CROSS-PLATFORM SWIFT

JP SIMARD - @SIMJP - TRY! SWIFT - TOKYO - MARCH 2, 2016

Realm is a mobile database hundreds of millions of people rely on

JAZZY & SWIFTLINT


```
1  func swiftLintTest() {
2      let someForceCast = NSNumber() as! Int
3      let colonOnWrongSide :Int = 0
4      // SwiftLint is smart enough to ignore comments and strings
5      // NSNumber() as! Int => no error
6      "let colonOnWrongSide :Int = 0" // => no error
7  }
8
```

Force Cast Violation (High Severity): Force casts should be avoided

Colon Violation (Low Severity): When specifying a type, always associate the colon with the identifier

Open Source  Swift

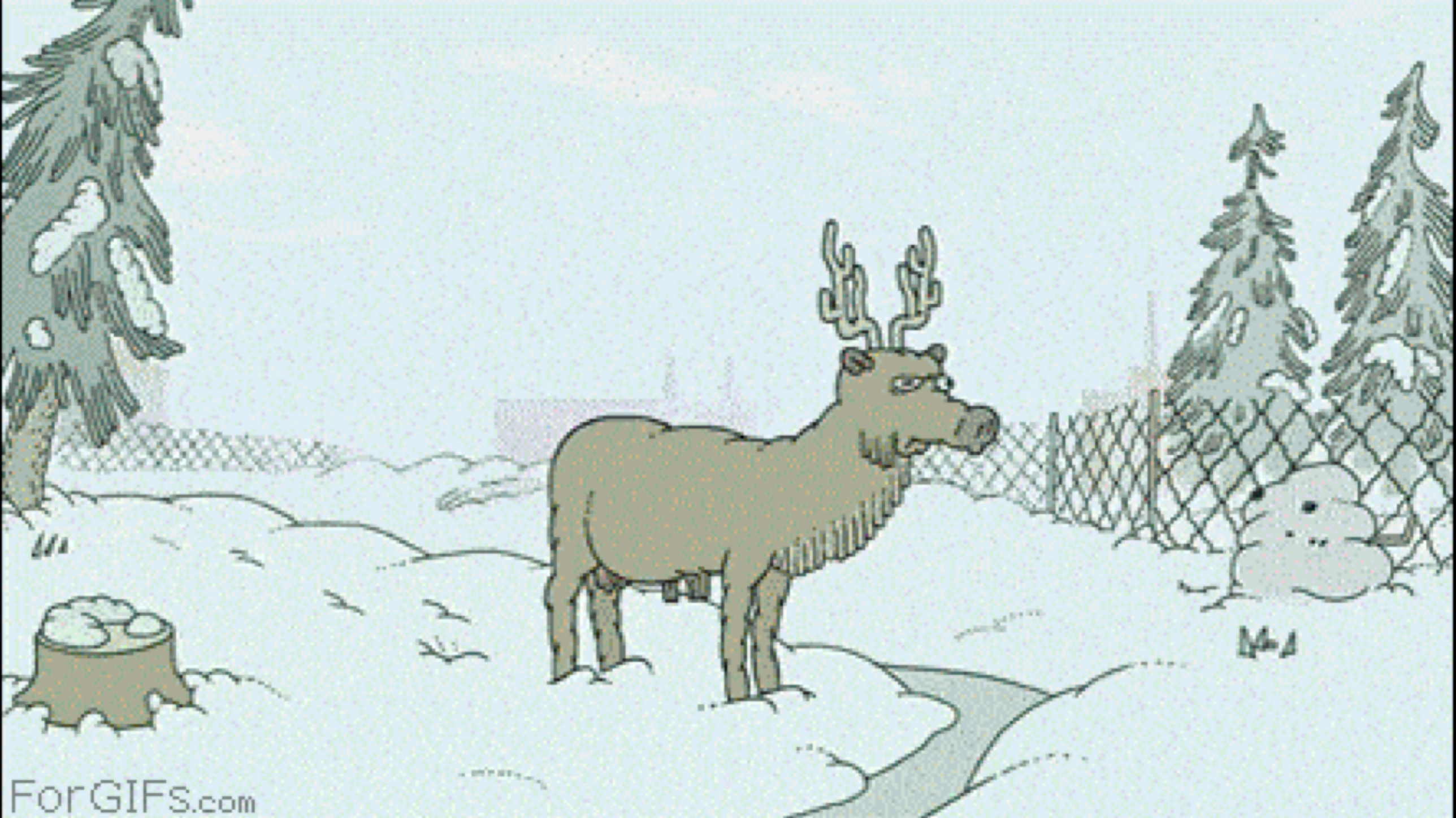
WE'LL GO OVER...

