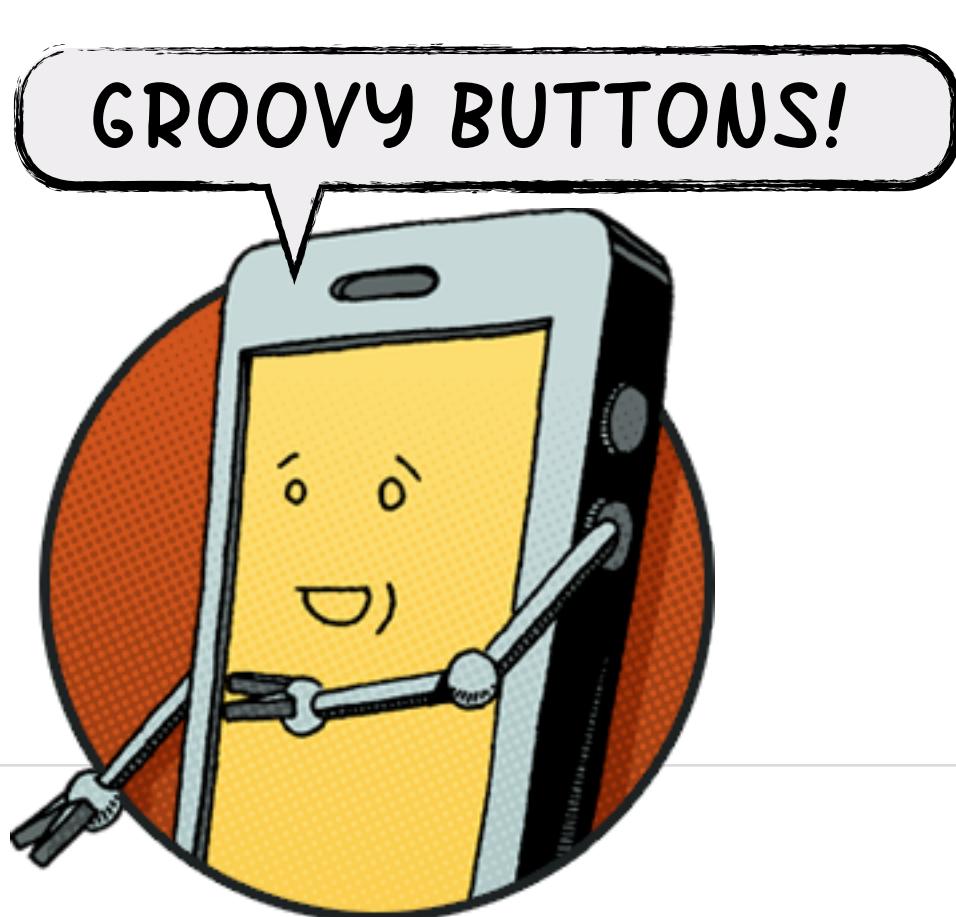
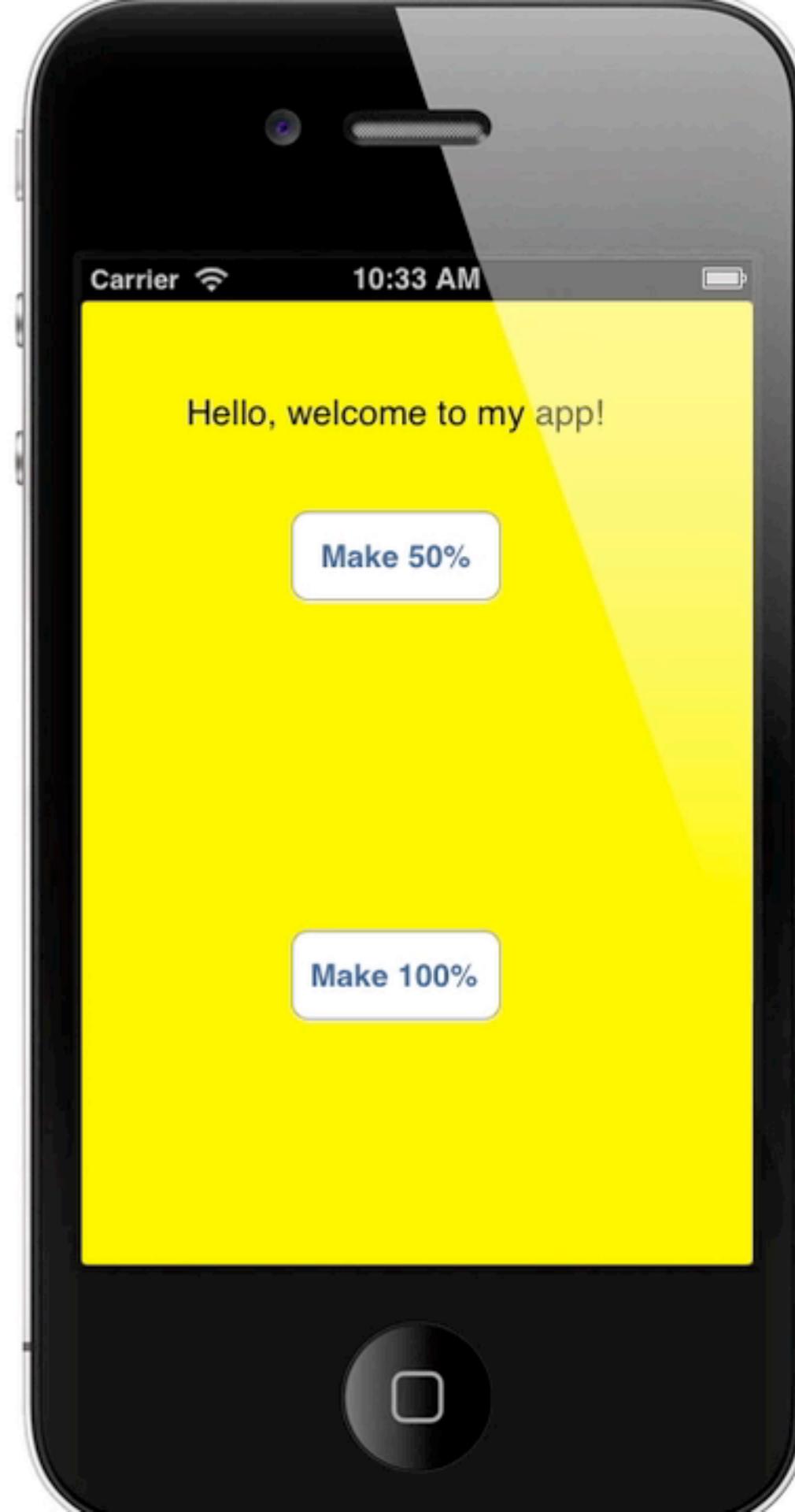


Completing alpha Functionality

```
- (void)buttonPressed:(UIButton *)sender
{
    NSLog(@"Button pressed, sender: %@", sender);

    if ([sender.titleLabel.text isEqualToString:@"Make 50%"]) {
        self.view.alpha = .5;
    } else {
        self.view.alpha = 1;
    }
}
```

Now they work!





Improving this Code

ViewController.m

```
- (void)viewDidLoad
{
    UIButton *firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    ...
    UIButton *secondButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    ...
}

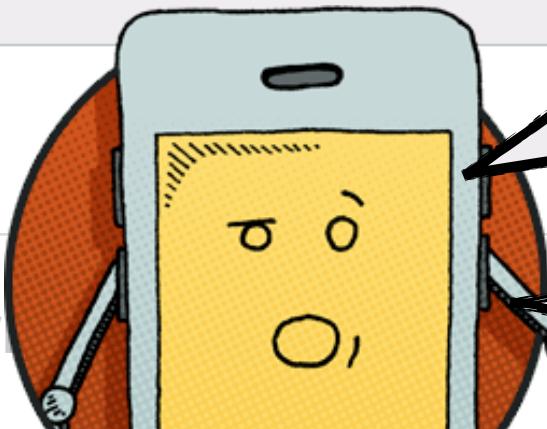
- (void)buttonPressed:(UIButton *)sender
{
    if ([sender.titleLabel.text isEqualToString:@"Make 50%"]) {
        ...
}
```



WE'RE DOING A COMPARISON
ON THE BUTTON TITLE.

THE VARIABLE NAMES ARE AWFUL.

Subv.



Try
iOS



Creating new properties

ViewController.h

```
@interface ViewController : UIViewController  
  
@property (weak, nonatomic) UIButton *fiftyPercentButton;  
  
@property (weak, nonatomic) UIButton *hundredPercentButton;
```



ViewController.m

```
- (void)viewDidLoad  
{  
  
    UIButton* firstButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];  
    firstButton.frame = CGRectMake(100, 100, 100, 44);  
    [firstButton setTitle:@"Make 50%" forState:UIControlStateNormal];  
    [self.view addSubview:firstButton];  
  
    [firstButton addTarget:self  
                      action:@selector(buttonPressed:)  
            forControlEvents:UIControlEventTouchUpInside];
```

old code





Updating the controller to use new property

ViewController.h

```
@interface ViewController : UIViewController  
  
    @property (weak, nonatomic) UIButton *fiftyPercentButton;  
  
    @property (weak, nonatomic) UIButton *hundredPercentButton;
```



ViewController.m

```
- (void)viewDidLoad  
{  
  
    self.fiftyPercentButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];  
    self.fiftyPercentButton.frame = CGRectMake(100, 100, 100, 44);  
    [self.fiftyPercentButton setTitle:@"Make 50%" forState:UIControlStateNormal];  
    [self.view addSubview:self.fiftyPercentButton];  
  
    [self.fiftyPercentButton addTarget:self  
                                action:@selector(buttonPressed:)  
                  forControlEvents:UIControlEventTouchUpInside];
```





Updating the buttonPressed conditional

ViewController.h

```
@interface ViewController : UIViewController  
  
@property (weak, nonatomic) UIButton *fiftyPercentButton;  
  
@property (weak, nonatomic) UIButton *hundredPercentButton;
```



ViewController.m

```
- (void)buttonPressed:(UIButton *)sender  
{  
    if ([sender.titleLabel.text isEqualToString:@"Make 50%"]) {  
        ...  
    }  
}
```





Updating the buttonPressed method

ViewController.h

```
@interface ViewController : UIViewController  
  
@property (weak, nonatomic) UIButton *fiftyPercentButton;  
  
@property (weak, nonatomic) UIButton *hundredPercentButton;
```

ViewController.m

```
- (void)buttonPressed:(UIButton *)sender  
{  
    if ([sender isEqual:self.fiftyPercentButton]) {  
        ...  
    }  
}
```



Level 3

Tabs, Images, and Scrolling



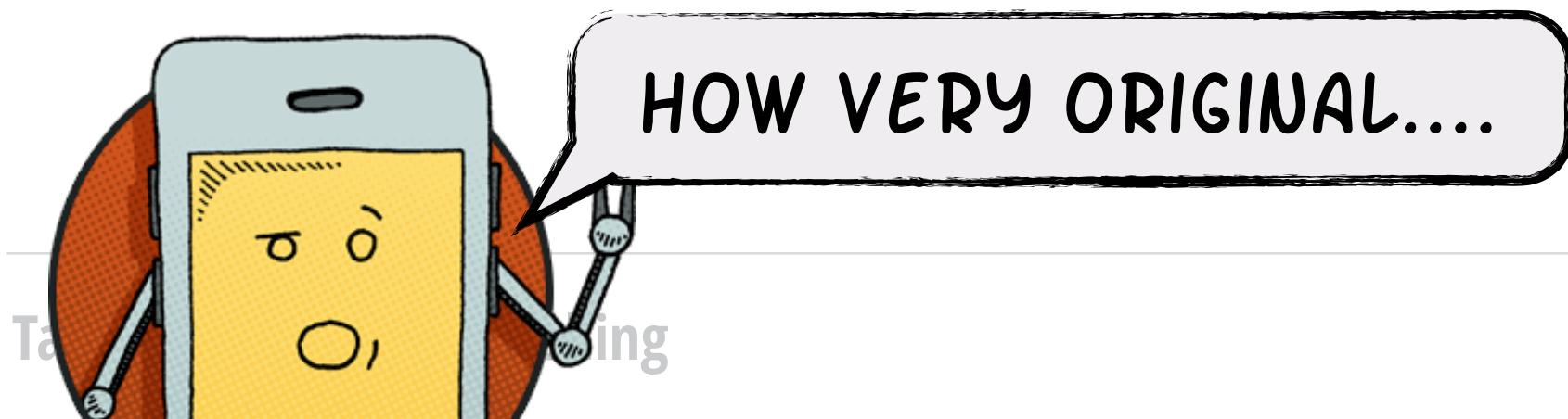
Introducing InstaPhoto



Photo Feed Show most recent photos from the Internet

Favorites Show your personal favorite photos, from the feed

Profile Show/Edit personal information



Ta

Try
iOS



Level 3

01 Introduction to the UITabBarController

02 Adding Images to the Tabs

03 Refactoring Our View Controllers

04 Adding Images to Our Application

05 Scroll View



Our Lonely ViewController

Window ViewController View



A VIEW CONTROLLER SHOULD
ONLY HAVE ONE PRIMARY JOB.



ViewControllers inside Clock app



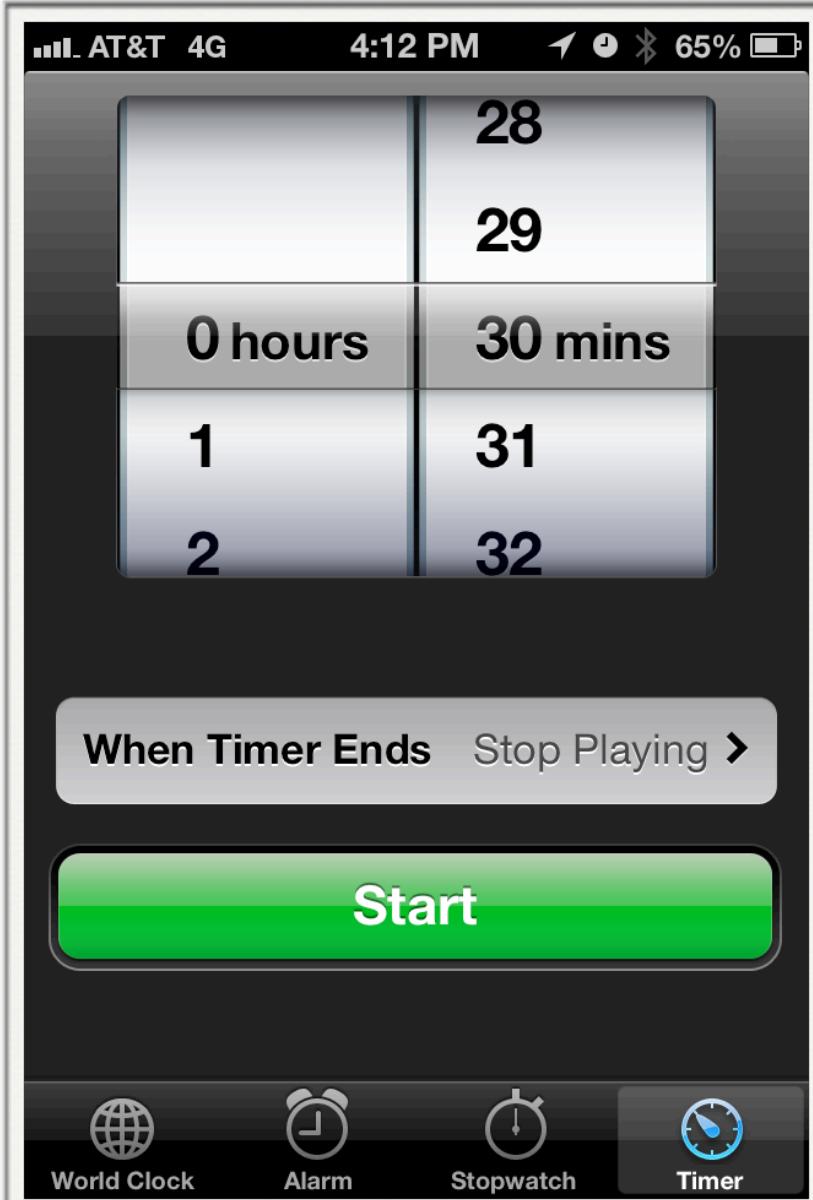
WorldClockVC



AlarmVC



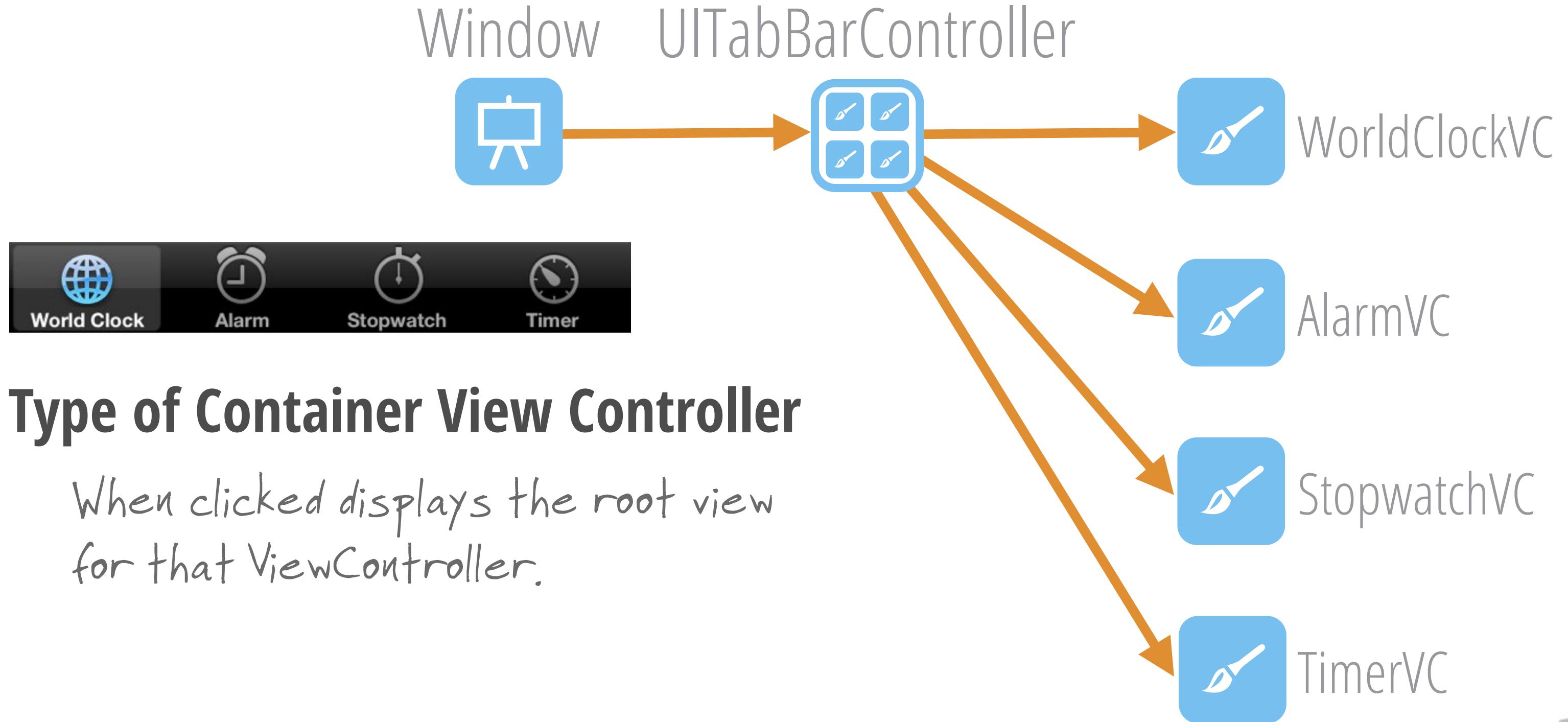
StopwatchVC



TimerVC



Introduction to the UITabBarController





Creating a UITabBarController

AppDelegate.m

```
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    UITabBarController *tabBarController = [[UITabBarController alloc] init];  
  
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];  
    self.window.rootViewController = tabBarController;  
    [self.window makeKeyAndVisible];  
  
    return YES;  
}
```

No need to create a subclass

Next, we need to create ViewControllers & add them to the tab



Adding ViewControllers to our Tab

AppDelegate.m

```
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{
```



```
    UIViewController *feedViewController = [[UIViewController alloc] init];  
    feedViewController.title = @"Feed";
```



```
    UIViewController *favoritesViewController = [[UIViewController alloc] init];  
    favoritesViewController.title = @"Favorites";
```

```
    feedViewController.view.backgroundColor = [UIColor blueColor];  
    favoritesViewController.view.backgroundColor = [UIColor redColor];
```



```
UITabBarController *tabBarController = [[UITabBarController alloc] init];
```

```
[tabBarController setViewControllers:@[feedViewController, favoritesViewController]];
```

... Add ViewControllers to the tabBar, and default to first.

