



M-Lab 2015 iOS-Tutorial

MOBIS, Uni Hamburg

Outline

- Swift
 - Basics
 - Beyond Basics
- iOS
- GIT
- Tutorial
 - Part 1
 - Part 2
- Literature



Swift Basics



Variables

```
var courseTitle: String = "M-Lab"
```



Constants

```
let courseTitle: String = "M-Lab"
```

Variables and Constants



```
let courseTitle: String = "M-Lab"  
var courseTerm: Int = 2015  
let isAwesome: Bool = true
```



Type Inference

```
let courseTitle = "M-Lab"    // inferred as String  
var courseTerm = 2015        // inferred as Int  
let isAwesome = true         // inferred as Bool
```



String Interpolation

```
let a = 3, b = 7
```

```
// Result: "3 times 7 is 21"
```

```
let result = "\(a) times \(b) is \(a * b)"
```



String Mutability

```
// "iOS and Swift"  
var varibaleString = "iOS"  
varibaleString += "and Swift"
```

```
// Error, Constant string cannot be changed  
let constantString = "iOS"  
constantString += "and Swift"
```



Arrays

```
var names = ["Anna", "Alex", "Brian"]
```



Array Modification

```
// ["Anna", "Alex", "Brian"]
var names = ["Anna", "Alex", "Brian"]

// ["Anna", "Alex", "Brian", "Marlo", "Mathias"]
names += ["Marlo", "Mathias"]

// ["Jan", "Alex", "Brian", "Marlo", "Mathias"]
names[0] = "Jan"

// ["Jan", "Markus", "Daniel", "Marlo", "Mathias"]
names[1...2] = ["Markus", "Daniel"]
```



Dictionaries

```
var numberOfLegs = ["ant": 6, "snake": 0,  
                    "cheetah": 4]
```

Dictionary Modification



```
var numberOfLegs = ["ant": 6, "snake": 0,  
                    "cheetah": 4]
```

```
numberOfLegs["spider"] = 273
```



Typed Collections

```
var names = ["Anna", "Alex", "Brian", true]
```

// An array of string values

```
var names: [String] = ["Anna", "Alex", "Brian"]
```



Loops

```
while !sated {  
    eatCake()  
}
```

```
for var i = 1; i <= 14; ++i {  
    eatCake()  
}
```



For-In: Ranges

```
for number in 1...4 {  
    print("\((number) * 4) = \((number * 4)")  
}  
  
// 1 * 4 = 4  
// 2 * 4 = 8  
// 3 * 4 = 12  
// 4 * 4 = 16
```



For-In: Arrays

```
for name in ["Anna", "Alex", "Brian"] {  
    print("Hello, \(name)!")  
}
```

```
// Hello, Anna!  
// Hello, Alex!  
// Hello, Brian!
```



For-In: Dictionaries

```
var number0fLegs = ["ant": 6, "snake": 0,  
                    "cheetah": 4]  
  
for (animalName, legs) in number0fLegs {  
    print("\(animalName)s have \(legs) legs")  
}  
  
// ants have 6 legs  
// snakes have 0 legs  
// cheetahs have 4 legs
```



Swift Beyond Basics



Optionals

```
let number0fLegs = ["ant": 6, "snake": 0,  
                    "cheetah": 4]
```

```
// number0fLegs["abc"] does not exists  
let possibleLegCount: Int? = number0fLegs["abc"]
```

- Used in situations where value may be absent
- Alternative for Objective-C "nil" passing
- Works with any type



Querying an Optional

```
let possibleLegCount: Int? = number0fLegs["abc"]

if possibleLegCount == nil {
    print ("abc was not found.")
} else {
    let legCount: Int = possibleLegCount!
    print("abc has \(legCount) legs.")
}
```



Querying an Optional

```
if possibleLegCount {  
    let legCount = possibleLegCount!  
    print("abc has \(legCount) legs.")  
}
```



Querying an Optional

```
if let legCount = possibleLegCount {  
    print("abc has \(legCount) legs.")  
}
```



If-Statement

```
if legCount == 0 {  
  
} else if legCount == 1 {  
  
} else {  
  
}
```



Switch-Statement

```
switch legCount {  
    case 0:  
        print("")  
    case 1:  
        print("")  
    default:  
        print("")  
}
```



Switch-Statement

```
switch legCount {  
    case 0:  
        print("")  
    case 1, 2, 3:  
        print("")  
    default:  
        print("")  
}
```



Switch-Statement

```
switch legCount {  
    case 0:  
        print("")  
    case 1...3:  
        print("")  
    default:  
        print("")  
}
```



Functions

```
func sayHello() {  
    print("Hello!")  
}
```

```
sayHello()
```

Functions with Parameters



```
func sayHello(name: String) {  
    print("Hello \(name)!")  
}
```

```
sayHello("M-Lab")
```

Functions with Default-Parameters



```
func sayHello(name: String = "M-Lab") {  
    print("Hello \(name)!")  
}  
  
sayHello()          // "Hello M-Lab!"  
sayHello(name: "Daniel") // "Hello Daniel!"
```



Return Values

```
func sayHello(name: String = "M-Lab") -> String {  
    return "Hello " + name  
}
```

```
let greeting = sayHello()  
print(greeting)
```

Return Multiple Values



```
func refreshWebPage() -> (Int, String) {  
    return(200, "Success")  
}
```

Decomposing a Tuple



```
func refreshWebPage() -> (Int, String) {  
    return(200, "Success")  
}
```

```
let (statusCode, message) = refreshWebPage()
```

```
print("Received \(statusCode): \(message)")
```



Classes

```
class Vehicle {  
    // properties  
    // methods  
    // initializers  
}
```



Class Inheritance

```
class Bicycle: Vehicle {  
}
```



Properties

```
class Vehicle {  
    var numberOfWheels = 0  
}
```



Dot-Syntax

```
let someVehicle = Vehicle()
```

```
println(someVehicle.numberOfWorks)
```

```
someVehicle.numberOfWorks = 2
```



Class Initialization

```
class Bicycle: Vehicle {  
    init() {  
        super.init()  
        numberOfWorkers = 2  
    }  
}
```

