

Lecture 9: Chapter 12: Add Item Screen

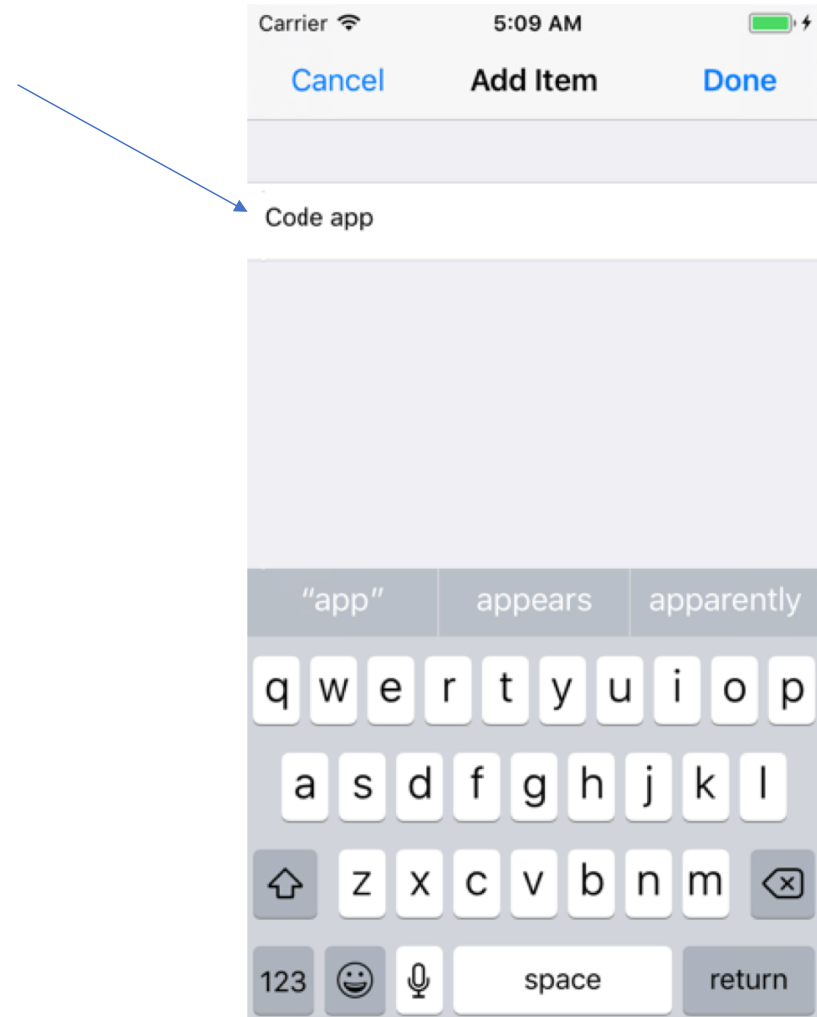
Ashley Williams

Outline

- Static Cells
- Programming To-Do List

Static Table Cells

- Good for when you know exactly what the cell is going to do and how many cells you will have.
- Can edit directly what the cell may say.
- Can make dynamic cell a static cell



Outline

- ~~Static Cells~~
- Programming To-Do List

Programming To-Do List: Chapter 12

- ~~Add a static table cell to the table for data entry~~
- Disable Cell Selection
- Read from the text field
- Polish it up

To-Do List: Disable Cell Selection

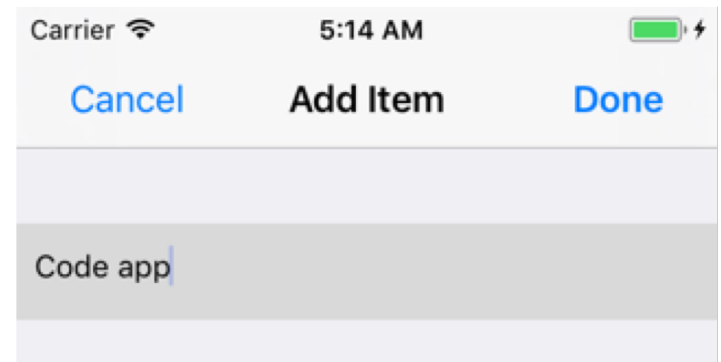
- We do not want the user to be able to select this cell. Add code to AddItemViewController.swift