With our UITabBarController



JUST ADD A POLICE SIREN,
AND YOU COULD PULL
SOMEONE OVER WITH THIS APP.



Let's add some Icons to our Tab

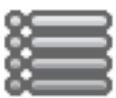


These icons are:

Sized 30x30 points

Usually PNG with only the Alpha channel

Imported into XCode project





Let's add some Icons to our Tabs

AppDelegate.m

```
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{
```



```
UIViewController *feedViewController = [[UIViewController alloc] init];  
feedViewController.title = @"Feed";
```



```
feedViewController.tabBarItem.image = [UIImage imageNamed:@"tab_icon_feed"];
```

...



A property of UIViewController

Look in cache/filesystem for tab_icon_feed.png



Our AppDelegate is Cluttered

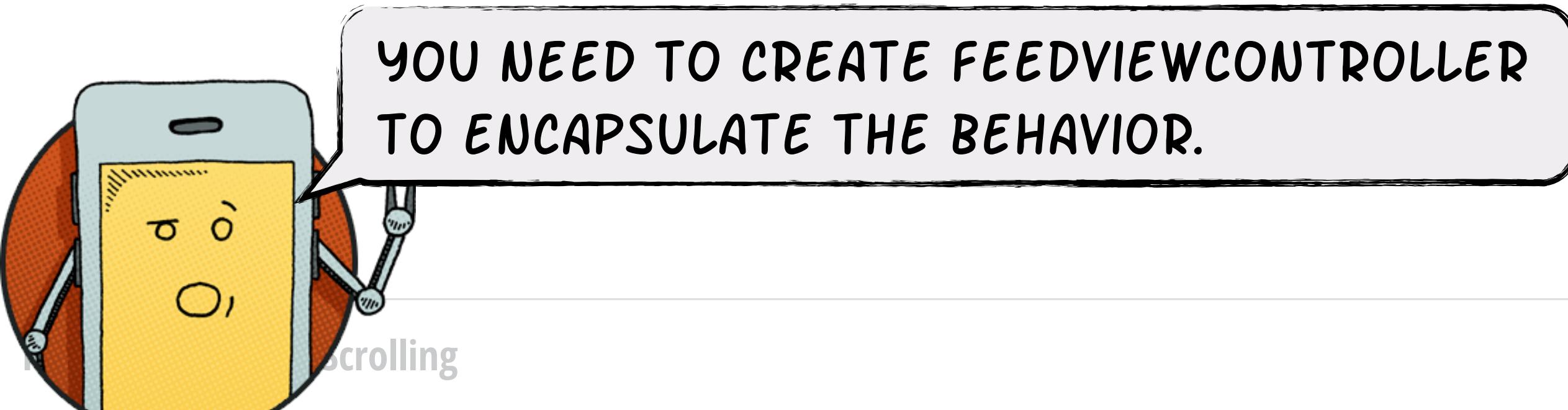
AppDelegate.m

```
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
```

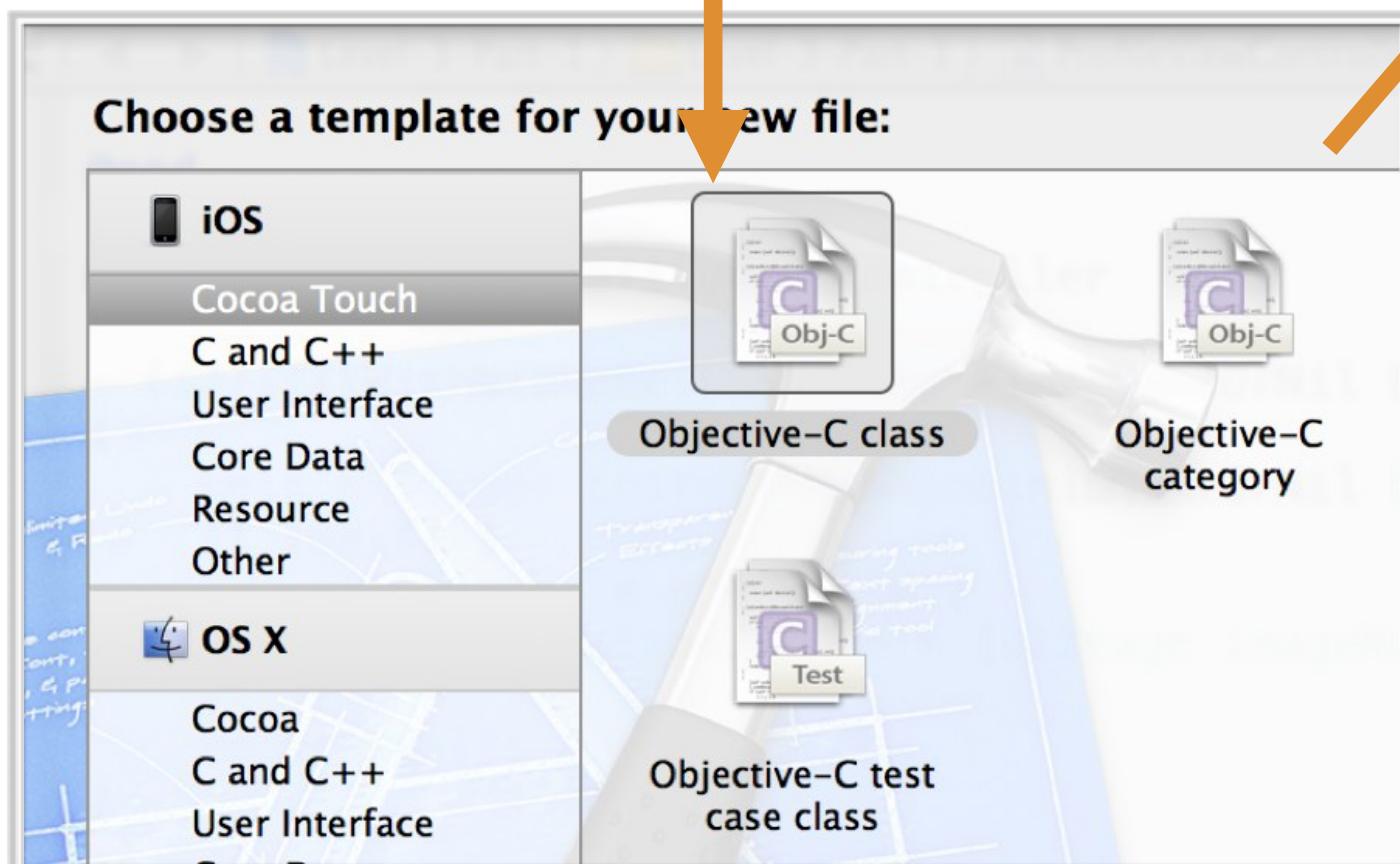
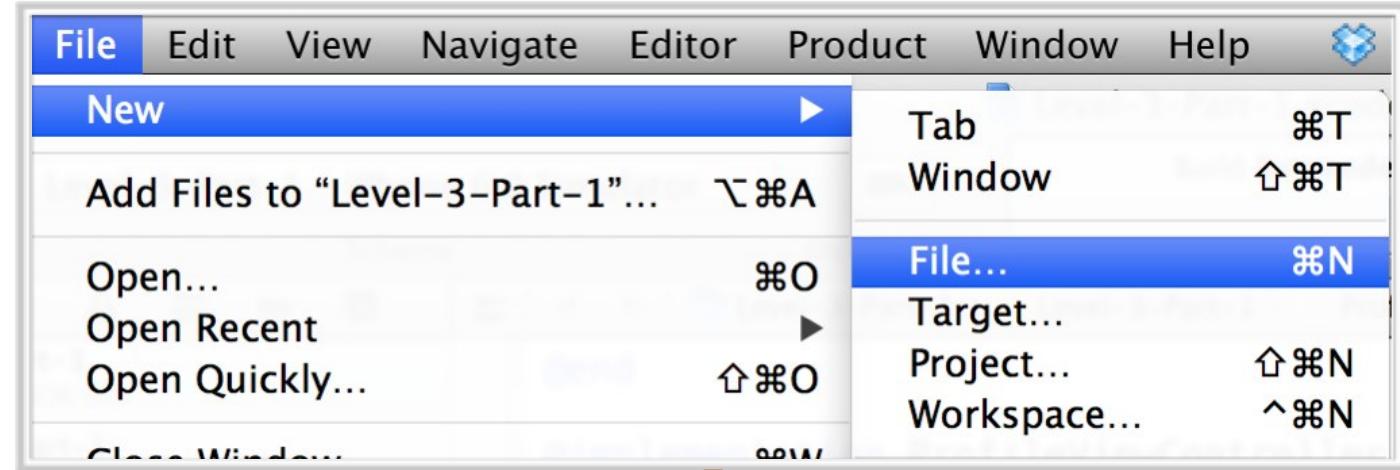


```
UIViewController *feedViewController = [[UIViewController alloc] init];  
feedViewController.title = @"Feed";  
feedViewController.tabBarItem.image = [UIImage imageNamed:@"tab_icon_feed"];
```

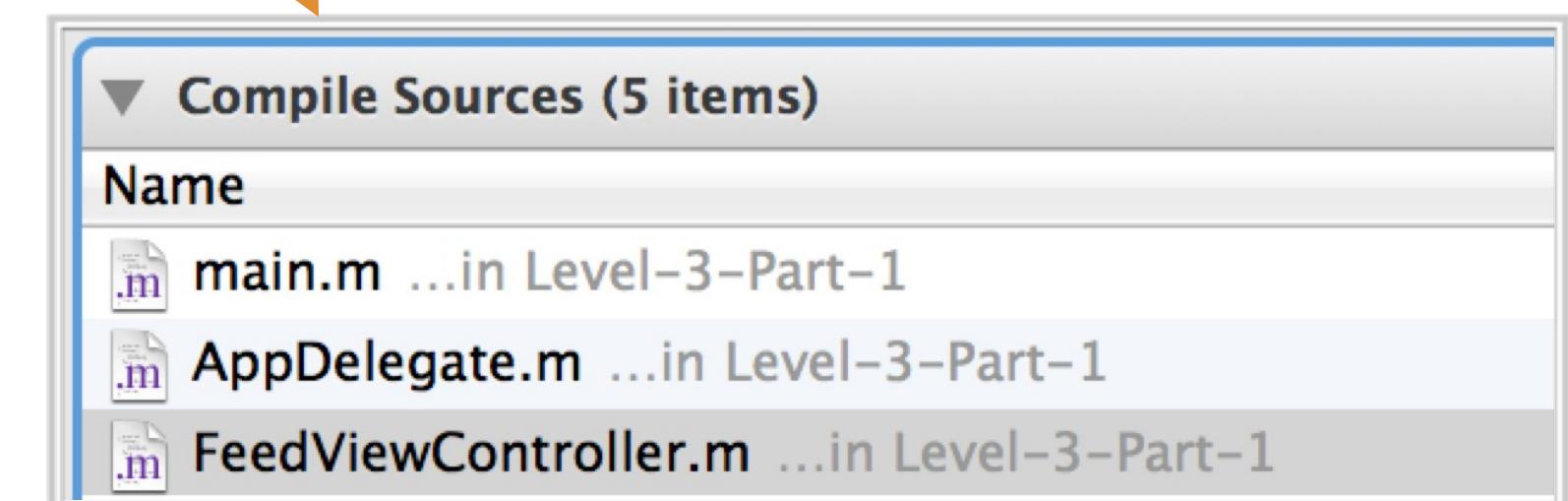
...



Create a New Class



FeedViewController.m
FeedViewController.h



Automatically adds this to files
to compile when you build



Using our new FeedViewController

AppDelegate.m

```
#import "AppDelegate.h"

@implementation AppDelegate

- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
     UIViewController *feedViewController = [[UIViewController alloc] init];
    feedViewController.title = @"Feed";
    feedViewController.tabBarItem.image = [UIImage imageNamed:@"tab_icon_feed"];
}
```

old code



We need to:

Import the new Class

Use the new Class



Using our new FeedViewController

AppDelegate.m

```
#import "AppDelegate.h"
#import "FeedViewController.h"

@implementation AppDelegate

- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    FeedViewController *feedViewController = [[FeedViewController alloc] init];
    feedViewController.title = @"Feed";
    feedViewController.tabBarItem.image = [UIImage imageNamed:@"tab_icon_feed"];
}
```



Move ViewController behavior into the ViewController

Inside our new FeedViewController



FeedViewController.m

```
...
@implementation FeedViewController
- (id)initWithNibName:(NSString *)NibNameOrNil bundle:(NSBundle *)nibBundleOrNilOrNil
{
    self = [super initWithNibName:nibNameOrNilOrNil bundle:nibBundleOrNilOrNil];
    if (self) {
        // Custom initialization
        self.title = @"Feed";
        self.tabBarItem.image = [UIImage imageNamed:@"tab_icon_feed"];
    }
    return self;
}
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    self.view.backgroundColor = [UIColor blueColor];
```

Initialization method (yeah, it's ugly)



.ios



Let's add a Picture In our Profile

ProfileViewController.m

```
...
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    self.view.backgroundColor = [UIColor blueColor];
```



```
UIImageView *greggView = [[UIImageView alloc] initWithImage:
    [UIImage imageNamed:@"gregg.jpg"]];
[greggView setContentMode:UIViewContentModeScaleAspectFit];
greggView.frame = self.view.frame;
[self.view addSubview:greggView];
```



Let's Break this Down



ProfileViewController.m

...



Creates a view for displaying an image



```
UIImageView *greggView = [[UIImageView alloc] initWithImage:  
[UIImage imageNamed:@"gregg.jpg"]];
```

If not a png we need to specify image extension

```
[greggView setContentMode:UIViewContentModeScaleAspectFit];
```

Determines how a view lays out its content

```
greggView.frame = self.view.frame;
```

The frame rectangle, which describes the view's location
and size in its superview's coordinate system

```
[self.view addSubview:greggView];
```

iOS

Let's Build Out Our Profile



Profile Image



Name



City



Biography



Member Since





Resizing Our Picture

ProfileViewController.m

```
...
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    self.view.backgroundColor = [UIColor blueColor];

    UIImageView *greggView = [[UIImageView alloc] initWithImage:
        [UIImage imageNamed:@"gregg.jpg"]];
    [greggView setContentMode:UIViewContentModeScaleAspectFit];
    greggView.frame = self.view.frame;
    [self.view addSubview:greggView];
```

Old Code



We need a smaller Profile Image



Resizing Our Picture

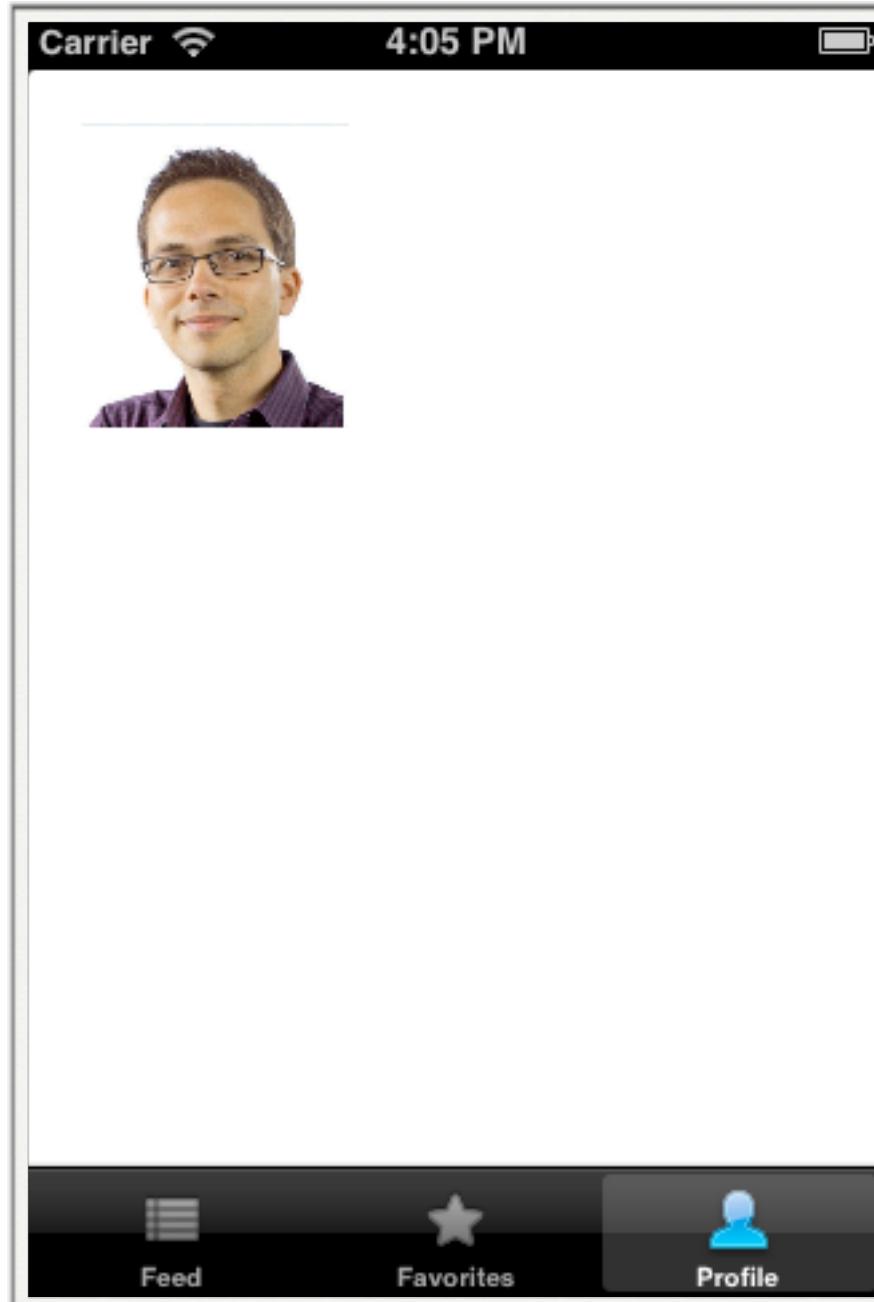
ProfileViewController.m

```
...
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    self.view.backgroundColor = [UIColor blueColor];

    UIImageView *greggView = [[UIImageView alloc] initWithImage:
        [UIImage imageNamed:@"gregg.jpg"]];
    greggView.frame = CGRectMake(20,20,100,114);
    [self.view addSubview:greggView];
```



