

Table of Contents

Swift 3

Hello World 6

SwiftUI 6

Programming Language

- [Xcode](#)
- [Swift](#)
- [SwiftUI](#)
- [Objective-C](#)
- [iOS](#)
- [MacOS](#)

Swift

Swift is a general-purpose, multi-paradigm, compiled programming language developed by Apple Inc. for iOS, macOS, watchOS, tvOS, Linux, and z/OS.

Swift (<i>programminglanguage</i>)	
Full Name	Swift
Short Name	Swift
Description	Swift is a general-purpose, multi-paradigm, compiled programming language
Company	Apple Inc
Web	Yes
Mobile	Yes
Enterprise	Yes
Embedded	Yes

What is Swift?

Swift is a programming language developed by Apple for building applications for iOS, macOS, watchOS, and tvOS. Swift was designed to be fast, safe, and easy to use, and it is intended to replace Objective-C as the primary programming language for Apple's platforms.

What platforms can I build with Swift?

Swift can be used to build applications for iOS, macOS, watchOS, and tvOS. In addition, Swift can also be used to build server-side applications using frameworks such as Vapor and Kitura.

What are some of the key features of Swift?

Some of the key features of Swift include:

- Optionals: a way of handling values that may be missing
- Generics: a way of writing reusable code that can work with different types
- Closures: a way of writing code that can be passed around and executed at a later time
- Type inference: a way of allowing the compiler to infer the type of a variable based on its value
- Automatic memory management: a way of managing memory automatically so that you don't

have to worry about deallocating objects manually

Is Swift open source?

Yes, Swift was open sourced by Apple in 2015, and it is now maintained by the Swift community as well as Apple. The Swift source code is available on GitHub.

What are some popular frameworks and libraries for Swift?

Some popular frameworks and libraries for Swift include:

- SwiftUI: a framework for building user interfaces for iOS, macOS, watchOS, and tvOS
- Combine: a framework for handling asynchronous events and data streams
- Alamofire: a library for handling network requests and responses
- Kingfisher: a library for downloading and caching images
- Realm: a database framework for iOS, macOS, watchOS, and tvOS

Can I use Swift with Objective-C?

Yes, you can use Swift and Objective-C together in the same project, and you can call Objective-C code from Swift and vice versa. This makes it easy to adopt Swift gradually in existing projects that use Objective-C.

Snippet from [Wikipedia](#): **Swift (programming language)**

Swift is a high-level general-purpose, multi-paradigm, compiled programming language developed by Apple Inc. and the open-source community. Swift compiles to machine code, as it is an LLVM-based compiler. Swift was first released in June 2014, and the Swift toolchain has shipped in Xcode since version 6, released in 2014.

Apple intended Swift to support many core concepts associated with Objective-C, notably dynamic dispatch, widespread late binding, extensible programming, and similar features, but in a "safer" way, making it easier to catch software bugs; Swift has features addressing some common programming errors like null pointer dereferencing and provides syntactic sugar to help avoid the pyramid of doom. Swift supports the concept of protocol extensibility, an extensibility system that can be applied to types, structs and classes, which Apple promotes as a real change in programming paradigms they term "protocol-oriented programming" (similar to traits and type classes).

Swift was introduced at Apple's 2014 Worldwide Developers Conference (WWDC). It underwent an upgrade to version 1.2 during 2014 and a major upgrade to Swift 2 at WWDC 2015. Initially a proprietary language, version 2.2 was made open-source software under the Apache License 2.0 on December 3, 2015, for Apple's platforms and Linux.

Through version 3.0 the syntax of Swift went through significant evolution, with the core team making source stability a focus in later versions. In the first quarter of 2018 Swift

surpassed Objective-C in measured popularity.

Swift 4.0, released in 2017, introduced several changes to some built-in classes and structures. Code written with previous versions of Swift can be updated using the migration functionality built into Xcode. Swift 5, released in March 2019, introduced a stable binary interface on Apple platforms, allowing the Swift runtime to be incorporated into Apple operating systems. It is source compatible with Swift 4.

Swift 5.1 was officially released in September 2019. Swift 5.1 builds on the previous version of Swift 5 by extending the stable features of the language to compile-time with the introduction of module stability. The introduction of module stability makes it possible to create and share binary frameworks that will work with future releases of Swift.

Swift 5.5, officially announced by Apple at the 2021 WWDC, significantly expands language support for concurrency and asynchronous code, notably introducing a unique version of the actor model.

Swift 5.9, was released in September 2023 and includes a macro system, generic parameter packs, and ownership features like the new consume operator.

The current version, Swift 5.10, was released in March 2024. This version improves the language's concurrency model, allowing for full data isolation to prevent data races. It is also the last release before Swift 6.

[Creative Commons Attribution-Share Alike 4.0](#)

GitHub Topics

- <https://github.com/topics/swift>

Swift is a general-purpose programming language built using a modern approach to safety, performance, and software design patterns. Inspired by many other modern programming languages, Swift is intended to be more resilient and expressive than Objective-C, its predecessor.

Free Programming Books

- [Essential Swift](#)
- [Hacking with Swift](#)
- [Swift Notes for Professionals](#)
- [The Swift Programming Language](#)
- [Using Swift with Cocoa and Objective-C](#)

Source: [EbookFoundation](#)

Hello World

```
println("Hello, world!") // Swift 1.x  
print("Hello, world!")   // Swift 2.x
```

Source: [Wikibooks](#)

SwiftUI


As of 2019, with Xcode 11, Apple has added a major new UI paradigm called **SwiftUI**. **SwiftUI** replaces the older Interface Builder paradigm with a new declarative development paradigm.

- <https://developer.apple.com/xcode/swiftui/>

See also: [Programming Languages](#)

[Apple](#), [language](#), [programminglanguage](#), [dev-apple](#)

ToDo

- 
- Optionals
- Control structures
- Functions
- Closures
- Object-oriented programming (OOP)
- Classes
- Inheritance
- Polymorphism
- Protocols
- Extensions
- Generics
- Error handling
- Defer statements
- Memory management
- ARC (Automatic Reference Counting)
- Properties
- Access control
- Structs
- Enums
- Initializers
- Control flow statements
- Collection types
- iOS development
- SwiftUI framework

- Combine framework

From:

<https://almbok.com/> - **ALMBoK.com**

Permanent link:

<https://almbok.com/dev/swift>

Last update: **2023/05/01 17:29**