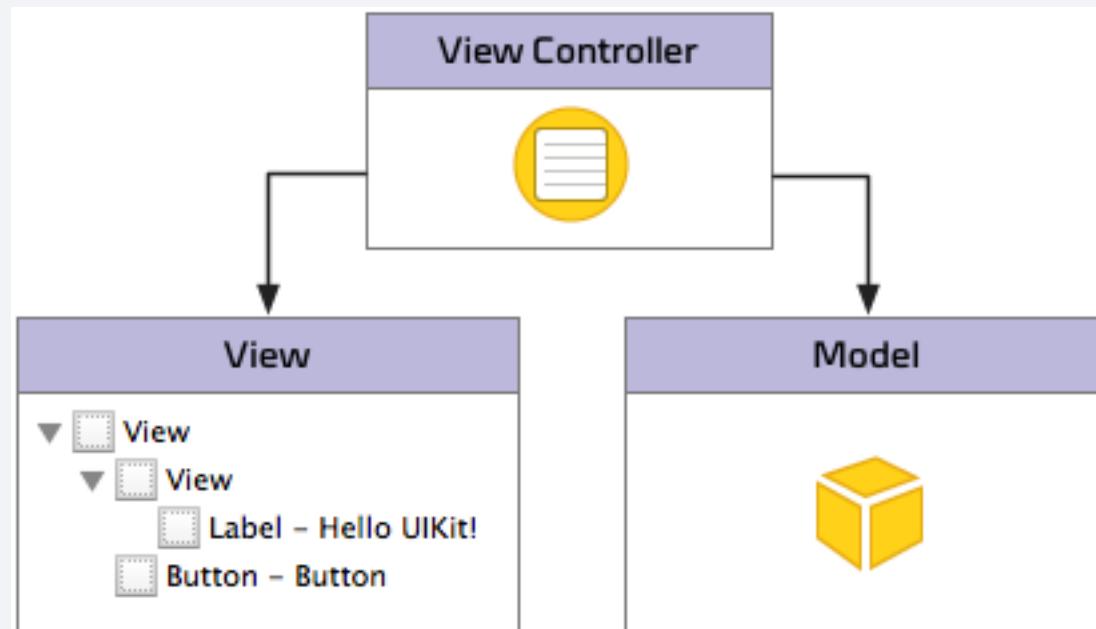


9

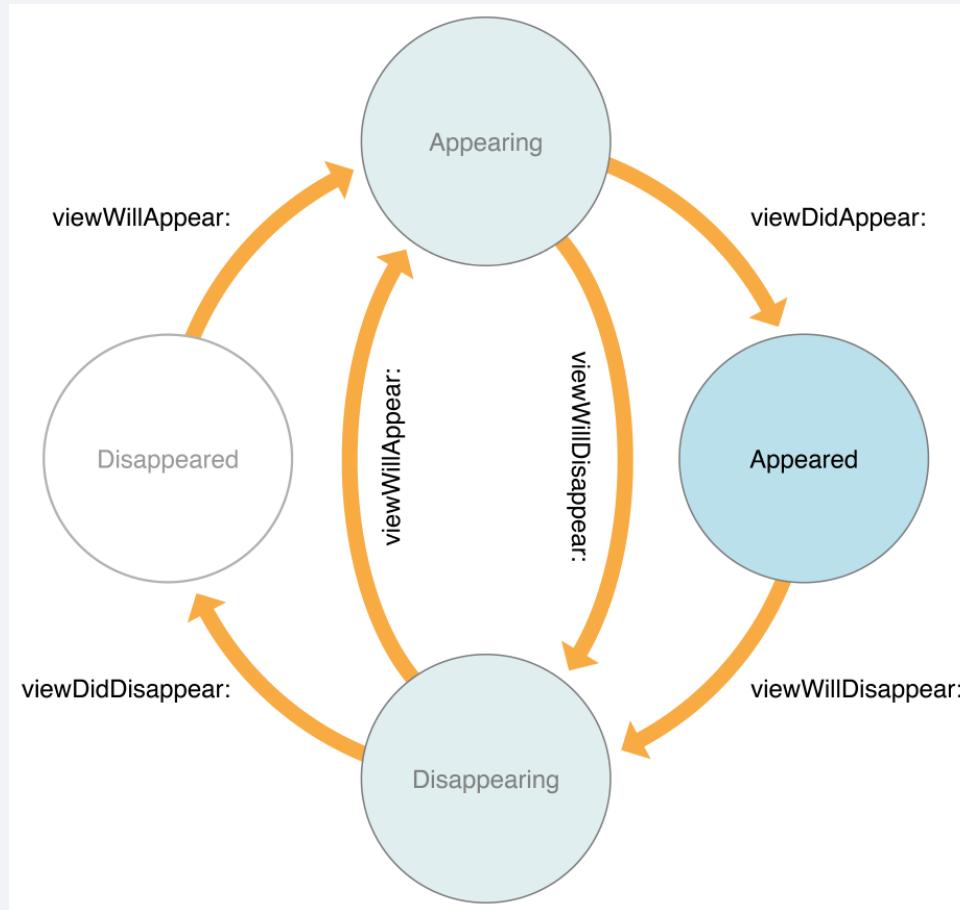
iOS



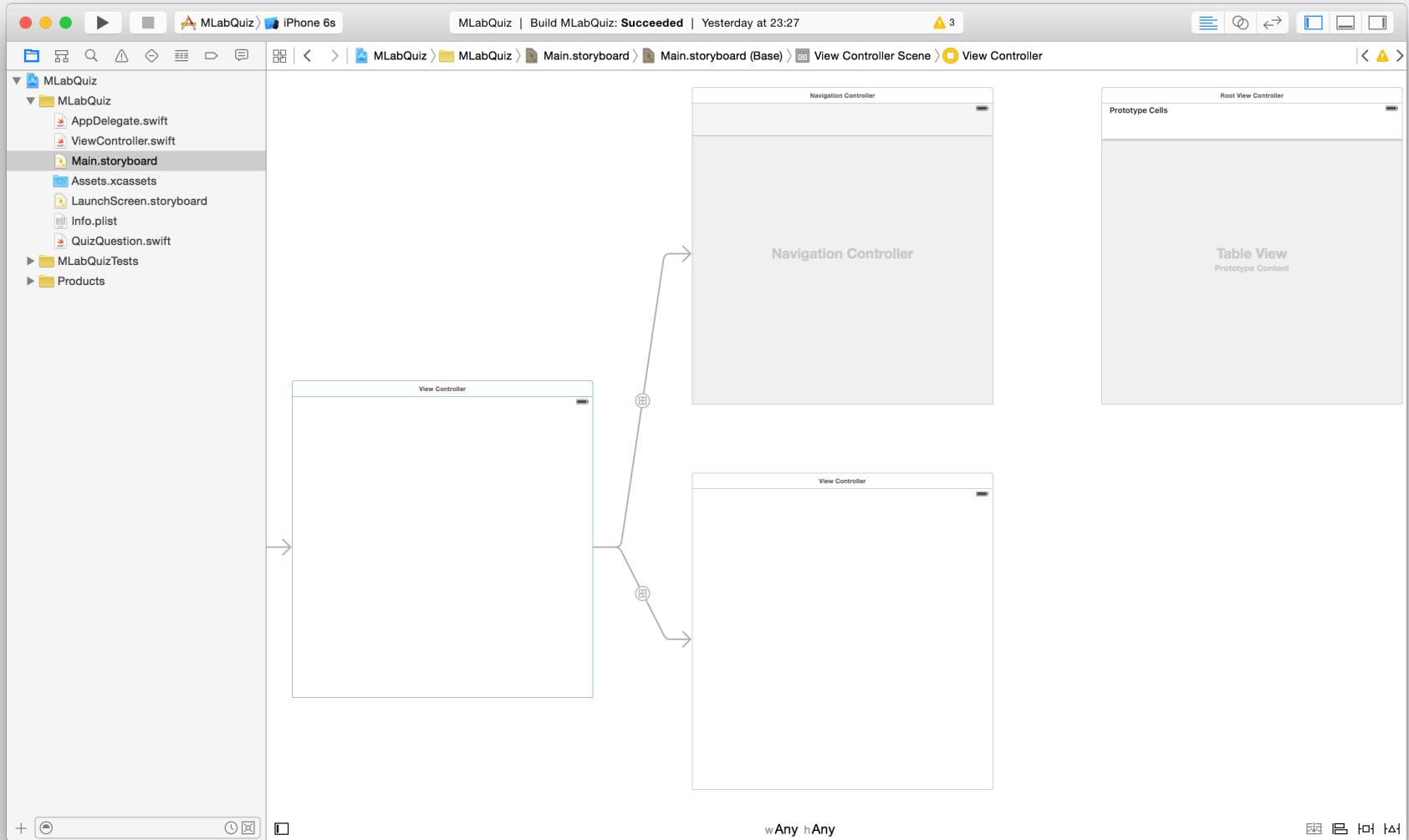
MVC-Pattern



View-Lifecycle



Storyboard



Auto Layout



The screenshot shows the Xcode interface with the project 'MLabQuiz' open. The storyboard file 'Main.storyboard' is selected. A 'Label' view is centered in the main canvas. In the bottom right corner of the storyboard, there is a small circular icon with a 'wAny hAny' label, indicating the current size class. A large grey arrow points from the top left towards the storyboard area.

The right side of the screen displays the 'Attributes Inspector' for the selected 'Label' view. The 'View' section shows the current position (X: 20, Y: 40) and size (Width: 560, Height: 21). The 'Constraints' section is open, showing an 'Add New Constraints' dialog. The dialog has '20' selected in the top dropdown. Below it, there are three dropdowns for 'Horizontal Space' (20), 'Vertical Space' (539), and 'Relative To' (560). The 'Vertical Space' dropdown is highlighted with a blue border. The 'Content Hugging Priority' and 'Content Compression Resistance Priority' sections are visible on the right, with various priority levels listed.

Auto Layout



The screenshot shows the Xcode interface with the project 'MLabQuiz' open. The storyboard file 'Main.storyboard' is selected. A 'Label' view is centered in the main canvas. In the bottom right corner of the storyboard, there is a floating 'Add New Constraints' popover. This popover has 'Height' checked and '21' entered in the height field. It also shows '560' for width and '539' for spacing to nearest neighbor. Below these fields are checkboxes for 'Constrain to margins', 'Width', 'Equal Widths', 'Equal Heights', 'Aspect Ratio', and 'Align'. At the bottom of the popover are buttons for 'Update Frames' (set to 'None'), 'Add 4 Constraints', and a close button.

On the right side of the Xcode window, the 'View' section of the Utilities panel is visible. It shows the 'Label' selected, with 'Preferred Width' set to 'Automatic' and 'Show Alignment Rectangle' checked. The alignment rectangle shows values: X=20, Y=40, Width=560, Height=21. Below this are sections for 'Arrange', 'Layout Margins', and 'Constraints'. The 'Constraints' section notes that no constraints are present and that constraints will be generated at build time if margins are preserved. The 'Content Hugging Priority' section is also shown.