Building Out Our Profile



Profile Image



Name

UILabel



City

UILabel



Biography

UITextView



Member Since

UILabel

Resizing Our Picture

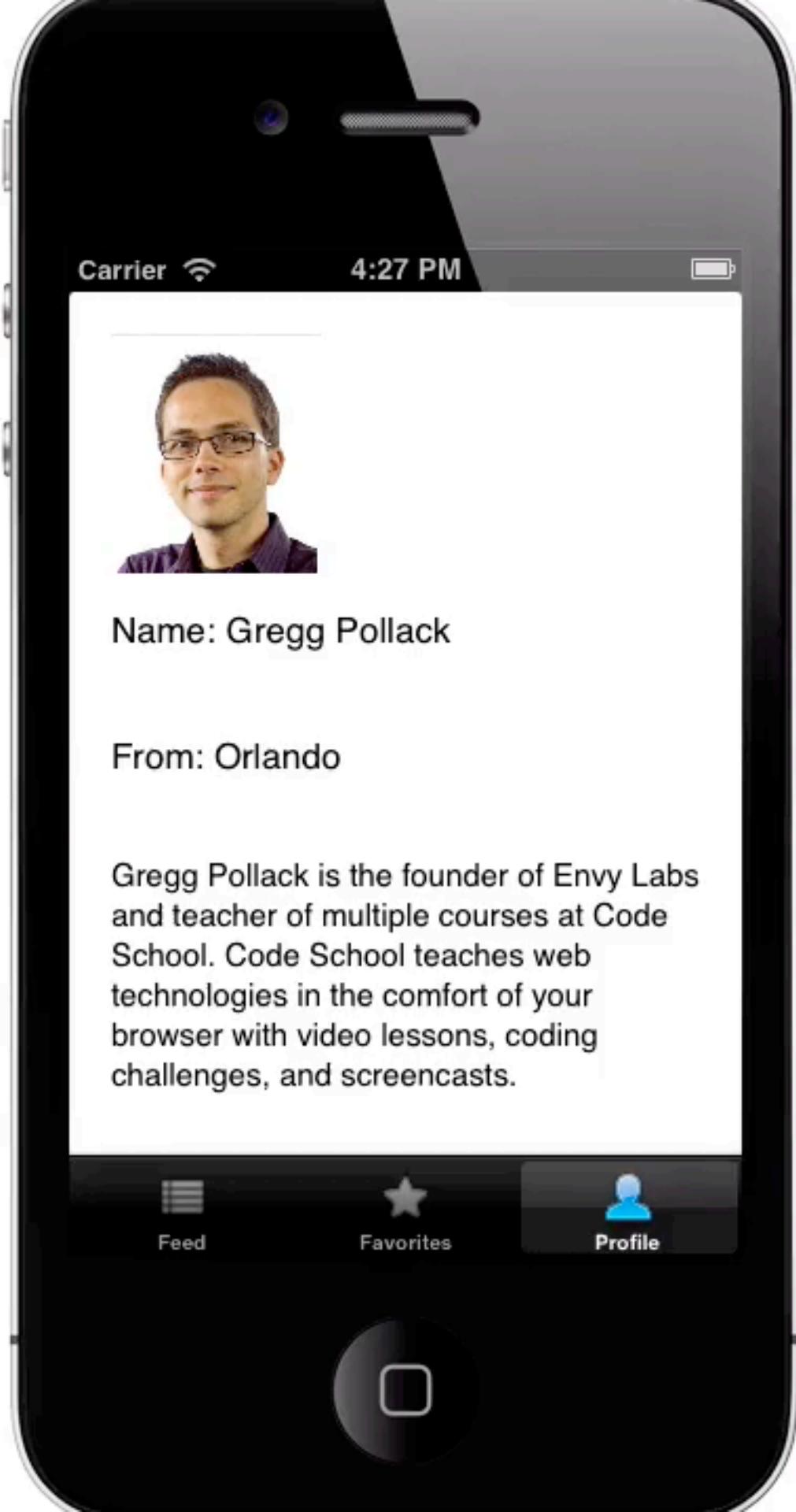


ProfileViewController.m

```
...  
  
UILabel *nameLabel = [[UILabel alloc] initWithFrame:CGRectMake(20,140,280,40)];  
nameLabel.text = @"Name: Gregg Pollack";  
[self.view addSubview:nameLabel];  
  
UILabel *cityLabel = [[UILabel alloc] initWithFrame:CGRectMake(20,200,280,40)];  
cityLabel.text = @"From: Orlando";  
[self.view addSubview:cityLabel];  
  
UITextView *biography = [[UITextView alloc] initWithFrame:CGRectMake(12,260,300,180)];  
biography.font = [UIFont fontWithName:@"Helvetica" size:15];  
biography.editable = NO;  
biography.text = @"Gregg Pollack is the ...";  
[self.view addSubview:biography];  
  
UILabel *memberSinceLabel = [[UILabel alloc] initWithFrame:CGRectMake(20,440,280,40)];  
memberSinceLabel.text = @"November 2012";  
[self.view addSubview:memberSinceLabel];
```

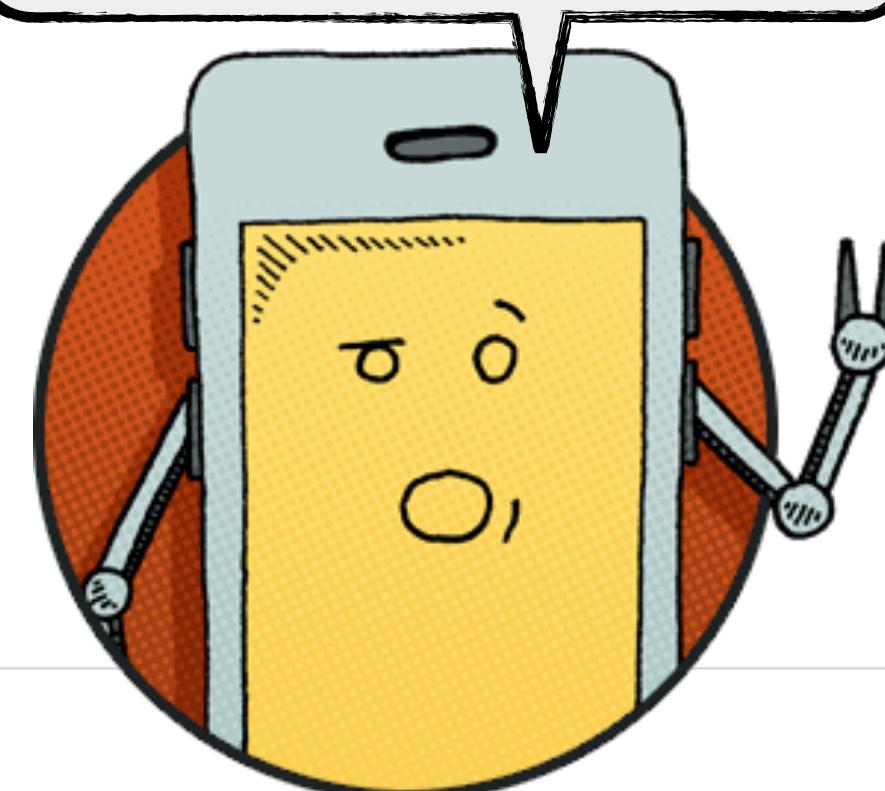


Our Profile Doesn't Scroll



Member Since

WHERE DID
MEMBER SINCE GO?



Try
iOS



Resizing Our Picture

ProfileViewController.h

```
@property (weak, nonatomic) UIScrollView *scrollView;
```

ProfileViewController.m

```
self.scrollView = [[UIScrollView alloc] initWithFrame:self.view.bounds];
self.scrollView.contentSize = CGSizeMake(320,480);
```

...

```
[self.scrollView addSubview:biography];
AddSubviews into the scrollView
```

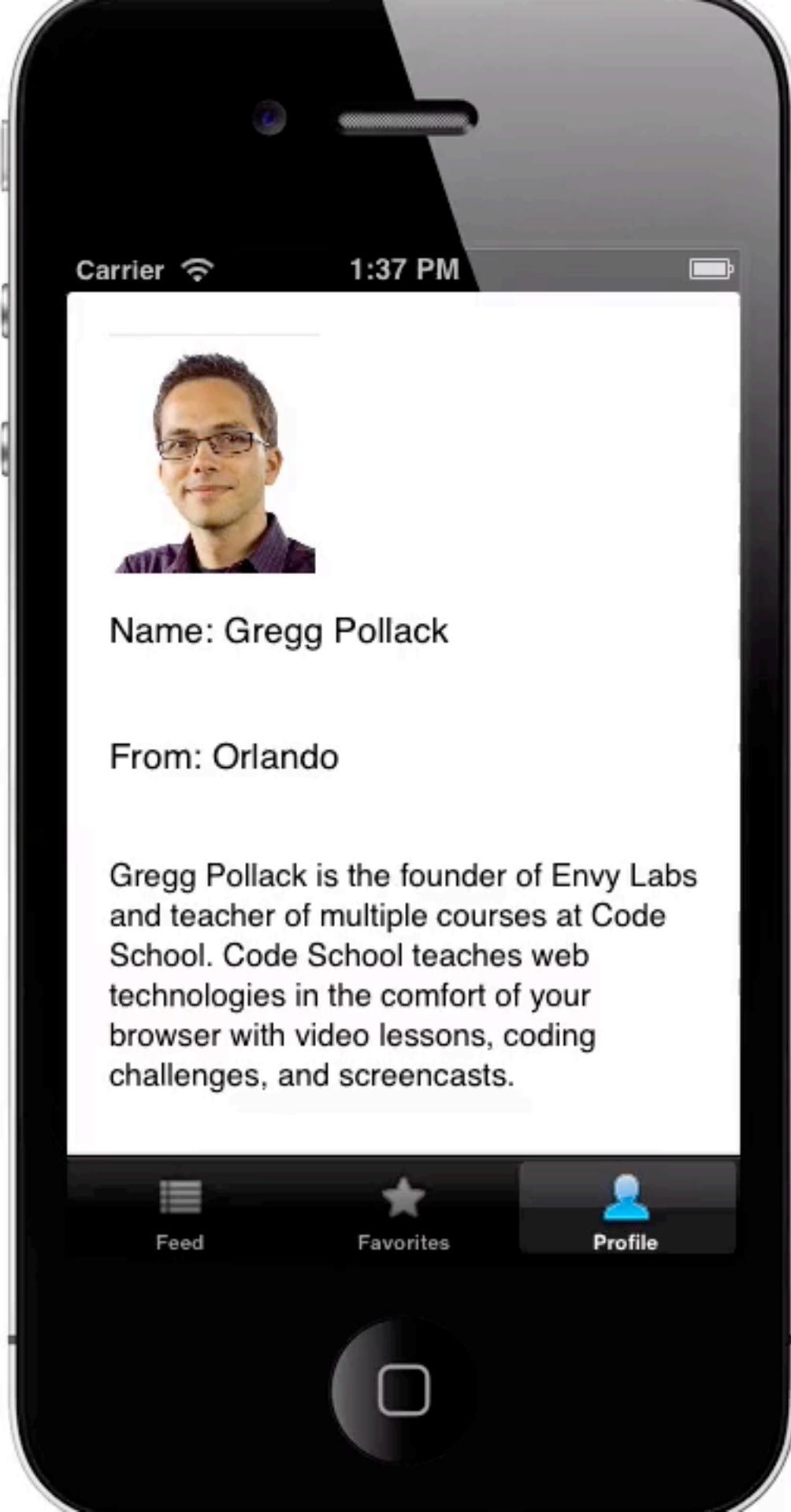
```
UILabel *memberSinceLabel = [[UILabel alloc] initWithFrame:CGRectMake(20,440,280,40)];
memberSinceLabel.text = @"November 2012";
[self.scrollView addSubview:memberSinceLabel];
```

```
[self.view addSubview:self.scrollView]; Add the scrollView onto the ViewController's View
```

$440 + 40$

height

Now with ScrollView

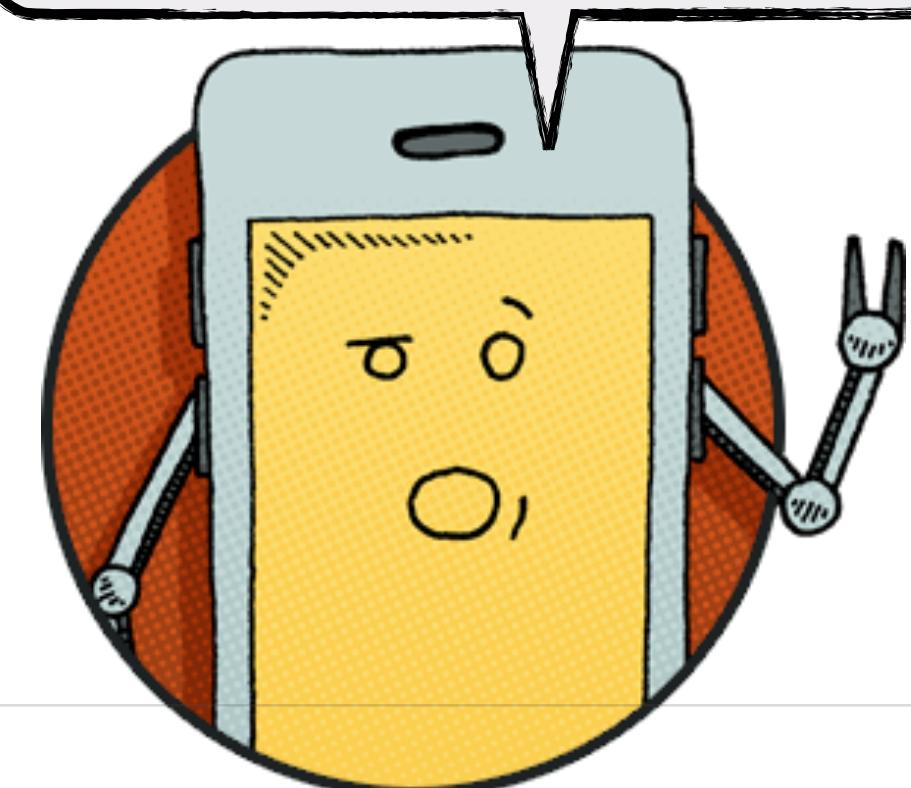


Name: Gregg Pollack

From: Orlando

Gregg Pollack is the founder of Envy Labs and teacher of multiple courses at Code School. Code School teaches web technologies in the comfort of your browser with video lessons, coding challenges, and screencasts.

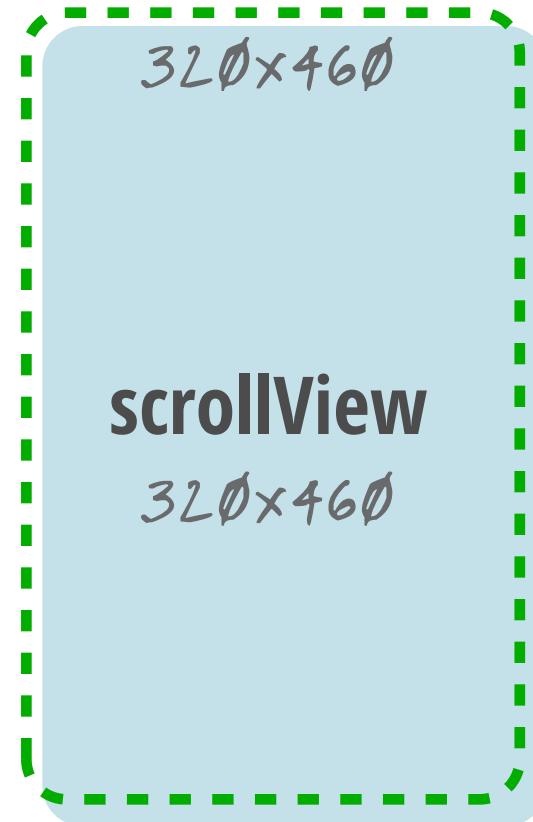
WHERE DID WE GO WRONG?
THE SCROLL DOESN'T STICK!



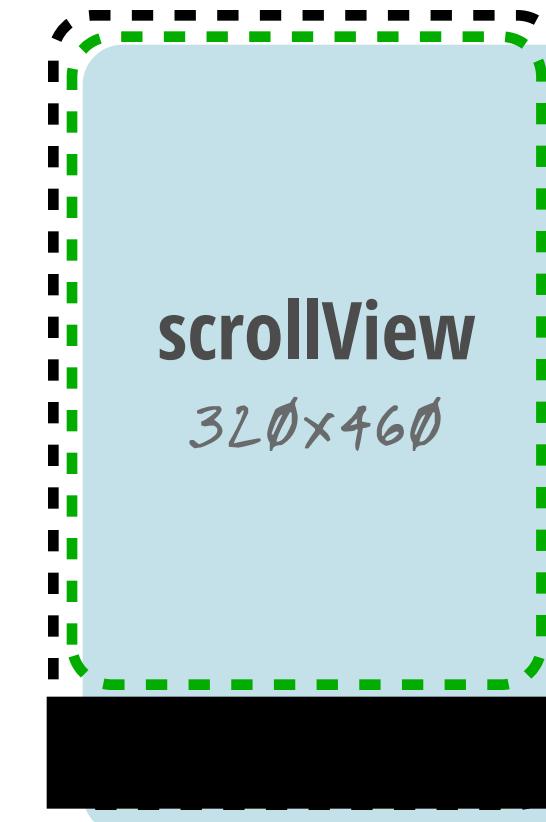
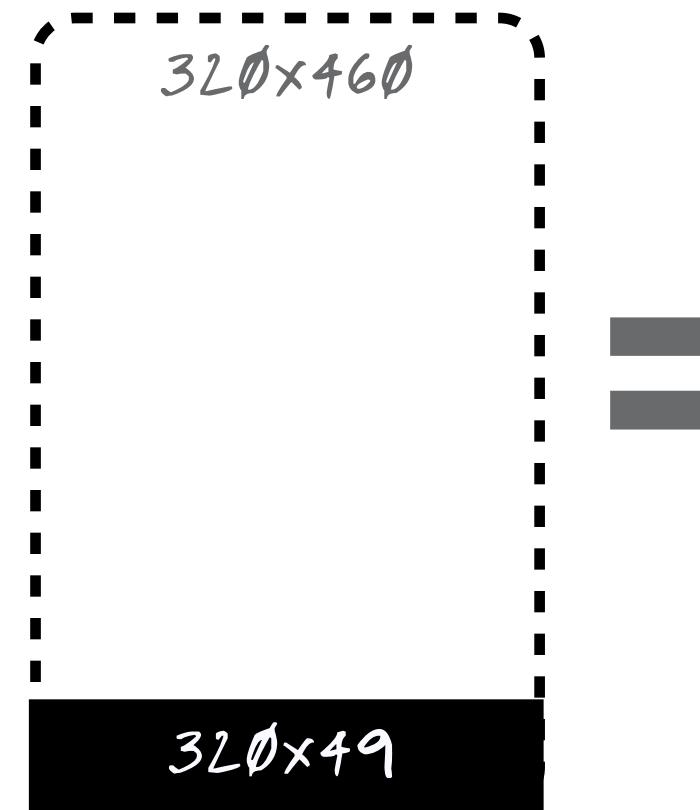


Resizing the ScrollView

profileViewController



tabBarController



profileViewController gets
resized to 320×411

scrollView Does not!



We need to tell scrollView what to do when it's superview resizes.



