GENERAL ASSEMBLY

MOBILE DEVELOPMENT

LESSON 07
SMALL WINS

#F6D3D2

GETTING STARTED

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- Recap of Tying IB to Code
- AIM Profile Project (cont.)
    - Pair Programming
    - Tying IB into Code
- View Controller Lifecycle
- String Tricks
- Switch Statements

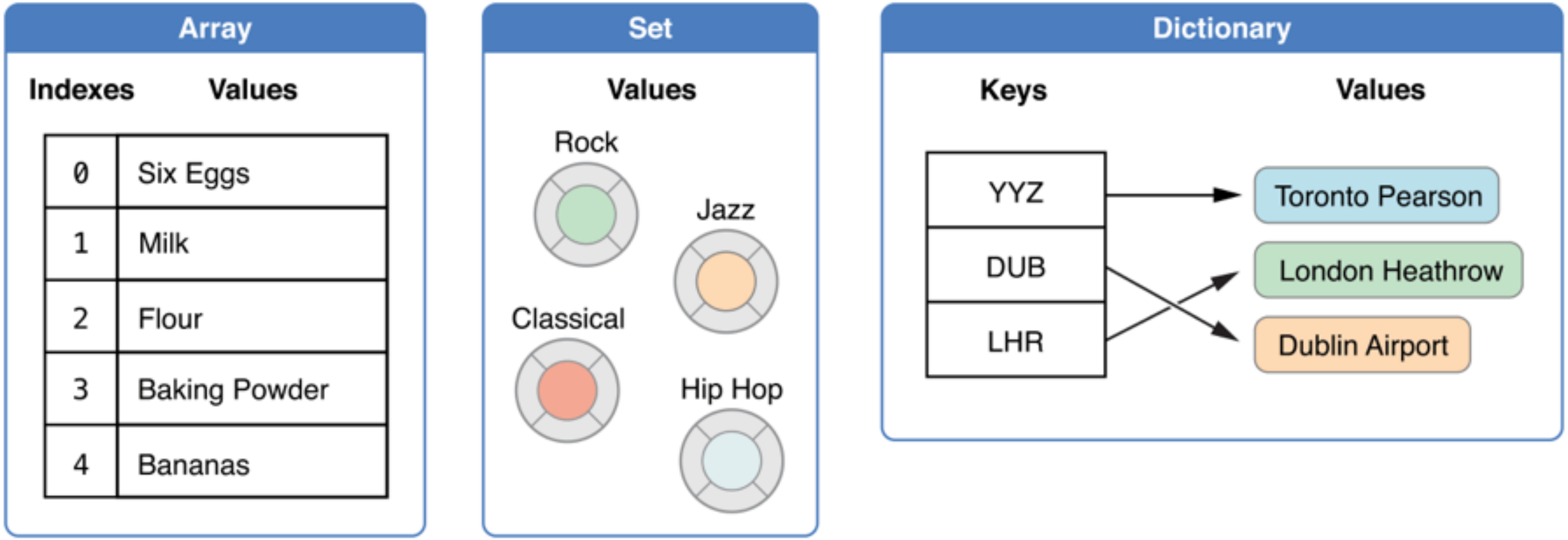
GETTING STARTED

COLLECTIONS

WHAT ARE COLLECTIONS?

- Collections, or Collection *Types, are ways to store collections of values.*
  - *Values are var, let, class instances, struct instances, other collections.*
- *What kind of collections exist?*
  - *Arrays*
  - *Dictionaries*
  - *Sets*
    - *We will not cover sets in class.*

WHAT ARE COLLECTIONS?

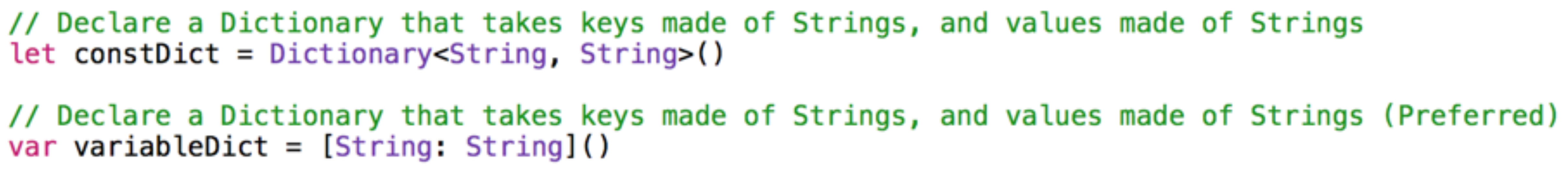


GETTING STARTED

DICTIONARIES

WHAT ARE DICTIONARIES?

