Level 5

Table View Controllers

CODE SCHOOL

Try

iOS

• PRESENTS •

Level 5

01 Introduction to the UITableViewController

02 Display data in a UITableView and detect taps in a UITableViewCell

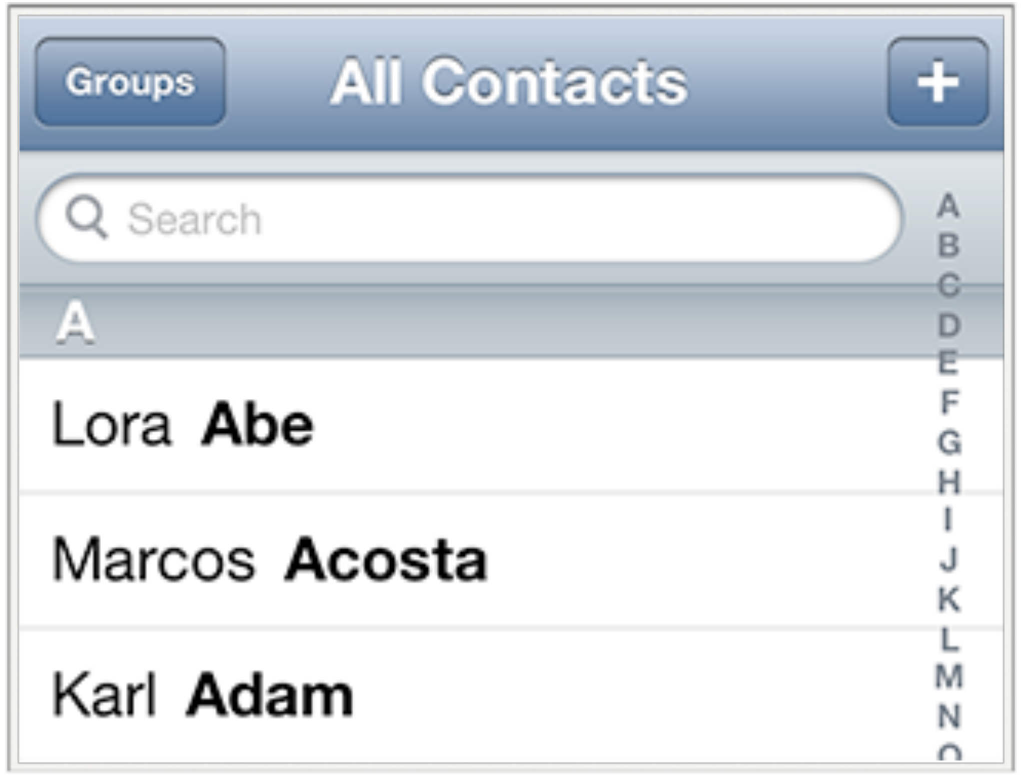
03 Pushing to another ViewController from a cell tap