The bottom status bar of the Xcode window displays the device orientation as 'wAny hAny'.

Auto Layout



The screenshot shows the Xcode interface with the project 'MLabQuiz' open. The storyboard file 'Main.storyboard' is selected. A 'Label' view is placed in the center of the main view. The Xcode interface includes:

- Left Navigator:** Shows the project structure with 'MLabQuiz' at the root, containing 'MLabQuiz' (with files 'AppDelegate.swift', 'ViewController.swift', and 'Main.storyboard'), 'Assets.xcassets', 'LaunchScreen.storyboard', 'Info.plist', and 'QuizQuestion.swift'. It also lists 'MLabQuizTests' and 'Products'.
- Top Bar:** Displays the project name 'MLabQuiz', build status 'Build MLabQuiz: Succeeded', and the date 'Yesterday at 23:27'. It also shows a warning count of '3'.
- Right Panel (Attributes Inspector):** Shows settings for the selected 'Label' view, including 'Preferred Width' set to 'Automatic' and 'View' settings for alignment (X: 20, Y: 40), width (560), and height (21).
- Bottom Bar:** Shows the size class 'wAny hAny' and various interface control buttons.
- Bottom Right Legend:** Describes three storyboard components:
 - View Controller** - A controller that manages a view.
 - Storyboard Reference** - Provides a placeholder for a view controller in an external storyboard.
 - Navigation Controller** - A controller that manages navigation through a hierarchy of views.



GIT

GIT



- Distributed version control system
- Main commands
 - git add
 - git checkout
 - git clone
 - git commit
 - git init
 - git pull
 - git push
 - git status

Clone

A screenshot of the GitHub desktop application interface. The window title is "GitHub". The top menu bar includes "File", "Edit", "Clone", "Create", and "Add". The status bar shows "No Uncommitted Changes" and "History". A sidebar on the left has buttons for "Add", "Create", and "Clone", with "Clone" being the active tab. The main content area features a lightbulb icon and the text: "Hey! Looks like you're not logged in to GitHub or GitHub Enterprise. You can [log in](#) now to clone repositories." To the right, there is a section titled "Add a Repository" with the sub-instruction "by creating or opening one".

Hey! Looks like you're not logged in to GitHub or GitHub Enterprise. You can [log in](#) now to clone repositories.

