Swift on Non-Mac Devices

**Best option: The end result is a Windows binary that runs your limited Swift code.**

1. Using the binaries whichi allow you to compile and build your Swift code requires your machine to be running a 64 bit processor
2. You're going to need the C++ Redistributable for Visual Studio 2015
3. You must have the Visual Studio 2015 SDK installed.
4. The process is able to compile Swift 2.x code.  Does not seem to support Swift 3.x syntax and commands
5. Not all libraries are available.  Things like UIKit and Foundation libraries are not fully available.
6. This builds a console application.  This does not build any graphic UI representation.  This is purely so you can learn basic Swift syntax and methods etc.
7. Learn Swift syntax
8. Learn to write pure Swift as you concentrate only on Swift -- As you're learning it's actually nice to only focus on the language itself first.  This will keep the noise of learning Cocoa (iOS UI library) and other items seperate.
9. The syntax and skills you learn will directly apply to code you will write when you build your iOS apps.
10. Location to access the libraries: [Release Swift for Windows (MSVC) · tinysun212/swift-windows · GitHub](https://github.com/tinysun212/swift-windows/releases/tag/swift-msvc-20160418)[[^](https://github.com/tinysun212/swift-windows/releases/tag/swift-msvc-20160418)]

**You could develop Swift on a virtual machine.**

1. The only restriction is the host computer has to be a Mac running macOS (or older MacOS X versions). If you attempt this on another host computer such as Windows or Linux the virtual macOS will refuse to start.

**Connecting XCode Cloud to Github**

When you first configure your project or workspace to use Xcode Cloud, you need to allow Xcode Cloud to access your Git repository. It uses this access to automatically build and test your code when you make changes to the codebase.

If you host your code as part of a GitHub organization, the person who first configures a project to use Xcode Cloud must be an organization owner. If you don’t use a GitHub organization, the person who first configures a project to use Xcode Cloud must have the admin permission. If you don’t have the required role or permission, see [Connect Xcode Cloud to an admin-managed Git repository](https://developer.apple.com/documentation/xcode/configuring-xcode-cloud-for-your-team#Connect-Xcode-Cloud-to-an-admin-managed-Git-repository).

Link of steps to connect: <https://irenebosque.com/how-to-xcode-and-github/>

**Memory Safety in Swift**

1. By default, Swift prevents unsafe behavior from happening in your code. For example, Swift ensures that variables are initialized before they’re used, memory isn’t accessed after it’s been deallocated, and array indices are checked for out-of-bounds errors.
2. Swift also makes sure that multiple accesses to the same area of memory don’t conflict, by requiring code that modifies a location in memory to have exclusive access to that memory. Because Swift manages memory automatically, most of the time you don’t have to think about accessing memory at all. However, it’s important to understand where potential conflicts can occur, so you can avoid writing code that has conflicting access to memory. If your code does contain conflicts, you’ll get a compile-time or runtime error.