04 Changing the display style of a UITableViewCell

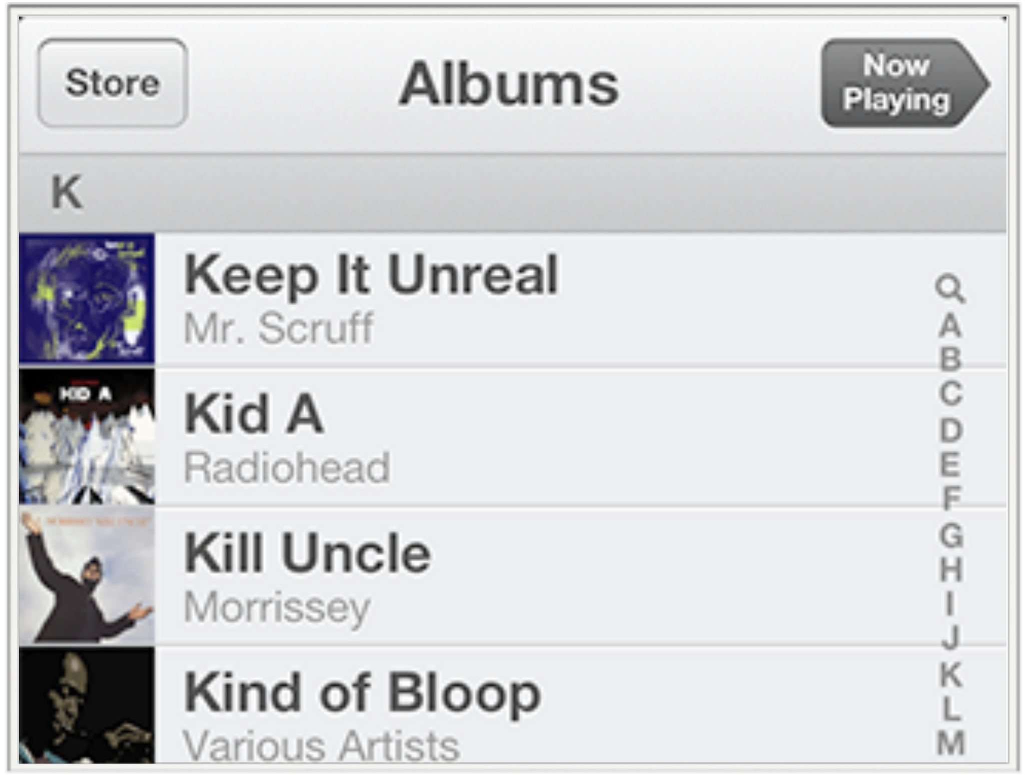
UITableView Examples

UITableViews are great for displaying lists of data

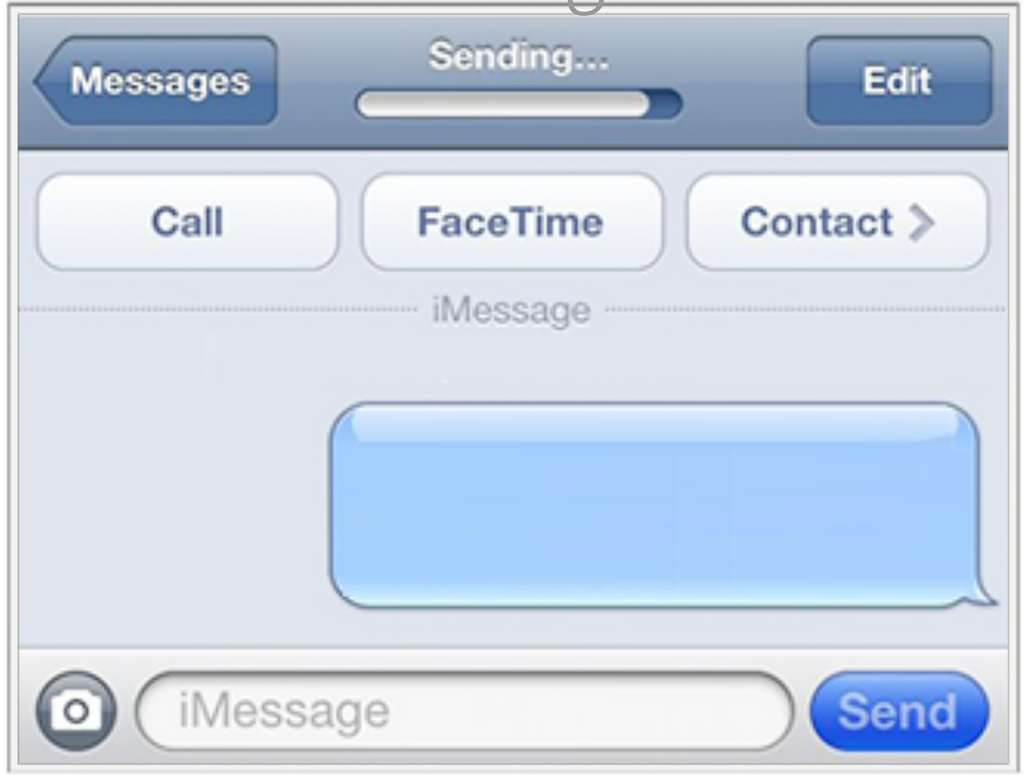
Address Book



Music

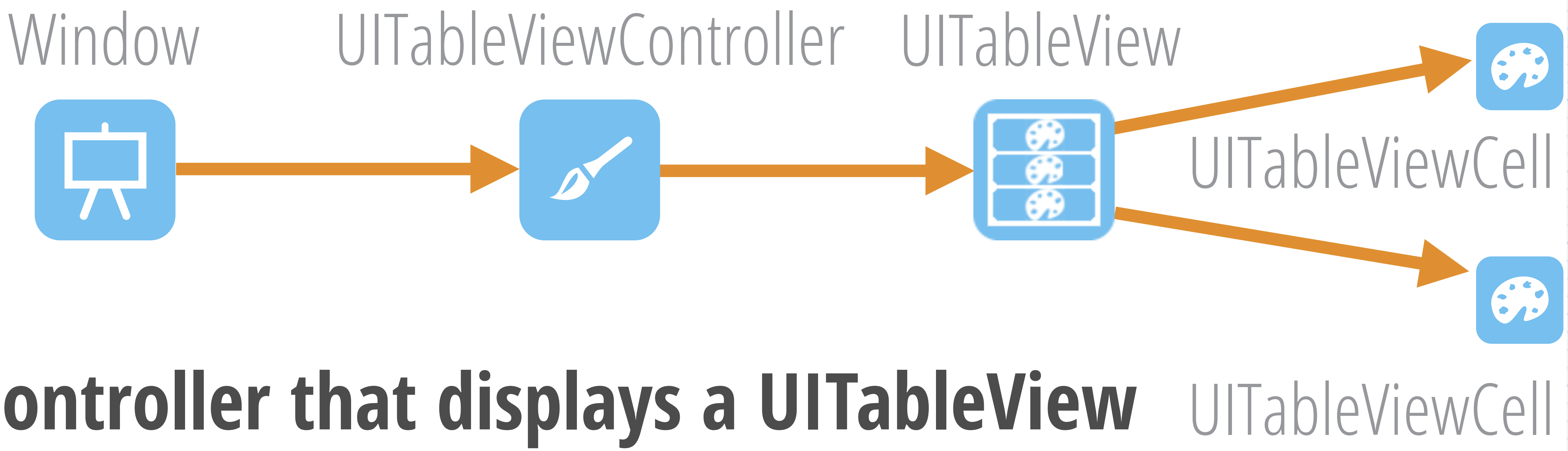


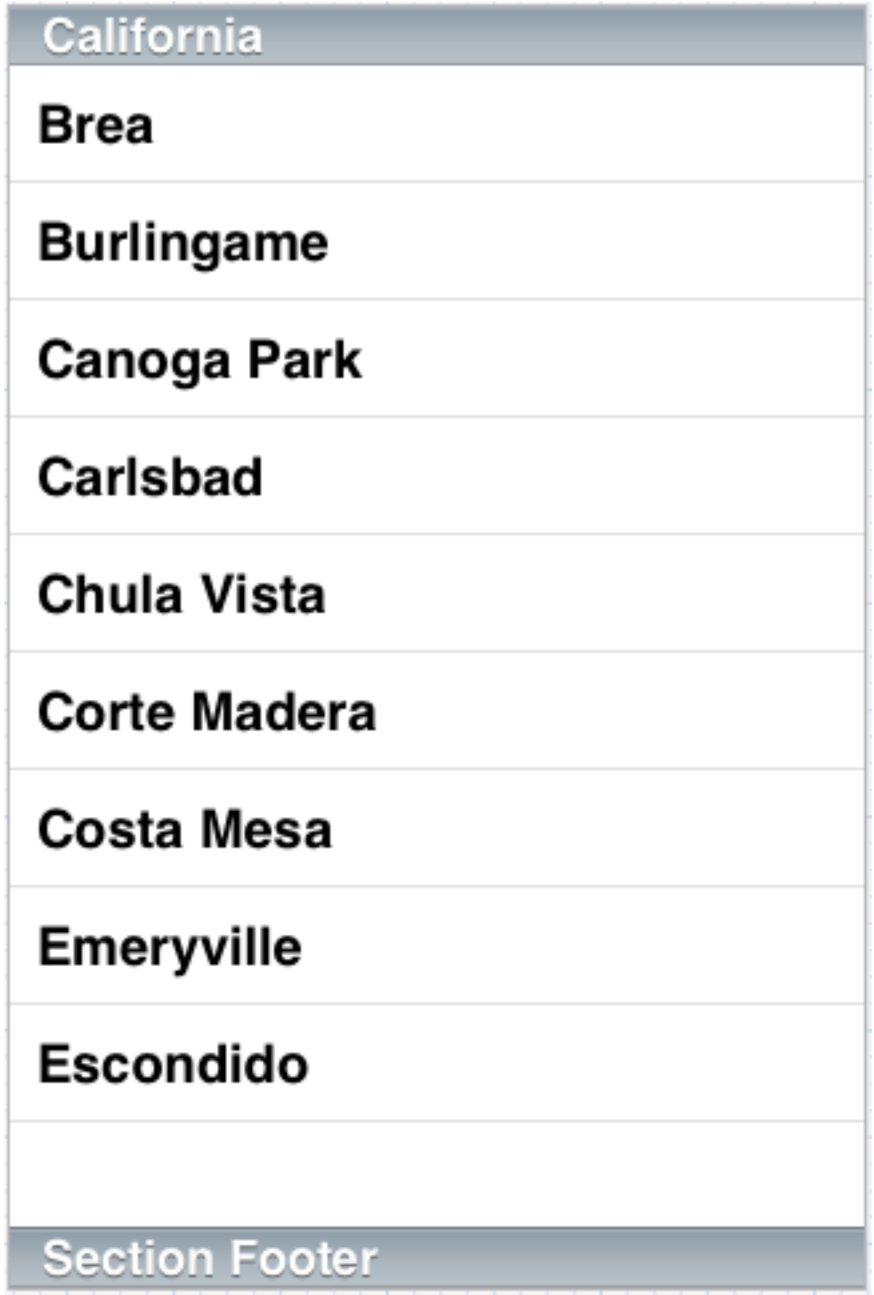
Messages



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

