

Full Stack iOS Entwicklung mit Swift

WPF im MIM - SS 17
Alexander Dobrynin M.Sc.

Heute

Über Mich
Über die Veranstaltung
Learning Outcomes

iOS Entwicklung
Übersicht von iOS
Swift

Ressourcen
Demo

Über Mich

- Alexander Dobrynin, M.Sc.
- Medieninformatik Bachelor & Informatik Master (SE)
- Werkstudent bei OPITZ CONSULTING GmbH
- Wissenschaftlicher Mitarbeiter im ADV-Labor
- Java- und Scala-Enterprise-Apps (Backend)
- iOS-Apps (Front- und Backend)
- User-Centered-Agile-Software-Development (UCASD)
- Vollblut Nerd



@alexdobry
alexander.dobrynin@th-koeln.de

Über die Veranstaltung

- 5 ECTS-P, wahlweise **WPF A oder B** im MIM
- Max. 10 Personen
- Entwicklung von **coolen iOS Apps mit Swift**
 - vom Design bis zur Implementierung
 - kennenlernen einer neuen Programmiersprache und Systemumgebung
 - anwenden verschiedener Konzepte aus der Informatik
 - kollaboratives und agiles Arbeiten im Team
- Vorausgesetzt werden Kenntnisse in der **objektorientierten Programmierung**
- Benötet wird ein durchzuführendes Projekt und die Abschlusspräsentation

Über die Veranstaltung

- Vorlesung
 - Mobile Entwicklung, iOS, Xcode und Tools
 - Programmiersprache Swift, seine Konzepte und Patterns
 - “Real-World” objektorientierte- und angewandte funktionale Programmierung, Networking, Storage, Multithreading, ...
 - UI- und Interaction-Design
- Durchführung eines frei zu wählenden Projektes (max. 2er Teams)
 - Erarbeiten einer neuen Programmiersprache und kennenlernen einer neuen Systemumgebung
 - Design und Implementierung einer iOS App mit Swift
 - Inklusive eines Backends, einer Extension und/oder einem ausgewählten CoreModule (kontextabhängig)
 - Projektmanagement, Organisation und “Dokumentation” auf GitHub
 - Zwischenaudits während des Projektes - aktueller Stand, Fragen und Feedback der Studierenden
- Abschlusspräsentation (Anwesenheitspflicht)
 - Thema, Design, Architektur, Durchführung, Herausforderungen, besondere Highlights, ...
 - Entscheidungen begründen
 - Live-Demo der App (eigenes Gerät oder Simulator)

Über die Veranstaltung

- Dienstag, von 15 bis 17 Uhr in Raum 1242
- Zunächst vom 25.04 bis zum 09.05 (3 Wochen)
- Hackathon in der Profil²-Woche ab 15.05
- Weitere Termine für Vorlesungseinheiten und Zwischenaudits - TBA
- Abschlusspräsentation in der vorlesungsfreien Zeit (TBA)

Woche	03.04. - 07.04.	10.04. - 14.04.	17.04. - 21.04.	24.04. - 28.04.	01.05. - 05.05.	08.05. - 12.05.	15.05. - 19.05.	22.05. - 26.05.	29.05. - 02.06.	05.06. - 09.06.	12.06. - 16.06.	19.06. - 23.06.	26.06. - 30.06.	03.07. - 07.07.	10.07. - 14.07.	17.07. - 21.07.	24.07. - 28.07.
alle Semester																	

Vorlesungszeiten	
entsprechend Ihrem Veranstaltungsplan	

Blockveranstaltungen / Projektwochen	
Blockwoche / Projektarbeiten	10.04.2017 - 21.04.2017
Profil ²	15.05.2017 - 19.05.2017

Prüfungszelträume	
alle Semester	03.04.2017 - 07.04.2017 24.07.2017 - 28.07.2017

Feiertage / vorlesungsfrei	
Ostern	14.04.2017 - 17.04.2017
Maifeiertag	01.05.2017
Chr. Himmelfahrt	25.05.2017
Pfingsten	02.06.2017 - 06.06.2017
Fronleichnam	15.06.2017