Setting the autoresizingMask

ProfileViewController.m

```
self.scrollView = [[UIScrollView alloc] initWithFrame:self.view.bounds];
self.scrollView.autoresizingMask = UIViewAutoresizingFlexibleHeight;
self.scrollView.contentSize = CGSizeMake(320, 480);
```

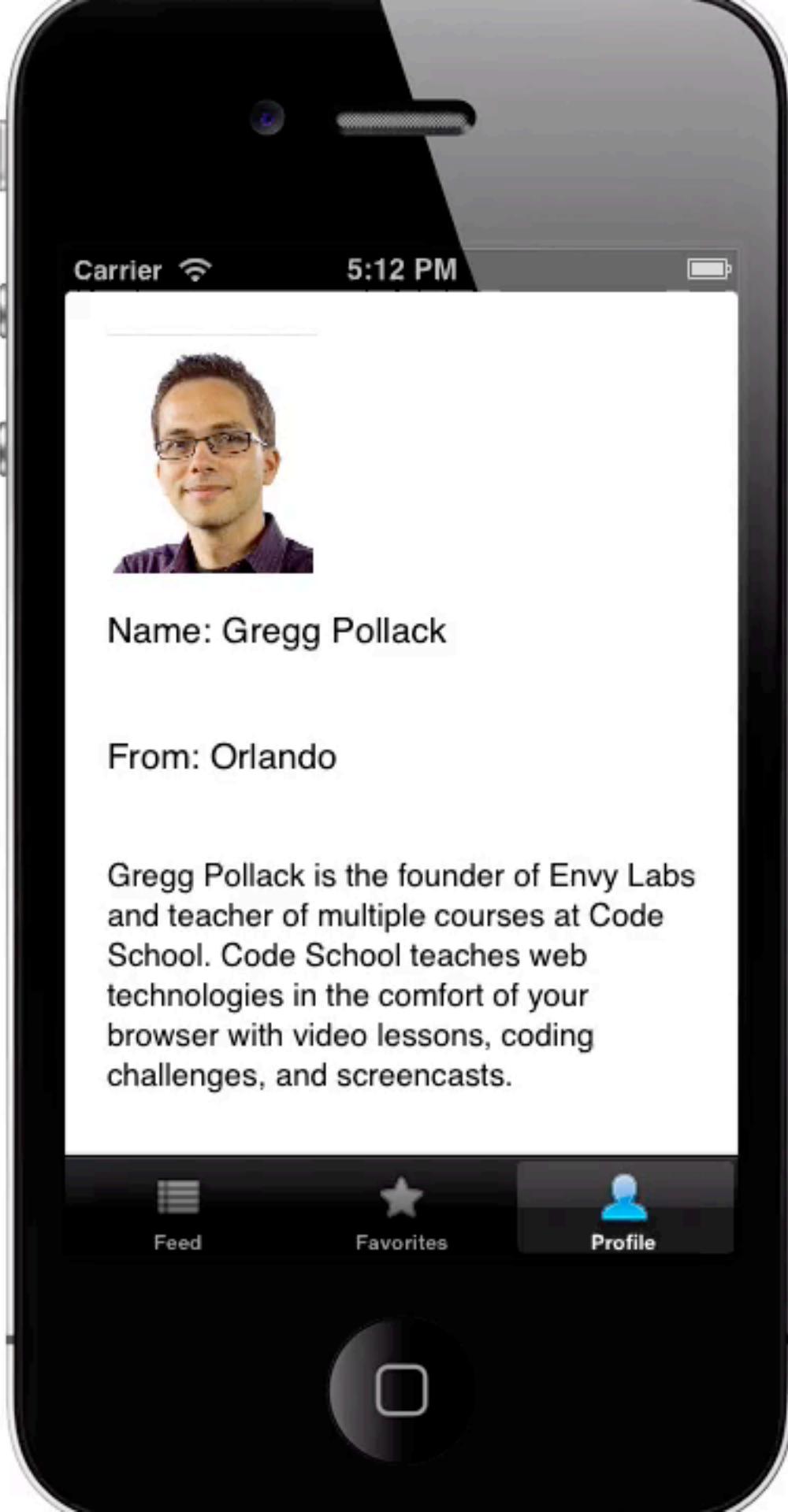
UIViewAutoresizingNone
UIViewAutoresizingFlexibleLeftMargin
UIViewAutoresizingFlexibleWidth
UIViewAutoresizingFlexibleRightMargin
UIViewAutoresizingFlexibleTopMargin
UIViewAutoresizingFlexibleHeight
UIViewAutoresizingFlexibleBottomMargin

To Resize also in Landscape View

```
self.scrollView.autoresizingMask =
    (UIViewAutoresizingFlexibleHeight | UIViewAutoresizingFlexibleWidth);
```



Now with UIScrollView!



Try
iOS

Level 4

UINavigationController
& Custom Buttons





01 Introduction to the UINavigationController

02 Creating a Custom Image Button

03 Creating View Controllers on the Fly

04 Using Both Navigation and TabBar Controllers

How else do we navigate ViewControllers?



SettingsViewController



GeneralViewController

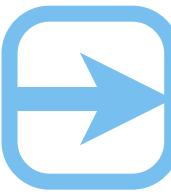


AboutViewController

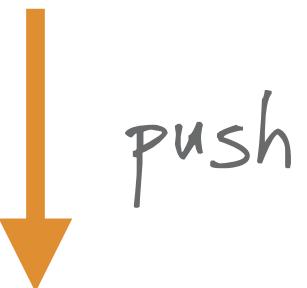
UINavigationController & Custom Buttons



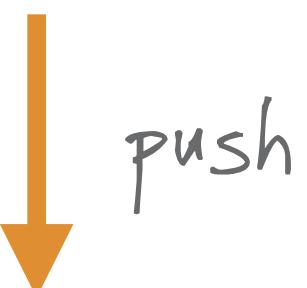
UINavigationController



SettingsViewController



GeneralViewController



AboutViewController

ns

Try
iOS



This is Our Old Code

AppDelegate.m

```
#import "AppDelegate.h"  
#import "FeedViewController.h"  
  
@implementation AppDelegate  
  
- (BOOL) application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    FeedViewController *feedViewController = [[FeedViewController alloc] init];  
  
    UITabBarController *tabBarController = [[UITabBarController alloc] init];  
  
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];  
  
    self.window.rootViewController = tabBarController;  
}
```

old code



Using the UINavigationController

AppDelegate.m

```
#import "AppDelegate.h"
#import "FeedViewController.h"

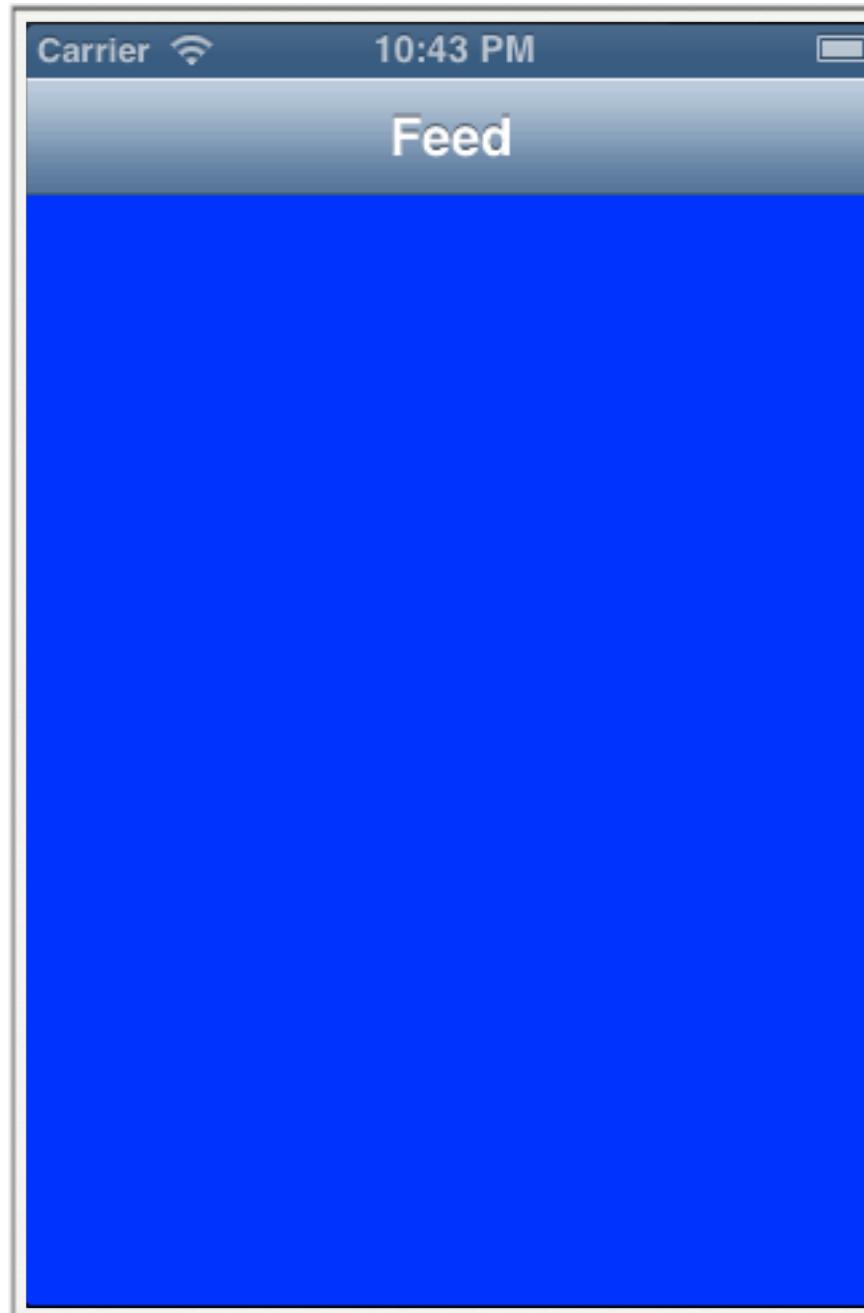
@implementation AppDelegate

- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    FeedViewController *feedViewController = [[FeedViewController alloc] init];

    UINavigationController *navController = [[UINavigationController alloc]
                                             initWithRootViewController:feedViewController];
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    self.window.rootViewController = navController;
}
```



First Screen of our UINavigationController



Takes the name of the controller by default

We can push to a new controller with

```
[self.navigationController
```

```
    pushViewController:<New View Controller> animated:YES];
```

FeedViewController



push

FavoritesViewController



Navigating to the FavoritesViewController

FeedViewController.m

```
...
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    self.view.backgroundColor = [UIColor blueColor];

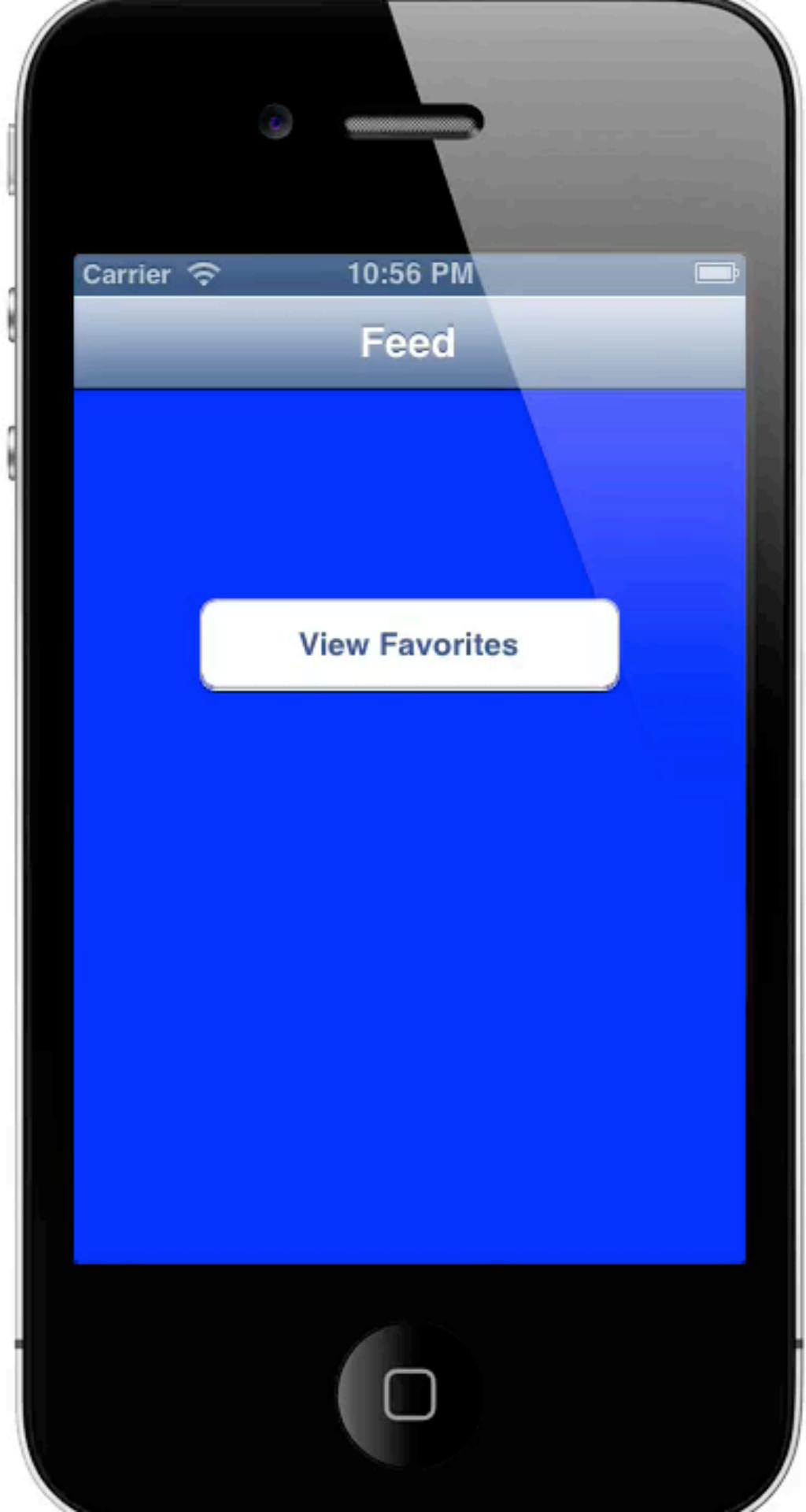
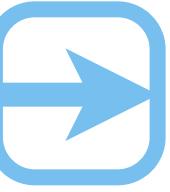
     UIButton *favoritesButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    favoritesButton.frame = CGRectMake(60, 100, 200, 44);
    [favoritesButton setTitle:@"View Favorites" forState:UIControlStateNormal];
    [self.view addSubview:favoritesButton];

    [favoritesButton addTarget:self action:@selector(showFavorites:)
                           forControlEvents:UIControlEventTouchUpInside];
}

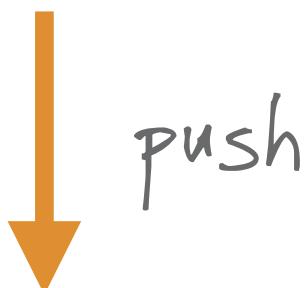
- (void)showFavorites:(UIButton *)sender
{
     FavoritesViewController *favoritesViewController =
        [[FavoritesViewController alloc] init];
     [self.navigationController pushViewController:favoritesViewController
                                         animated:YES];
}
```

OS

With UINavigationController

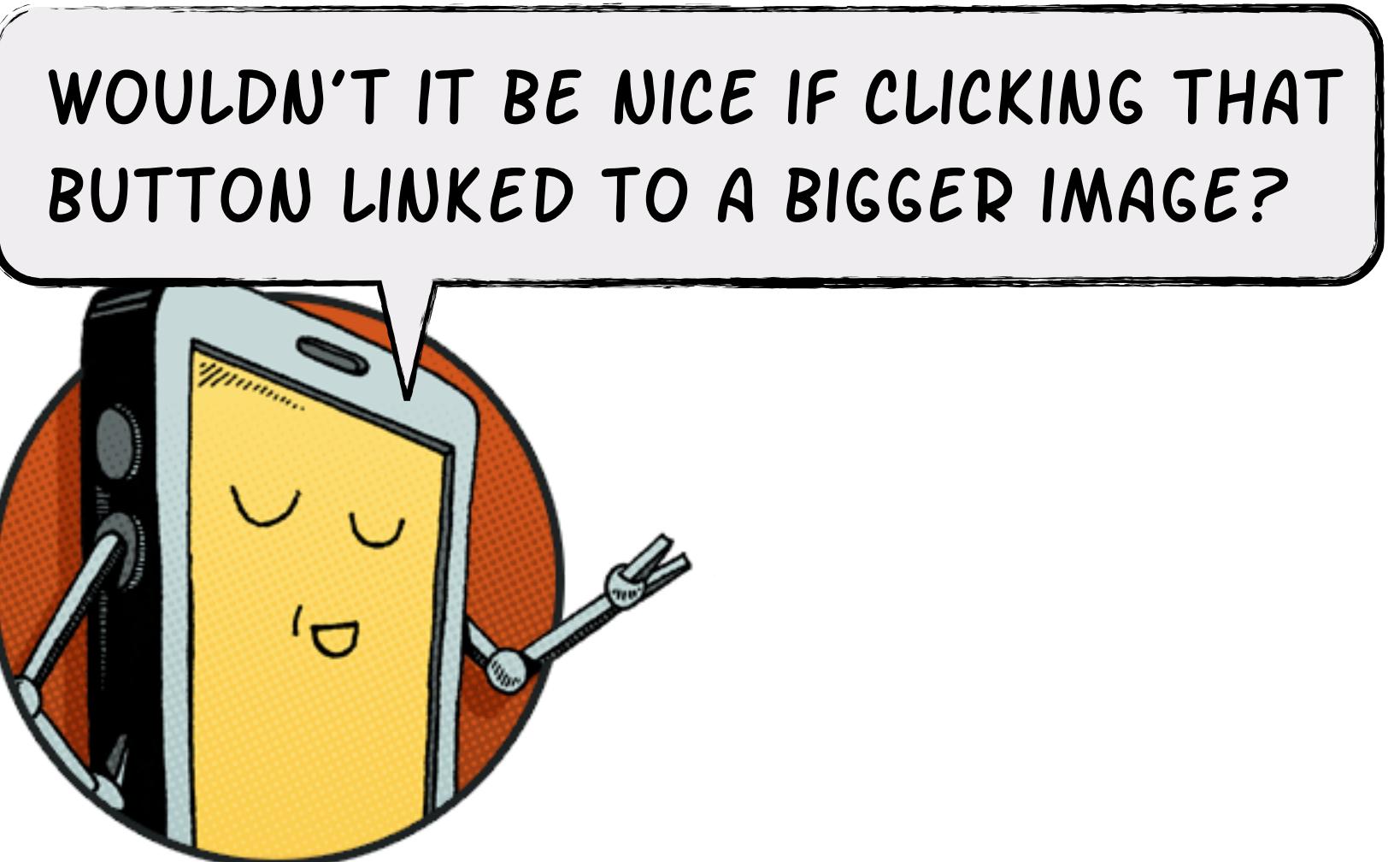


FeedViewController



FavoritesViewController

Custom Buttons





Changing an Image to a Button

FavoritesViewController.m

```
...
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    self.view.backgroundColor = [UIColor blueColor];

    UIImageView *tryiosView = [[UIImageView alloc] initWithImage:
        [UIImage imageNamed:@"tryios.gif"]];
    [tryiosView setContentMode:UIViewContentModeScaleAspectFit];
    tryiosView.frame = self.view.frame;
    [self.view addSubview:tryiosView];
```

old code



Lets add a Picture In our Profile



FavoritesViewController.m

```
...
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    self.view.backgroundColor = [UIColor blueColor];

    UIButton *tryiosView = [UIButton buttonWithType:UIButtonTypeCustom];
    [tryiosView setImage:[UIImage imageNamed:@"tryios.gif"]
        forState:UIControlStateNormal];
    [tryiosView setImage:[UIImage imageNamed:@"tryios.gif"]
        forState:UIControlStateHighlighted];
    [tryiosView setFrame:CGRectMake(15, 15, 100, 100)];
    [self.view addSubview:tryiosView];
    [tryiosView addTarget:self
        action:@selector(showZoomedPicture:)
        forControlEvents:UIControlEventTouchUpInside];
```



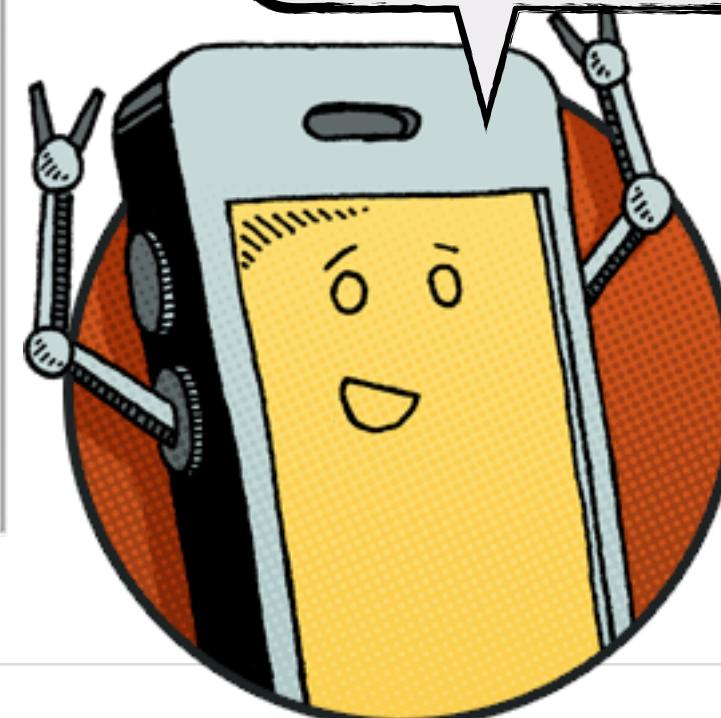
y
OS

We have an Image Button



But now we need a ViewController filled with
an image to pass control to....

BUT HOLD YOUR HORSES,
YOU DON'T NEED ANOTHER
SUBCLASS!





