https://github.com/GroundSpeed/SwiftlyBreakingGround

BREAKING GROUND WITH IOS

Don Miller
Founder of GroundSpeed™
Co-founder of Seed Coworking
@donmiller

SCHEDULE

Introductions

Discuss Terminology, Xcode Overview

Create Your First App: Hello World!

MVC Overview

Color Me Demo

Swift Basics and the Playground

Storyboards

UITableView

JSON & APIs

GroundSpeed™
rapid web + mobile software

TERMINOLOGY

- iOS
- iPhone
- iPad
- iPod
- MacBook
- iMac
- SSD
- SDK
- IDE
- Xcode
- OSX
- Mavericks
- Mountain Lion
- Lion
- Snow Leopard, etc.
- Retina

GroundSpeed™
rapid web + mobile software

XCODE 7.X & IOS 9.X

GroundSpeed™
rapid web + mobile software

HELLO WORLD!

GroundSpeed™
rapid web + mobile software

CREATE A NEW SINGLE VIEW APPLICATION

GroundSpeed™
rapid web + mobile software

ENTER THE PRODUCT NAME

GroundSpeed™
rapid web + mobile software

UPDATE THE GENERAL PROJECT ATTRIBUTES

HelloWorld
2 targets, iOS SDK 8.1
- HelloWorld
  - AppDelegate.swift
  - ViewController.swift
  - Main.storyboard
  - Images.xcassets
  - LaunchScreen.xib
  - Supporting Files
- HelloWorldTests
- Products

General | Capabilities | Info | Build Settings | Build Phases | Build Rules

PROJECT
- HelloWorld

TARGETS
- HelloWorld
- HelloWorldTests

Identity

Bundle Identifier	com.groundspeedhq.HelloWorld
Version	1.0
Build	1
Team	None

Deployment Info

Deployment Target	8.1
Devices	iPhone
Main Interface	Main
Device Orientation	☑ Portrait ☐ Upside Down ☐ Landscape Left ☐ Landscape Right
Status Bar Style	Default
	☐ Hide status bar

Identity and Type

Name	HelloWorld
Location	Absolute
	HelloWorld.xcodeproj
Full Path	/Users/donmiller/Documents/Projects/iOS/Playground/HelloWorld/HelloWorld.xcodeproj

Project Document

Project Format	Xcode 3.2-compatible
Organization	GroundSpeed
Class Prefix	

Text Settings

Indent Using	Spaces	
Widths	4 (Tab)	4 (Indent)

☑ Wrap lines

Source Control

Repository	--
Type	--
Current Branch	--
Version	--
Status	No changes
Location	

GroundSpeed™
rapid web + mobile software

PROJECT STRUCTURE

- Folders do not match physical layout on file system.
- Actually you will not find any sub folder named like Supporting Files in your physical project folder.
- The folder hierarchy in the XCode's navigator is simply a logical grouping of the resources.
- At this stage, we would be interested only on the Main.storyboard file.

GroundSpeed™
rapid web + mobile software

FILE INSPECTOR & SCHEMA SETTINGS

GroundSpeed™
rapid web + mobile software

DEVICES & TARGETS

General | Capabilities | Resource Tags | Info | Build Settings | Build Phases | Build Rules

PROJECT
- HelloWorld

TARGETS
- HelloWorld

Identity

Bundle Identifier: com.groundspeedhq.HelloWorld
Version: 1.0
Build: 1
Team: None

Deployment Info

Deployment Target: 9.2
Devices: iPhone
Main Interface: Main
Device Orientation:
- [x] Portrait
- [ ] Upside Down
- [ ] Landscape Left
- [ ] Landscape Right

Status Bar Style: Default
- [ ] Hide status bar
- [ ] Requires full screen

App Icons and Launch Images

App Icons Source: AppIcon
Launch Images Source: Use Asset Catalog
Launch Screen File: LaunchScreen

GroundSpeed™
rapid web+mobile software

FILE OWNER, VIEW, & FIRST RESPONDER

GroundSpeed™
rapid web + mobile software

RUN THE APPLICATION BY PRESSING PLAY

GroundSpeed™
rapid web + mobile software

ATTRIBUTE INSPECTOR

GroundSpeed™
rapid web + mobile software

INSPECTORS

GroundSpeed™
rapid web + mobile software

CREATE "HELLO WORLD!" LABEL

GroundSpeed™
rapid web + mobile software

Lab #1

Hello World!

