Reading Assignment I: Grundlagen von Swift

Ziele

Das Ziel des ersten Reading Assignments ist es, die Grundlagen der neuen Sprace Swift zu erlernen. Wir decken grundlegende Dinge wie Variablen und Control Flow, aber auch etwas komplexere Dinge wie Closures (Funktionen als Typen) ab.

Die meisten von Ihnen haben wahrscheinlich keine Erfahrung in Objective-C, machen Sie sich darüber aber keine Gedanken, da nichts in der Swift Dokumentation dies voraussetzt.

Lesen alles referenzierte Material bis zur Woche 3.

Material

- Die Reading Assignments stammen aus zwei Online Dokumenten: The Swift Progamming Language und den Swift API Guidelines.

Swift Programming Language

Lesen Sie die unten beschriebenen Sektionen in Swift Progamming Language. Um Ihre wertvolle Zeit besser zu nutzen und auf die wichtgen Konzepte zu konzentrieren, wurden der Lesestoff in vier Kategorien unterteilt.

Rote Sektionen sind SEHR WICHTIG und sind wahrscheinlich schwieriger zu verstehen. Lesen Sie diese sehr aufmerksam.

Gelbe Sektionen sind wichtig, sind aber nicht schwierig zu verstehen.

Grüne Sektionen sind wichtig, decken aber einfache Grundlagen ab (das meiste ist wie C).

Ausgeraute Sektionen müssen Sie für dieses Assignemt (noch) nicht lesen. Sie sind vielleicht später im Semester wichtig.

Überfliegen Sie nicht den NOTE Text (in den grauen Kästen) - viele dieser Dinge sind sehr wichtig. Wenn sich eine NOTE jedoch auf Objective-C oder Bridging bezieht, können Sie sie überspringen.

Wenn ein Link im Text auf eine andere Sektion verweist, müssen Sie diesem nicht folgen, außer der verlinkte Teil ist ebenfalls Teil des Reading Assignments.

Im Language Guide Bereich, lesen Sie die Sektionen in den folgenden Kapiteln:

The Basics

Constants and Variables
Comments
Semicolons
Integers
Floating-Point Numbers
Type Safety and Type Inference
Numeric Literals
Numeric Type Conversion
Type Aliases
Booleans
Tuples
Optionals
Error Handling
Assertions

Unicode Variablen- und Konstantennamen können Spaß machen, Sie sind jedoch verantwortlich für die Qualität Ihrer Benamung (jeder Art) und Lesbarkeit Ihres Codes. Setzen Sie keine Semikolons ans Ende von Zeilen (verwenden Sie diese nur (sehr selten), um zwei Statements in einer Zeile zu trennen).

Basic Operators

Das meiste in dieser Sektion ist identisch zu C (daher sind die meisten Sektionen in grün).

Terminology
Assignment Operator (machen Sie sich keine Gedanken zu Referenzen auf Tuples)
Arithmetic Operators
Compound Assignment Operators
Comparison Operators
Ternary Conditional operator
Nil Coalescing Operator
Range Operators
Logical Operators

Strings and Characters

String Literals
Initializing an Empty String
String Mutability
Strings Are Value Types
Working with Characters
Concatenating Strings and Characters
String Interpolation
Unicode
Counting Characters
Accessing and Modifying a String
Comparing Strings
Unicode Representations of Strings

Collection Types

Mutability of Collections
Arrays
Sets (ignorieren Sie das erste NOTE Kästchen in dieser Sektion)
Performing Set Operations
Dictionaries

Wenn Sie versuchen auf einen Index in einem Array zuzugreifen, der höher als dessen `count` ist, stürzt Ihr Programm ab.

Es existiert auch eine Funktion `last` in `Array`, die ein `Optional` zurück gibt (d.h. sie gibt `nil` zurück wenn und nur wenn das Array `isEmpty`).

Beachten Sie, dass der `+=` Array Operator ein weiteres Array rechtsseitig annimmt (nicht ein Element, welches dem Array hinzugefügt wird.

