GENERAL ASSEMBLY

OUTLETS

Zeke Abuhoff

Lead iOS Instructor, General Assembly

LEARNING OBJECTIVES

+ Connect storyboards to code with outlets
+ Get data from the UI using IBOutlets and IBActions
+ Change the UI using IBOutlets

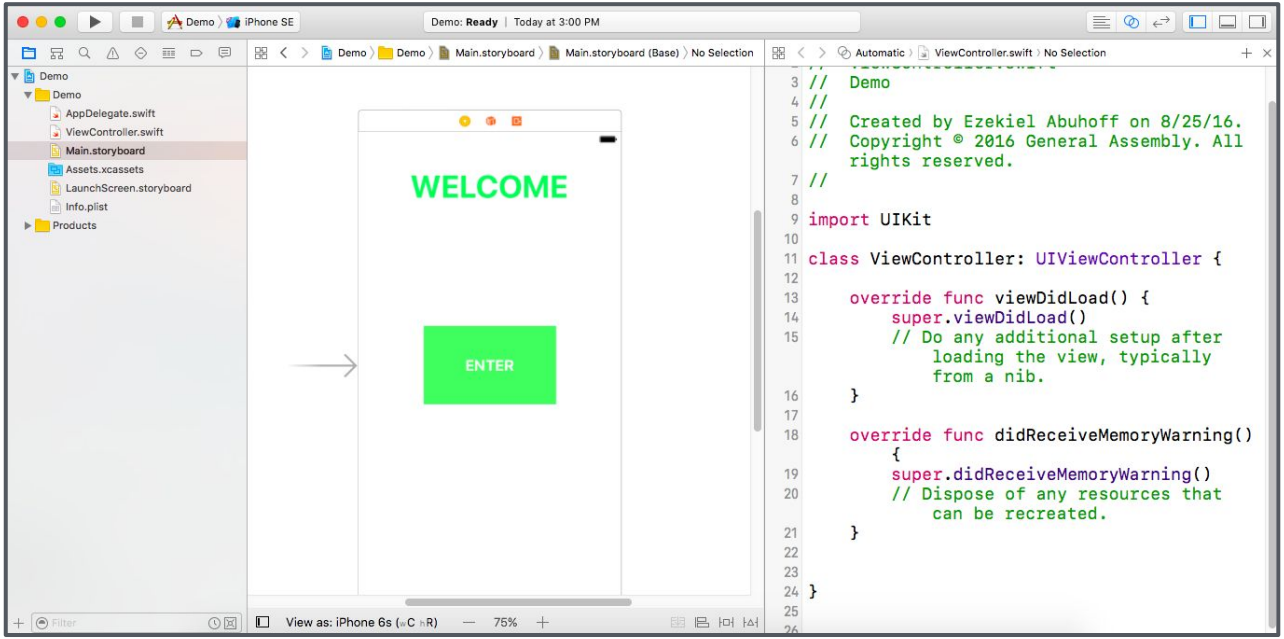
OUTLETS

STAR-CROSSED LOVERS

WILL STORYBOARDS AND CODE EVER BE TOGETHER?