Take 15 Minutes to Build It

GroundSpeed™
rapid web + mobile software

SWIFT BASICS

Xcode Playgrounds

Variables & Constants (Mutable and Immutable)

Types

- Strings, Integers, Floats, Booleans

Operators

- Binary and Unary Operators

Collection Types

- Arrays and Dictionaries

Control Flow

- For-In, While, For-Condition, If, Switch, Comparison and Logical Operators

GroundSpeed™
rapid web + mobile software

EXAMPLE: HAVE FUN IN THE PLAYGROUND

GroundSpeed™
rapid web + mobile software

ABOUT MUTABLE AND IMMUTABLE OBJECTS OR VARIABLES AND CONSTANTS

```
3  import UIKit
4
5  let immutable = "This cannot be changed"          "This cannot be changed"
6  immutable = "This will give an error"
7
8  var mutable = "This cannot be changed"            "This cannot be changed"
9  mutable = "This will NOT give an error"           "This will NOT give an error"
10
```

GroundSpeed™
rapid web + mobile software

DECLARING TYPES: STRINGS, INTEGERS, FLOATS, AND BOOLEANS

```
5  let string1 = "This is a string"                    "This is a string"
6  let string2: String = "This is string typed"        "This is string typed"
7
8  var intNumber1 = 1                                  1
9  intNumber1 = 2.5
10     Type 'Int' does not conform to protocol 'FloatLiteralConvertible'
11 let intNumber2: Int = 3                             3
12
13 var floatNumber1 = 1.0                              1.0
14 floatNumber1 = 2                                    2.0
15 floatNumber1 = 2.5                                  2.5
16
17 let floatNumber2: Float = 1                         1.0
18
19 let bool1 = true                                    true
20 let bool2: Bool = false                             false
```

GroundSpeed™
rapid web + mobile software

OPERATORS: BINARY AND UNARY
...AND MAYBE A TERNARY TOO

```
3   import UIKit
4
5   // binary
6   var a = 1                          1
7   a = a + 1                          2
8   a = a - 1                          1
9   a = a / 1                          1
10  a = a * 4                          4
11  a = a % 3                          1
12
13  // unary
14  var b = 1                          1
15  ++b                                2
16  b++                                2
17  b                                  3
18  b--                                3
19  b                                  2
20
21  // ternary
22  var bool = true                    true
23  var c = (bool ? 1 : 3)             1
24  bool = false                       false
25  c                                  1
26  c = (bool ? 1 : 3)                 3
```

GroundSpeed™
rapid web + mobile software

ARRAYS

```
5  var arrayFish = ["Bass", "Bluegill",
       "Crappie", "Muskie", "Walley"]
6
7  arrayFish.count
8  arrayFish[3]
9  arrayFish[5]   Execution was interrupted, reason: EXC_...
```

```
["Bass", "Bluegill", "Crappie", "Muskie", "Walley"]

5
"Muskie"
error
```

GroundSpeed™
rapid web + mobile software

CREATING ARRAYS AND DICTIONARIES WITH SWIFT

```
// Arrays
let arrayFish = ["Bass", "Bluegill", "Crappie", "Muskie", "Walley"]

arrayFish.count
arrayFish[3]


// Dictionaries
let dictStates = ["OH": "Ohio", "MI": "Michigan", "KY": "Kentucky", "IN": "Indiana", "IL":
    "Illinois", "PA": "Pennsylvania", "NY": "New York"]

dictStates.count
dictStates["OH"]!

// For Loop through dictionary
for (stateCode, stateName) in dictStates {
    print("\(stateCode): \(stateName)")
}

for stateCode in dictStates.keys {
    print("\(stateCode)")
}

for stateName in dictStates.values {
    print("\(stateName)")
}

// Array of stateCodes
let stateCodes = [String](dictStates.keys)
```

GroundSpeed™
rapid web + mobile software

CONTROL FLOW - LOOPING

```
// For-In
for index in 1...5 {
    print("\(index) times 5 is \(index * 5)")
}

let arrayFish = ["Bass", "Bluegill", "Crappie", "Muskie", "Walley"]
for fish in arrayFish {
    print("Fish array: \(fish)")
}

// For
for var index = 0; index < 5; ++index {
    print("For Index is \(index)")
}

// While
var index = 0
var max = 5
while index < max {
    print("While Index is \(index++)")
}

// Repeat-While
index = 0
max = 5
repeat {
    print("Repeat-While Index is \(index++)")
} while index < max
```

Console Output

```
1 times 5 is 5
2 times 5 is 10
3 times 5 is 15
4 times 5 is 20
5 times 5 is 25
Fish array: Bass
Fish array: Bluegill
Fish array: Crappie
Fish array: Muskie
Fish array: Walley
For Index is 0
For Index is 1
For Index is 2
For Index is 3
For Index is 4
While Index is 0
While Index is 1
While Index is 2
While Index is 3
While Index is 4
Do-While Index is 0
Do-While Index is 1
Do-While Index is 2
Do-While Index is 3
Do-While Index is 4
```