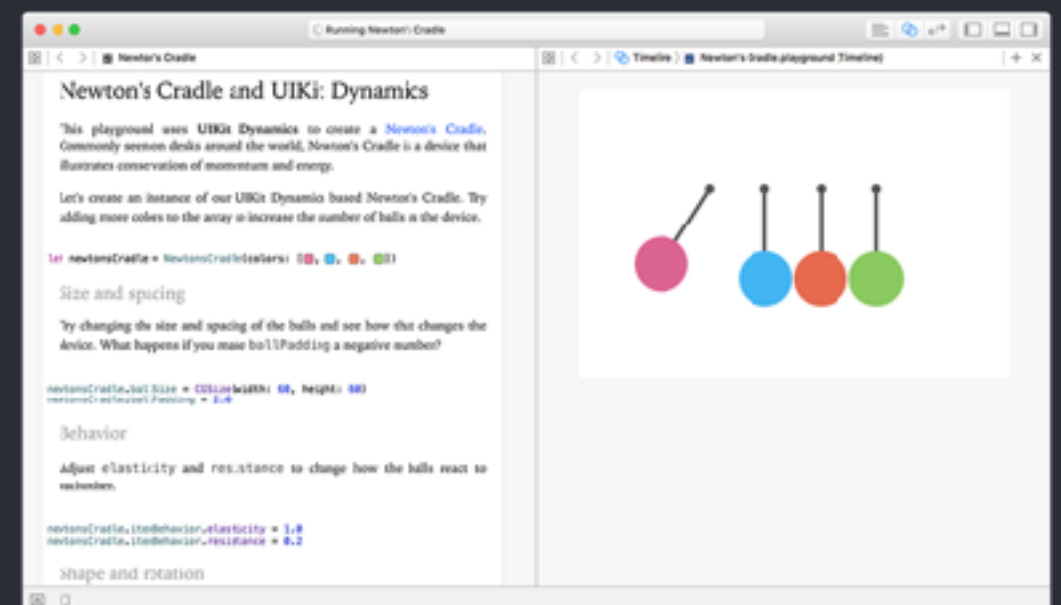
Termine Schulferien im Vorlesungszeitraum	
Osterferien	10.04.2017 - 22.04.2017
Sommerferien	17.07.2017 - 29.08.2017

Learning Outcomes

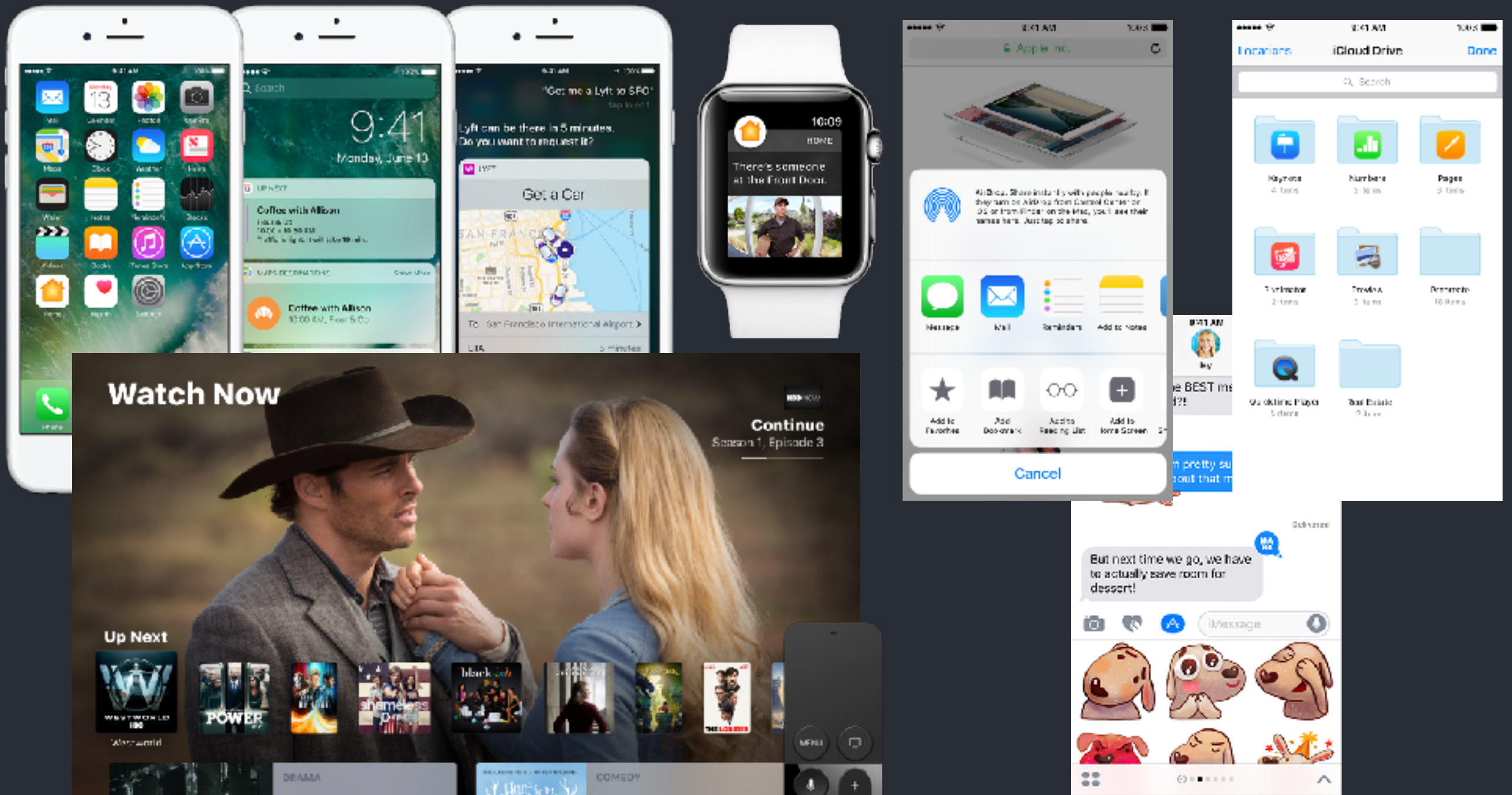
- Probleme Analysieren und Dekonstruieren, Lösungen Konzipieren, Fertigen und Bewerten
- Bestehende Konzepte der Informatik konsolidieren
- Neue Programmiersprache, Konzepte und Sprachfeatures
- Mobile (iOS-lastige) Patterns und Programmierstile
- Funktionale Denkweise
- Mobile Entwicklung im Allgemeinen
- iOS Entwicklung im Speziellen
- Zusammenspiel von Design und Implementierung
- Zusammenspiel von Mobile- und Backend App
- Kollaborative Arbeiten mit Git und GitHub
- Agile Softwareentwicklung in einem Team