Add a Repository
by creating or opening one

Clone



GitHub

No Uncommitted Changes History

Filter Repositories

Clone As: Tags:
Where:

Add a Repository
by creating or opening one

A screenshot of a GitHub application window showing a cloning dialog. The dialog has fields for 'Clone As' (set to 'mlab-quiz'), 'Tags' (empty), and 'Where' (set to 'Develop'). It includes 'Cancel' and 'Clone' buttons. Below the dialog, there's a section titled 'Add a Repository' with the sub-instruction 'by creating or opening one'.

Clone



A screenshot of the GitHub desktop application interface. The window title is "dplusm/mlab-quiz". The top navigation bar includes "master" (branch dropdown), "No Uncommitted Changes", "History", and "Pull Request". On the left, there's a sidebar with "Filter Repositories" and a list of repositories under "GitHub", with "mlab-quiz" selected. The main content area shows a dark header with "Compare" and "master". Below it, a large section displays "0 Changes" with a checkmark icon. To the right, a message says "No Changes" and "ProTip! Select a commit above to view its changes". At the bottom, there's a summary section with a user icon, a "Summary" button, a "Description" input field, and a "Commit to master" button.

Clone

Command-Line



```
$ git clone [Repository-URL]
```

Add



GitHub master 14 Uncommitted Changes History Pull Request

Filter Repositories Compare Publish

mlab-quiz 14 Changes MLabQuiz.xcodeproj/project.pbxproj

MLabQuiz.xcodeproj/project.pbxproj

MLabQuiz.xcodeproj/xcuserdata

MLabQuiz.xcodeproj/xcscheme

MLabQuiz.xcodeproj/nagement.plist

MLabQuiz/AppDelegate.swift

MLabQuiz/Assets/Contents.json

MLabQuiz/Base.lproj storyboard

MLabQuiz/Base.latin storyboard

Summary Description Commit to master

```
@@ -0,0 +1,404 @@
+ // !$*UTF8*$!
+ {
+     archiveVersion = 1;
+     classes = {
+     };
+     objectVersion = 46;
+     objects = {
+
+         /* Begin PBXBuildFile section */
+         07B3F0CE1BAD9F0500A67252 /* AppDelegate.swift in Sources */ = {isa = PBXBuildFile; fileRef =
+ 07B3F0CD1BAD9F0500A67252 /* AppDelegate.swift */; };
+         07B3F0D01BAD9F0500A67252 /* ViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef =
+ 07B3F0CF1BAD9F0500A67252 /* ViewController.swift */; };
+         07B3F0D31BAD9F0500A67252 /* Main.storyboard in Resources */ = {isa = PBXBuildFile; fileRef =
+ 07B3F0D11BAD9F0500A67252 /* Main.storyboard */; };
+         07B3F0D51BAD9F0500A67252 /* Assets.xcassets in Resources */ = {isa = PBXBuildFile; fileRef =
+ 07B3F0D41BAD9F0500A67252 /* Assets.xcassets */; };
+         07B3F0D81BAD9F0500A67252 /* LaunchScreen.storyboard in Resources */ = {isa = PBXBuildFile; fileRef =
+ 07B3F0D61BAD9F0500A67252 /* LaunchScreen.storyboard */; };
```

Add

Command-Line



```
$ git add [FileName/Directory]
```

Commit



GitHub master 14 Uncommitted Changes History Pull Request Publish

Filter Repositories GitHub mlab-quiz 14 Changes MLabQuiz.xcodeproj/project.pbxproj

MLabQuiz.xcodeproj/project.pbxproj
MLabQuiz.xcodeproj/xcuserdata
MLabQuiz.xcodeproj/xcuserstate
MLabQuiz.xcodeproj/xcscheme
MLabQuiz.xcodeproj/nagement.plist
MLabQuiz/AppDelegate.swift
MLabQuiz/Assets.xcassets/Contents.json
MLabQuiz/Base.lproj/en storyboard
MLabQuiz/Base.lproj/zh storyboard

Adds initial Xcode project

Description

Commit to master

```
@@ -0,0 +1,404 @@
+ // !$*UTF8*$!
+ {
+     archiveVersion = 1;
+     classes = {
+     };
+     objectVersion = 46;
+     objects = {
+
+         /* Begin PBXBuildFile section */
+         07B3F0CE1BAD9F0500A67252 /* AppDelegate.swift in Sources */ = {isa = PBXBuildFile; fileRef =
+ 07B3F0CD1BAD9F0500A67252 /* AppDelegate.swift */; };
+         07B3F0D01BAD9F0500A67252 /* ViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef =
+ 07B3F0CF1BAD9F0500A67252 /* ViewController.swift */; };
+         07B3F0D31BAD9F0500A67252 /* Main.storyboard in Resources */ = {isa = PBXBuildFile; fileRef =
+ 07B3F0D11BAD9F0500A67252 /* Main.storyboard */; };
+         07B3F0D51BAD9F0500A67252 /* Assets.xcassets in Resources */ = {isa = PBXBuildFile; fileRef =
+ 07B3F0D41BAD9F0500A67252 /* Assets.xcassets */; };
+         07B3F0D81BAD9F0500A67252 /* LaunchScreen.storyboard in Resources */ = {isa = PBXBuildFile; fileRef =
+ 07B3F0D61BAD9F0500A67252 /* LaunchScreen.storyboard */; };
```

Commit



A screenshot of the GitHub desktop application interface. The window title is "dplusm/mlab-quiz". The repository name "mlab-quiz" is selected in the sidebar. The main view shows a commit history for the "master" branch. The first commit is highlighted with a blue header: "Adds initial Xcode..." by Daniel Martens, committed just now. The commit message is "Adds initial Xcode project". The file being edited is "MLabQuiz.xcodeproj/project.pbxproj". The code editor shows the following content:

```
... ...
@@ -0,0 +1,404 @@
+ // !$*UTF8*$!
+ {
+     archiveVersion = 1;
+     classes = {
+     };
+     objectVersion = 46;
+     objects = {
+
+         /* Begin PBXBuildFile section */
+         07B3F0CE1BAD9F0500A67252 /* AppDelegate.swift in Sources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0CD1BAD9F0500A67252 /* AppDelegate.swift */; };
+         07B3F0D01BAD9F0500A67252 /* ViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0CF1BAD9F0500A67252 /* ViewController.swift */; };
+         07B3F0D31BAD9F0500A67252 /* Main.storyboard in Resources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0D11BAD9F0500A67252 /* Main.storyboard */; };
+         07B3F0D51BAD9F0500A67252 /* Assets.xcassets in Resources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0D41BAD9F0500A67252 /* Assets.xcassets */; };
+     };
+ }
```