GroundSpeed™
rapid web + mobile software

CONTROL FLOW - DECISION

```
// IF
var temperatureInFahrenheit = 90
if temperatureInFahrenheit <= 32 {
    print("It's very cold. Consider wearing a scarf.")
} else if temperatureInFahrenheit >= 86 {
    print("It's really warm. Don't forget to wear sunscreen.")
} else {
    print("It's not that cold. Wear a t-shirt.")
}

// SWITCH
let someCharacter: Character = "a"
switch someCharacter {
case "a", "e", "i", "o", "u":
    print("\(someCharacter) is a vowel")
case "b", "c", "d", "f", "g", "h", "j", "k", "l", "m",
"n", "p", "q", "r", "s", "t", "v", "w", "x", "y", "z":
    print("\(someCharacter) is a consonant")
default:
    print("\(someCharacter) is not a vowel or a consonant")
}
```

```
90

"It's really warm. Don't forget to wear sunscreen.\n"

"a"

"a is a vowel\n"
```

GroundSpeed™
rapid web + mobile software

CLASS ILLUSTRATIONS

```
5  class Person {
6      var ssn: String!
7      var name: String!
8      var city: String!
9      var state: String!
10 }
11
12 class Employee: Person {
13     var empId: String!
14     var department: String!
15     var healthInsurance: String!
16
17     func printChecks() {
18         println("Printing Checks")
19     }
20 }
21
22 class SalariedEmployee: Employee {
23     var salary: String!
24 }
25
26 class HourlyEmployee: Employee {
27     var hourlyRate: String!
28 }
29
```

```
30 let hourlyEmployee = HourlyEmployee()
31 hourlyEmployee.name = "Don Miller"
32 hourlyEmployee.empId = "1234"
33 hourlyEmployee.hourlyRate = "9.99"
34
35 var output = "Name: \(hourlyEmployee.name) \r"
36 output += "Employee ID is \(hourlyEmployee.empId) \r"
37 output += "Hourly wage is \(hourlyEmployee.hourlyRate) \r"
38
39 println("\(output)")
40 hourlyEmployee.printChecks()
41
```

Console Output

```
Name: Don Miller
Employee ID is 1234
Hourly wage is 9.99
Printing Checks
```

GroundSpeed™
rapid web + mobile software

Need a Break?

GroundSpeed™
rapid web + mobile software

MVC, Storyboards, and Events

GroundSpeed™
rapid web + mobile software

IOS APPLICATION RUNTIME CYCLE

GroundSpeed™
rapid web + mobile software

COMMON TYPES OF EVENTS

Event type	Delivered to...	Notes
Touch	The view object in which the event occurred	Views are responder objects. Any touch events not handled by the view are forwarded down the responder chain for processing.
Remote control Shake motion events	First responder object	Remote control events are for controlling media playback and are generated by headphones and other accessories.
Accelerometer Magnetometer Gyroscope	The object you designate	Events related to the accelerometer, magnetometer, and gyroscope hardware are delivered to the object you designate.
Location	The object you designate	You register to receive location events using the Core Location framework. For more information about using Core Location, see *Location and Maps Programming Guide*.
Redraw	The view that needs the update	Redraw events do not involve an event object but are simply calls to the view to draw itself. The drawing architecture for iOS is described in *Drawing and Printing Guide for iOS*.

GroundSpeed™
rapid web + mobile software

MODEL - VIEW - CONTROLLER

GroundSpeed™
rapid web + mobile software

MODEL - VIEW - CONTROLLER

The Model: The M in MVC stands for the Model: For the Model, we will need to develop an independent class (or a set of classes) that will serve the data requirements and standard business procedures.

The View: In this context the View is the UI screen's layout (the canvas). Here, we configure many User Interface items (also known as UI controls on the screen).

The Controller: This is a class that connects the view with the model. In its simplest form, it contains the name of the UIControls (that would be used in the app), and it also includes the signatures and the detail implementations of event handlers.

GroundSpeed™
rapid web + mobile software

STORYBOARD OVERVIEW

Storyboards are graphic organizers displayed in sequence for the purpose of pre-visualizing a motion picture or animation. The storyboarding process, in the form it is known today, is used by developers to give the designer, developer, and end user an opportunity to see what the application or web site will look like at a high level.