Introduction to the UITableViewController




California

Brea

Burlingame

Canoga Park

Carlsbad

Chula Vista

Corte Madera

Costa Mesa

Emeryville

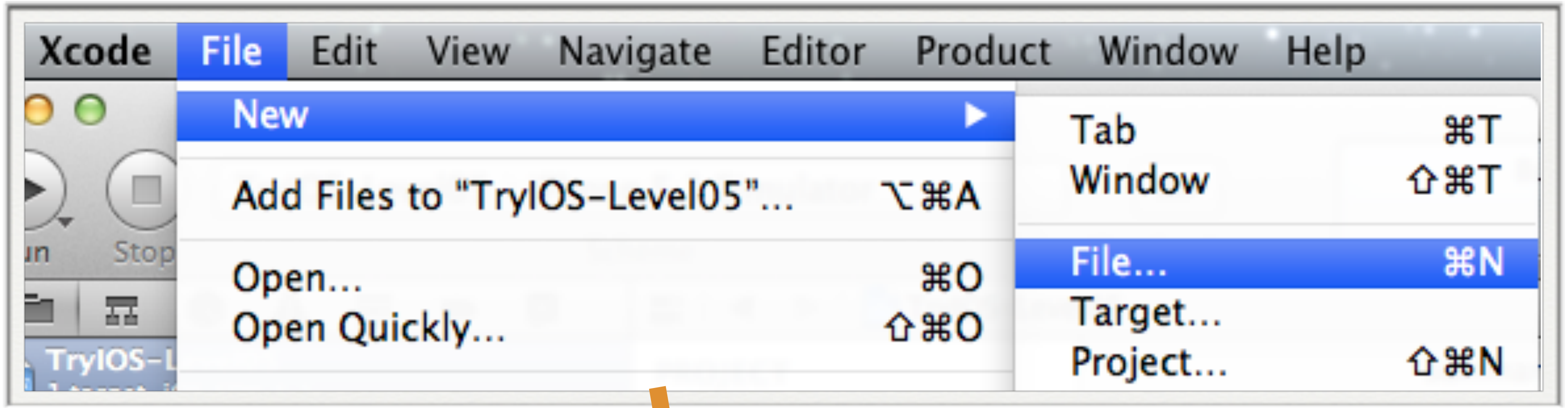
Escondido

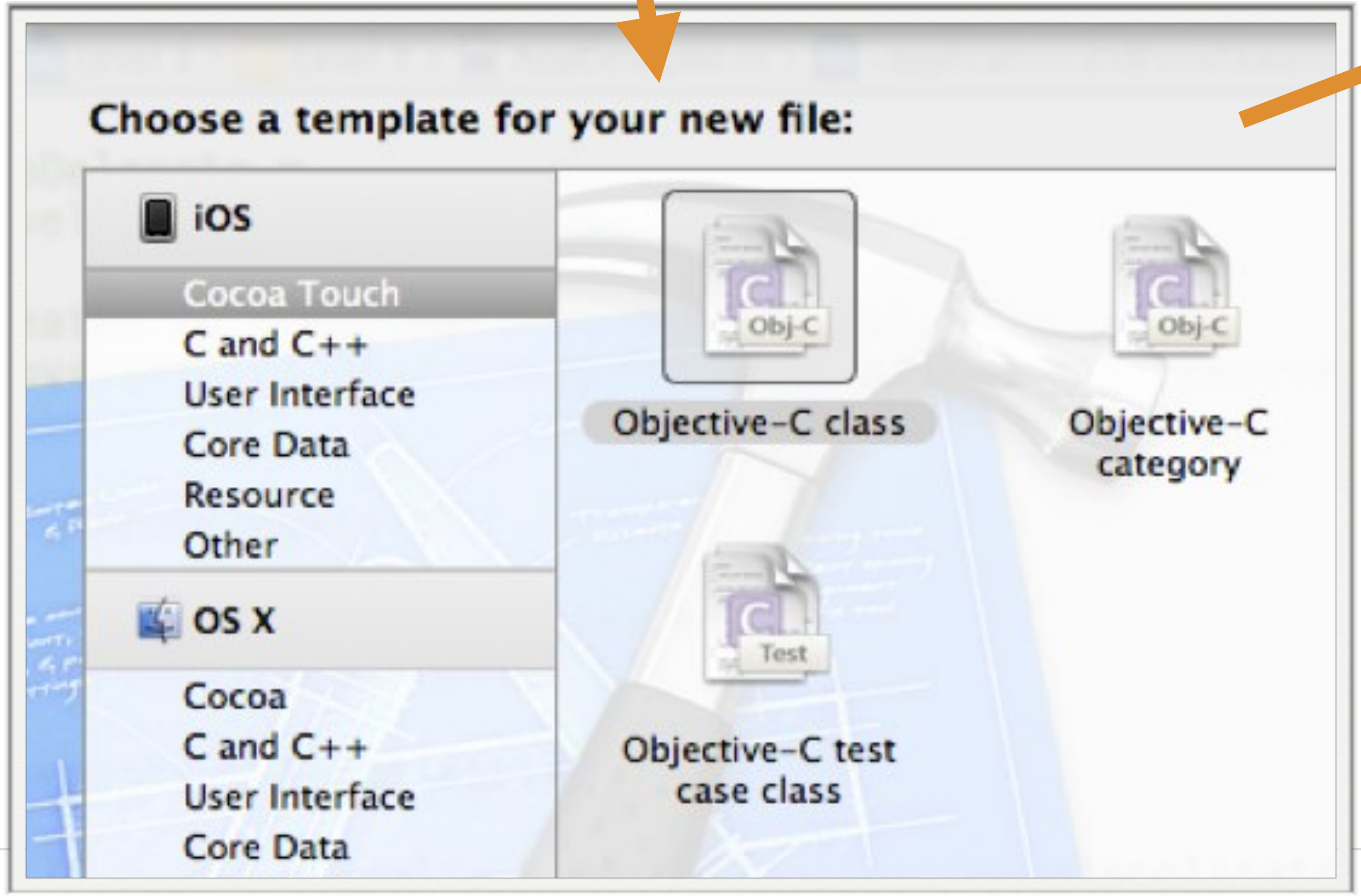
Section Footer

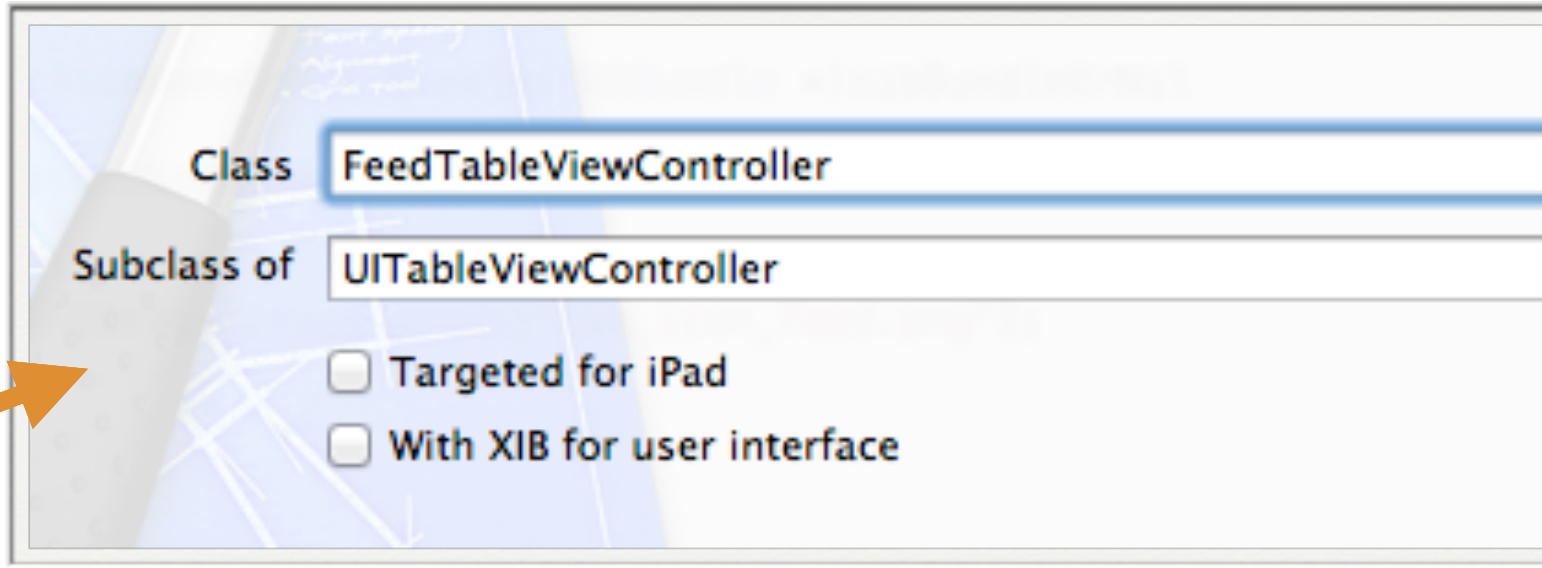
Controller that displays a UITableView

Inherits from the superclass UIViewController

Create a UITableViewController class




Choose a template for your new file:


Class: FeedTableViewController

Subclass of: UITableViewController

☐ Targeted for iPad

☐ With XIB for user interface

FeedTableViewController.h
FeedTableViewController.m

Adding a Table View Controller

AppDelegate.m

old code ⚠

```
#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];
    ...
}
```