Commit

Command-Line



```
$ git commit -m "Commit-Message"
```

Push



GitHub master No Uncommitted Changes History Pull Request Sync

Filter Repositories Compare master

Adds initial Xcode project Just now by Daniel Martens 14

Adds initial Xcode project Daniel Martens c39fbf1 Just now

MLabQuiz.xcodeproj/project.pbxproj

```
@@ -0,0 +1,404 @@
+ // !$*UTF8*$!
+ {
+     archiveVersion = 1;
+     classes = {
+     };
+     objectVersion = 46;
+     objects = {
+
+ /* Begin PBXBuildFile section */
+         07B3F0CE1BAD9F0500A67252 /* AppDelegate.swift in Sources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0CD1BAD9F0500A67252 /* AppDelegate.swift */; };
+         07B3F0D01BAD9F0500A67252 /* ViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0CF1BAD9F0500A67252 /* ViewController.swift */; };
+         07B3F0D31BAD9F0500A67252 /* Main.storyboard in Resources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0D11BAD9F0500A67252 /* Main.storyboard */; };
+         07B3F0D51BAD9F0500A67252 /* Assets.xcassets in Resources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0D51BAD9F0500A67252 /* Assets.xcassets */; };
+     };
+ }
```

Push



GitHub dplusm/mlab-quiz master No Uncommitted Changes History Pull Request Sync

Filter Repositories Compare master

mlab-quiz Adds initial Xcode project Just now by Daniel Martens 14 Adds initial Xcode project Daniel Martens c39fbf1 Just now Adds initial Xcode project Just now by dplusm 14

MLabQuiz.xcodeproj/project.pbxproj

```
@@ -0,0 +1,404 @@
+ // !$*UTF8*$!
+ {
+     archiveVersion = 1;
+     classes = {
+     };
+     objectVersion = 46;
+     objects = {
+
+         /* Begin PBXBuildFile section */
+         07B3F0CE1BAD9F0500A67252 /* AppDelegate.swift in Sources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0CD1BAD9F0500A67252 /* AppDelegate.swift */; };
+         07B3F0D01BAD9F0500A67252 /* ViewController.swift in Sources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0CF1BAD9F0500A67252 /* ViewController.swift */; };
+         07B3F0D31BAD9F0500A67252 /* Main.storyboard in Resources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0D11BAD9F0500A67252 /* Main.storyboard */; };
+         07B3F0D51BAD9F0500A67252 /* Assets.xcassets in Resources */ = {isa = PBXBuildFile; fileRef =
+             07B3F0D51BAD9F0500A67252 /* Assets.xcassets */; };
+     };
+ }
```

Push

Command-Line



```
$ git push origin master
```



Tutorial Part 1

Task 1



1. Create a new iOS Single View Application
2. Create an additional class QuizQuestion

Create an Xcode Project



Choose a template for your new project:

iOS	Application	Master-Detail Application	Page-Based Application	Single View Application	Tabbed Application
watchOS	Application				
OS X	Framework & Library				
Other	Framework & Library				
	System Plug-in				

Game

Single View Application

This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a storyboard or nib file that contains the view.

No Selection

Cancel Previous Next

No Matches

Create an Xcode Project



Choose options for your new project:

Product Name: MLabQuiz
Organization Name: organization
Organization Identifier: com.organization
Bundle Identifier: com.organization.MLabQuiz
Language: Swift
Devices: iPhone

Use Core Data
 Include Unit Tests
 Include UI Tests

Cancel Previous Next

No Selection

No Matches

Include
unit
tests

Create an Xcode Project



Deselect landscape boxes

MLabQuiz: Ready | Today at 15:48

MLabQuiz

MLabQuiz

MLabQuiz

MLabQuizTests

Products

General Capabilities Resource Tags Info Build Settings Build Phases Build Rules

PROJECT TARGETS

MLabQuiz MLabQuiz Tests

Identity

Bundle Identifier com.organization.MLabQuiz

Version 1.0

Build 1

Team None

Deployment Info

Deployment Target 9.0

Devices iPhone

Main Interface Main

Device Orientation 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

Embedded Binaries

Add embedded binaries here

Identity and Type

Name MLabQuiz

Location Absolute

MLabQuiz.xcodeproj

Full Path /Users/martensd/Develop/dplsm/MLabQuiz/MLabQuiz.xcodeproj

Project Document

Project Format Xcode 3.2-compatible

Organization organization

Class Prefix

Text Settings

Indent Using Spaces

Widths Tab 4 Indent 4

Wrap lines

Source Control

No Matches

Create a Class



MLabQuiz: Ready | Today at 16:12

Choose a template for your new file:

iOS

- Source (selected)
- User Interface
- Core Data
- Apple Watch
- Resource
- Other

watchOS

- Source
- User Interface
- Core Data
- Resource
- Other

OS X

- Source
- User Interface
- Core Data
- Resources

Swift File (highlighted)

UI Test Case Class

Unit Test Case Class

Playground

Objective-C File

Header File

C File

C++ File

Metal File

Swift File
An empty Swift file.

Cancel Previous Next

Identity and Type

Name: MLabQuiz

Location: Relative to Group

MLabQuiz

Full Path: /Users/martensd/Develop/dplsm/mlab/MLabQuiz/MLabQuiz

Text Settings

Indent Using: Spaces

Widths: 4 Tab Indent

Wrap lines

No Matches



Create a Class

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure for "MLabQuiz".
- Editor:** Displays a Swift file snippet for creating a class named "QuizQuestion".
- Save As:** Dialog box showing "Save As: QuizQuestion".
- File Browser:** Shows the current folder is "MLabQuiz".
- Identity and Type:** Inspector panel showing the project name is "MLabQuiz".
- Text Settings:** Inspector panel showing indentation settings.
- Targets:** A modal dialog titled "Select test target" is open, showing the "Targets" section. It lists "MLabQuiz" and "MLabQuizTests". The checkbox for "MLabQuizTests" is checked and highlighted with a red box.
- Search:** Shows "No Matches".