Lets push to a New ViewController

FavoritesViewController.m

```
...
- (void)showZoomedPicture:(UIButton *)sender
{
    UIViewController *tryiosImageViewController = [[UIViewController alloc] init];
    tryiosImageViewController.view.frame = self.view.frame;
    tryiosImageViewController.title = @"Try IOS Logo";

    UIImageView *tryiosView = [[UIImageView alloc]
                                initWithImage:[UIImage imageNamed:@"tryios.gif"]];
    [tryiosView setContentMode:UIViewContentModeScaleAspectFit];

    tryiosView.frame = tryiosImageViewController.view.frame;
    [tryiosImageViewController.view addSubview:tryiosView];

    [self.navigationController pushViewController:tryiosImageViewController
                                         animated:YES];
}
```

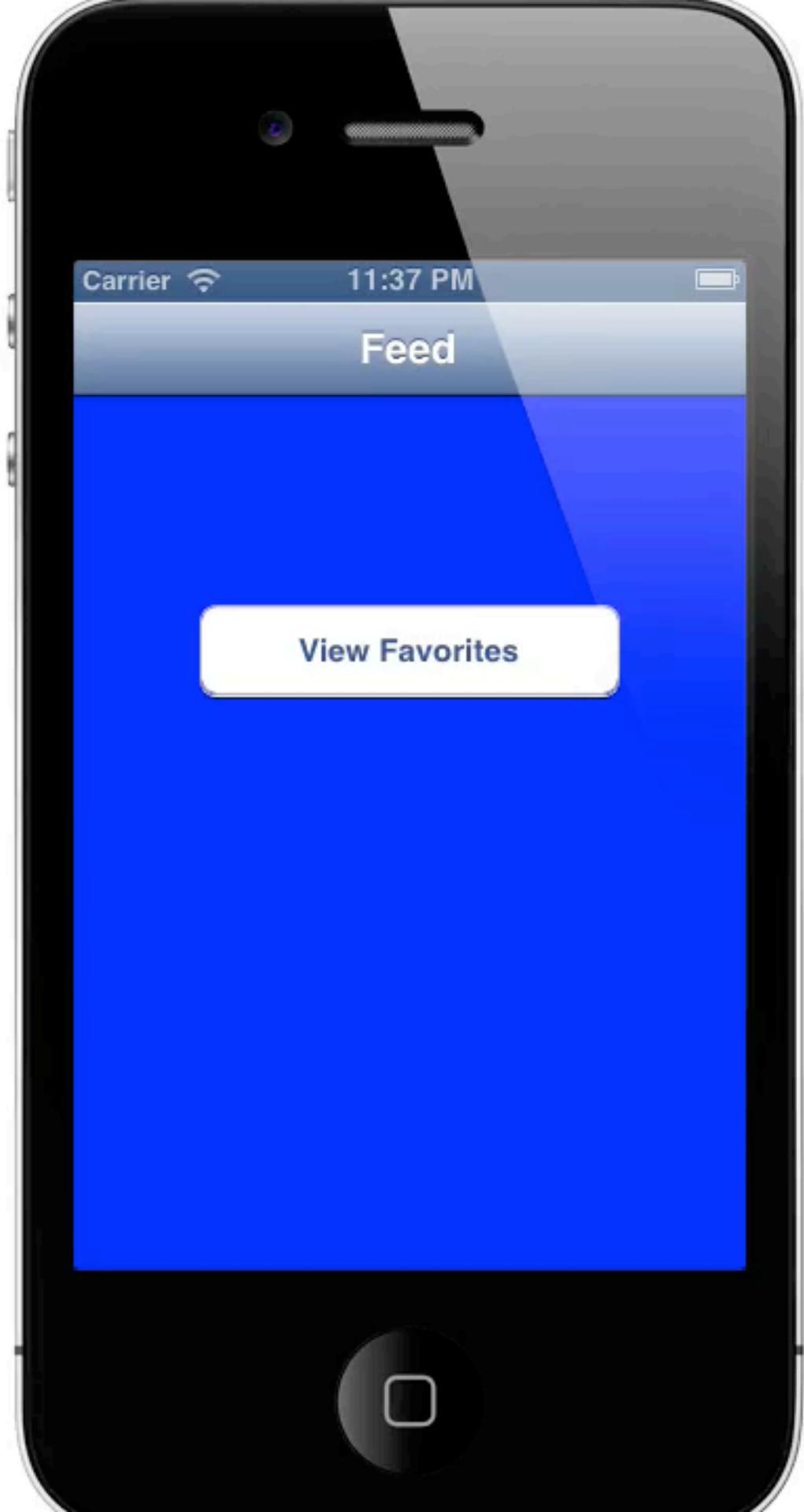
Create a ViewController on the fly!



Push Control to our new ViewController



With Our Custom Button



S

FeedViewController

push

FavoritesViewController

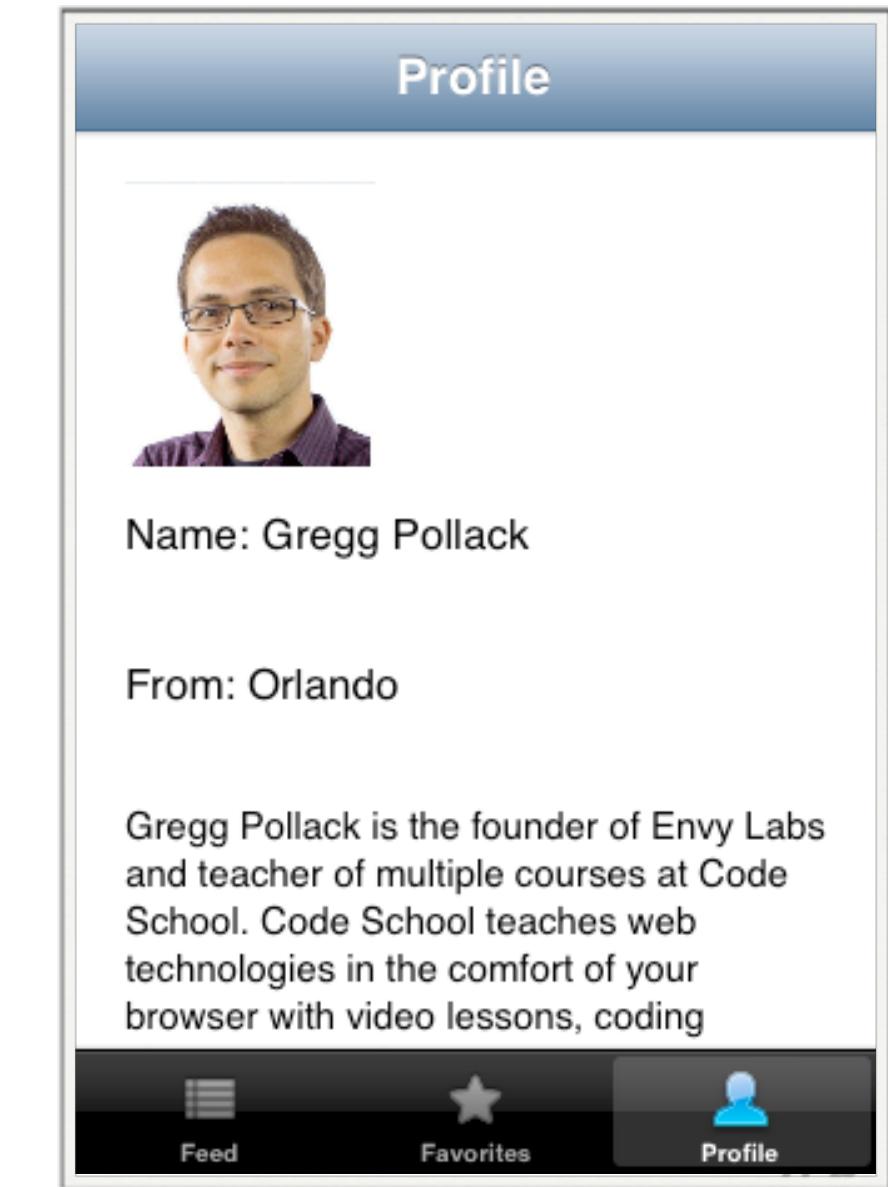
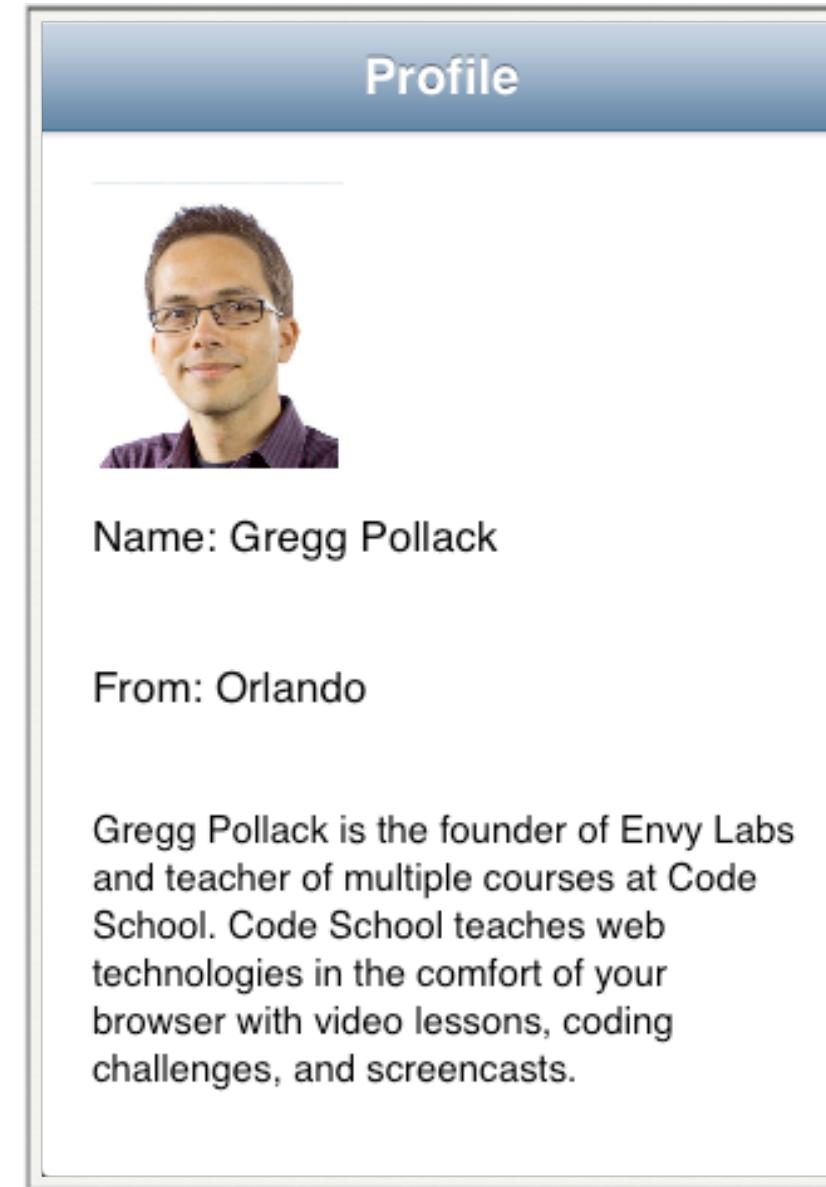
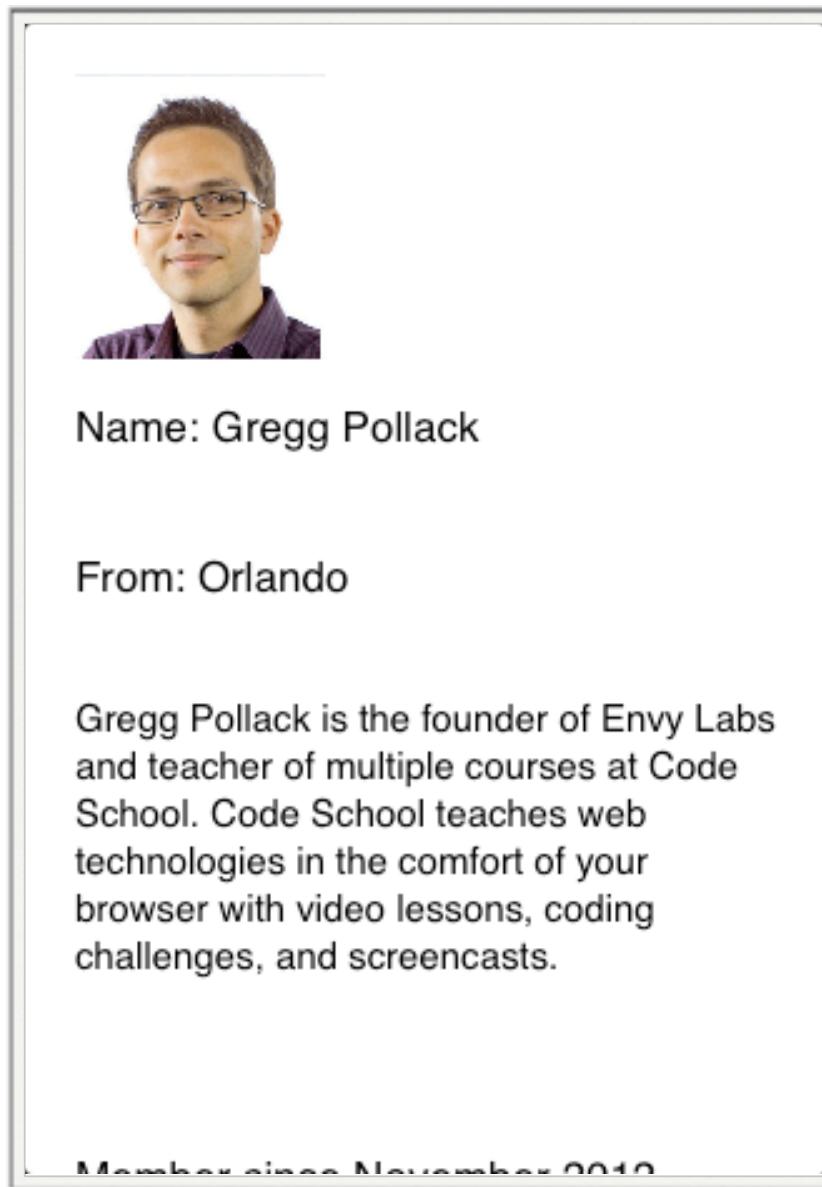
push

tryiosImageViewController

Lets Combine Both Nav + Tab Containers



wrap inside
ProfileViewController → ProfileNavController → tabBarController

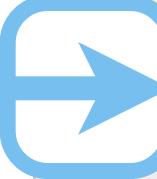




Using the UINavigationController

AppDelegate.m

```
@implementation AppDelegate

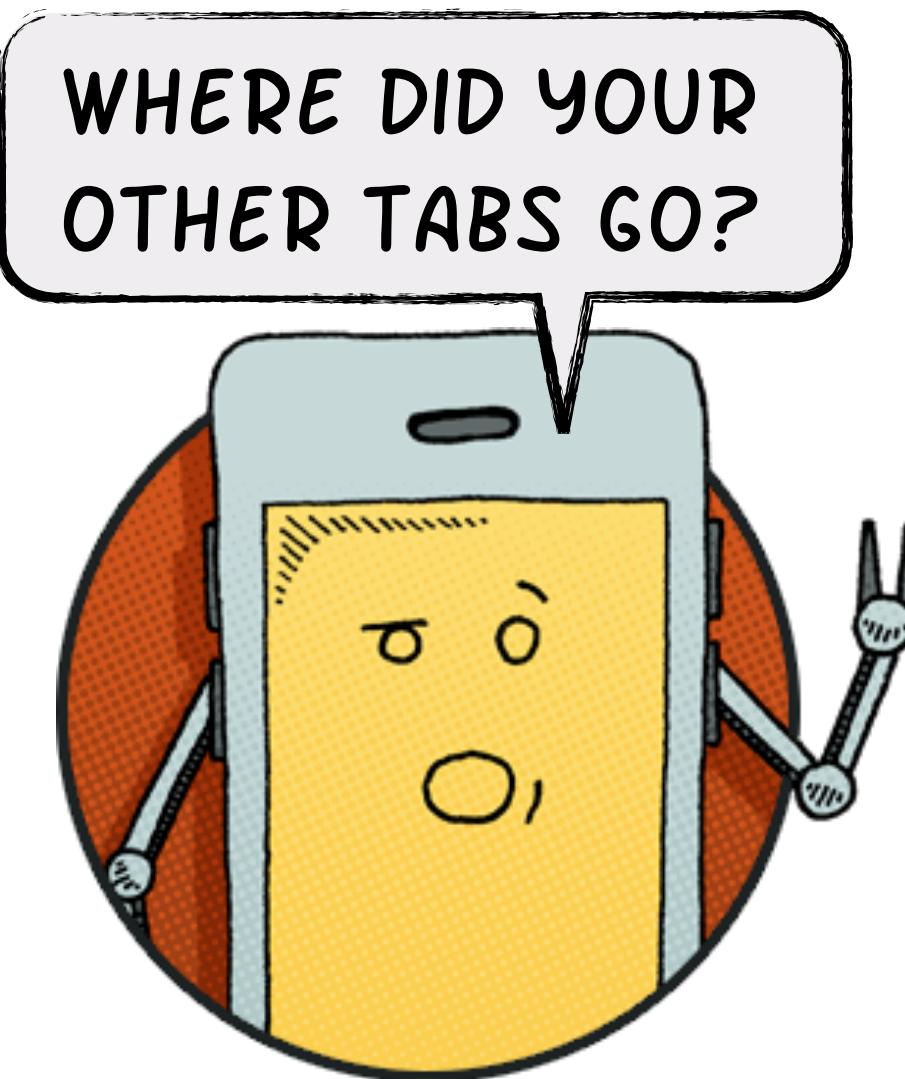
- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
     ProfileViewController *profileViewController = [[ProfileViewController alloc] init];
     UINavigationController *profileNavController = [[UINavigationController alloc]
                                                initWithRootViewController:profileViewController];
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    UITabBarController *tabBarController = [[UITabBarController alloc] init];
     [tabBarController setViewControllers:@[profileNavController]];
    self.window.rootViewController = tabBarController;
}
```

Both Containers Together!