In iOS, a Storyboard is a container of screens that may have embedded view controllers, navigation controllers, and tab bar controllers. It allows the user to graphically connect screens to one another through embedded objects on the screen. These connections are called segues (pronounced seg-wey). The storyboard eliminates the need for xib files and has one file that contains all of your screens and transitions. It is possible to connect two storyboard files with one another; however, unlikely with smaller applications.

PROS OF STORYBOARDS

Pros of using Storyboards:

- With a storyboard you have a better conceptual overview of all the screens in your app and the connections between them. It's easier to keep track of everything because the entire design is in a single file, rather than spread out over many separate nibs.

- The storyboard describes the transitions between the various screens. These transitions are called "segues" and you create them by simply ctrl-dragging from one view controller to the next. Thanks to segues you need less code to take care of your UI.

- Storyboards make working with table views a lot easier with the new prototype cells and static cells features. You can design your table views almost completely in the storyboard editor, something else that cuts down on the amount of code you have to write.

GroundSpeed™
rapid web + mobile software

CONS OF STORYBOARDS

Cons of using Storyboards:

- It's not easy to work with storyboards in a team, since only one participant can work on the storyboard at once (because it's one binary file).

- If you need to do things storyboard doesn't offer, it's not quite easy to get storyboard mixed with programmatically created views)

- Since all objects are in one file, a larger project could cause the storyboard to be very slow and unresponsive. It loads the entire thing into memory. It loads the entire item into memory.

GroundSpeed™
rapid web + mobile software

STORYBOARD OVERVIEW

GroundSpeed™
rapid web + mobile software

ANOTHER QUICK DEMO - [COLOR ME]

GroundSpeed™
rapid web + mobile software

CREATE A SINGLE VIEW APPLICATION

GroundSpeed™
rapid web + mobile software

CONFIGURE STORYBOARD BY DISABLING AUTO LAYOUT AND SIZE CLASSES

GroundSpeed™
rapid web + mobile software

DRAG A LABEL TO THE STORYBOARD

GroundSpeed™
rapid web + mobile software

CREATE A BLANK LABEL FOR OUTPUT

GroundSpeed™
rapid web + mobile software

CREATE TWO BUTTONS

GroundSpeed™
rapid web + mobile software

ADD OUTLETS TO CONTROLLER

```swift
//
//  ViewController.swift
//  ColorMe
//
//  Created by Don Miller on 12/30/14.
//  Copyright (c) 2014 GroundSpeed. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    @IBOutlet var lblMessage: UILabel!
    @IBOutlet var btnGreen: UIButton!
    @IBOutlet var btnRed: UIButton!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }


}
```

GroundSpeed™
rapid web + mobile software

ADD ACTIONS TO CONTROLLER AND UPDATE VIEWDIDLOAD

```
17      override func viewDidLoad() {
18          super.viewDidLoad()
19          // Do any additional setup after loading the view, typically from a nib.
20          lblMessage.text = ""
21      }
22
23      override func didReceiveMemoryWarning() {
24          super.didReceiveMemoryWarning()
25          // Dispose of any resources that can be recreated.
26      }
27
28      @IBAction func displayGreen(sender: AnyObject) {
29          lblMessage.text = "You wanted green"
30          self.view.backgroundColor = UIColor.greenColor()
31      }
32
33      @IBAction func displayRed(sender: AnyObject) {
34          lblMessage.text = "You wanted red"
35          self.view.backgroundColor = UIColor.redColor()
36      }
37  }
```

GroundSpeed™
rapid web + mobile software

CONNECTING THE OUTLETS AND ACTIONS

GroundSpeed™
rapid web + mobile software

Lab #2 - Create Color Me Storyboard App

GroundSpeed™
rapid web + mobile software

STORYBOARD DEMO
STEP 1: CREATE A NEW SINGLE VIEW APPLICATION

GroundSpeed™
rapid web + mobile software

STEP 2: OBSERVE THE SUMMARY PAGE AND TURN OFF SIZE CLASSES

Identity

Bundle Identifier: com.groundspeedhq.Storyboard
Version: 1.0
Build: 1
Team: None

Deployment Info

Deployment Target: 8.1
Devices: iPhone
Main Interface: Main
Device Orientation: ☑ Portrait ☐ Upside Down ☐ Landscape Left ☐ Landscape Right
Status Bar Style: Default
☐ Hide status bar

Identity and Type

Name: Main.storyboard
Type: Default - Interface Build...
Location: Relative to Group
Base.lproj/Main.storyboard
Full Path: /Users/donmiller/Documents/Projects/iOS/Playground/HelloWorld/HelloWorld/Base.lproj/Main.storyboard