Task 2



1. Add an answer and question property to QuizQuestion
2. Create an initializer for QuizQuestion which accepts both question and answer as arguments

Solution 2



```
01 import Foundation
02
03 class QuizQuestion {
04
05     // MARK: Properties
06     var question: String
07     var answer: String
08
09
10    // MARK: Quiz Question
11    init(question: String, answer: String) {
12        self.question = question
13        self.answer = answer
14    }
15 }
```

Task 3



Unit Tests

1. Create a testcase for the constructor
2. Create a testcase for the question and answer getter

Solution 3



```
01 func testQuestion() {  
02     let q: QuizQuestion = QuizQuestion(question: "a", answer: "b")  
03  
04     XCTAssertEqual("a", q.question)  
05     XCTAssertEqual("b", q.answer)  
06 }
```

Task 4



Unit Tests

1. Create an mutable string-typed array
2. Put a couple of strings to in the array
3. Use the third string from the array to pass it to the QuizQuestion initializer

Hint: ⌘ + MouseClick toggles documentation

Solution 4



```
01 func testQuestion() {  
02     let a: [String] = ["a", "b", "c", "d"]  
03  
04     let q: QuizQuestion = QuizQuestion(question: "a", answer: a[2])  
05  
06     XCTAssertEqual("a", q.question)  
07     XCTAssertEqual(a[2], q.answer)  
08 }
```



Tutorial Part 2

Create a UI



The screenshot shows the Xcode interface with a storyboard for an iPhone 6s. A single 'Label' view is selected in the center of the storyboard canvas. The left sidebar displays the project structure, including files like AppDelegate.swift, ViewController.swift, and Main.storyboard. The right sidebar contains a list of UI components with their descriptions:

- Label** - Preferred Width: Automatic / Explicit
- View** - Show Alignment Rectangle: X: 20, Y: 40, Width: 560, Height: 21
- Label** - Label - A variably sized amount of static text.
- Button** - Intercepts touch events and sends an action message to a target object when it's tapped.
- Segmented Control** - Displays multiple segments, each of which functions as a discrete button.
- Text Field** - Displays editable text and sends an action message to a target object when Return is tapped.
- Slider** - Displays a continuous range of values and allows the selection of a single value.
- Switch** - Displays an element showing the boolean state of a value. Allows tapping the control to toggle..
- Activity Indicator View** - Provides feedback on the progress of a task or process of unknown duration.
- Progress View** - Depicts the progress of a task over time.
- Page Control** - Displays a dot for each open page in an application and

Create a UI



The screenshot shows the Xcode interface for creating a user interface. On the left, the Project Navigator displays the project structure for 'MLabQuiz' with 'Main.storyboard' selected. The main canvas shows a storyboard scene with a navigation bar at the top and a single 'Label' view in the center. A constraint preview is visible, indicating a horizontal space of 40 points between the label and its superview. The Utilities panel on the right is open to the 'View' section, showing alignment settings for the label: X: 20, Y: 101, Width: 560, Height: 30. Below this, the 'Label' view is described as 'A variably sized amount of static text'. Other view types like 'Button' and 'Segmented Control' are listed in the sidebar.

MLabQuiz | Build MLabQuiz: Succeeded | Yesterday at 23:27

MLabQuiz

Main.storyboard

Label

Add New Constraints

40

0 0 469

Constrain to margins

Width: 560

Height: 30

Equal Widths

Equal Heights

Aspect Ratio

Align: Leading Edges

Update Frames: None

Add Constraints

wAny hAny

View

Show Alignment Rectangle

X: 20 Y: 101

Width: 560 Height: 30

Arrange Position View

Layout Margins Default

Label - A variably sized amount of static text.

Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control - Displays multiple segments, each of which functions as a discrete button.

TextField - Displays editable text and sends an action message to a target object when Return is tapped.

Range - Displays a continuous range of values and allows the selection of a value.

Switch - Displays an element showing the boolean state of a value. Tapping the control to toggle...

Progress Indicator View - Provides a visual cue on the progress of a task or a series of unknown duration.

Progress View - Depicts the progress of a task over time.

Control - Displays a dot for a given page in an application and supports sequential navigation through

Create a UI



The screenshot shows the Xcode interface with the project 'MLabQuiz' open. The 'Main.storyboard' file is selected in the left sidebar. In the storyboard preview, a single 'Label' is visible. A constraint editor panel on the right is showing a 'Vertical Space Constraint' between the 'Label' and a 'Round Style Text Field'. The constraint details are: First Item: Round Style Text Field.Top, Relation: Equal, Second Item: Label.Bottom, Constant: 40, Priority: 1000, Multiplier: 1. Below this, there is a list of UI element types with their descriptions:

- Label** - A variably sized amount of Label static text.
- Button** - Intercepts touch events and sends an action message to a target object when it's tapped.
- Segmented Control** - Displays multiple segments, each of which functions as a discrete button.
- Text Field** - Displays editable text and sends an action message to a target object when Return is tapped.
- Slider** - Displays a continuous range of values and allows the selection of a single value.
- Switch** - Displays an element showing the boolean state of a value. Allows tapping the control to toggle...
- Activity Indicator View** - Provides feedback on the progress of a task or process of unknown duration.
- Progress View** - Depicts the progress of a task over time.
- Page Control** - Displays a dot for each open page in an application and supports sequential navigation thro...

Create a UI



The screenshot shows the Xcode interface with the storyboard editor open. The project is named 'MLabQuiz' and has build succeeded. The storyboard file 'Main.storyboard' is selected. A large arrow points from the storyboard editor towards a preview window on the right.

Storyboard Editor:

- Project Navigator: Shows files like AppDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, and QuizQuestion.swift.
- Outline: Shows the View Controller Scene and View Controller.
- Preview: Shows a placeholder for the question text and three buttons labeled "Feedback", "Überprüfe Antwort", and "Zeige richtige Antwort".
- Object Library: Shows a Segmented Control component with two segments labeled "1" and "2". A callout bubble indicates it displays multiple segments, each of which functions as a discrete button.