Control Flow

For Loops (`for-in` ist vielleicht für einige von Ihnen neu, genauso wie `ranges`, z.B. `1..5`)
While Loops
Conditional Statements (speziell `switch`, ignorieren Sie Tuples, Value Bindings & Where)
Control Transfer Statements (ignorieren Sie aber Labeled Statements)
Early Exit
Checking API Availability

Das `switch` Statement ist viel wichtiger in Swift als Sie dies von C oder anderen Sprachen gewohnt sind.

Functions

Defining and Calling Functions
Function Paramenters and Return Values (ignorieren Sie Functions with Multiple Return Values und Optional Tuple Return Types)
Function Argument Labels and Parameter Names (ignorieren Sie Variadic und In-Out Parameters)
Function Types (dies ist wahrscheinlich die herausfordernste Sektion für die meisten von Ihnen)
Nested Functions

Wenn Sie dieses Assignment nach der ersten Vorlesung begonnen haben, möchten Sie an dieser Stelle vielleicht pausieren bis nach dem Ende der zeiten Vorlesung (in der wir Closures, Enums und die Unterschiede zwischen classes und structs besprechen).

Closures

Zu verstehen, dass die Description (Signatur) einer Funktion (d.h. deren Parameter und Rückgabewert) ein first-class Typ (genau wie ein `Array` oder ein `Int`) in Swift sein kann, ist sehr wichtig. Viele iOS APIs haben Closures als Parameter.

Closure Expressions (stellen Sie sicher, dass Sie Function Types von oben zuerst verstehen)
Trailing Closures
Capturing Values
Closures are Reference Types
Nonescaping Closures
Autoclosures

Enumerations

Enumeration Syntax
Matching Enumeration Values with a Swift Statement
Associated Values
Raw Values
Recursive Enumerations

Classes and Structures

Comparing Classes and Structures
Structures and Enumerations are Value Types
Classes are Reference Types
Choosing Between Classes and Structures
Assignment and Copy Behavior for Strings, Arrays and Dictionaries

Beachten Sie, dass wir in dieser Woche nur zwei Arten kennen um eine class oder struct zu erstellen: durch schreiben des Namens der `class` oder `struct`s gefolgt von leeren Klammern, z.B. `let x = VideoMode()` oder, nur für `struct`s, der Nutzung des Namens des `struct`s mit all seinen Variablen die einen Wert bekommen, z.B. `let hd = Resolution(width: 1920, height: 1080)`. Wir lernen noch mächtigere Wege kennen `class`es und `struct`s zu initialisieren.

Properties

Stored Properties (ignorieren Sie Lazy Stored Properties und Stored Properties and Instance Variables)
Computed Properties
Property Observers
Global and Local Variables
Type Properties

Swift API Guidelines

Lesen Sie das Swift API Guidelines Dokument vollständig.

Da Sie wahrscheinlich noch keine Erfahrungen in Swift haben, sind einige Dinge in diesem Dokument anfänglich wahrscheinlich schwer zu verstehen. Sich aber mit den Inhalten des Dokumentes vertraut zu machen ist ausschlaggebend guten Code in Swift zu schreiben. In diesem Assignment ist das Ziel zu sehen was alles existiert, statt die kompletten Guidelines von Beginn an vollständig zu verstehen. Während das Semester fortschreitet, werden Sie wahrscheinlich zu einem Experten für Benamung von Properties, Methoden oder andere Swift Konstrukte. Dies verlangt von Ihnen oft in diesem Dokument nachzuschlagen.

Stellen Sie sicher überall drauf zu klicken wo “MORE DETAIL” steht.

Richten Sie Ihre Aufmerksamkeit auf die “Write a documentation comment” Sektion.

Richten Sie Ihre Aufmerksamkeit auf die “Follow case conventions” Sektion.

Richten Sie Ihre Aufmerksamkeit auf die gesamte “Argument Labels” Sektion.

Sie können (für jetzt) Punkte ignorieren, die sich auf Protocols beziehen. Wenn wir in den nächsten Wochen etwas über Protocols gelernt haben, stellen Sie sicher, dass Sie dies dann noch mal im Dokument nachlesen.

Da Sie noch keine Initializer in Swift gesehen haben, macht die Sektion “Initializer and factory method calls” noch keine Sinn für Sie, erinnern Sie sich aber daran, wenn wir Initializer kennengelernt haben.

Sie können ebenfalls die letzte Sektion (Special Instructions) für jetzt ignorieren.