OUTLETS

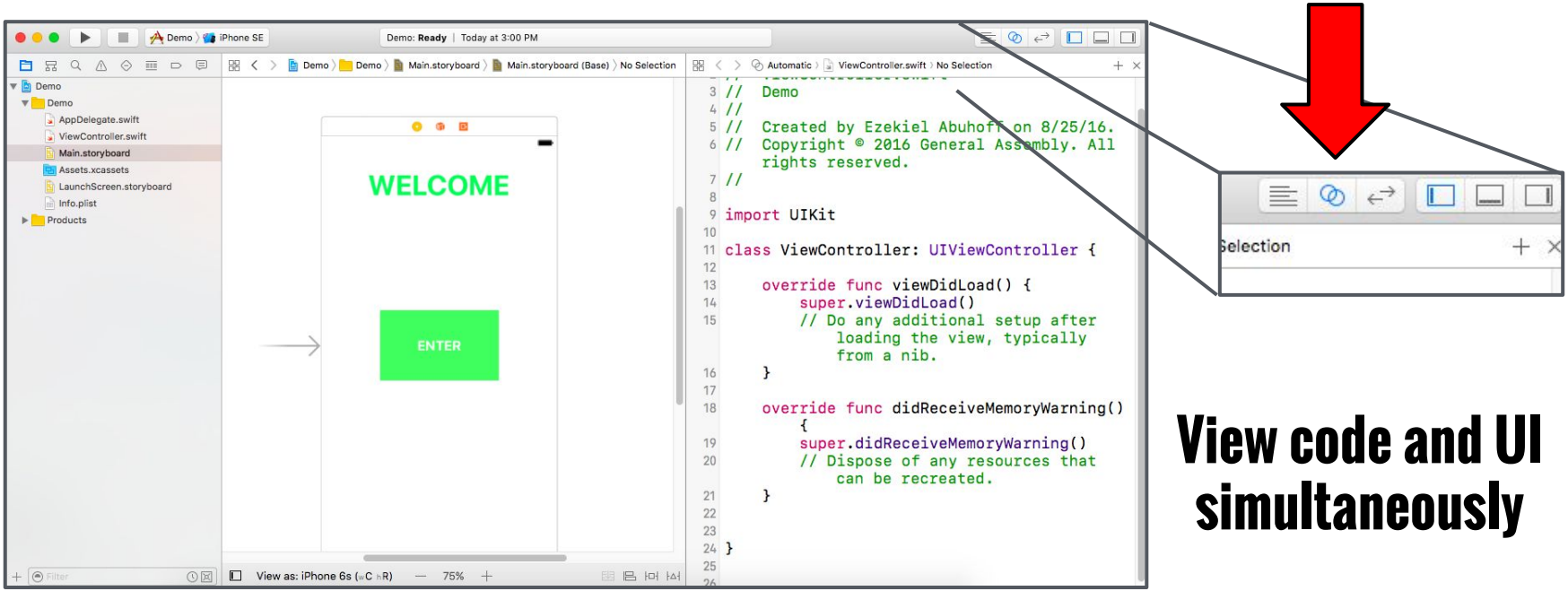
ASSISTANT EDITOR



View code and UI simultaneously

OUTLETS

ASSISTANT EDITOR



View code and UI simultaneously

CREATING OUTLETS

Ctl + Click + Drag

Connect a UI element to your view controller file to create a code reference to that element.

```
@IBOutlet weak var welcomeLabel: UILabel!
```

CREATING ACTIONS

Ctl + Click + Drag

When you connect a UI element with interactivity (like a button) to your code, you have the option of making an IBAction rather than an IBOutlet.

```
@IBAction func enterButtonPressed(_ sender: AnyObject) {
    // Execute code for entering the main menu
}
```

CREATING OUTLETS AND ACTIONS

Practice

1) In a new Xcode project, add a button, a label and an image view to your initial view controller.
2) Connect the UI elements in your view controller to your view controller code as outlets.
3) Connect the button to your view controller code as an action. Make that action set the label's text to "Connected!"

VIEW CONTROLLER CODE

Each view controller in your app will have behavior defined by a subclass of UIViewController.

```
class ViewController: UIViewController { ... }
```

The file you've used called "ViewController" features just one subclass of UIViewController to start you off.

When adding new view controllers with new, unique behavior, subclass UIViewController again.

VIEW CONTROLLER CODE

Your view controller subclasses will override certain methods that come from its UIViewController superclass.

Here are some frequently used ones:

```
override func viewDidLoad() { }
override func prepare(for segue: UIStoryboardSegue, sender: AnyObject?) { }
override func viewWillAppear(_ animated: Bool) { }
override func viewDidAppear(_ animated: Bool) { }
```

OUTLETS

Practice

1) In the view controller subclass you used for the last set of practice exercises, add another button. This one should lead to a new view controller.
2) Using interface builder, set your segue's reuse identifier to the string "myOneSegue".
3) Using an override of your first view controller's prepareForSegue method, print the reuse identifier of the segue in progress.
4) Using the second view controller's viewDidAppear method, print "We've arrived!" after the segue is complete.