Preview Window:

- Device: iPhone 6s - iPhone 6s / iOS 9.0 (13A340)
- Carrier: (no carrier shown)
- Time: 10:02 AM
- Content:
 - Text: Frage
 - Text: Feedback
 - Text: Überprüfe Antwort
 - Text: Zeige richtige Antwort

Add/Connect Properties



The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure with files like AppDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, and QuizQuestion.swift.
- Editor:** Displays the ViewController.swift file and the Main.storyboard.
- Main.storyboard:** Shows a single view controller with a label named "Frage". A blue line connects the "Frage" label to the "viewDidLoad" method in the code.
- Code Editor (ViewController.swift):**

```
1 // ViewController.swift
2 // MLabQuiz
3 // Created by Daniel Martens on 22/09/15.
4 //
5 // Copyright © 2015 dplum. All rights reserved.
6 //
7 //
8 import UIKit
9
10 class ViewController: UIViewController {
11     override func viewDidLoad() {
12         super.viewDidLoad()
13         // Do any additional setup after loading the
14         // view, typically from a nib.
15     }
16
17     override func didReceiveMemoryWarning() {
18         super.didReceiveMemoryWarning()
19         // Dispose of any resources that can be
20         // recreated.
21     }
22
23
24 }
25
26 }
```
- Assistant View:** Shows the "Frage" label selected in the storyboard, with its properties listed in the right panel.
- Properties Panel (Label):**
 - Text: Plain
 - Color: Black
 - Font: System 17.0
 - Alignment: Center
 - Lines: 1
 - Behavior: Enabled
 - Baseline: Align Baselines
 - Line Breaks: Truncate Tail
 - Autoshrink: Fixed Font Size
 - Highlighted: Default
 - Shadow: Default
 - Shadow Offset: 0 (Horizontal), -1 (Vertical)
- Properties Panel (View):**
 - Mode: Left
 - Semantic: Unspecified
 - Tag: 0
 - Interaction: User Interaction Enabled, Multiple Touch
- Callouts:**
 - Label**: Label - A variably sized amount of static text.
 - Button**: Button - Intercepts touch events and sends an action message to a target object when it's tapped.
 - Segmented Control**: Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Add/Connect Properties



Screenshot of Xcode showing the ViewController.swift file and the storyboard.

The ViewController.swift file contains the following code:

```
// ViewController.swift
// MLabQuiz
//
// Created by Daniel Martens on 22/09/15.
// Copyright © 2015 dplum. All rights reserved.

import UIKit

class ViewController: UIViewController {

    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.
    }
}
```

The storyboard shows a single view controller with a label. A connection line is being drawn from the label to the code, indicating the creation of an outlet named "questionLabel".

The right-hand panel displays the storyboard object inspector and a callout for the "questionLabel" outlet, showing its properties:

- Label**:
 - Text: Plain
 - Frame: [x: 100, y: 100, width: 200, height: 50]
 - Color: Default
 - Font: System 17.0
- Alignment: Center
- Lines: 1
- Behavior: Enabled
- Baseline: Align Baselines
- Line Breaks: Truncate Tail
- Autoshrink: Fixed Font Size
- Highlighted: Default
- Shadow: Default
- Shadow Offset: 0 (Horizontal), -1 (Vertical)

- View**:
- Mode: Left
- Semantic: Unspecified
- Tag: 0
- Interaction**:
- User Interaction Enabled
- Multiple Touch
- Label** - A variably sized amount of static text.
- Button** - Intercepts touch events and sends an action message to a target object when it's tapped.
- Segmented Control** - Displays multiple segments, each of which functions as a discrete button.

Add/Connect Properties



iPhone 6s Finished running MLabQuiz on iPhone 6s 3

MLabQuiz

Main.storyboard

```
// ViewController.swift
// MLabQuiz
//
// Created by Daniel Martens on 22/09/15.
// Copyright © 2015 dplum. All rights reserved.

import UIKit

class ViewController: UIViewController {

    // MARK: Properties

    @IBOutlet weak var questionLabel: UILabel!
    @IBOutlet weak var feedbackLabel: UILabel!
    @IBOutlet weak var answerTextField: UITextField!
    @IBOutlet weak var validateButton: UIButton!
    @IBOutlet weak var showAnswerButton: UIButton!

    // MARK: View Life Cycle

    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.
    }
}
```

Frage

Feedback

Überprüfe Antwort

Zeige richtige Antwort

Button

Type: System

State Config: Default

Title: Plain

Font: System 15.0

Text Color: Default

Shadow Color: Default

Image: Default Image

Background: Default Background Image

Shadow Offset: 0 0

Width: Height

Reverses On Highlight

Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break: Truncate Middle

Edge: Content

Inset: 0 0

Left: Top

Bottom: Right

Control

Alignment:

Label - A variably sized amount of static text.

Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Add/Connect Actions



The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure with files like AppDelegate.swift, ViewController.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, and QuizQuestion.swift.
- Document Outline:** Shows the storyboard scene hierarchy: Frage, Feedback, Überprüfe Antwort, and Zeige rich.
- Code Editor:** Displays the ViewController.swift file with the following code:

```
// ViewController.swift
// MLabQuiz
//
// Created by Daniel Martens on 22/09/15.
// Copyright © 2015 dplum. All rights reserved.

import UIKit

class ViewController: UIViewController {

    // MARK: Properties

    @IBOutlet weak var questionLabel: UILabel!
    @IBOutlet weak var feedbackLabel: UILabel!
    @IBOutlet weak var answerTextField: UITextField!
    @IBOutlet weak var validateButton: UIButton!
    @IBOutlet weak var showAnswerButton: UIButton!

    // MARK: View Life Cycle

    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.
    }
}
```

- Assistant Editor:** Shows the Validate Button selected in the storyboard, with its action configuration dialog open. The "Action" tab is selected, showing the connection to the ViewController object and the validateButtonPressed method.
- Attributes Inspector:** Shows the properties for the Validate Button, including Type (System), State Config (Default), Title (Plain), and Font (System 15.0). It also includes sections for "Überprüfe Antwort" and "Feedback".
- Utilities Inspector:** Shows the Control, Label, and Segmented Control sections.

Add/Connect Actions



iPhone 6s Finished running MLabQuiz on iPhone 6s 3

MLabQuiz

Main.storyboard

Frage

Feedback

Überprüfe Antwort

Zeige richtige Antwort

```
1 // ViewController.swift
2 // MLabQuiz
3 //
4 //
5 // Created by Daniel Martens on 22/09/15.
6 // Copyright © 2015 dplum. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     // MARK: Properties
14
15     @IBOutlet weak var questionLabel: UILabel!
16     @IBOutlet weak var feedbackLabel: UILabel!
17     @IBOutlet weak var answerTextField: UITextField!
18     @IBOutlet weak var validateButton: UIButton!
19     @IBOutlet weak var showAnswerButton: UIButton!
20
21
22     // MARK: View Life Cycle
23
24     override func viewDidLoad() {
25         super.viewDidLoad()
26         // Do any additional setup after loading the
27         // view, typically from a nib.
28     }
29
30     override func didReceiveMemoryWarning() {
31         super.didReceiveMemoryWarning()
32         // Dispose of any resources that can be
33         // recreated.
34     }
35
36     // MARK: Actions
37
38     @IBAction func validateButtonPressed(sender: AnyObject) {
39
40     }
41
42     @IBAction func showAnswerButtonPressed(sender: AnyObject) {
43
44     }
45 }
```

Identity and Type

Name: ViewController.swift

Type: Default - Swift Source

Location: Relative to Group

ViewController.swift

Full Path: /Users/martensd/Develop/dplum/mlab/MLabQuiz/MLabQuiz/ViewController.swift

On Demand Resource Tags

Only resources are taggable

Target Membership

MLabQuiz (checked)

MLabQuizTests

Text Settings

Text Encoding: Default - Unicode (UTF-8)

Line Endings: Default - OS X / Unix (LF)

Indent Using: Spaces

Widths: 4 Tab 4 Indent

Wrap lines (checked)

Source Control

Repository --

Label

Label - A variably sized amount of static text.