- A dictionary stores associations between keys of the same type and values of the same type in a collection with no defined ordering



```
// Declare a Dictionary that takes keys made of Strings, and values made of Strings
let constDict = Dictionary<String, String>()

// Declare a Dictionary that takes keys made of Strings, and values made of Strings (Preferred)
var variableDict = [String: String]()
```

WHAT ARE DICTIONARIES?

- Dictionaries have a lot of cool properties and methods. We'll go over some of them:
    - Properties
        - isEmpty
        - count
        - keys
        - values
    - Methods
        - removeAll()
        - removeValueForKey(_:)

GETTING STARTED

BEGINNING PROGRAMMATIC VIEW CONTROLLER TRANSITIONS

GETTING STARTED

TABLE VIEW CONTROLLERS

WHAT IS A UITABLEVIEWCONTROLLER?

- Table views are a one dimensional list
- Vocabulary:
  - Section: All table views contain multiple sections
  - Row: Every section has a number of rows, which are entries in that section
  - Index path: The combination of a section and row that is a unique entry in a table view
  - Cell: The view that is displayed for an index path (the class UITableViewCell is a subclass of UIView)

WHAT IS A UITABLEVIEWCONTROLLER?

- Table views must have a number of sections, a number of cells in each section, and (optionally), the cells themselves
- Tableviews have a data source and a delegate
    - Data source: Provides cells, number of cells and number of sections
    - Delegate: Gets called when things happen to the table view, provides some views (e.g. header and footer)

IN-CLASS ASSIGNMENT

EXERCISE

KEY OBJECTIVE(S)

You will create a **secret playbook**. The app will have three screens:

1. Main screen. The main screen will have a button that will take the user to a table view controller.
2. Table view controller that will display all of the player roster.
3. Secret notes view. There will be no link to this view from the main view controller, however when the player swipes left, this view controller will display modally.
4. Bonus: Error view. If the player swipes right instead of left, display a view controller (modally) that flashes red. This will lock out the user for a bad attempt. The user will only be able to dismiss this view controller by swiping up.

TIMING

45 min	1. Code with partner
5 min	2. Debrief

VIEW CONTROLLER LIFECYCLE DEMO

GETTING STARTED

VIEW CONTROLLER LIFECYCLE INFO

BRIEF REVIEW ON VIEW

- “View controllers are a vital link between an app’s data and its visual appearance. Whenever an iOS app displays a user interface, the displayed content is managed by a view controller or a group of view controllers coordinating with each other. Therefore, view controllers provide the skeletal framework on which you build your apps.”

- Source: https://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/Introduction/Introduction.html

THE LIFECYCLE

- loadView()
- viewDidLoad()
- viewWillAppear()
- viewDidAppear()
- viewWillDisappear()
- viewDidDisappear()

VIEW CONTROLLER LIFECYCLE

LOADVIEW()