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:cellForRowAtIndexPath:
```

Initialize and setup each cell in your tableView

Required Methods for UITableViewController

FeedTableViewController.m

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

Returns the number of sections

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

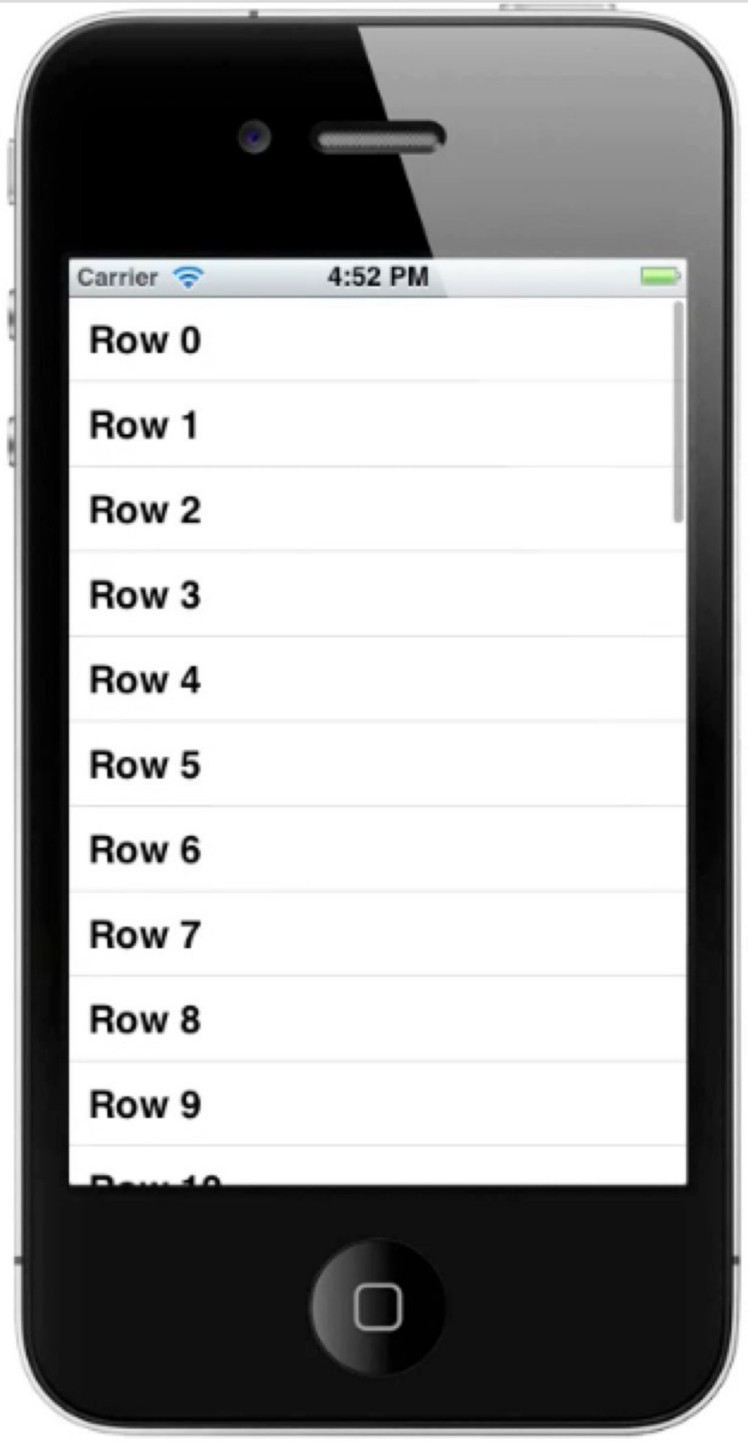
Returns the number of rows (cells)

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

Initialize and setup each cell in your tableView

Reusing TableView Cells

UITableViewCell



Carrier 4:52 PM

Row 0

Row 1

Row 2

Row 3

Row 4

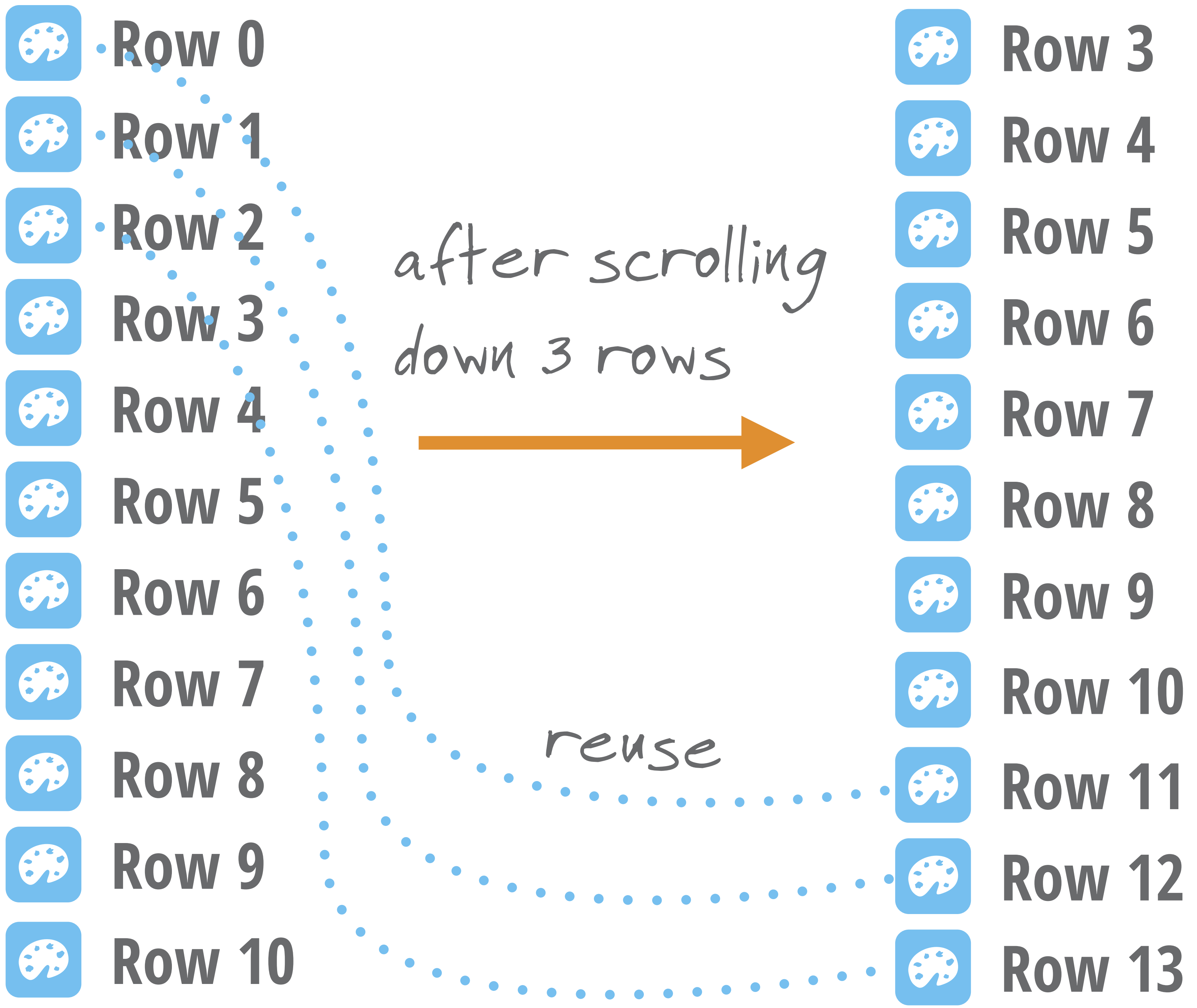
Row 5

Row 6

Row 7

Row 8

Row 9



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

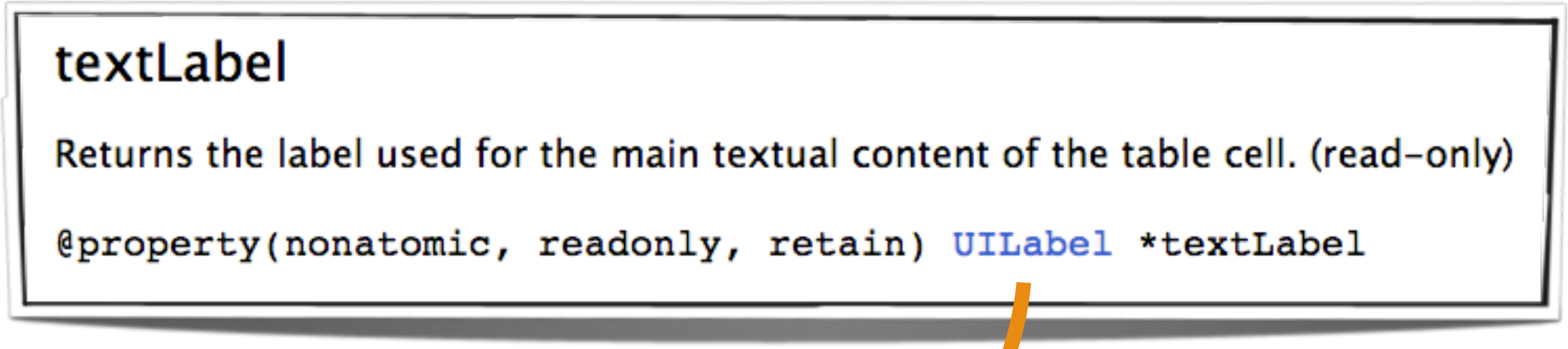
```
    if(cell == nil) {
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault
                                      reuseIdentifier:@"Title"];
    }
```

Nope, no reusable cell, so lets allocate a new one