Carrier 5:30 PM

Profile

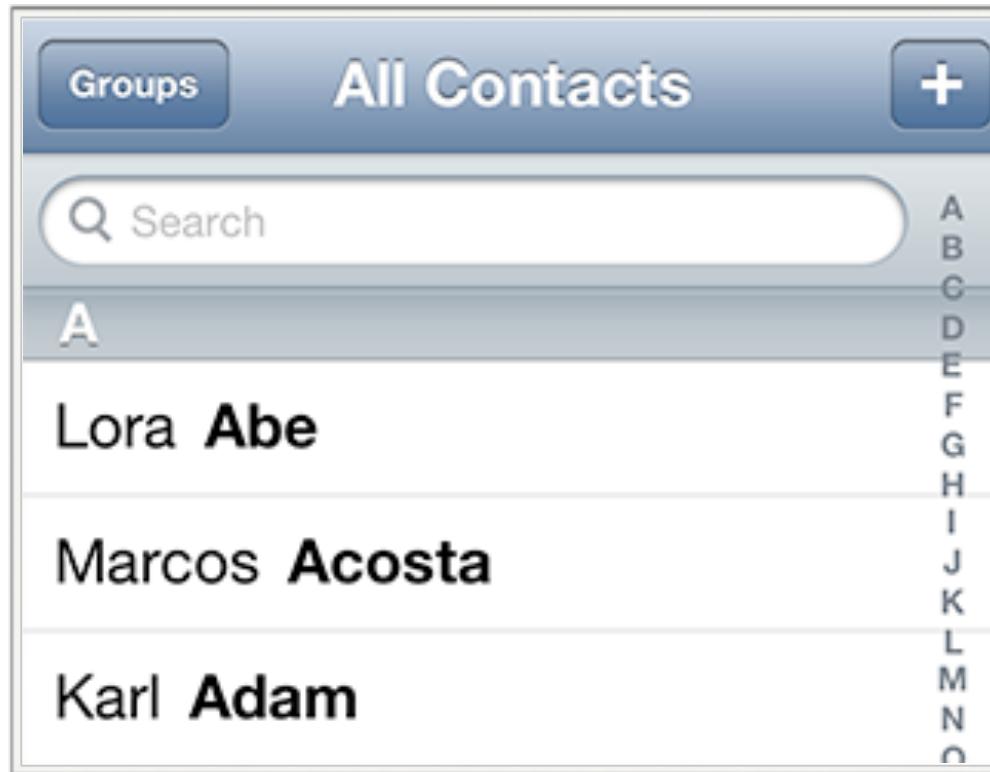


UITableView Examples

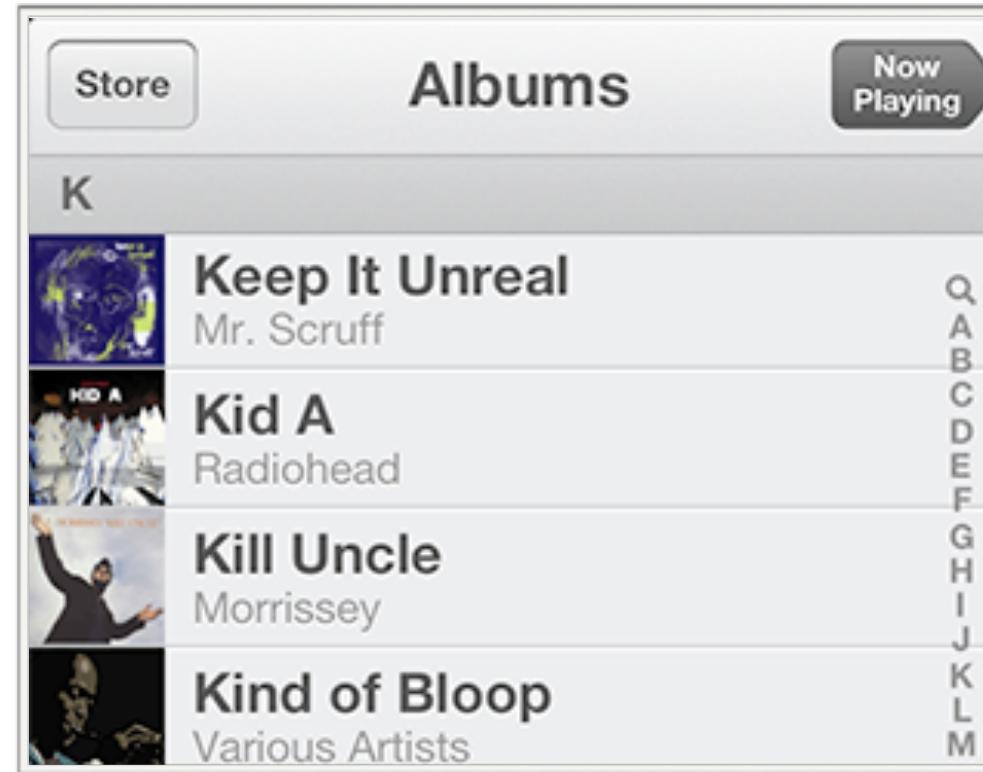


UITableViews are great for displaying lists of data

Address Book



Music



Messages



IT'S HARD TO FIND A NON-GAME APP
THAT DOESN'T CONTAIN AT LEAST ONE
UITABLEVIEW.

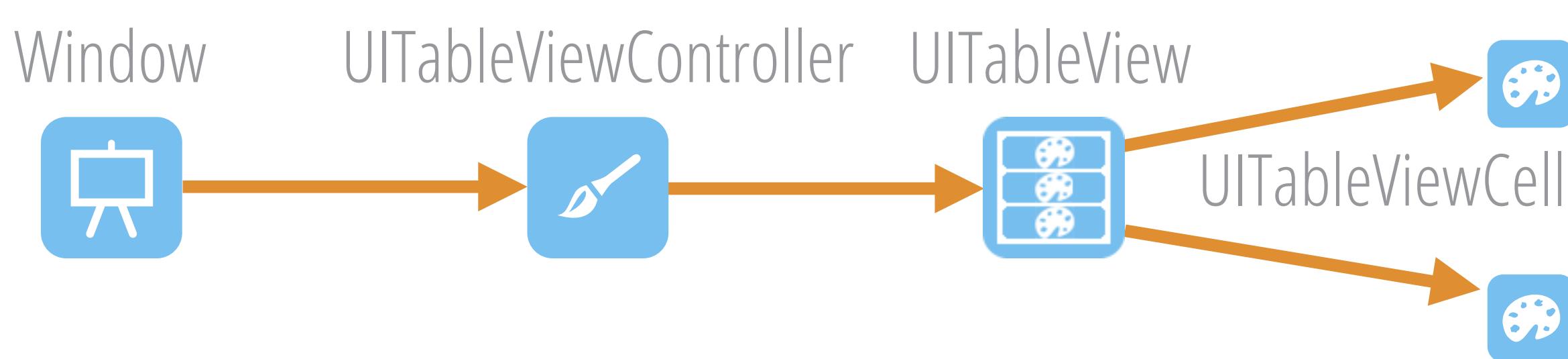


Table
Views

Try
iOS



Introduction to the UITableViewController

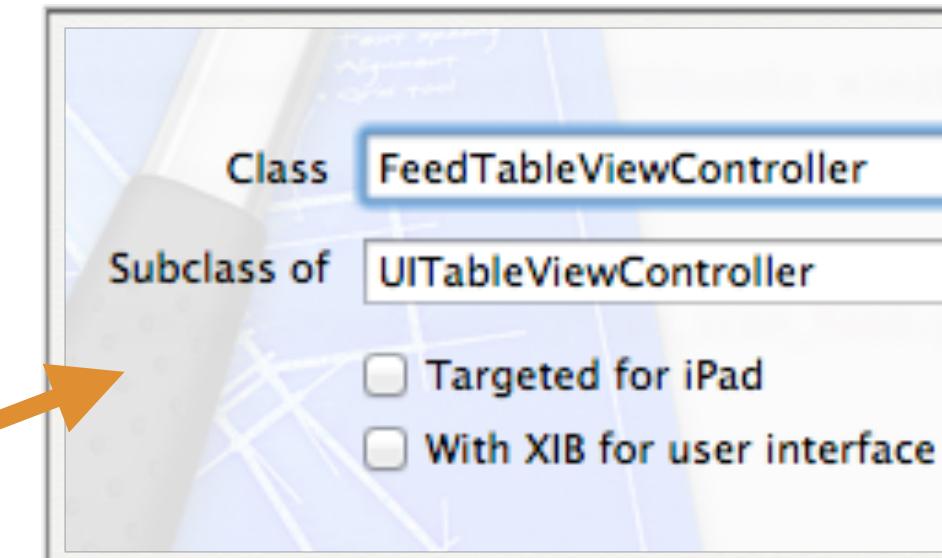
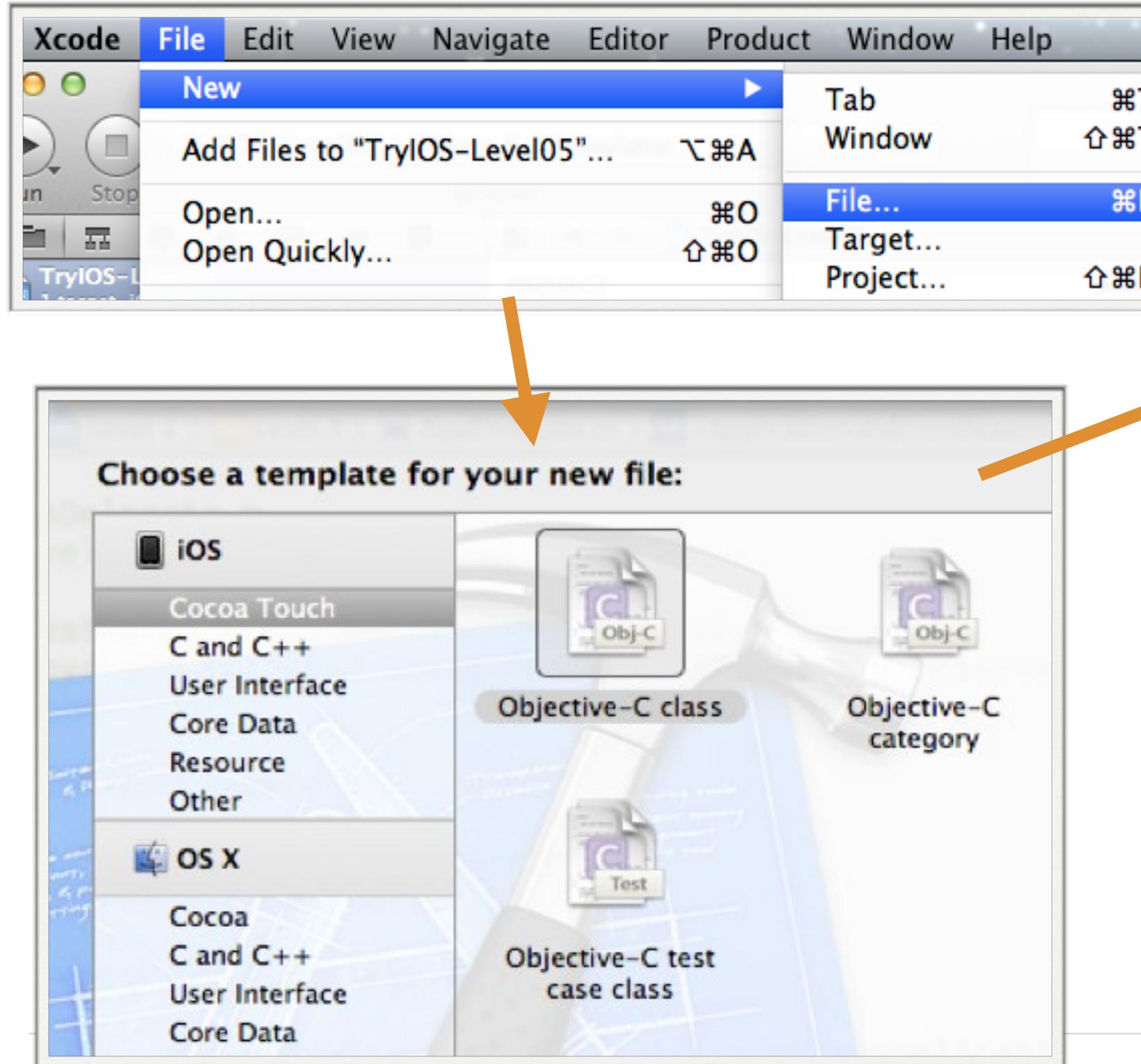


Controller that displays a UITableView

Inherits from the superclass UIViewController

California
Brea
Burlingame
Canoga Park
Carlsbad
Chula Vista
Corte Madera
Costa Mesa
Emeryville
Escondido
Section Footer

Create a UITableViewController class



FeedTableViewController.h
FeedTableViewController.m



Adding a Table View Controller

AppDelegate.m

```
#import "AppDelegate.h"
#import "FeedViewController.h"

@implementation AppDelegate

- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    ...
    FeedViewController *feedViewController = [[feedViewController alloc] init];
    UINavigationController *feedNavController = [[UINavigationController alloc]
        initWithRootViewController:feedViewController];
    ...
}
```

old code 

Need to replace FeedViewController with FeedTableViewController



Adding a Table View Controller

AppDelegate.m

```
#import "AppDelegate.h"
#import "FeedTableViewController.h"

@implementation AppDelegate

- (BOOL) application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    ...
    FeedTableViewController *feedTableViewController = [[feedTableViewController alloc]
        initWithStyle:UITableViewStylePlain];
    ...
    UINavigationController *feedNavController = [[UINavigationController alloc]
        initWithRootViewController:feedTableViewController];
    ...
}
```

UITableViewStyleGrouped Alternate style : bubble shaped cells like on settings

Required Methods for UITableViewController



FeedTableViewController.h
FeedTableViewController.m

Start with a lot of boilerplate code

`numberOfSectionsInTableView:`

Returns the number of sections

`tableView:numberOfRowsInSection:`

Returns the number of rows (cells)

`tableView:cellForRowIndexPath:`

Initialize and setup each cell in your tableView



Required Methods for UITableViewController

FeedTableViewController.m

```
...
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView
 numberOfRowsInSection:(NSInteger)section {
    return 5;
}

- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath
```

Returns the number of sections

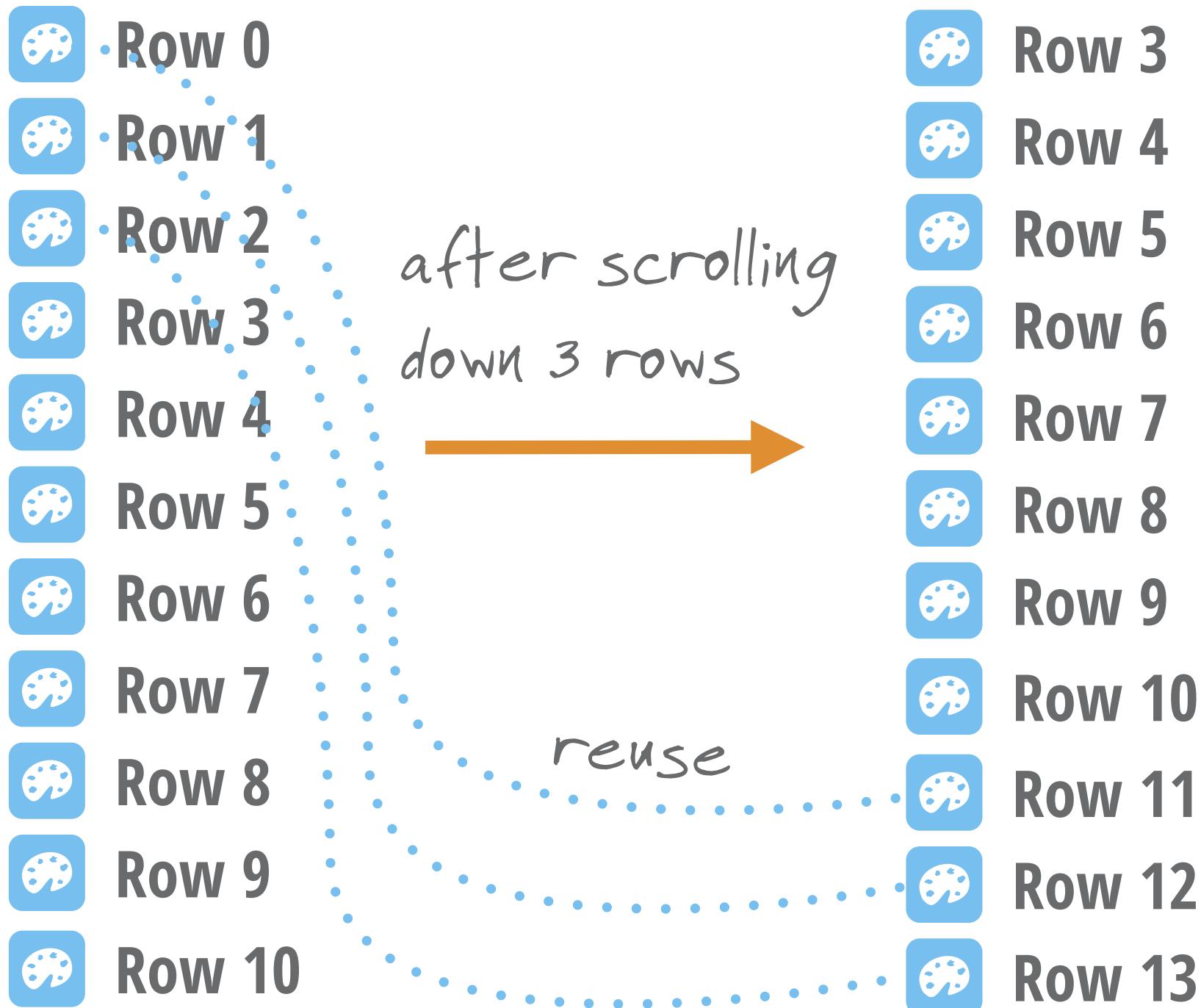
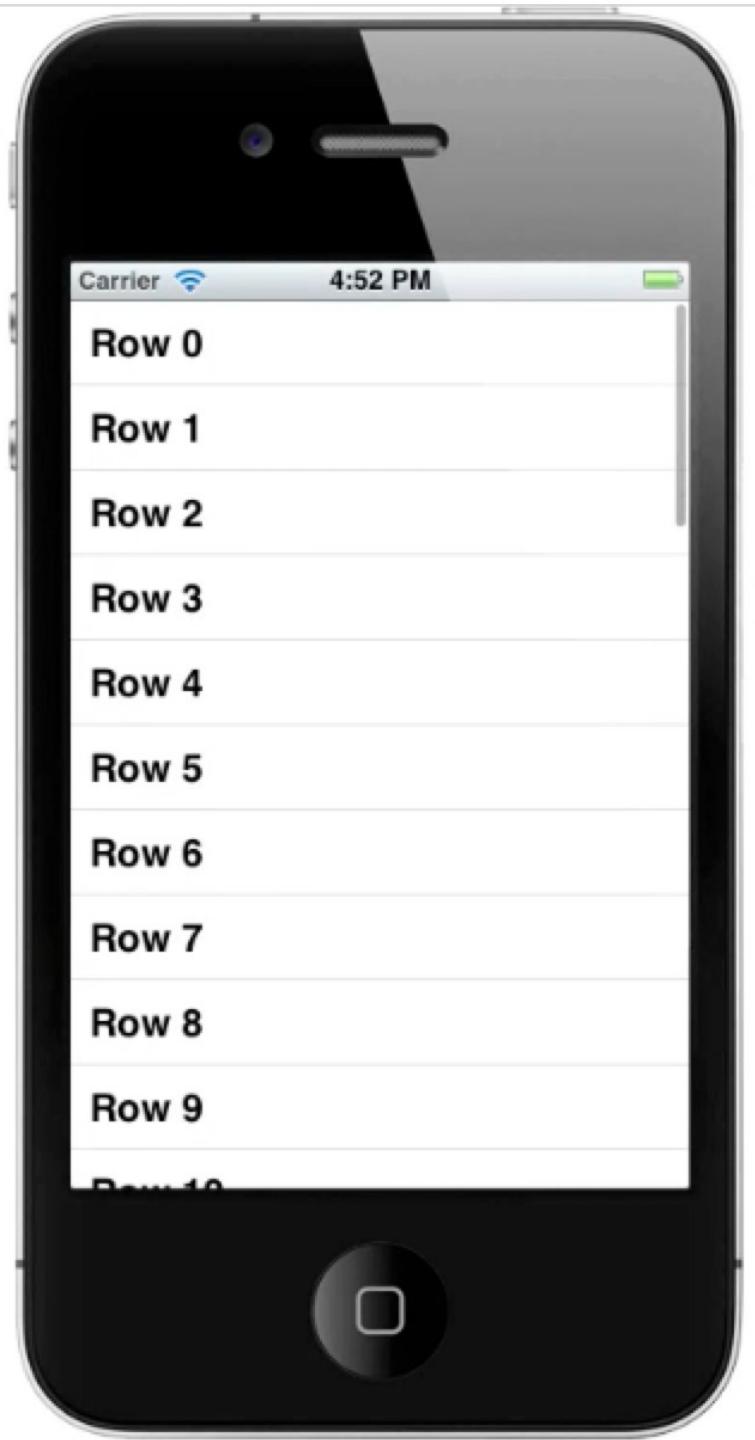
Returns the number of rows (cells)

Initialize and setup each cell in your tableView

Reusing UITableView Cells



UITableViewCell





Digging into tableView:cellForRowAtIndexPath:

FeedTableViewController.m This method runs one time for each row, when it becomes visible

```
- (UITableViewCell *)tableView:(UITableView *)tableView  
    cellForRowAtIndexPath:(NSIndexPath *)indexPath  
{  
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"Title"];  
    Check to see if there are any cells of style "Title" we can reuse  
  
    if(cell == nil) {  
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault  
                                reuseIdentifier:@"Title"];  
    }  
    Nope, no reusable cell, so lets allocate a new one  
  
    return cell;  
}
```



Setting the text in a UITableViewCell

Documentation for UITableViewCell

textLabel

Returns the label used for the main textual content of the table cell. (read-only)

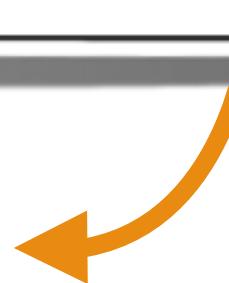
```
@property(nonatomic, readonly, retain) UILabel *textLabel
```

Documentation for UILabel

text

The text displayed by the label.

```
@property(nonatomic, copy) NSString *text
```



```
cell.textLabel.text = @"my awesome cell";
```



Reviewing NSArray

Creating an NSArray of NSStrings

```
NSArray *names = @[@"Gregg", @"Eric", @"Chris", @"Jon"];
```