Interface Builder Document

Opens in: Default (6.0)
Builds for: Project Deployment Tar...
View as: iOS 7.0 and Later
☐ Use Auto Layout
☐ Use Size Classes
Global Tint: Default

STEP 3: ADD 2 BUTTONS TO THE VIEW CONTROLLER
STEP 4: LET'S MAKE THEM THE SAME WIDTH IN SIZE INSPECTOR

GroundSpeed™
rapid web+mobile software

STEP 5: CHANGE THE BUTTON LABELS TEXT TO RED AND GREEN

GroundSpeed™
rapid web + mobile software

STEP 6: ADD TWO VIEW CONTROLLERS TO THE STORYBOARD BY DRAGGING THEM OVER

GroundSpeed™
rapid web + mobile software

STEP 7: CONNECT THE BUTTONS TO THE NEW VIEW CONTROLLERS BY HOLDING DOWN THE [CONTROL] KEY AND DRAGGING FROM THE BUTTON TO THE NEW VIEW CONTROLLER

GroundSpeed™
rapid web + mobile software

STEP 8: EMBED A NAVIGATION CONTROLLER ON YOUR PRIMARY VIEW

GroundSpeed™
rapid web + mobile software

STEP 9: CHANGE THE TITLE OF THE NAVIGATION BAR

GroundSpeed™
rapid web + mobile software

STEP 10: ADD A BAR BUTTON ITEM TO THE NAVIGATION BAR AND CHANGE THE TITLE TO BLUE

GroundSpeed™
rapid web + mobile software

STEP 11: ADD OUR THIRD AND FINAL VIEW CONTROLLER TO THE STORYBOARD

GroundSpeed™
rapid web + mobile software

STEP 12: CONNECT OUR BAR BUTTON ITEM TO THE NEW VIEW CONTROLLER AND SELECT PUSH. REMEMBER TO HOLD DOWN THE [CONTROL] BUTTON WHILE CLICKING AND DRAGGING

GroundSpeed™
rapid web + mobile software

STEP 13: NOW WE NEED TO DO A LITTLE PROGRAMMING. ADD A CONTROLLER OBJECT AND NAME IT "GREENVIEWCONTROLLER" AS SHOWN BELOW

STEP 14: ASSIGN YOUR VIEW IN THE STORYBOARD TO THE NEW OBJECT VIEW CONTROLLER YOU JUST CREATED

GroundSpeed™
rapid web + mobile software

STEP 15: SHOW THE ASSISTANT EDITOR TO HAVE OUR CODE GENERATED FOR US BY CLICKING

GroundSpeed™
rapid web+mobile software

STEP 16: ADD A BUTTON TO THE GREEN VIEW

GroundSpeed™
rapid web + mobile software

STEP 17: HOLD DOWN THE CONTROL WHEN CLICKING FROM THE BUTTON TO YOU HEADER FILE AND DRAG IT UNDER THE INTERFACE LINE. CHANGE THE CONNECTION TO ACTION AND THE NAME TO BTNEXIT

GroundSpeed™
rapid web + mobile software

STEP 18: ADD THE FOLLOWING LINE OF CODE IN THE BTNEXIT METHOD TO DISMISS THE MODAL WINDOW

```
9   import UIKit
10
11  class GreenViewController: UIViewController {
12
13      override func viewDidLoad() {
14          super.viewDidLoad()
15
16          // Do any additional setup after loading the view.
17      }
18
19      override func didReceiveMemoryWarning() {
20          super.didReceiveMemoryWarning()
21          // Dispose of any resources that can be recreated.
22      }
23
24      @IBAction func btnExit(sender: AnyObject) {
25          self.dismissViewControllerAnimated(true, completion: nil)
26      }
27  }
```

GroundSpeed™
rapid web + mobile software

STEP 19: FINAL TOUCHES. CHANGE THE VIEW CONTROLLER TITLES AND BACKGROUNDS TO REFLECT THE COLOR DESCRIBED IN THE CALLING BUTTON

GroundSpeed™
rapid web + mobile software

STEP 20: RUN IT!

