Reading Assignment I: Intro to Swift

Objective

The goal of our first week of reading assignments is to start to get a handle on this new language you must learn called Swift. This week covers basic stuff like variables and control flow, but also more trickier topics like Optionals, manipulating Strings and the syntax for defining classes and struct and their methods and properties.

Most of you have not had experience with Objective-C, but don't worry about that. Nothing in the Swift documentation really assumes that. However, if you have never programmed in C (or C++ or any other variant), then Swift might be extremely new to you (but hopefully still not too steep a hill to climb to learn).

Read all of the material referenced here by the start of Lecture 3. Set aside sufficient time because there's quite a bit of reading here. You will only have reading assignments in the first few weeks of this course.

Materials

- The reading in this assignment comes from two on-line documents: the Swift Programming Language and the Swift API Guidelines.

Swift Programming Language

Read the sections described below in the Swift Programming Language. To better utilize your valuable time and to emphasize important concepts, the sections in the reading have been annotated with three colors:

Red sections are VERY IMPORTANT and might be more difficult to understand. Read these carefully.

Yellow sections are important, but probably won't be as difficult to understand.

Grayed-out sections are not required reading (this week). They may be in future weeks.

Don't gloss over reading any NOTE text (inside gray boxes)—many of those things are quite important. However, if a NOTE refers to Objective-C or bridging, you can ignore it.

If there is a link to another section in the text, you don't have to follow that link unless what it links to is also part of this week's reading assignment.

Note that a random sampling of the topics in the list below have links. There are not link destinations available for all topics, unfortunately, but for ones that exist, the link is included. This is just a way to help you jump to the "ballpark" of where a topic is. Linked topics are not any more or less important than any other topic.

In the **Language Guide** area, read the following sections in the following chapters:

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 and Preconditions

Basic Operators

- Terminology
- Assignment Operator
- 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
- Substrings
- Comparing Strings
- Unicode Representations of Strings

Collection Types

- Mutability of Collections
- Arrays
- Sets
- Performing Set Operations
- Dictionaries

Control Flow

- For-In Loops
- While Loops
- Conditional Statements
  - If
  - Switch
  - No Implicit Fallthrough
  - Interval Matching
  - Tuples
  - Value Bindings
  - Where
  - Compound Cases
- Control Transfer Statements
  - Continue
  - Break
  - Fallthrough
  - Labeled Statements
- Early Exit
- Checking API Availability

Functions

- Defining and Calling Functions
- Function Parameters and Return Values
  - Functions Without Parameters
  - Functions With Multiple Parameters
  - Functions Without Return Values
  - Functions With Multiple Return Values
- Function Argument Labels and Parameter Names
  - Specifying Argument Labels
  - Omitting Argument Labels
  - Default Parameter Values
  - Variadic Parameters
  - In-Out Parameters
- Function Types
- Nested Functions

Closures

Enumerations

Classes and Structures

- Comparing Classes and Structures
- Structures and Enumerations Are Value Types (ignore enumerations)
- Classes Are Reference Types
- Choosing Between Classes and Structures
- Assignment and Copy Behavior for Strings, Arrays and Dictionaries

Properties

- Stored Properties
- Computed Properties
- Property Observers
- Global and Local Variables
- Type Properties

Methods

- Instance Methods
  - The self Property
  - Modifying Value Types from Within Instance Methods
  - Assigning to self Within a Mutating Method
- Type Methods

Subscripts

Inheritance

- Defining a Base Class
- Subclassing
- Overriding
  - Accessing Superclass Methods, Properties, and Subscripts
  - Overriding Methods
  - Overriding Properties
- Preventing Overrides

Initialization

Setting Initial Values for Stored Properties
Customizing Initialization
Default Initializers
Class Inheritance and Initialization
Failable Initializers
Required Initializers
Setting a Default Property Value with a Closure or Function

Unicode variable and constant names (e.g., 🔥) can be fun, but you will be held accountable for the quality of your naming (of all kinds) and readability in your code.

Do not put semicolons at the ends of lines (only use them to (very rarely, if ever) separate two statements on a single line).

Swift API Guidelines

Read the Swift API Guidelines document in its entirety.

Given that you are completely new to Swift, some of what is in this document will be a bit hard to fully absorb at first. But familiarizing yourself with what is in this document is crucial to writing good Swift code. So, for this assignment, the goal is to know what's there rather than completely and fully master the guidelines right off the bat. As the quarter progresses, you should eventually become an expert namer of properties, methods and other Swift constructs. This will require you to refer back to this document often.

Be sure to click everywhere that it says "MORE DETAIL".

Pay special attention to the "Write a documentation comment" section.

Pay special attention to the "Follow case conventions" section.

Pay special attention to the entire "Argument Labels" section.

You can ignore (for now), points that reference Protocols. When we learn about Protocols next week, be sure to check back with this document after that.

You can also ignore the final section (Special Instructions) for now.