Reading values from an NSArray

```
names[1]; // reads the 2nd object in the array, or "Eric"
```

```
int myInt = 3;  
names[myInt]; // reads the 4th object in the array, or "Jon"
```

Determining the length of an NSArray

```
names.count; // returns 4
```



Detecting when a cell is tapped

FeedTableViewController.m

```
- (void) tableView:(UITableView *)tableView  
didSelectRowAtIndexPath:(NSIndexPath *)indexPath  
{  
}
```

indexPath.row

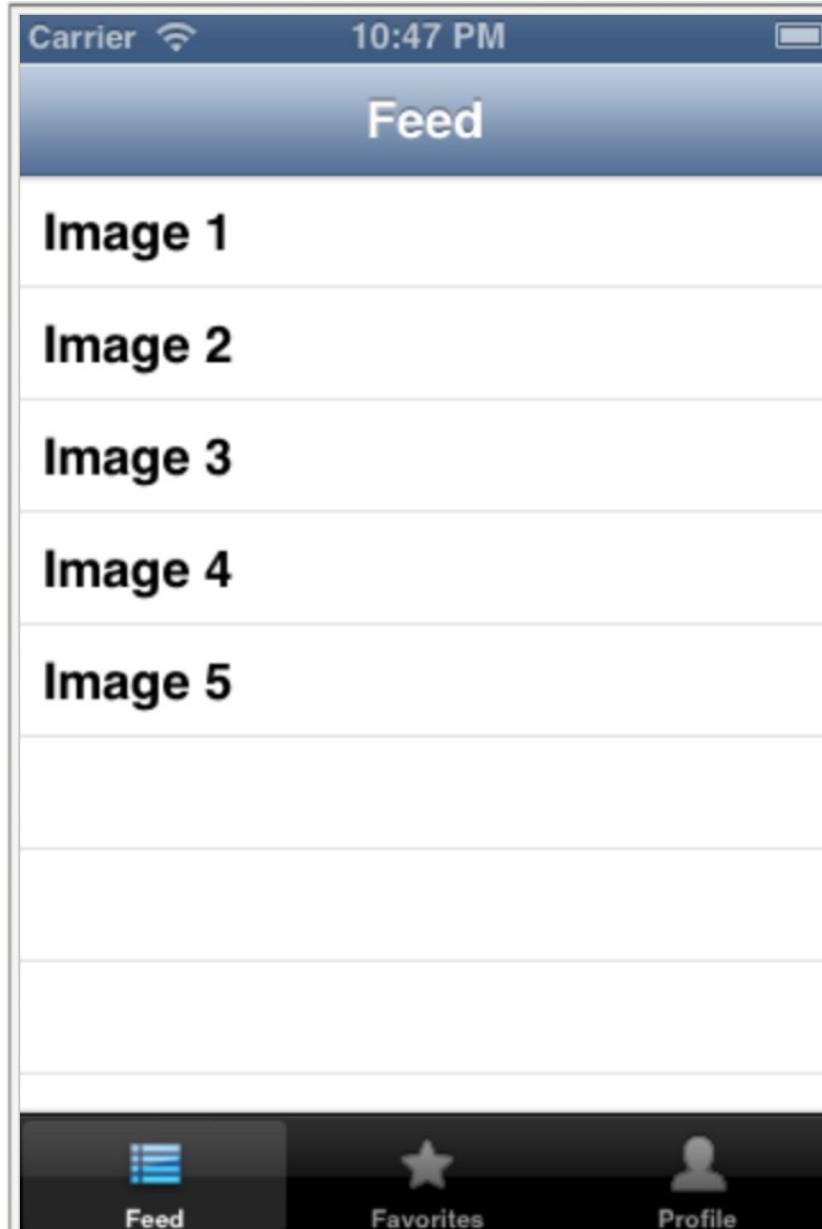
indexPath.section

This method runs each time a cell is tapped

contains an integer representing the index of the tapped cell

contains an integer representing the section containing the tapped cell

How do we make cells push to another Controller?



FeedTableViewController.m

Push to Image



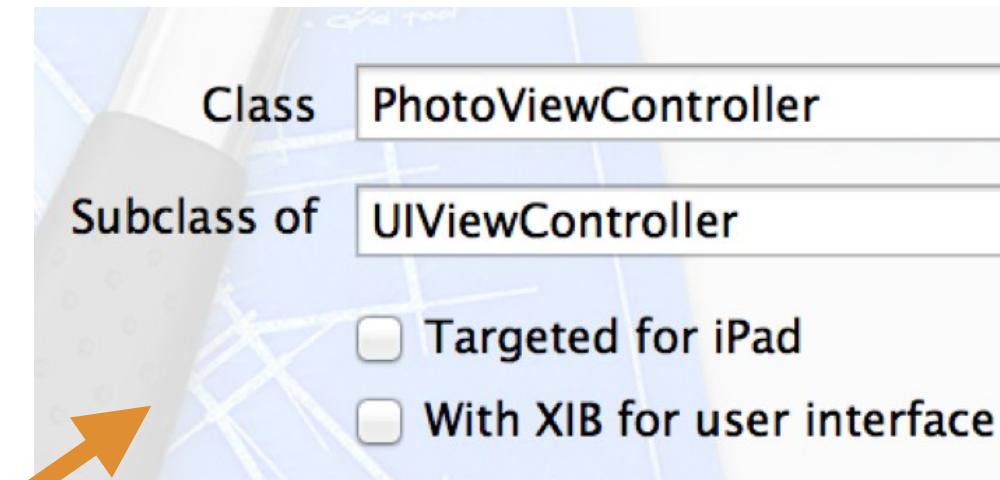
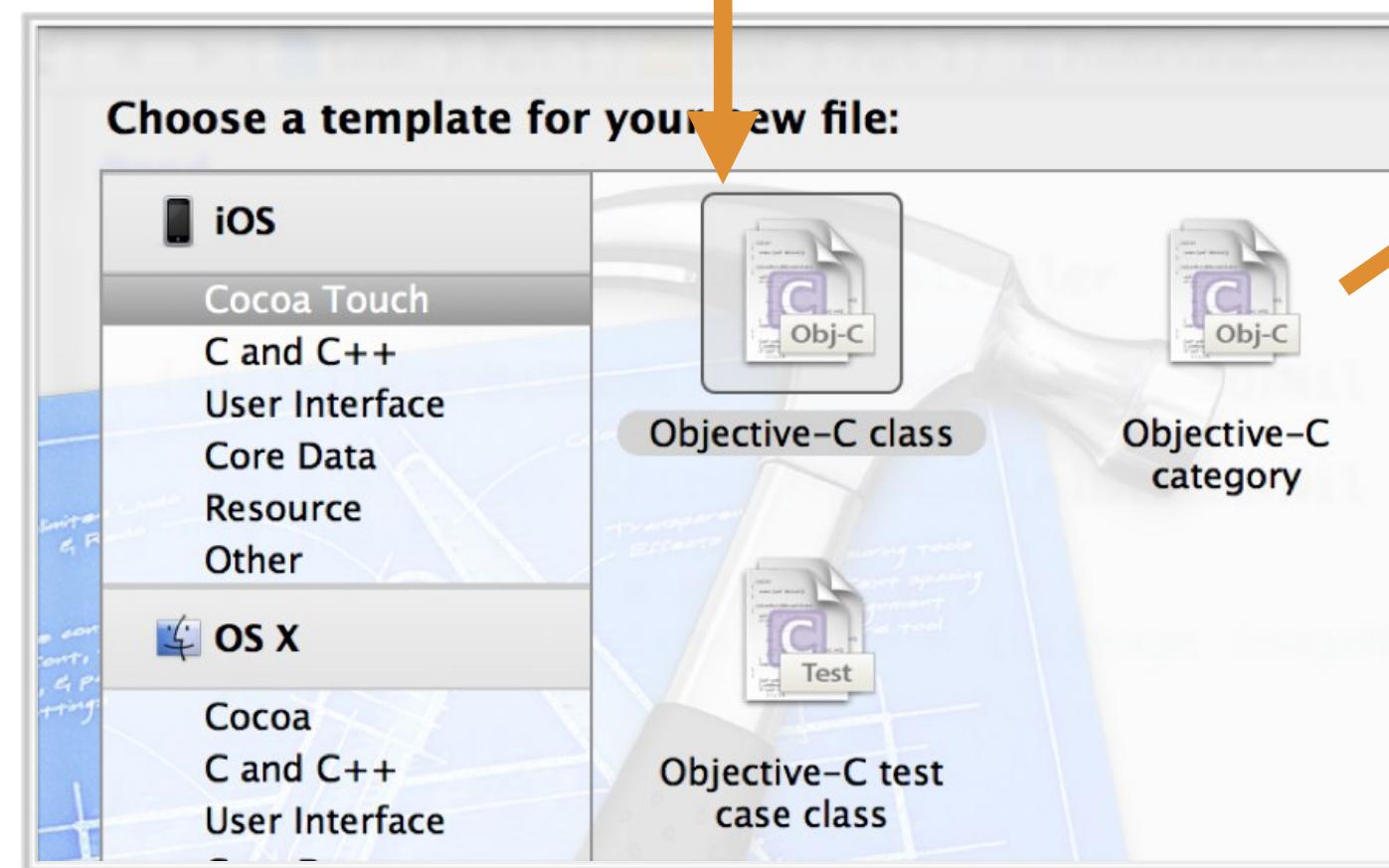
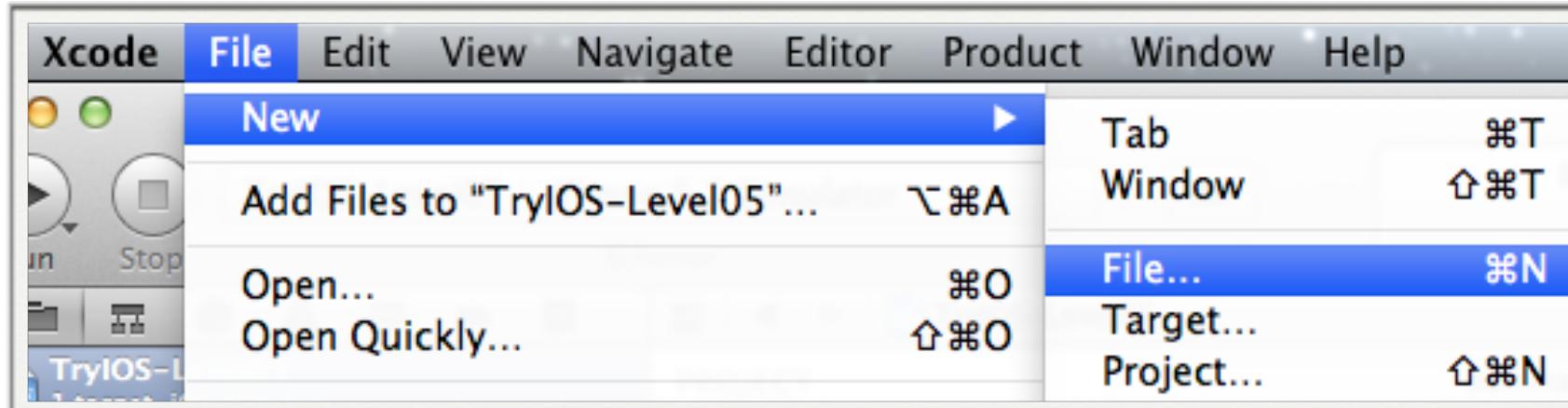
PhotoViewController.m



Pushing from didSelectRowAtIndexPathIndexPath

1. Create the PhotoViewController class.
2. Define an @property in PhotoViewController.h to store the imageName.
3. Create a UIImageView in PhotoViewController.m to show the image.
4. Modify our TableView's didSelectRowAtIndexPathIndexPath to instantiate a PhotoViewController, set the proper imageName and push VC.

Create a PhotoViewController



PhotoViewController.m
PhotoViewController.h

Define an @property to store the imageFileName

PhotoViewController.h

```
#import "<UIKit/UIKit.h>\n\n@implementation PhotoViewController : UIViewController\n\n@property (strong, nonatomic) NSString *imageFileName;\n\n@end
```

We will pass a value into
imageFileName from
FeedTableViewController

PhotoViewController.m

```
- (void)viewDidLoad {\n    UIImageView *imageView = [[UIImageView alloc] initWithImage:[UIImage\n        imageNamed:self.imageFileName]];\n    imageView.frame = CGRectMake(10, 10, 300, 300);\n\n    [self.view addSubview:imageView];\n}
```

Setting location, width, and height

Pass an `imageFileName` into `PhotoViewController`

`FeedTableViewController.m`

```
#import "PhotoViewController.h"

...
- (void) tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    PhotoViewController *photoVC = [[PhotoViewController alloc] init];
    photoVC.imageFileName = imageFilenames[indexPath.row];
    Set the image filename so photoVC knows what to display
    [self.navigationController pushViewController:photoVC animated:YES];
}
```



Level 5

- 01 Introduction to the UITableViewController
- 02 Display data in a UITableView and detect taps in a UITableViewCell
- 03 Pushing to another ViewController from a cell tap

04 Changing the display style of a UITableViewCell



UITableViewCell Styles

```
cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault  
                           reuseIdentifier:@"Cell"];
```



UITableViewCellStyleDefault
UITableViewCellStyleValue1
UITableViewCellStyleValue2
UITableViewCellStyleSubtitle

```
cell.detailTextLabel.text = @"My Subtitle";
```

```
cell.imageView.image = [UIImage imageNamed:@"image1thumb.jpg"];
```

To set thumbnail

Level 6

AFNetworking





Level 6

01 Introduction to the AFNetworking

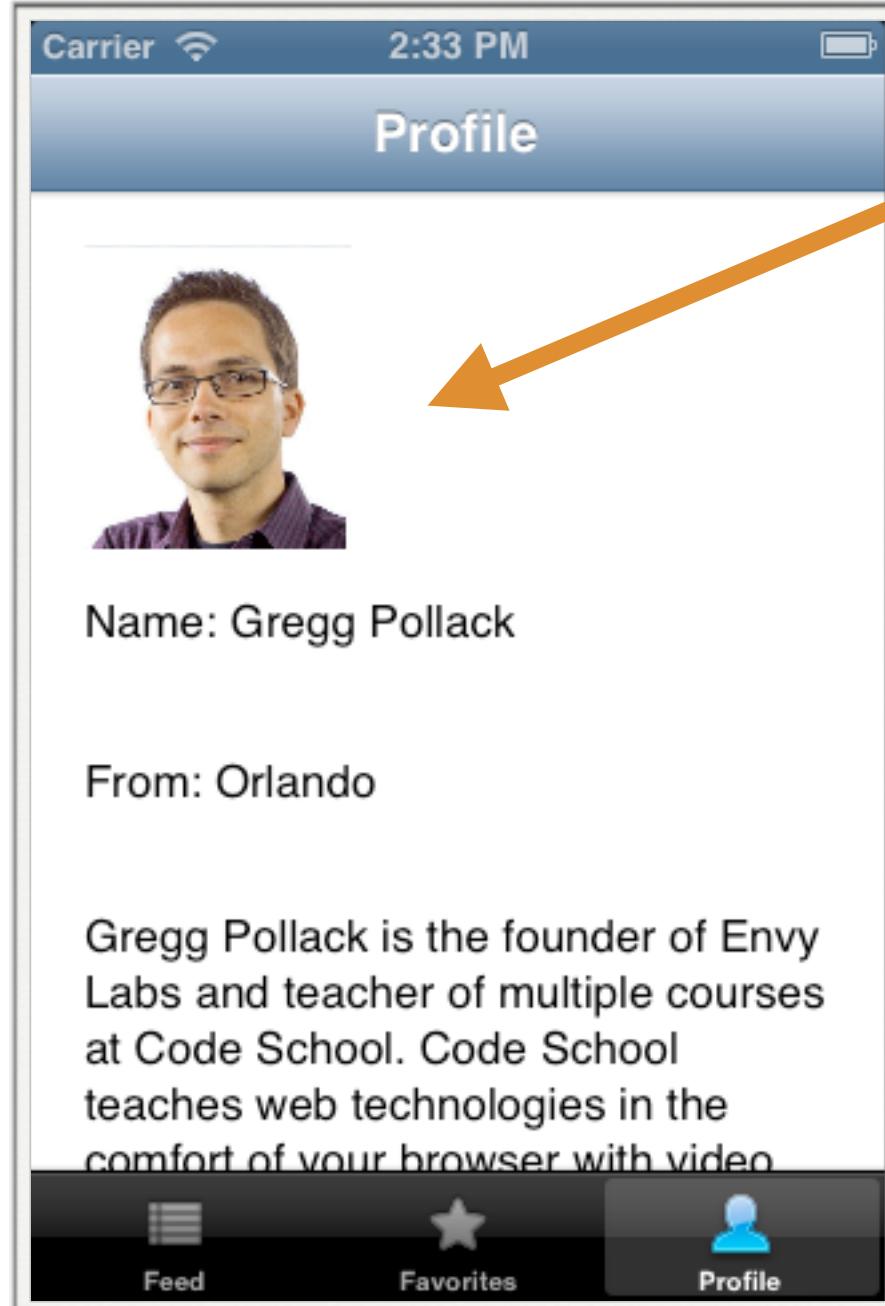
02 Fetching Images from the Internet

03 Fetching JSON data from the Internet

04 The NSDictionary Data Structure

05 Making our FeedTableViewController Real

Remember our ProfileViewController



What if we had a URL for this photo?



