

KEYWORDS

DEBUGGING

- breakpoint
- UI debugging
- Instruments
- Console application

DEPENDENCY

- Cocoapods
 - `pod init`
 - `pod install`
 - `pod update`
 - use `.xcworkspace`
- Carthage
- Swift package manager

How To's

AutoLayout

- you need to practice it!!! ;)

Extras

UIScrollView

- add UIScrollView to UIViewController in Storyboard
- use PIN tool, add top, leading, trailing, bottom constraints with constant 0
- add UIView inside UIScrollView
- use PIN tool, add top, leading, trailing, bottom constraints with constant 0
- select both UIScrollView and UIView
- use PIN tool, add equal widths
- start adding your UI elements to UIView, you need to connect your UI elements with constraints from top to bottom, in order for UIScrollView to calculate its intrinsic content size.
- scroll view will only scroll if the size of its content is greater than the size of scroll view itself
- use my example and run it on iPhone SE

GCD

- Grand Central Dispatch

```
DispatchQueue.main.asyncAfter(deadline: .now() + 2) {  
    // Your code with delay  
}
```

UINavigationController

- will cover on next lecture but here are some tips if you are impatient
- UINavigationController is used for navigating between UIViewControllers and keeping them on stack, back and forth :)
- select your Login UIViewController from Storyboard
- Editor -> Embed In -> Navigation Controller
- create Register button action handler for event TouchUpInside
- open Login UIViewController subclass from code, go to Signup button action handler
- use this function to navigate to your new Registration UIViewController

```
navigationController?.pushViewController(<#T##viewController:  
UIViewController##UIViewController#>, animated: <#T##Bool#>)
```

- as you can see, this function takes a UIViewController subclass as a first parameter
- so you will need to load your new Registration UIViewController from Storyboard using

```
let bundle = Bundle.main  
let storyboard = UIStoryboard(name: "Main", bundle: bundle)  
let registrationViewController =  
storyboard.instantiateViewController(withIdentifier:  
"IdentifierThatYouNeedToSetUpInStoryboard")
```

- you can setup UIViewController identifier inside Storyboard
- the same location where you add your custom subclass
- under section `Identity`

Keyboard Events

```
NotificationCenter
    .default
    .addObserver(forName: UIResponder.keyboardWillShowNotification, object: nil, queue:
.main) { [weak self] notification in
    // keyboard is about to show, handle UIScrollView contentInset, e.g.
    self?.scrollView.contentInset = UIEdgeInsets(top: <T##CGFloat#>, left:
<T##CGFloat#>, bottom: <T##CGFloat#>, right: <T##CGFloat#>)
    }
NotificationCenter
    .default
    .addObserver(forName: UIResponder.keyboardWillHideNotification, object: nil, queue:
.main) { [weak self] notification in
    // keyboard is about to hide, handle UIScrollView contentInset, e.g.
    self?.scrollView.contentInset = .zero
    }
```