```
    return cell;
}
```

Setting the text in a UITableViewCell

Documentation for UITableViewCell

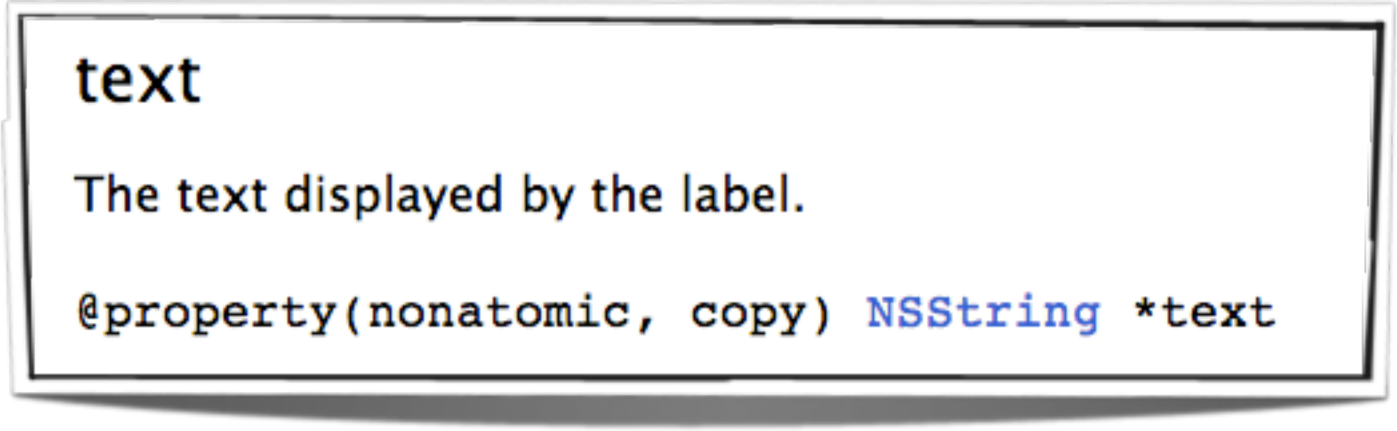


textLabel

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
{

}
```

This method runs each time a cell is tapped

```
indexPath.row
```

contains an integer representing the index of the tapped cell

```
indexPath.section
```

contains an integer representing the section containing the tapped cell

How do we make cells push to another Controller?