GroundSpeed™
rapid web + mobile software

Lab #3 - Create Amortize Payment App

GroundSpeed™
rapid web + mobile software

CREATE AN APP TO CALCULATE AN AMORTIZATION PAYMENT

1. Create 5 Labels
2. Create 4 Text Boxes
3. Create 2 Buttons
   A. Clear clears all text boxes and total label
   B. Calculate creates payment amount on top label

$$P = A \cdot \frac{1 - \left(\frac{1}{1+r}\right)^n}{r}$$

GroundSpeed™
rapid web + mobile software

```
    func getMonthlyPayment() {
        // A = payment Amount per period
        // P = initial Principal (loan amount)
        // r = interest rate per period
        // n = total number of payments or periods

        let principal : Float = Float(txtAmount.text!)! -
Float(txtDownPayment.text!)!
        let payments = Float(txtTerm.text!)!*12
        let rate = Float(txtInterestRate.text!)!/12/100
        let amount = calculatPMTWithRatePerPeriod(rate, numberOfPayments:
payments, loanAmount: principal, futureValue: 0, type: 0)

        lblPayment.text = String(format: "%.02f", amount)
    }

    func calculatPMTWithRatePerPeriod (ratePerPeriod: Float, numberOfPayments:
Float, loanAmount: Float, futureValue: Float, type: Float) -> Float {

        var q : Float

        q = pow(1 + ratePerPeriod, numberOfPayments)

        let returnValue = (ratePerPeriod * (futureValue + (q * loanAmount))) /
((-1 + q) * (1 + ratePerPeriod * (type)))

        return returnValue

    }
```

$$P = A \cdot \frac{1 - \left(\frac{1}{1+r}\right)^n}{r}$$

GroundSpeed™
rapid web + mobile software

Lunch
Part 2 Afterwards

Don Miller

don@GroundSpeedHQ.com

@donmiller

http://github.com/donmiller

http://github.com/groundspeed

GroundSpeed™
rapid web + mobile software

Tab Bar and Navigation Controllers

GroundSpeed™
rapid web + mobile software

CREATE A NEW TABBED APPLICATION

GroundSpeed™
rapid web + mobile software

CREATE A NEW TABBED APPLICATION

Choose options for your new project:

Product Name: NavDemo

Organization Name: GroundSpeed

Organization Identifier: com.groundspeedhq

Bundle Identifier: com.groundspeedhq.NavDemo

Language: Swift

Devices: iPhone

☐ Include Unit Tests

☐ Include UI Tests

Cancel Previous Next

GroundSpeed™
rapid web + mobile software

NOTICE THE NEW TAB BAR AND THE CONNECTIONS TO THE TWO DEFAULT SCREENS

GroundSpeed™
rapid web + mobile software

THE IMAGES OF THE TAB BAR BUTTONS ARE CONFIGURED ON EACH INDIVIDUAL VIEW CONTROLLER

GroundSpeed™
rapid web + mobile software

THE TAB VIEW CAN CALL ANOTHER FORM BY EMBEDDING IT INTO A NAVIGATION CONTROLLER

GroundSpeed™
rapid web + mobile software

NAVIGATION TO THE NEW FORM CAN BE HANDLED BY CREATING A SEGUE FROM THE BUTTON TO THE NEW FORM (CONTROL-CLICK-DRAG)

GroundSpeed™
rapid web + mobile software

Lab #4 - Convert App to Tabbed View

GroundSpeed™
rapid web + mobile software

ADD A TAB BAR CONTROLLER TO THE AMORTIZATION PAYMENT APP

1. Open Amortized App
2. Open storyboard and drag the tab controller to it
3. Delete the two default screens
4. Connect tab controller to amortized app
5. Add TableView Controller and connect tab controller
6. Change the images and labels on each of the tab bar items

Download Images At:
http://sl.gshq.co/percent
http://sl.gshq.co/calculator

GroundSpeed™
rapid web + mobile software

UITableViews

GroundSpeed™
rapid web + mobile software

UITABLEVIEW OVERVIEW

GroundSpeed™
rapid web + mobile software

UITABLEVIEW OVERVIEW

A Simple UITableView with 4 Rows

A Sectioned UITableView

- Programmatically, the UIKit identifies rows and sections through their index number.
- Sections are numbered 0 through n-1 from the top of a table
- Rows are numbered 0 through n-1 within a section
- As Mentioned earlier, the table view comes in one of two basic styles: plain or grouped.

GroundSpeed™
rapid web+mobile software

TABLE VIEW EXAMPLE

In this example, we will develop a simple table view to display the names of our fish from the example earlier. The completed project will run as shown.

GroundSpeed™
rapid web + mobile software

STEP 1: CREATE A SINGLE VIEW BASED APP.

STEP 2: TURN OFF SIZE CLASSES

GroundSpeed™
rapid web + mobile software

STEP 3: DRAW TABLE VIEW

- **Open the Main.storyboard file.**
- **Draw a label and change its text to "Types of Fish".**
- **Now draw a TableView as shown**

STEP 4: UPDATE PROTOTYPE CELL WITH REUSE IDENTIFIER

GroundSpeed™
rapid web + mobile software

STEP 5: DATASOURCE & DELEGATE

Open the Property Inspector of the Table View, then drag its dataSource and delegate outlets and drop those on the File's Owner.

