

Go vs. Swift, The Languages of The Modern Tech Giants

Jake Rockland
jakerockland@gmail.com

December 5, 2016

Abstract

This project stands as a comparative exploration of Go and Swift, the recent flagship languages developed by Google and Apple, respectively. Specifically, I would like to explore first what pushed these modern tech behemoths to develop two new programming languages and second how these two languages are similar and how they differ, given that they were released at relatively similar times.

1 Introduction

The current technology landscape is often driven by two major companies: Apple and Google. With market capitalizations of \$520.93 billion¹ and \$586.02 billion², respectively, Apple (\$AAPL) and Google (\$GOOG) are the two highest valued companies in the world at the time of writing. Along with making top-class products that millions of consumers use on a regular basis, both Apple and Google have continually contributed to an ecosystem of tools built for developers to create new software, whether for their platforms or not. In the past decade, one way this has manifested is in the development of two new programming languages, Swift and Go.

The similar statuses of Apple and Google as technology behemoths of the modern world, as well as the similar timing of the releases of Swift and Go, drives us to explore first what pushed these companies to develop two new programming languages and second how these two languages are similar and how they differ. We will explore this by first looking at the historical context and initial goals of these languages, then look to the outside influences of each language, followed by a comparison of the design decisions made for each, finally we will conclude with a consideration of the future prospects for each language.

2 Historical Context

Development for Swift began in early 2010 by Chris Lattner, the main author of LLVM and the Clang compiler³, who joined Apple in 2005 to integrate these tools into Apple's products at a production quality level⁴. Lattner was later joined by other developers at Apple in late 2011 and by 2013 Swift was a main focus of the Apple Developer Tools team. By the summer of 2014, Swift was ready for public release and was first introduced at WWDC on June 2nd⁵. The next year, on December 3, 2015, Apple announced that they would open Swift up as an open source project, welcoming collaboration from the public⁶.

¹<http://finance.yahoo.com/quote/aapl>

²<http://finance.yahoo.com/quote/goog>

³<http://llvm.org>

⁴<http://nondot.org/sabre/>

⁵<https://developer.apple.com/videos/wwdc2014/>

⁶www.apple.com/pr/library/2015/12/03Apple-Releases-Swift-as-Open-Source.html

The origins of Go date back slightly earlier, to 2007, when Robert Griesemer, Rob Pike and Ken Thompson, all working at Google, started outlining the goals for a new language⁷. By January 2008, Thompson, who had previously designed much of the initial UNIX operating system and the B programming language, had started working on a compiler to implement some of these new goals, which generated C code as its output. By the middle of that year, Go had become a full-time project at Google and late the following year, on November 10, 2009, Go was publicly announced as an open source project⁸. The language is now used in many of Google’s production level systems as well as in many open source tools⁹.

Looking at both of these stories, it is interesting to note that Swift was initially a private project and it wasn’t over until a year after its initial release that it was made open source, whereas Go was released as open source from the get-go. Go was open-sourced under a BSD- style license¹⁰, whereas Swift was open source under an Apache 2.0 license with a Runtime Library Exception¹¹. Another interesting thing to note is the influence of notable past Bell Labs employees Thompson and Pike on the development of Go, which may give some explanation as to why Go appears to be more C-influenced than Swift.

3 Initial Goals

Before Swift, software for iOS and OS X was written in Objective-C. Objective-C was powerful in that it allowed the high-performance benefits of a C-based language; however, it was also cumbersome to write software in, as it was essentially an extension of C to allow object oriented support built on top of basic C, which made for many points of confusion and error in programming¹². Swift was born out of a desire to make it easier to write software for iOS, macOS, watchOS, and tvOS. It takes inspiration from many other languages and was written to be easy to write software in, like a scripting language, but still take full advantage of the hardware it was being run on by being a compiled language like C++. With Swift, interaction with Objective-C code and libraries is still supported, while allowing for safer and more modern development of new software.

Alternatively, Go was born out of Google’s need for a better systems programming language for scalable development¹³. Previous to the creation of Go, Google had used a software stack of C++, Python, and Java applications. Programmers often were left to make a decision between efficient compilation, efficient execution, or ease of programming, due to restrictions of these mainstream languages. The desire for a simpler way to write software for highly

⁷<https://golang.org/doc/faq#history>

⁸<https://techcrunch.com/2009/11/10/google-go-language/>

⁹https://golang.org/doc/faq#Is_Google_using_go_internally

¹⁰<https://golang.org/LICENSE>

¹¹<https://swift.org/LICENSE.txt>

¹²<https://www.quora.com/What-are-the-reasons-that-Swift-was-created-given-that-Objective-C-was-used>

¹³https://golang.org/doc/faq#creating_a_new_language

scalable network servers and distributed systems lead to the creation of Go, a language designed to combine the ease of programming found in interpreted dynamically typed languages with the efficiency of compiled statically typed languages, including built-in support for things like multi-core processing and simple dependency analysis¹⁴.

4 Outside Influence

The design of Swift was influenced by a variety of existing programming languages, primarily C#, CLU, D, Haskell, Objective-C, Python, Rust, and Ruby¹⁵. Go takes much influence from the C-based languages with respect to its basic syntax while incorporating aspects from the Pascal/Modula/Oberon family as well as ideas from Newsqueak and Limbo¹⁶. Because of their shared roots in C and other C-based languages, Swift and Go have many similarities in their syntax and the expressiveness that it allows.

¹⁴https://talks.golang.org/2012/splash.article#TOC_4

¹⁵<http://nondot.org/sabre/>

¹⁶<https://golang.org/doc/faq#ancestors>