- **override func** tableView(_ tableView: UITableView, willSelectRowAt indexPath: IndexPath)

```
-> IndexPath? {
```

```
return nil
```

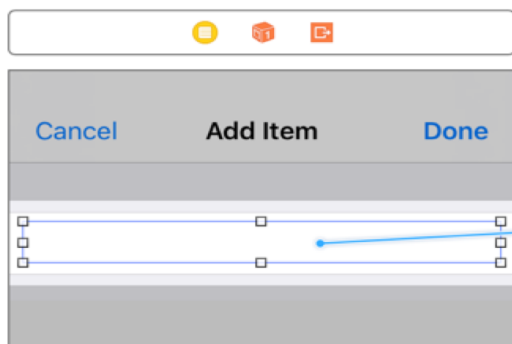
```
}
```



To-Do List: Read from the text field

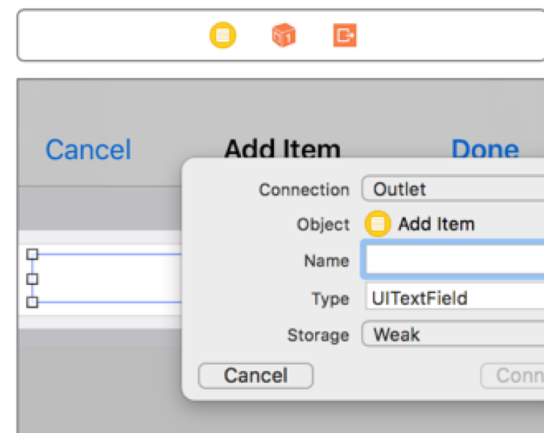
First, connect the text field to `@IBOutlet weak var textField: UITextField`

1



```
1 //
2 /**
3  * Copyright (c) 2017 Razeware LLC
4  */
5
6 import UIKit
7
8 class AddItemViewController: UITableViewController {
9
10  override func viewDidLoad() {
11      super.viewDidLoad()
12      navigationItem.largeTitleDisplayMode = .never
13  }
14 }
```

2



```
1 //
2 /**
3  * Copyright (c) 2017 Razeware LLC
4  */
5
6 import UIKit
7
8 class AddItemViewController: UITableViewController {
9
10  override func viewDidLoad() {
11      super.viewDidLoad()
12      navigationItem.largeTitleDisplayMode = .never
13  }
14 }
```

Read from the text file: the Code

- Add code to AddItemViewController.swift

```
@IBAction func done() {  
    // Add the following line  
    print("Contents of the text field: \(textField.text!)"  
    navigationController?.popViewController(animated:true)  
}
```

To Do List:

- ~~Add a static table cell to the table for data entry~~
- ~~Disable Cell Selection~~
- ~~Read from the text field~~
- Polish it up

To-Do List: Polish it up

- Give the text field focus on screen opening
- Style text field
- Handle the keyboard done button
- Disallow empty input

Give the text field focus on screen opening

- Add this code to AddItemViewController.swift

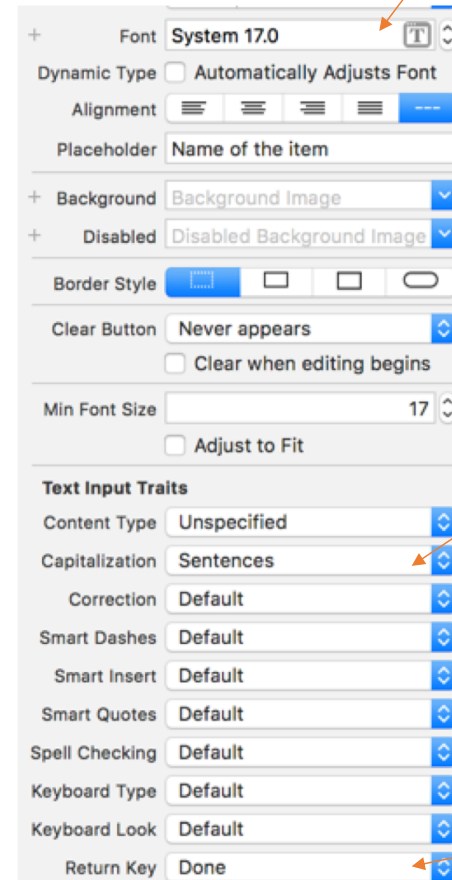
```
override func viewWillAppear(_ animated: Bool) {  
    super.viewWillAppear(animated)  
    textField.becomeFirstResponder()  
}
```