GroundSpeed™
rapid web+mobile software

STEP 6: OPEN THE VIEWCONTROLLER.SWIFT FILE AND INSERT THE FOLLOWING CODE

```
class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

    let arrayFish = ["Bass", "Bluegill", "Crappie", "Muskie", "Walley"]

    override func viewDidLoad() {
        super.viewDidLoad()
    }

    func numberOfSectionsInTableView(tableView: UITableView) -> Int {
        return 1
    }

    func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return self.arrayFish.count
    }

    func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->
UITableViewCell {
        let cell = tableView.dequeueReusableCellWithIdentifier("reuseIdentifier", forIndexPath:
indexPath) as UITableViewCell

        // Configure the cell...
        cell.textLabel?.text = self.arrayFish[indexPath.row]

        return cell
    }
}
```

GroundSpeed™
rapid web + mobile software

STEP 7: OPTIONAL HEADER AND FOOTER TITLE OF SECTIONS: YOU MAY IMPLEMENT THE FOLLOWING METHODS FOR THE HEADER AND FOOTER TITLES

```
func tableView(tableView: UITableView, titleForHeaderInSection section: Int) -> String?
{
    return "Types of Fish"
}

func tableView(tableView: UITableView, titleForFooterInSection section: Int) -> String?
{
    return "End of List"
}
```

GroundSpeed™
rapid web + mobile software

YOU ARE DONE!

GroundSpeed™
rapid web + mobile software

Lab #5 -

Prepare a static table to handle the rates data from Zillow

GroundSpeed™
rapid web + mobile software

ADD A TAB BAR CONTROLLER TO THE AMORTIZATION PAYMENT APP

1. Open Amortized App
2. Select the tableview
3. Change content to Static cells
4. Set Sections to 2
5. Set Style to Grouped
6. Add the labels to each of the sections
7. Create IB Connections to each of the labels on the right
8. Add some test data

JSON & APIs

GroundSpeed™
rapid web + mobile software

Sample API Output (JSON)


```
cb( {
"request" : {
    "output" : "json",
    "callback" : "cb"
},

"message" : {
    "text" : "Request successfully processed",
    "code" : "0"
},

"response" : {
    "today" : {
        "thirtyYearFixed" : "5.91",
        "thirtyYearFixedCount" : "1252",
        "fifteenYearFixed" : "5.68",
        "fifteenYearFixedCount" : "839",
        "fiveOneARM" : "5.49"
        "fiveOneARMCount" : "685"
    },

    "lastWeek" : {
        "thirtyYearFixed" : "6.02",
        "thirtyYearFixedCount" : "8933",
        "fifteenYearFixed" : "5.94",
        "fifteenYearFixedCount" : "5801",
        "fiveOneARM" : "5.71"
        "fiveOneARMCount" : "3148"
    }
}
} );
```

JSON

An *object* is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).

An *array* is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma).

A *value* can be a *string* in double quotes, or a *number*, or `true` or `false` or `null`, or an *object* or an *array*. These structures can be nested.

GroundSpeed™
rapid web + mobile software

Lab #6 - Fetch Zillow Rates from API

GroundSpeed™
rapid web + mobile software

STEP 1: OPEN AMORTIZED APP

```
9  import UIKit
10
11 class Rates {
12
13     var thirtyYearFixed: String
14     var fifteenYearFixed: String
15     var fiveOneARM: String
16
17     init(thirtyYearFixed: String, fifteenYearFixed: String, fiveOneARM: String) {
18         self.thirtyYearFixed = thirtyYearFixed
19         self.fifteenYearFixed = fifteenYearFixed
20         self.fiveOneARM = fiveOneARM
21     }
22
23 }
24
```


```
    },
  - response: {
      - today: {
            thirtyYearFixed: "3.78",
            thirtyYearFixedCount: "162620",
            fifteenYearFixed: "3.01",
            fifteenYearFixedCount: "156259",
            fiveOneARM: "3.0",
            fiveOneARMCount: "128583"
        },
      - lastWeek: {
            thirtyYearFixed: "3.8",
            thirtyYearFixedCount: "1825657",
            fifteenYearFixed: "2.99",
            fifteenYearFixedCount: "1835503",
            fiveOneARM: "2.99",
            fiveOneARMCount: "1479766"
        }
    }
}
```


GroundSpeed™
rapid web + mobile software

STEP 2: CREATE APIHELPER CLASS TO RETURN A DICTIONARY OF VALUES

```
11 class ApiHelper {
12
13     func getRatesFromZillow() -> Dictionary<String, String> {
14         var results : Dictionary<String, String> = Dictionary()
15
16 |       }
17         return results
18     }
19 }
```