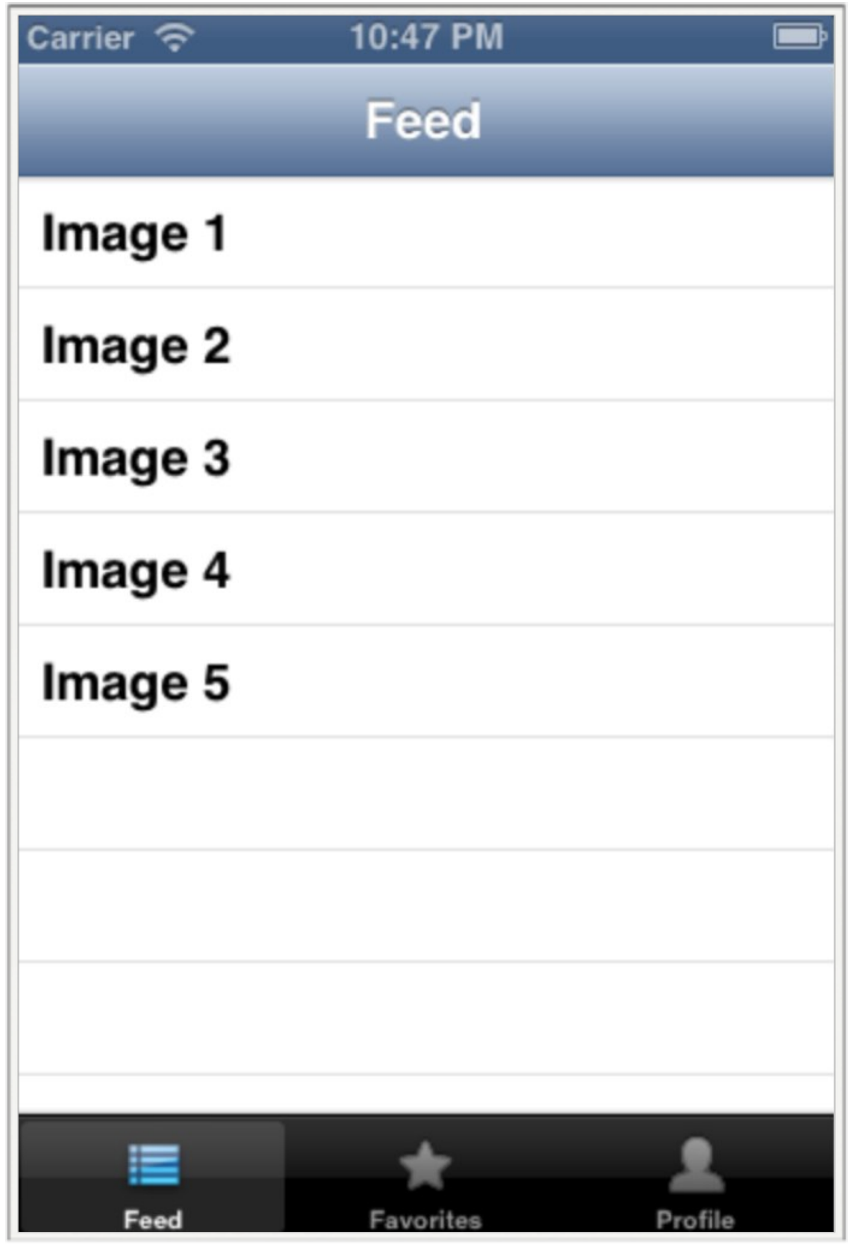


Image 1

Image 2

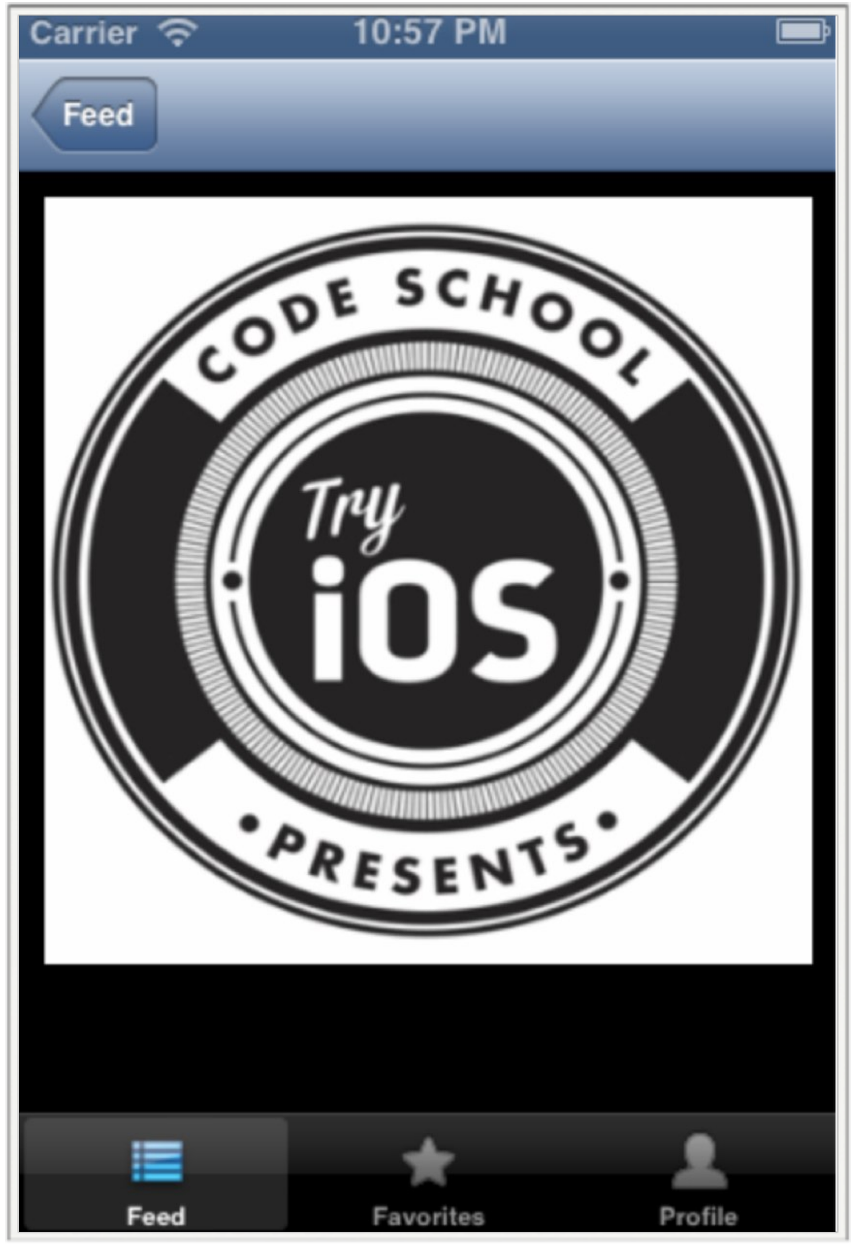
Image 3

Image 4

Image 5

FeedTableViewController.m

Push to Image



PhotoViewController.m

Pushing from didSelectRowAtIndexPath

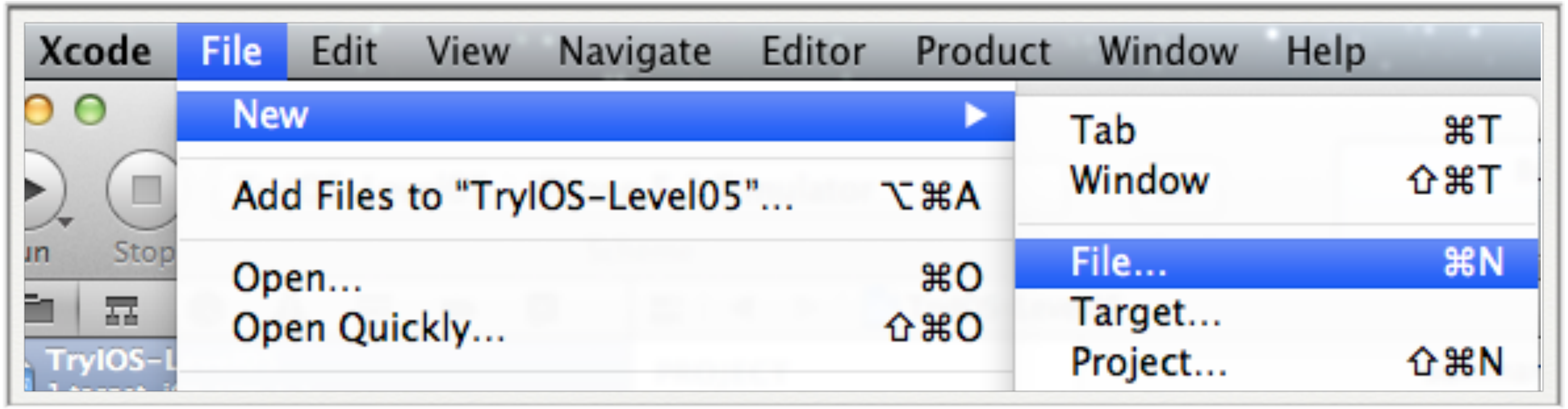
1. Create the PhotoViewController class.

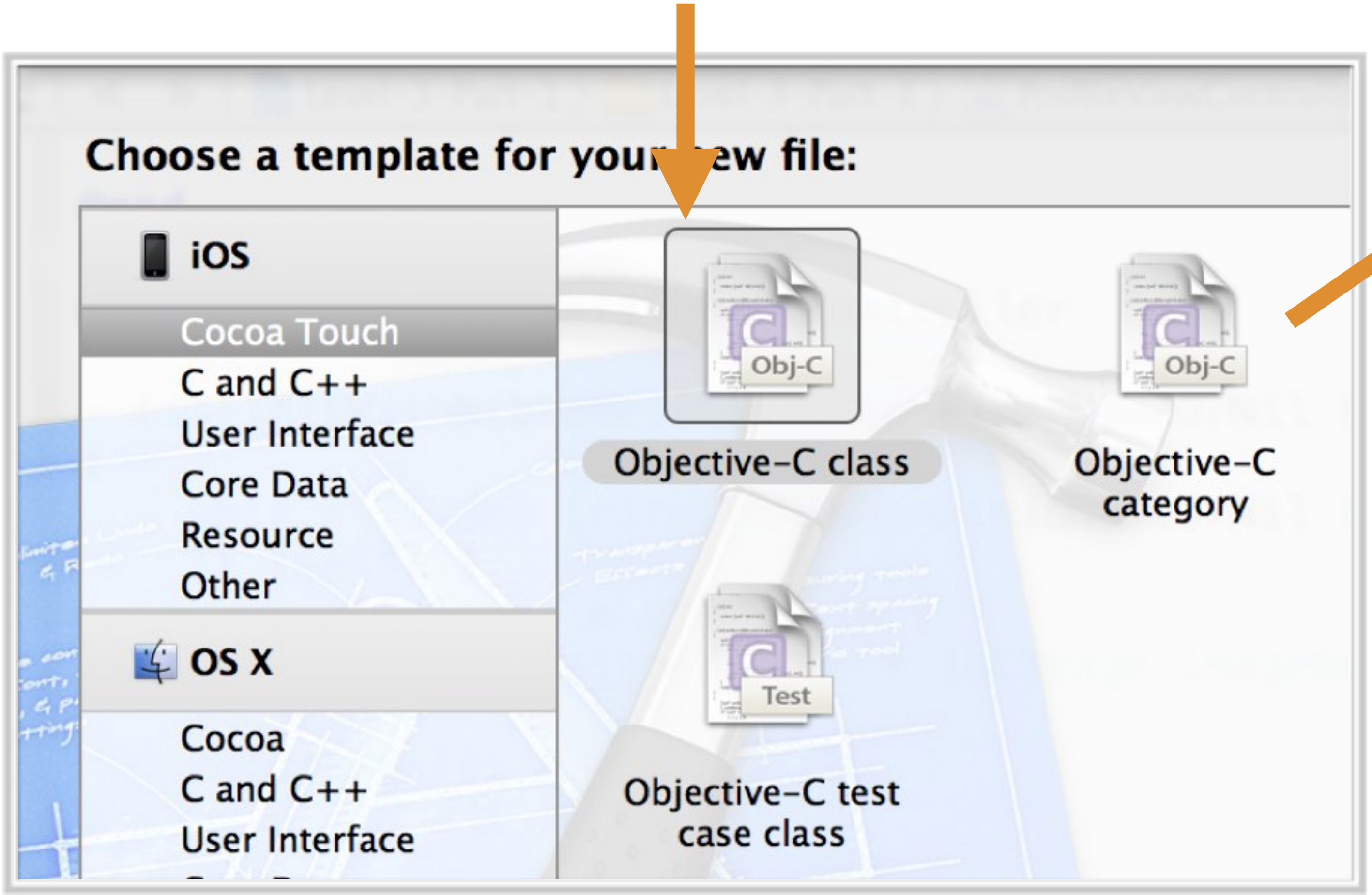
2. Define an @property in PhotoViewController.h to store the imageFileName.