iOS Entwicklung

- Kommt, wird hacken coole Apps!
- Mobiler Computer, mobile Lösungen
- Sensoren, Kamera, mobiles Internet, usw.
- Mit einfachen UIs komplexe Probleme effizient lösen
- Integration und Interprozesskommunikation mit anderen Apps und Systemen
- Weitreichendes Ökosystem



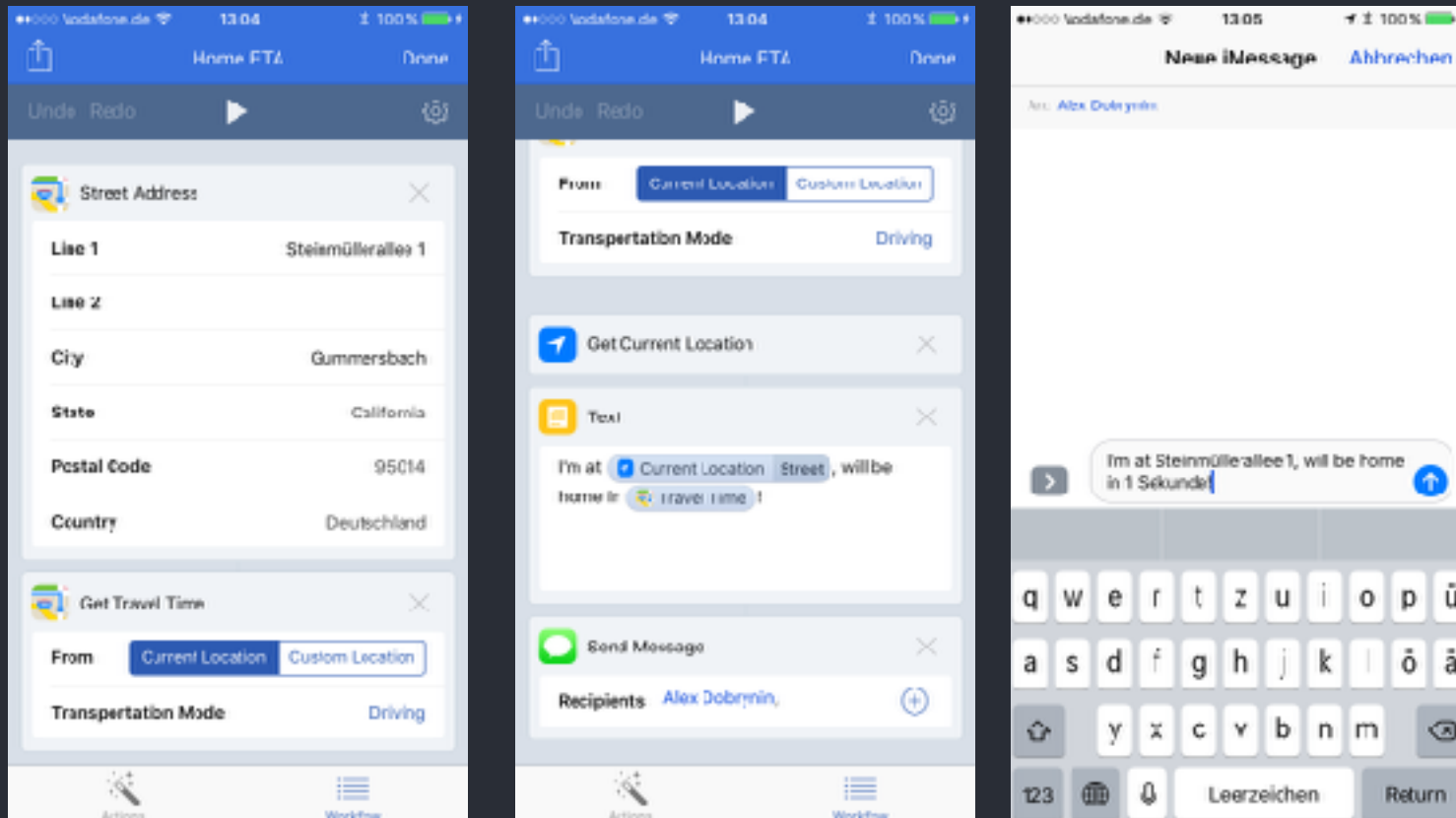
iOS Entwicklung



<https://developer.apple.com/ios/human-interface-guidelines/extensions/messaging/>
<https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/>
<https://developer.apple.com/watchos/human-interface-guidelines/overview/>
<https://developer.apple.com/tvos/human-interface-guidelines/overview/>

Full Stack iOS Entwicklung

- App + Backend = mobile Development™
- App trägt zum Erfolg des Geschäftes bei
- "Build Features, Not Apps"



https://commons.wikimedia.org/wiki/File:Car2go_logo.svg
https://en.wikipedia.org/wiki/File:Tinder_logo.png
<https://outbankapp.com>
<https://box.fileee.com/en/>
https://de.wikipedia.org/wiki/Datei:Instagram_logo_2016.svg
<https://workflow.is/static/landing/img/help-icon.png>

Übersicht von iOS

Cocoa Touch

UIKit, MapKit, App Extensions, ...

Media

Core Graphics, Core Audio, AVFoundation, ...

Core Services

SQLite, Foundation, Core Location, ...

Core OS

Kernel, Security, Filesystem, ...

Übersicht von iOS

- 100% objektorientiert (zumindest der Kern von Cocoa-Touch)
- Objective-C und Swift als Programmiersprachen
- Xcode als IDE
- Swift Playgrounds als “interaktive REPL”
- Profiling mit Instruments
- Speicherverwaltung mit Automatic-Reference-Counting (ARC)
- Code-Design mit Model-View-Controller (MVC)
- UI-Design mit Storyboards (in Xcode)

Übersicht von iOS

- Verschiedene Geräte, Gerätegrößen und OS-Versionen
- Verschiedene Plattformen (iOS, tvOS, watchOS, macOS)
- Event-Getriebene Eingaben auf dem Main-Thread
- Eingeschränkte Ressourcen
- Zentraler Einstiegspunkt und definierter Lebenszyklus

```
