

# Getting Started with Swift on Windows

One can build and run Swift natively or through the Windows Subsystem for Linux.

## Native Windows

Currently there are two supported ways to build Swift for Windows.

1. To cross-compile Swift for Windows from another host operating system (using `clang`), see [Cross Compiling for Windows](#)
2. To build on Windows using Microsoft Visual C++ (MSVC), see [Building on Windows](#)

## Windows Subsystem for Linux (WSL)

[Windows Subsystem for Linux](#) is an Ubuntu environment. Follow the instructions for building on [Linux](#)

There two versions of WSL as of November 23, 2019:

- WSL1: is the current stable version. Both Swift compilation and execution work but REPL and debugging (LLDB) hang on startup.
- WSL2: Both REPL and debugging work with WSL2. Although, WSL2 is still in development, it is available by downloading an insider build. Installing WSL2 is pretty simple if WSL1 was already installed (switch to insider, download an insider build and run some scripts). WSL2 can be installed [by following this link](#).