- In the simulator, if the keyboard does not open, press +K

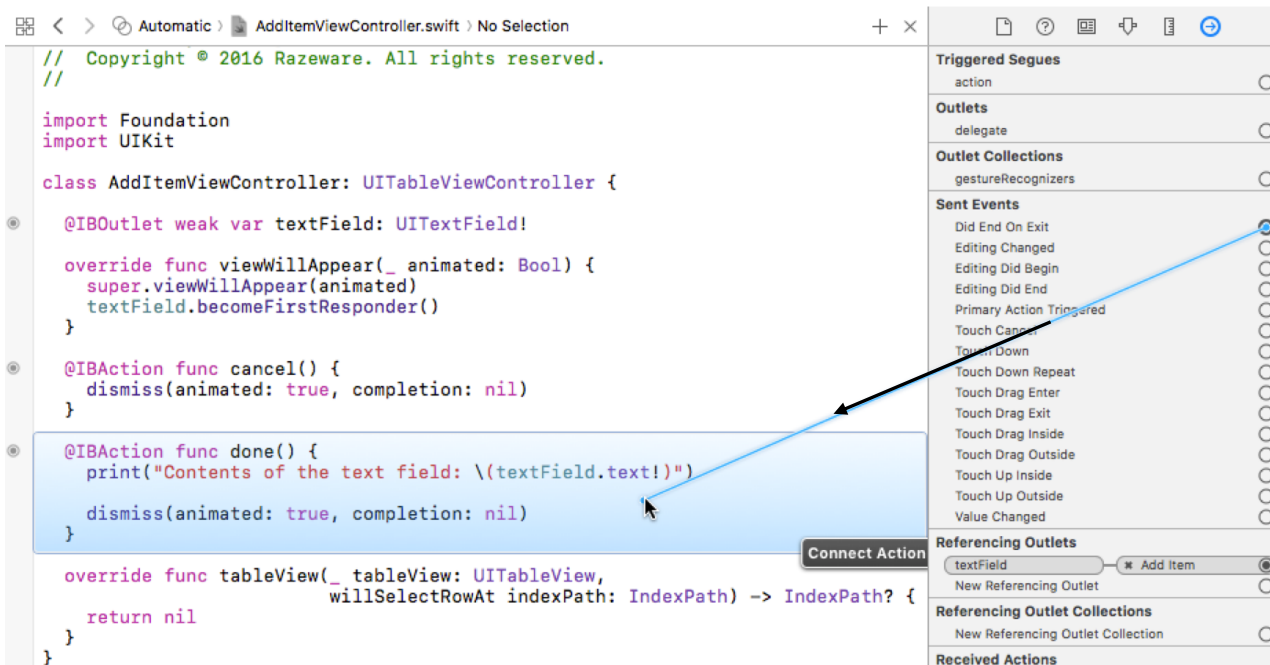
Style the text field

In Attributes Inspector:

- Placeholder: Name of the Item
- Font: System 17
- Adjust to Fit: Uncheck this
- Capitalization: Sentences
- Return Key: Done



Handle the keyboard done button



The screenshot shows the Xcode interface with a Swift file named `AddItemViewController.swift` open. The code defines a class `AddItemViewController` that inherits from `UITableViewController`. It includes a `UITextField` outlet and three methods: `viewWillAppear`, `cancel`, and `done`. The `done` method is highlighted in blue, and a blue arrow points from the `Done` button in the keyboard image to the `done` method. The Actions inspector on the right shows the `Done` event selected under the `Sent Events` section.

```
// Copyright © 2016 Razeware. All rights reserved.
//

import Foundation
import UIKit

class AddItemViewController: UITableViewController {

    @IBOutlet weak var textField: UITextField!

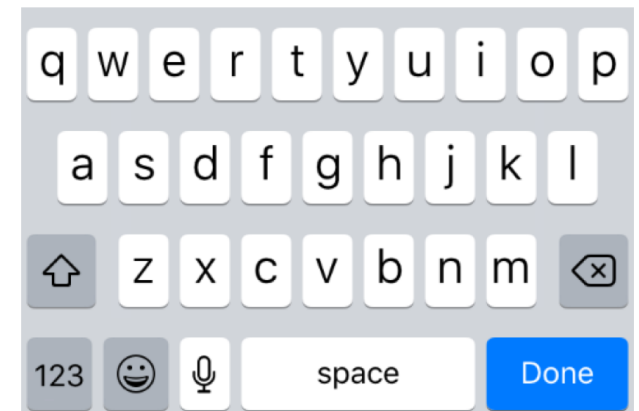
    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)
        textField.becomeFirstResponder()
    }

    @IBAction func cancel() {
        dismiss(animated: true, completion: nil)
    }

    @IBAction func done() {
        print("Contents of the text field: \(textField.text!)")
        dismiss(animated: true, completion: nil)
    }

    override func tableView(_ tableView: UITableView,
        willSelectRowAt indexPath: IndexPath) -> IndexPath? {
        return nil
    }
}
```