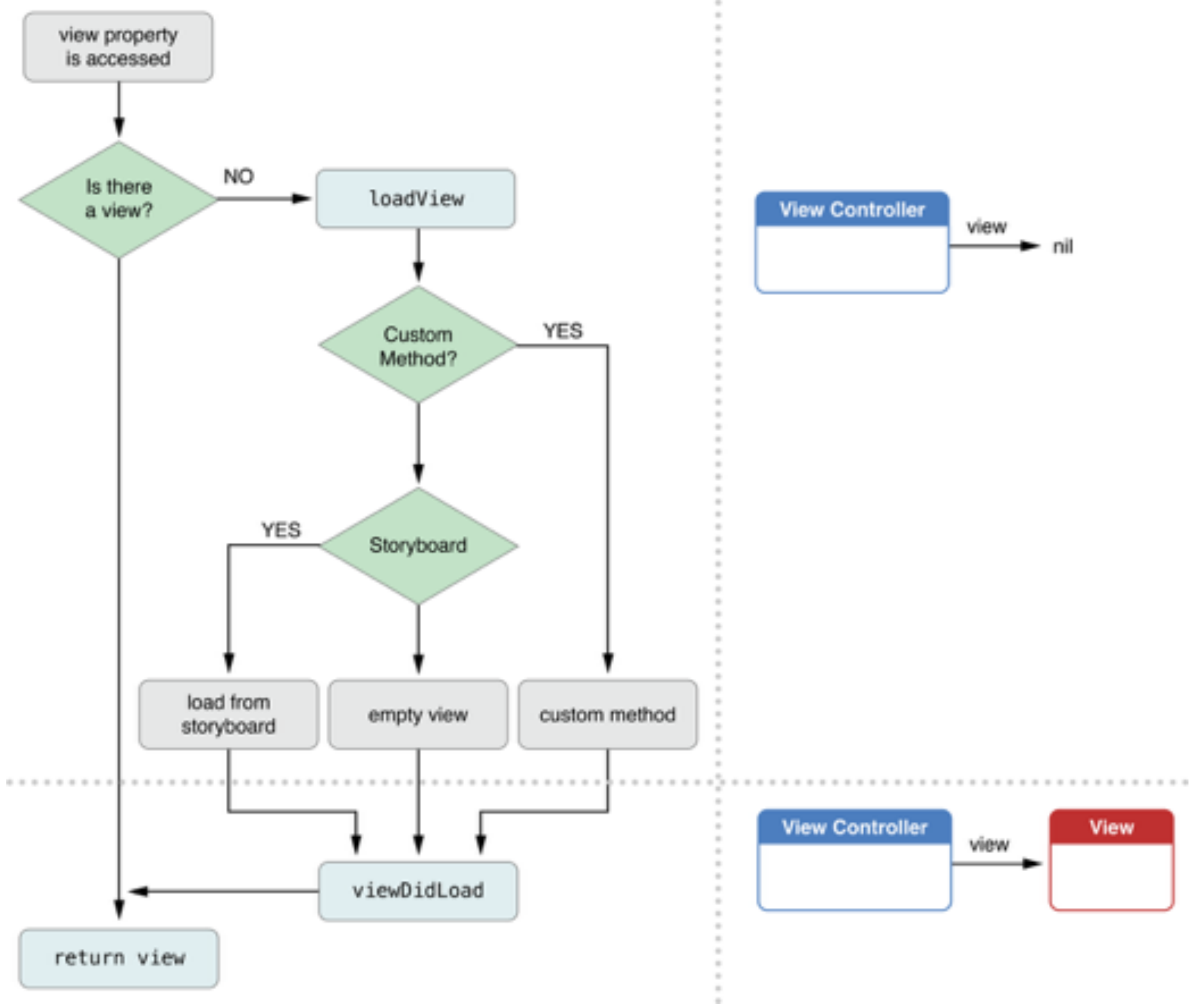
- Creates and links your view for you.
- This method is called IF-AND-ONLY-IF your view controller is backed by a xib or a storyboard file.
- If you are creating your view controller programmatically (e.g., without using storyboards or xibs), then this method will not be called.

- Do not override this method directly.

VIEW CONTROLLER LIFECYCLE

LOADVIEW()

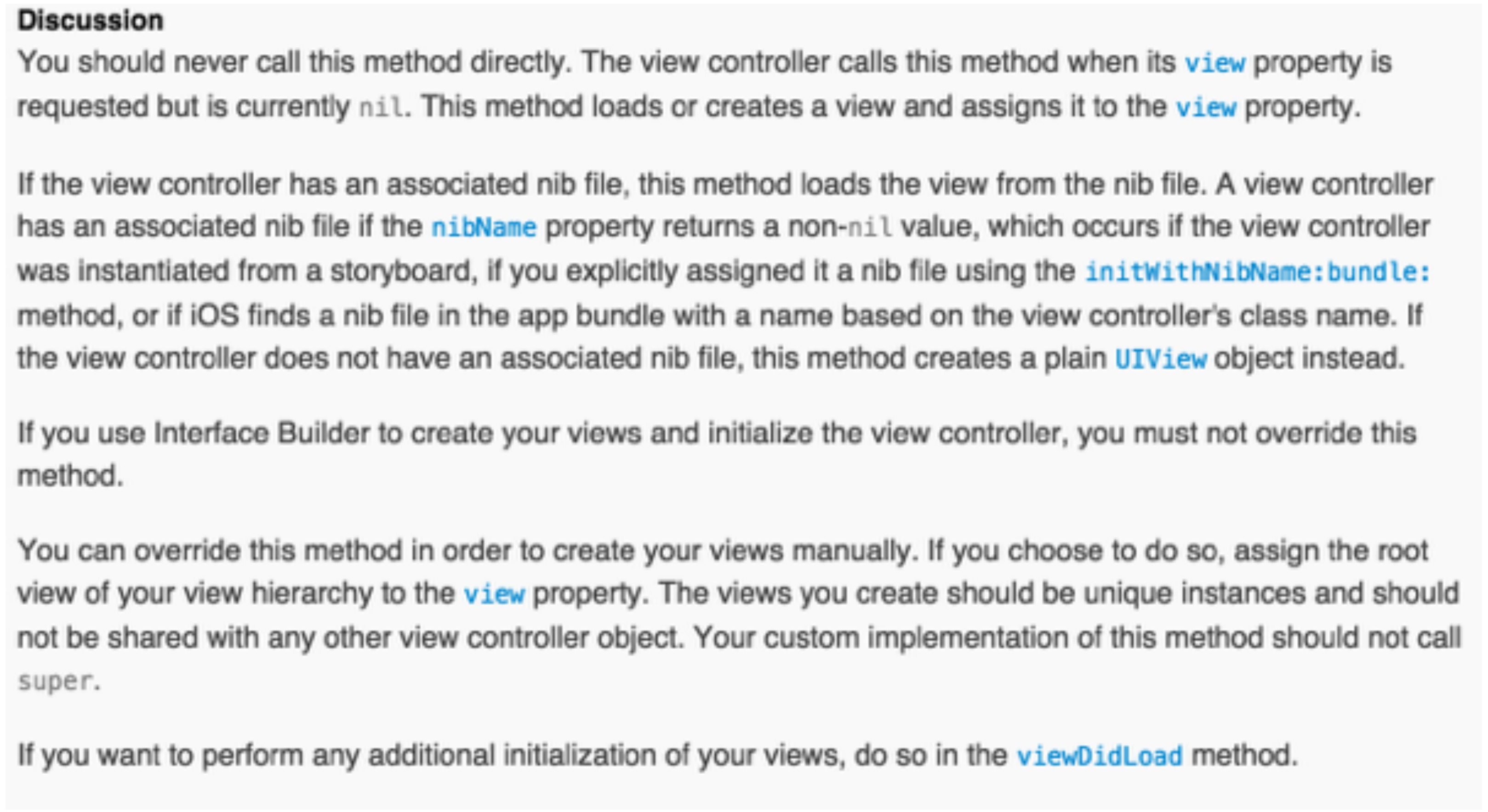
- More info here:
- https://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/ViewLoadingandUnloading/ViewLoadingandUnloading.html#//apple_ref/doc/uid/TP40007457-CH10-SW2



