SWIFT

Swift is a powerful and intuitive programming language for macOS, iOS, watchOS, tvOS and beyond. Writing Swift code is interactive and fun, the syntax is concise yet expressive, and Swift includes modern features developers love. Swift code is safe by design, yet also produces software that runs lightning-fast.

Swift is the result of the latest research on programming languages, combined with decades of experience building Apple platforms. Named parameters are expressed in a clean syntax that makes APIs in Swift even easier to read and maintain. Even better, you don’t even need to type semi-colons. Inferred types make code cleaner and less prone to mistakes, while modules eliminate headers and provide namespaces. To best support international languages and emoji, Strings are Unicode-correct and use a UTF-8 based encoding to optimize performance for a wide-variety of use cases. Memory is managed automatically using tight, deterministic reference counting, keeping memory usage to a minimum without the overhead of garbage collection.

Swift has many other features to make your code more expressive:

* Generics that are powerful and simple to use
* Protocol extensions that make writing generic code even easier
* First class functions and a lightweight closure syntax
* Fast and concise iteration over a range or collection
* Tuples and multiple return values
* Structs that support methods, extensions, and protocols
* Enums can have payloads and support pattern matching
* Functional programming patterns, e.g., map and filter
* Native error handling using try / catch / throw

Designed for Safety

Swift eliminates entire classes of unsafe code. Variables are always initialized before use, arrays and integers are checked for overflow, memory is automatically managed, and enforcement of exclusive access to memory guards against many programming mistakes. Syntax is tuned to make it easy to define your intent — for example, simple three-character keywords define a variable ( var ) or constant ( let ). And Swift heavily leverages value types, especially for commonly used types like Arrays and Dictionaries. This means that when you make a copy of something with that type, you know it won’t be modified elsewhere.

Another safety feature is that by default Swift objects can never be nil. In fact, the Swift compiler will stop you from trying to make or use a nil object with a compile-time error. This makes writing code much cleaner and safer, and prevents a huge category of runtime crashes in your apps. However, there are cases where nil is valid and appropriate. For these situations Swift has an innovative feature known as optionals. An optional may contain nil, but Swift syntax forces you to safely deal with it using the ? syntax to indicate to the compiler you understand the behavior and will handle it safely.

Fast and Powerful

From its earliest conception, Swift was built to be fast. Using the incredibly high-performance LLVM compiler technology, Swift code is transformed into optimized native code that gets the most out of modern hardware. The syntax and standard library have also been tuned to make the most obvious way to write your code also perform the best whether it runs in the watch on your wrist or across a cluster of servers.

Swift is a successor to both the C and Objective-C languages. It includes low-level primitives such as types, flow control, and operators. It also provides object-oriented features such as classes, protocols, and generics, giving Cocoa and Cocoa Touch developers the performance and power they demand.

Great First Language

Swift can open doors to the world of coding. In fact, it was designed to be anyone’s first programming language, whether you’re still in school or exploring new career paths. For educators, Apple created free curriculum to teach Swift both in and out of the classroom. First-time coders can download Swift Playgrounds—an app for iPad that makes getting started with Swift code interactive and fun.

Source and Binary Compatibility

With Swift 5, you don’t have to modify any of your Swift 4 code to use the new version of the compiler. Instead you can start using the new compiler and migrate at your own pace, taking advantage of new Swift 5 features, one module at a time. And Swift 5 now introduces binary compatibility for apps. That means you no longer need to include Swift libraries in apps that target current and future OS releases, because the Swift libraries will be included in every OS release going forward. Your apps will leverage the latest version of the library in the OS, and your code will continue to run without recompiling. This not only makes developing your app simpler, it also reduces the size of your app and its launch time.

Playgrounds and Read-Eval-Print-Loop (REPL)

Much like Swift Playgrounds for iPad, playgrounds in Xcode make writing Swift code incredibly simple and fun. Type a line of code and the result appears immediately. You can then Quick Look the result from the side of your code, or pin that result directly below. The result view can display graphics, lists of results, or graphs of a value over time. You can open the Timeline Assistant to watch a complex view evolve and animate, great for experimenting with new UI code, or to play an animated SpriteKit scene as you code it. When you’ve perfected your code in the playground, simply move that code into your project. Swift is also interactive when you use it in Terminal or within Xcode’s LLDB debugging console. Use Swift syntax to evaluate and interact with your running app, or write new code to see how it works in a script-like environment.

Package Manager

Swift Package Manager is a single cross-platform tool for building, running, testing and packaging your Swift libraries and executables. Swift packages are the best way to distribute libraries and source code to the Swift community. Configuration of packages is written in Swift itself, making it easy to configure targets, declare products and manage package dependencies. New to Swift 5, the swift run command now includes the ability to import libraries in a REPL without needing to build an executable. Swift Package Manager itself is actually built with Swift and included in the Swift open source project as a package.

Objective-C Interoperability

You can create an entirely new application with Swift today, or begin using Swift code to implement new features and functionality in your app. Swift code co-exists alongside your existing Objective-C files in the same project, with full access to your Objective-C API, making it easy to adopt.