Button

Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control

Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Task 5



1. Run your application in the simulator
2. Tell us your observations

Hide the keyboard



```
01 class ViewController: UIViewController, UITextFieldDelegate {  
  
01 override func viewDidLoad() {  
    super.viewDidLoad()  
  
03  
04     self.answerTextField.delegate = self  
05 }  
  
01 func textFieldShouldReturn(textField: UITextField) -> Bool {  
02     textField.resignFirstResponder()  
03     return true;  
04 }
```

Show a Question on App-Start



```
01 @IBOutlet weak var questionLabel: UILabel!
02 var question: QuizQuestion?

01 override func viewDidLoad() {
02     super.viewDidLoad()
03
04     self.answerTextField.delegate = self
05
06     self.question = QuizQuestion(question: "Question", answer: "Answer")
07     if let questionString = self.question?.question {
08         questionLabel.text = questionString
09     }
10 }
```



Validate Answer

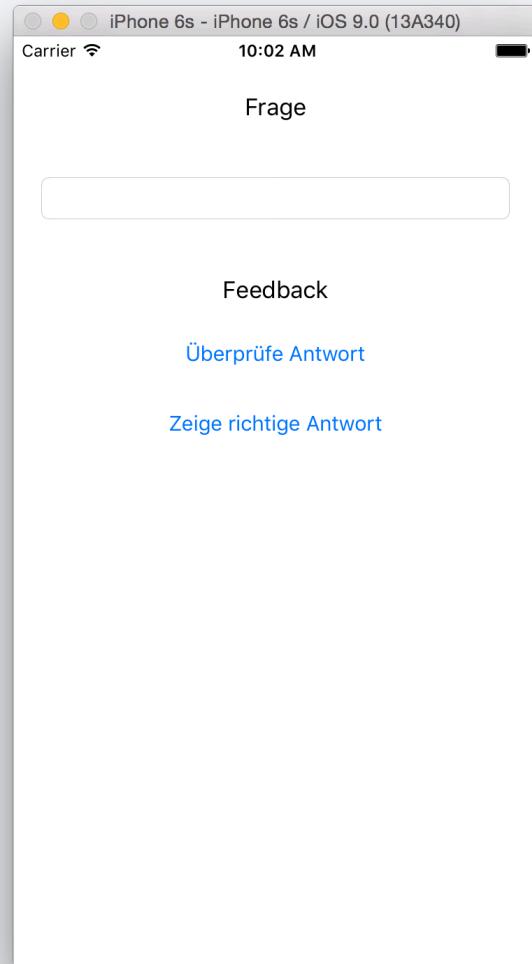
Compare two string using "==" operator

```
01 let firstString: String = "a"  
02 let secondString: String = "a"  
03 var isEqual: Bool = (firstString == secondString)
```

Task 5



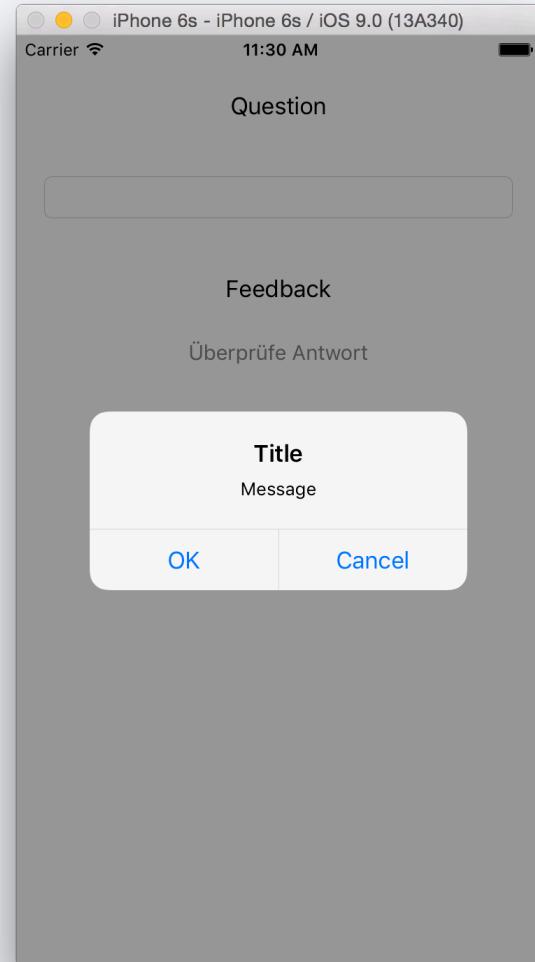
1. Implement the applications functionality



Task 6



1. Do not show the answer immediately on "Show Answer" button



Solution 6



```
01 let alert = UIAlertController(title: "Title", message:"Message",
02                             preferredStyle: .Alert)
03
04 let okAction = UIAlertAction(title: "OK", style: .Default) { _ in
05     // OK button action
06 }
07 let cancelAction = UIAlertAction(title: "Cancel", style: .Default) { _ in
08     // Cancel button action
09 }
10
11 alert.addAction(okAction)
12 alert.addAction(cancelAction)
13 self.presentViewController(alert, animated: true) {}
```

Task 7



1. Additional task: Use multiple questions



Literature

Literature

- **The Swift Programming Language**
<https://itunes.apple.com/de/book/swift-programming-language/id881256329?mt=11>
- **The Swift Programming Language (Swift 2)**
https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/GuidedTour.html
- **WWDC 2015 Videos**
<https://developer.apple.com/videos/wwdc/2015/>
- **ProGit Book**
<https://progit.org>

Who we are



Daniel Martens
M.Sc. Inf., 4th Semester



Jan Hennings
B.Sc. MCI, 5th Semester