VIEW CONTROLLER LIFECYCLE

LOADVIEW()

- More info here:
- https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIViewController_Class/index.html#//apple_ref/occ/instm/UIViewController/loadView



Discussion

You should never call this method directly. The view controller calls this method when its `view` property is requested but is currently `nil`. This method loads or creates a view and assigns it to the `view` property.

If the view controller has an associated nib file, this method loads the view from the nib file. A view controller has an associated nib file if the `nibName` property returns a non-`nil` value, which occurs if the view controller was instantiated from a storyboard, if you explicitly assigned it a nib file using the `initWithNibName:bundle:` method, or if iOS finds a nib file in the app bundle with a name based on the view controller's class name. If the view controller does not have an associated nib file, this method creates a plain `UIView` object instead.

If you use Interface Builder to create your views and initialize the view controller, you must not override this method.

You can override this method in order to create your views manually. If you choose to do so, assign the root view of your view hierarchy to the `view` property. The views you create should be unique instances and should not be shared with any other view controller object. Your custom implementation of this method should not call `super`.

If you want to perform any additional initialization of your views, do so in the `viewDidLoad` method.

VIEW CONTROLLER LIFECYCLE

VIEWDIDLOAD()

- All view and sub-view configuration begins here, irrespective of how you instantiate your view controller (e.g., with or without storyboards/xibs).

- Always call super.viewDidLoad() before doing any other tasks.

VIEW CONTROLLER LIFECYCLE

VIEWWILLAPPEAR()

- Called every time the view controller becomes visible.
  - e.g. If you push back on a navigation controller, and it loads this view-controller, this method will be entered BEFORE that view becomes visible to the end user.

VIEW CONTROLLER LIFECYCLE

VIEWDIDAPPEAR()

- Called every time the view controller becomes visible.
  - e.g. If you push back on a navigation controller, and it loads this view-controller, this method will be entered AFTER that view becomes visible to the end user.

VIEW CONTROLLER LIFECYCLE

VIEWWILLDISAPPEAR()

- Called every time the view controller is removed from the screen.
  - e.g. If you push back on a navigation controller, and this view-controller is the one being dismissed/removed, this method will be entered right BEFORE that view is dismissed/removed.

VIEW CONTROLLER LIFECYCLE

VIEWDIDDISAPPEAR()

- Called every time the view controller is removed from the screen.
  - e.g. If you push back on a navigation controller, and this view-controller is the one being dismissed/removed, this method will be entered right AFTER that view is dismissed/removed.

GETTING STARTED

QUESTIONS ABOUT ANYTHING WE'VE LEARNED IN THE LAST 7 CLASSES?