Connect the Did End On Exit to the func done()



To-Do List: Polish it up

- ~~Give the text field focus on screen opening~~
- ~~Style text field~~
- ~~Handle the keyboard done button~~
- Disallow empty input

Disallow Empty Input

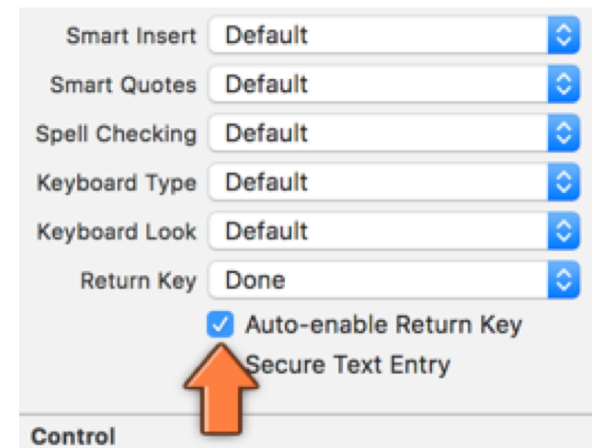
- Keyboard Done Button
- Navigation Done Button

Disallow empty text: Keyboard Done Button

- First, check the Auto-enable Return Key
- Disabled the done button on the keyboard

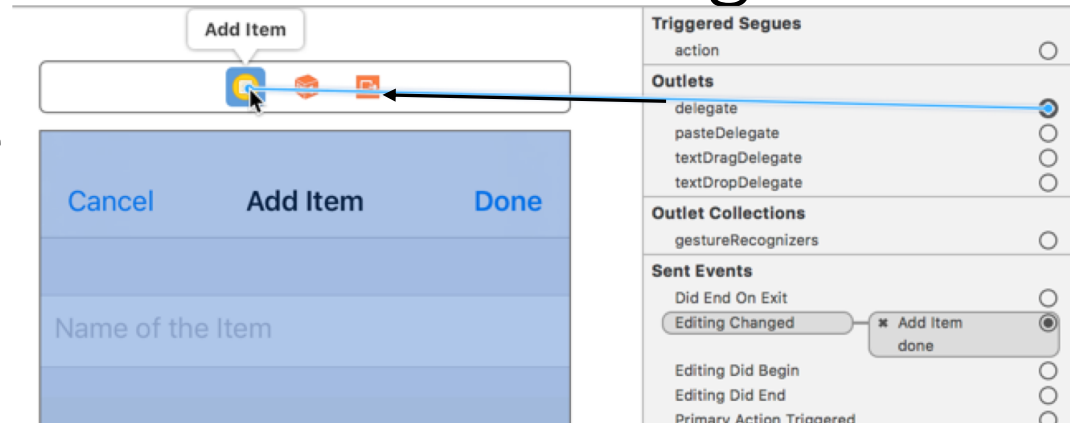
But what about Done button in the Navigation bar?

- We need to make the view controller into a delegate for the text field



Navigation Done Button: Become a delegate:

- Connect the text field's delegate outlet to the view controller



- Create a new outlet in AddItemViewController.swift

`@IBOutlet weak var doneBarButton: UIBarButtonItem!`

And connect it to the Done bar button

Navigation Done Button: The Code

- Add code to AddItemViewController.swift right before the last curly brace

```
let oldText = textField.text!
```

```
let stringRange = Range(range, in:oldText)!
```

```
let newText = oldText.replacingCharacters(in:stringRange, with: string)
```