An iOS & OS X networking framework

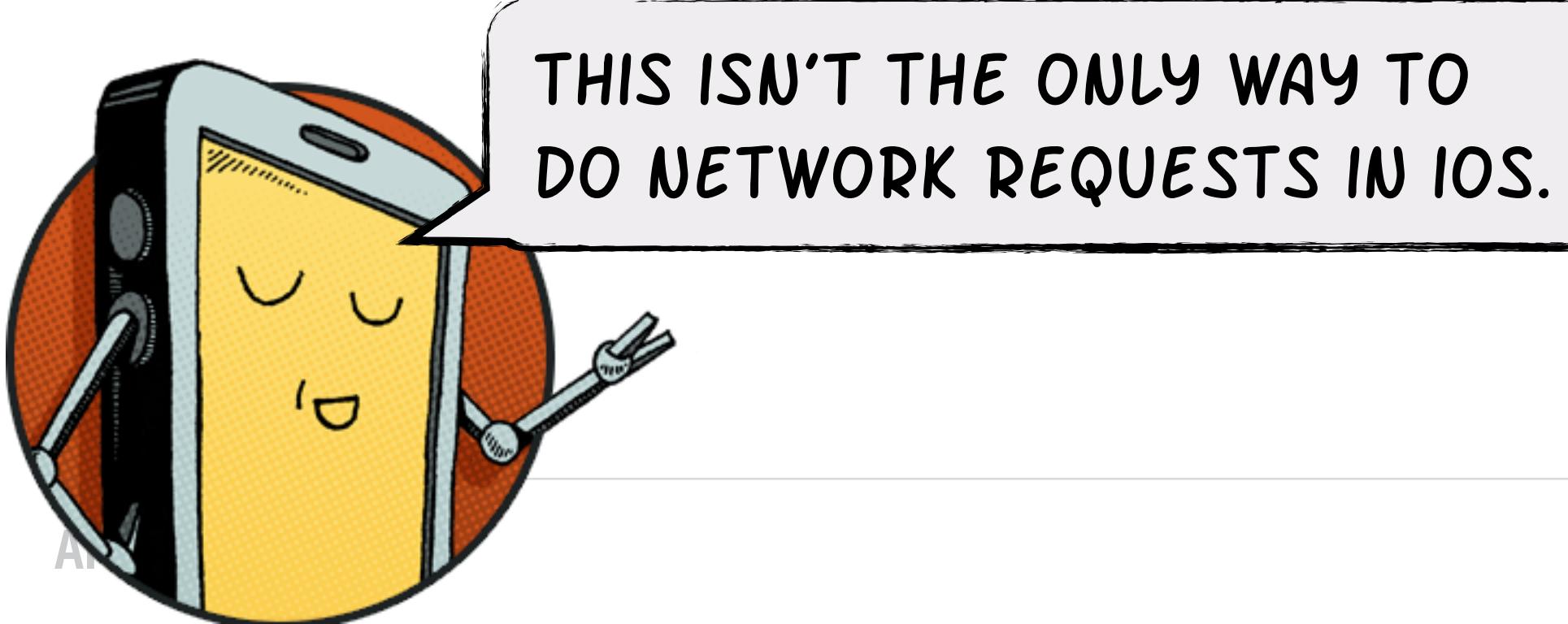
<http://afnetworking.com/>



Mattt Thompson



Scott Raymond



Try
iOS

Installing AFNetworking



A delightful iOS and OS X networking framework — [Readme](#)
<http://afnetworking.com>

Clone in Mac ZIP HTTP Git Read-

New

Add Files to "TryiOS-Level06" ⌘A

AFNetworking

Destination Copy items into destination group's folder (if needed)

Folders Create groups for any added folders
 Create folder references for any added folders

Add to targets TryiOS-Level06

New Folder Cancel Add



Pulling an Image from the Internet

ProfileViewController.m

old code

```
UIImageView *profileImageView = [[UIImageView alloc]
                                initWithImage:[UIImage imageNamed:@"gregg"]];
profileImageView.frame = CGRectMake(20,20,100,114);
[self.scrollView addSubview:profileImageView];
```

URL for image is <http://example.com/gregg.png>

Also need a placeholder image



placeholder.png



Pulling an Image from the Internet

ProfileViewController.m

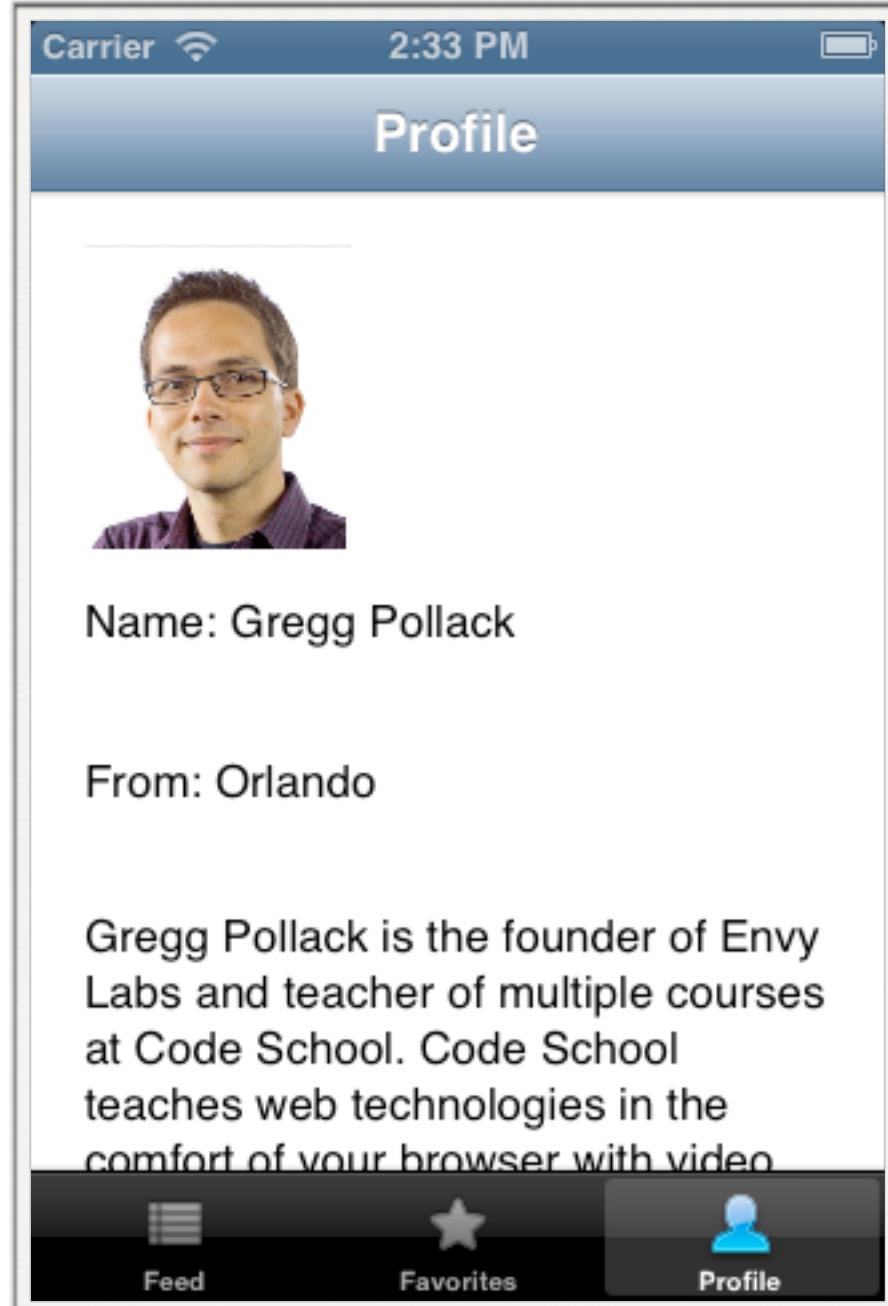
```
#import "UIImageView+AFNetworking.h"
...
UIImageView *profileImageView = [[UIImageView alloc] init];
profileImageView.frame = CGRectMake(20,20,100,114);
[profileImageView setImageWithURL:[NSURL URLWithString:@"http://example.com/gregg.png"]
placeholderImage:[UIImage imageNamed:@"placeholder.png"]];
[self.scrollView addSubview:profileImageView];
```



USING A PLACEHOLDER IMAGE ISN'T
REQUIRED, BUT IT'S A GOOD PRACTICE.

A

Pulling All Profile Data from the Internet?



<http://tryios.codeschool.com/users/3.json>

```
{  
  firstName: "Gregg",  
  lastName: "Pollack",  
  city: "Orlando",  
  profilePhoto: "http://example.com/gregg.png",  
  profilePhotoThumbnail: "http://example.c...".  
  biography: "...",  
  memberSince: "November 2012"  
}
```





Thinking Asynchronously

Synchronous Networking Code

Go fetch JSON code from the internet
Once returned load as a Subview
Add other subviews



Asynchronous Networking Code

Go fetch JSON code from the internet
Once returned load as a Subview
Add other subviews



Callback



In Objective C .. Asynchronous = Blocks

ProfileViewController.m

```
#import "AFJSONRequestOperation.h"
...
- (void)viewDidLoad
{
    ...
    Create a NSURL set to our endpoint to http://example.com/users/3.json
    Create a NSURLRequest with our NSURL
    Create a AFJSONRequestOperation which will do the call to the Internet
    If request has success
        // Read the data from the returned JSON object
    If request has failure
        NSLog(@"%@",error.localizedDescription);
    start the operation
```



In Objective C .. Asynchronous = Blocks

ProfileViewController.m

```
NSURL *url = [[NSURL alloc] initWithString:@"http://example.com/users/3.json"];  
  
NSURLRequest *request = [[NSURLRequest alloc] initWithURL:url];  
  
AFJSONRequestOperation *operation = [AFJSONRequestOperation  
JSONRequestOperationWithRequest:request  
success:^(NSURLRequest *request, NSHTTPURLResponse *response, id JSON) {  
    // Read the data from the returned JSON object  
  
} failure:^(NSURLRequest *request, NSHTTPURLResponse *response,  
           NSError *error, id JSON) {  
    NSLog(@"%@", error.localizedDescription);  
}];  
  
[operation start];
```



<Challenge 1>

Notice below in the ProfileView Controller we've already set the NSURL in a variable called imgURL, and a placeholder UIImage in a variable called placeholder.

Call the `setImageWithURL:placeholderImage:` method sending in these parameters to properly display the profile image.



<Challenge 2>

Log out the JSON which returns when we successfully make a JSON Request
by using `NSLog(@"Returned JSON is %@", JSON);`.



Your Log Message

```
NSLog(@"%@", JSON);
```

```
{  
    firstName = "Gregg";  
    lastName = "Pollack";  
    city = "Orlando";  
    profilePhoto = "http://example.com/gregg.png";  
    profilePhotoThumbnail = "http://example.c...";  
    biography = "...";  
    memberSince = "November 2012";  
}
```

```
NSLog(@"%@", [JSON class]);
```

NSDictionary



NSDICTIONARY IS A KEY VALUE
STORE. KINDA LIKE A RUBY HASH.



NSDictionary A Key Value Store

```
NSLog(@"%@", [JSON allKeys]);
```

```
{  
    firstName,  
    lastName,  
    city,  
    profilePhoto,  
    profilePhotoThumbnail,  
    biography,  
    memberSince  
}
```

```
NSLog(@"%@", [JSON allValues]);
```

```
{  
    "Gregg",  
    "Pollack",  
    "Orlando",  
    "http://example.com/gregg.png",  
    "http://example.c...",  
    "...",  
    "November 2012"  
}
```

```
NSLog(@"firstName is %@", JSON[@"firstName"]);
```

First Name is Gregg

```
NSLog(@"memberSince is %@", JSON[@"memberSince"]);
```

memberSince is November 2012



Completing our Network Call

ProfileViewController.h

To store the JSON NSDictionary from our request
@property (strong, nonatomic) NSDictionary *userProfile;

ProfileViewController.m

```
...
AFJSONRequestOperation *operation = [AFJSONRequestOperation
JSONRequestOperationWithRequest:request
success:^(NSURLRequest *request, NSHTTPURLResponse *response, id JSON) {
    self.userProfile = JSON;
    [self requestSuccessful];
```

Store the JSON and call method

```
- (void)requestSuccessful {
```

Populate the image and label with the proper data



Refactoring Our Code

ProfileViewController.m Populate the image and label with the proper data

```
- (void)requestSuccessful {  
    UIImageView *profileImageView = [[UIImageView alloc] init];  
    profileImageView.frame = CGRectMake(20, 20, 100, 114);  
    [profileImageView setImageWithURL:[NSURL URLWithString:@"http://example.com/gregg.png"]  
        placeholderImage:[UIImage imageNamed:@"placeholder.png"]];  
    ...  
    UILabel *nameLabel = [[UILabel alloc] init];  
    nameLabel.frame = CGRectMake(20, 140, 280, 40);  
    nameLabel.text = @"Name: Gregg Pollack";  
}
```



Refactoring Our Code

ProfileViewController.m Populate the image and label with the proper data

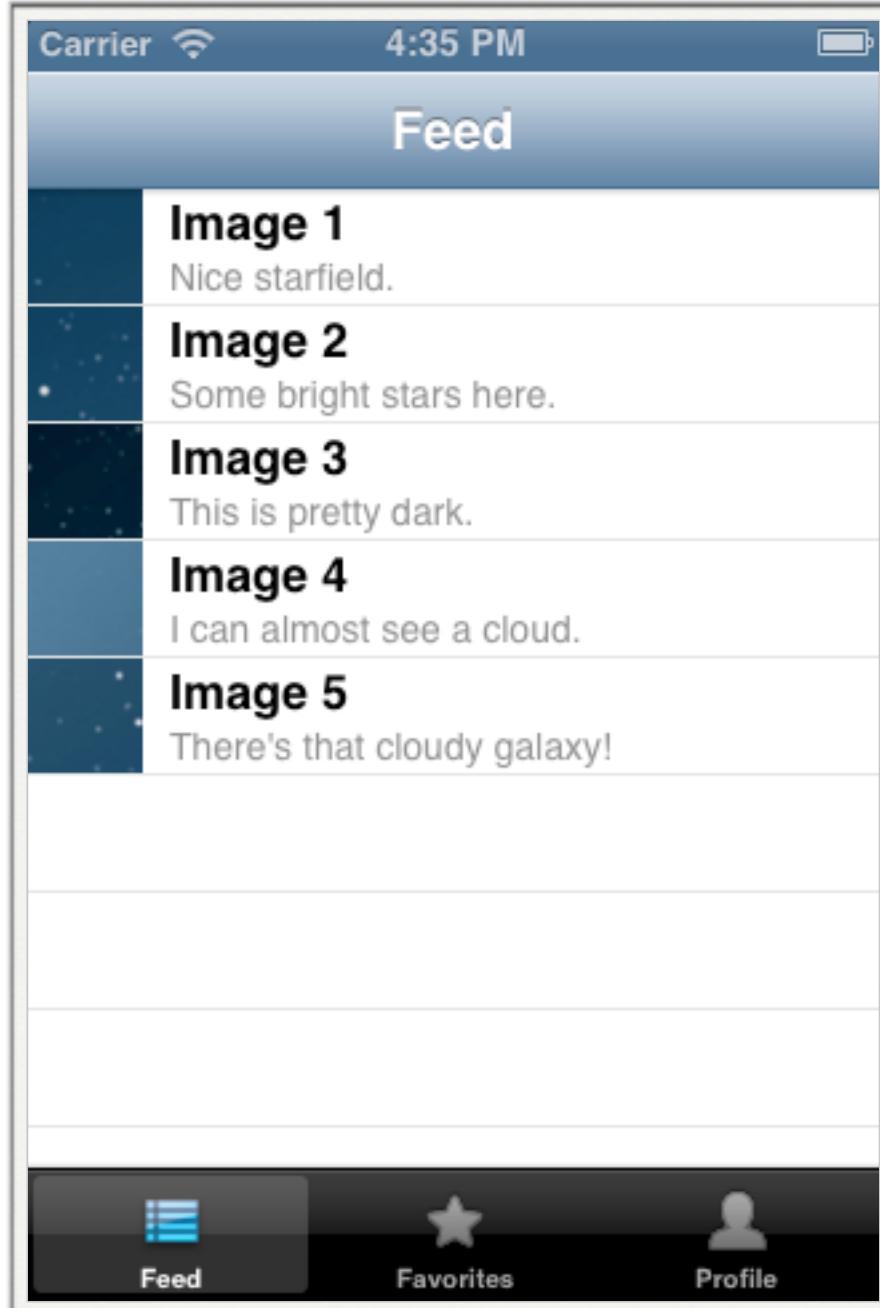
```
- (void)requestSuccessful {
    UIImageView *profileImageView = [[UIImageView alloc] init];
    profileImageView.frame = CGRectMake(20, 20, 100, 114);
    [profileImageView setImageWithURL:[NSURL URLWithString:self.userProfile[@"profilePhoto"]]
                           placeholderImage:[UIImage imageNamed:@"placeholder.png"]];
    ...
    UILabel *nameLabel = [[UILabel alloc] init];
    nameLabel.frame = CGRectMake(20, 140, 280, 40);
    nameLabel.text = [NSString stringWithFormat:@"Name: %@ %@", self.userProfile[@"firstName"], self.userProfile[@"lastName"]];
```



<Challenge 3>

Practice using NSDictionaries by finishing off the rest of the
requestSuccessful Function

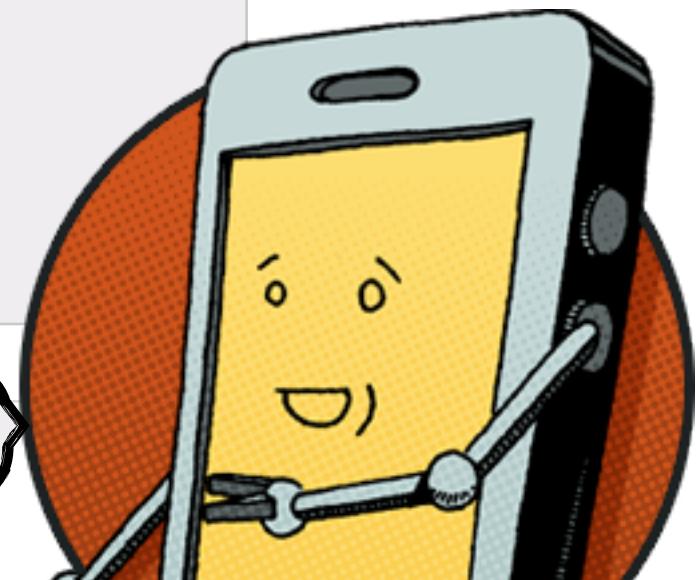
Pulling All Feed Data from the Internet?



<http://tryios.codeschool.com/images.json>

```
[  
  {  
    "title": "Image 1",  
    "detail": "Nice starfield.",  
    "filename": "http://...",  
    "thumbnail": "http://..."  
  },  
  {  
    "title": "Image 2",  
    "detail": "Some bright stars here.",  
    "filename": "http://...",  
    "thumbnail": "http://..."  
  },  
  ...  
]
```

THAT'S AN NSARRAY OF NSDICTIONARIES





Pulling All Feed Data from the Internet?

<http://tryios.codeschool.com/images.json>

```
[  
  {  
    "title": "Image 1",  
    "detail": "Nice starfield.",  
    "filename": "http://.../",  
    "thumbnail": "http://.../"  
  },  
  {  
    "title": "Image 2",  
    "detail": "Some bright stars here.",  
    "filename": "http://.../",  
    "thumbnail": "http://.../"  
  },  
  ...  
]
```

An Array of NSDictionaries

JSON[0] [@“title”]

Image 1

JSON[1] [@“title”]

Image 2

JSON[1] [@“detail”]

Some bright stars here.



Refactoring the FeedViewController

FeedViewController.h

`@property (strong, nonatomic) NSArray *photos;`

To store the Array of NSDictionaries from our request

FeedViewController.m

```
...
AFJSONRequestOperation *operation = [AFJSONRequestOperation
    JSONRequestOperationWithRequest:request
    success:^(NSURLRequest *request, NSHTTPURLResponse *response, id JSON) {
        self.photos = JSON;
        [self.tableView reloadData]; Redraw the table
    } failure:^(NSURLRequest *request, NSHTTPURLResponse *response,
               NSError *error, id JSON) {
        NSLog(@"NSError: %@", error.localizedDescription);
    }];
[operation start];
```



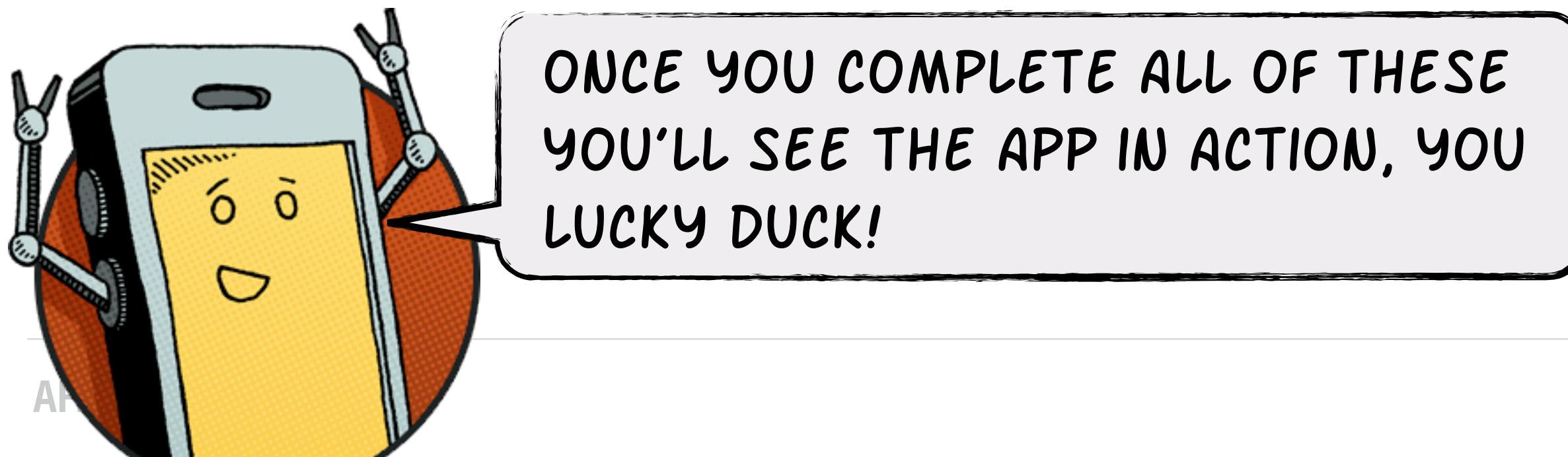
Refactoring the rest of FeedViewController

FeedViewController.h

```
@property (strong, nonatomic) NSArray *photos;
```

To store the Array of NSDictionaries from our request

1. Revise `tableView:numberOfRowsInSection:` to return the number of photos in the Array.
2. Revise `tableView:cellForRowAtIndexPath:` to access the particular NSDictionary at `indexPath.row`.
3. Revise `tableView:didSelectRowAtIndexPath:` to pass the proper values to PhotoViewController.



An

Try
iOS