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    func application(UINavigationController, didFinishLaunchingWithOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool
    func applicationDidEnterBackground(UINavigationController)
    func applicationWillEnterForeground(UINavigationController)
    func applicationDidReceiveMemoryWarning(UINavigationController)
}
```

- Zustände, Transitionen und “Callbacks”

Swift

- Vorgestellt auf der WWDC am 02.06.2014
- Statisch typisierte, multiparadigmen Programmiersprache
- Mittlerweile in der Version 3.1 und Open-Source
- “Objective-C without the C”
- Auf lange Sicht als primäre Programmiersprache für iOS und MacOS Anwendungen

*“Swift combines the best in **modern language thinking** with wisdom from the wider Apple engineering culture. The compiler is **optimized for performance**, and the language is **optimized for development**, without compromising on either. It’s **designed to scale** from “hello, world” to an entire operating system. All this makes Swift a sound future **investment for developers and for Apple.**”*

Ressourcen

- swift.org
- github.com/apple/swift
- github.com/apple/swift-evolution/blob/master/proposals
- developer.apple.com/swift/programming_language
- itunes.apple.com/de/book/the-swift-programming-language-swift-3-1
- developer.apple.com/swift/getting_started
- developer.apple.com/sample_code
- developer.apple.com/ios/human-interface-guidelines
- github.com/ochococo/Design-Patterns-In-Swift

Nächste Woche

Model-View-Controller (MVC)
DataSource, Delegate, Target-Action, Notification

Demo - MatchingCardGame

Demo - Card

Xcode, Projekt erstellen, App im Simulator ausführen
UI-Design im Storyboard und Auto-Layout
AppDelegate und Lebenszyklus einer App
Klassen, Variablen und Funktionen/Methoden in Swift
Verbindungen zwischen Storyboard (UI) und ViewController (Code)
UIButton, UILabel und Target-Action