GroundSpeed™
rapid web + mobile software

STEP 3: SETUP THE REQUEST TO THE API TO RETRIEVE ZILLOW RATES

```
16        let API_KEY = "X1-ZWz1f3t3bvl4i3_1brbp"
17        let postEndpoint = "http://www.zillow.com/webservice/GetRateSummary.htm?zws-id=\
              (API_KEY)&output=json"
18        print("URL \(postEndpoint)")
19
20        let url = NSURL(string: postEndpoint)!
21        let session = NSURLSession.sharedSession()
22
23        // Create the request
24        let request = NSMutableURLRequest(URL: url)
25
26        request.HTTPMethod = "GET"
27        request.setValue("application/json", forHTTPHeaderField: "Content-Type")
28        request.setValue("application/json", forHTTPHeaderField: "Accept")
```

GroundSpeed™
rapid web + mobile software

STEP 4: CREATE API SESSION

```
// Make the POST call and handle it in a completion handler
session.dataTaskWithRequest(request, completionHandler: { ( data: NSData?, response:
    NSURLResponse?, error: NSError?) -> Void in

    do
    {
        if let postString = try NSJSONSerialization.JSONObjectWithData(data!, options: []) as?
            NSDictionary
        {
            print("output: \(postString)")
            let response : Dictionary<String, AnyObject> = postString["response"] as!
                Dictionary<String, AnyObject>
            let today : Dictionary<String, String> = response["today"] as! Dictionary<String,
                String>
            let todayRate = Rates.init(thirtyYearFixed: today["thirtyYearFixed"]!,
                fifteenYearFixed: today["fifteenYearFixed"]!, fiveOneARM: today["fiveOneARM"]!)
            results["todayThirtyYearFixed"] = todayRate.thirtyYearFixed
            results["todayFifteenYearFixed"] = todayRate.fifteenYearFixed
            results["todayFiveOneARM"] = todayRate.thirtyYearFixed

            let lastWeek : Dictionary<String, String> = response["lastWeek"] as! Dictionary<
                String, String>
            let lastWeekRate = Rates.init(thirtyYearFixed: lastWeek["thirtyYearFixed"]!,
                fifteenYearFixed: lastWeek["fifteenYearFixed"]!, fiveOneARM:
                lastWeek["fiveOneARM"]!)
            results["lastWeekThirtyYearFixed"] = lastWeekRate.thirtyYearFixed
            results["lastWeekFifteenYearFixed"] = lastWeekRate.fifteenYearFixed
            results["lastWeekFiveOneARM"] = lastWeekRate.thirtyYearFixed
        }
        else
        {
            let jsonStr = NSString(data: data!, encoding: NSUTF8StringEncoding)
            // No error thrown, but not NSDictionary
            print("NODICT: Error could not parse JSON: \(jsonStr)")
        }
    }
    catch let parseError
    {
        print(parseError)
        // Log the error thrown by `JSONObjectWithData`
        let jsonStr = NSString(data: data!, encoding: NSUTF8StringEncoding)
        print("CATCH: Error could not parse JSON: '\(jsonStr)'")
    }
}).resume()
```

GroundSpeed™
rapid web + mobile software

STEP 5: CALL API METHOD FROM WITHIN THE TABLEVIEWCONTROLLER

```
26      override func viewDidLoad() {
27          super.viewDidLoad()
28
29          loadTextFields()
30      }
31
32      // MARK: - Table view data source
33      override func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
34          return 3
35      }
36
37      func loadTextFields() {
38
39          let zillowRates : Dictionary<String, String> = ApiHelper().getRatesFromZillow()
40
41          thirtyYearFixedToday.text = zillowRates["todayThirtyYearFixed"]
42          fifteenYearFixedToday.text = zillowRates["todayFifteenYearFixed"]
43          fiveOneArmToday.text = zillowRates["todayFiveOneARM"]
44          thirtyYearFixedLastWeek.text = zillowRates["lastWeekThirtyYearFixed"]
45          fifteenYearFixedLastWeek.text = zillowRates["lastWeekFifteenYearFixed"]
46          fiveOneArmLastWeek.text = zillowRates["lastWeekFiveOneARM"]
47
48      }
```

GroundSpeed™
rapid web + mobile software

YOU ARE DONE!

GroundSpeed™
rapid web + mobile software

Don Miller

don@GroundSpeedHQ.com

@donmiller

http://github.com/donmiller

http://github.com/groundspeed

GroundSpeed™
rapid web + mobile software