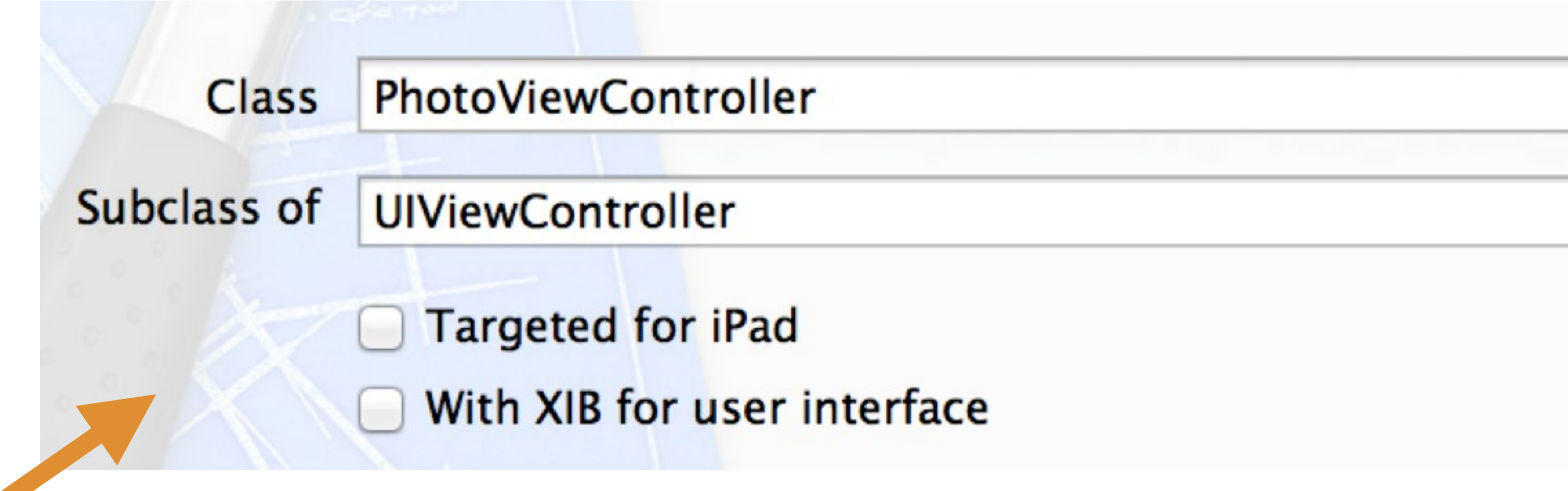
3. Create a UIImageView in PhotoViewController.m to show the image.

4. Modify our TableView's didSelectRowatIndexPath to instantiate a PhotoViewController, set the proper imageFileName and push VC.

Create a PhotoViewController







Class PhotoViewController

Subclass of UIViewController

☐ Targeted for iPad

☐ With XIB for user interface

PhotoViewController.m

PhotoViewController.h

Define an @property to store the imageFileName

PhotoViewController.h

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

@implementation PhotoViewController : UIViewController

@property (strong, nonatomic) NSString *imageFileName;

@end
```

We will pass a value into imageFileName from FeedTableViewController

PhotoViewController.m

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

    [self.view addSubview:imageView];
}
```

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

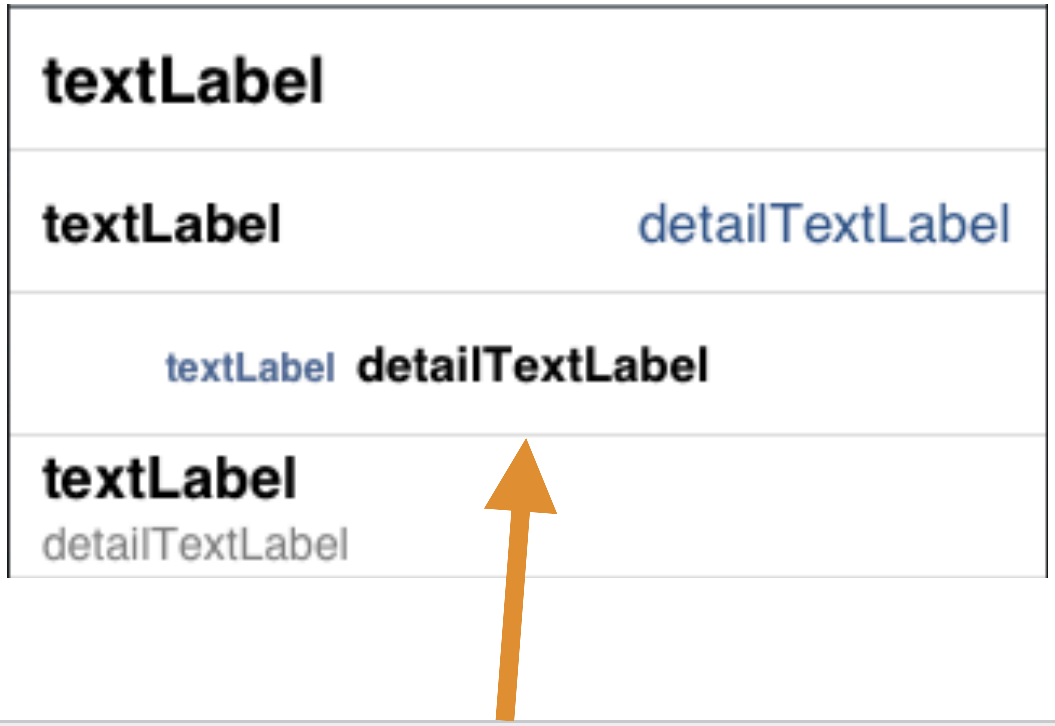
02 Display data in a UITableView and detect taps in a UITableViewCell

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"];
```



UITableCellStyleDefault

UITableCellStyleValue1

UITableCellStyleValue2

UITableCellStyleSubtitle

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

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

To set thumbnail