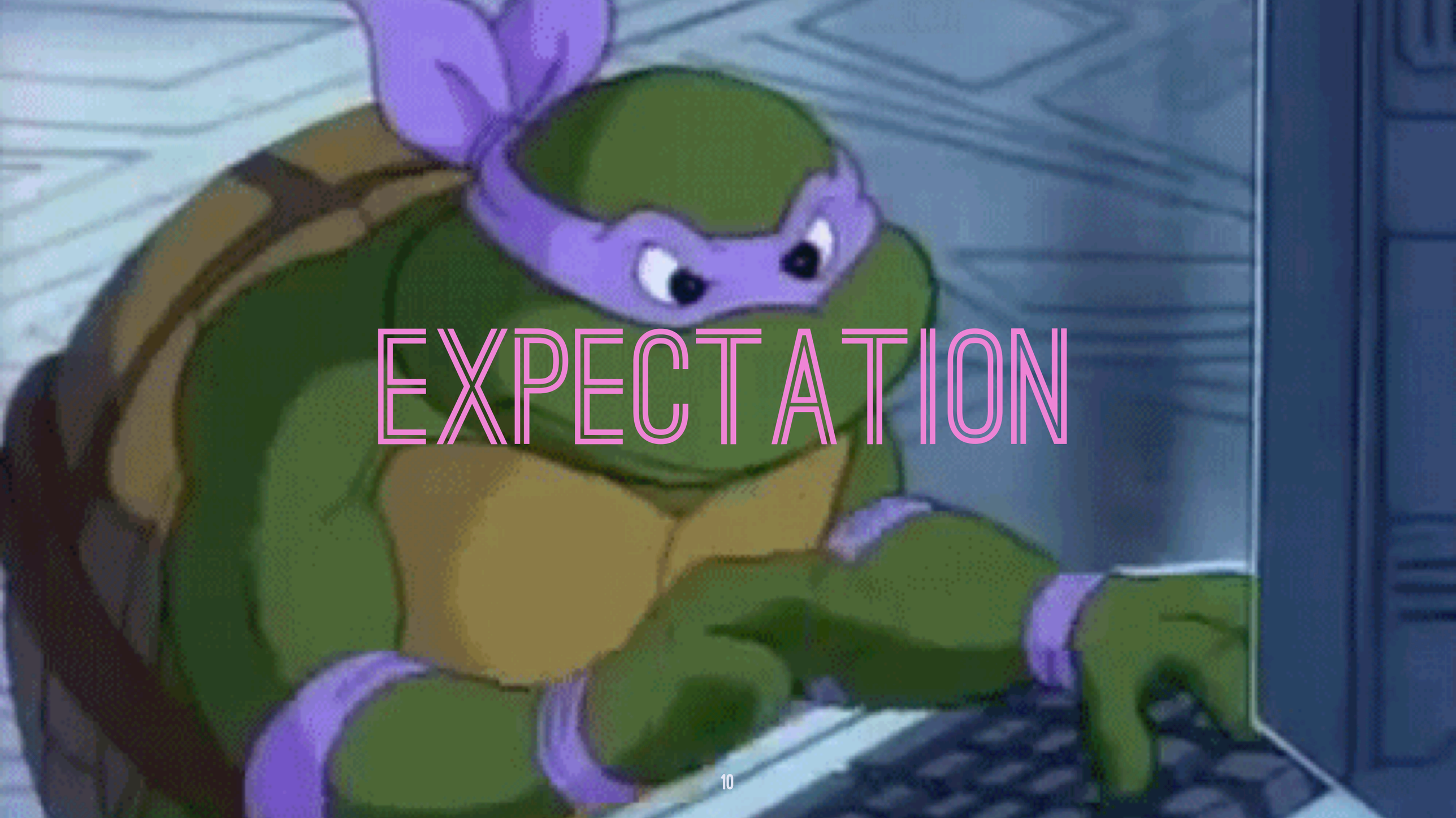
- DEVELOPMENT ENVIRONMENT
- SWIFT PACKAGE MANAGER
- TESTING
- CONTINUOUS INTEGRATION

EXACTLY 2 PLATFORMS

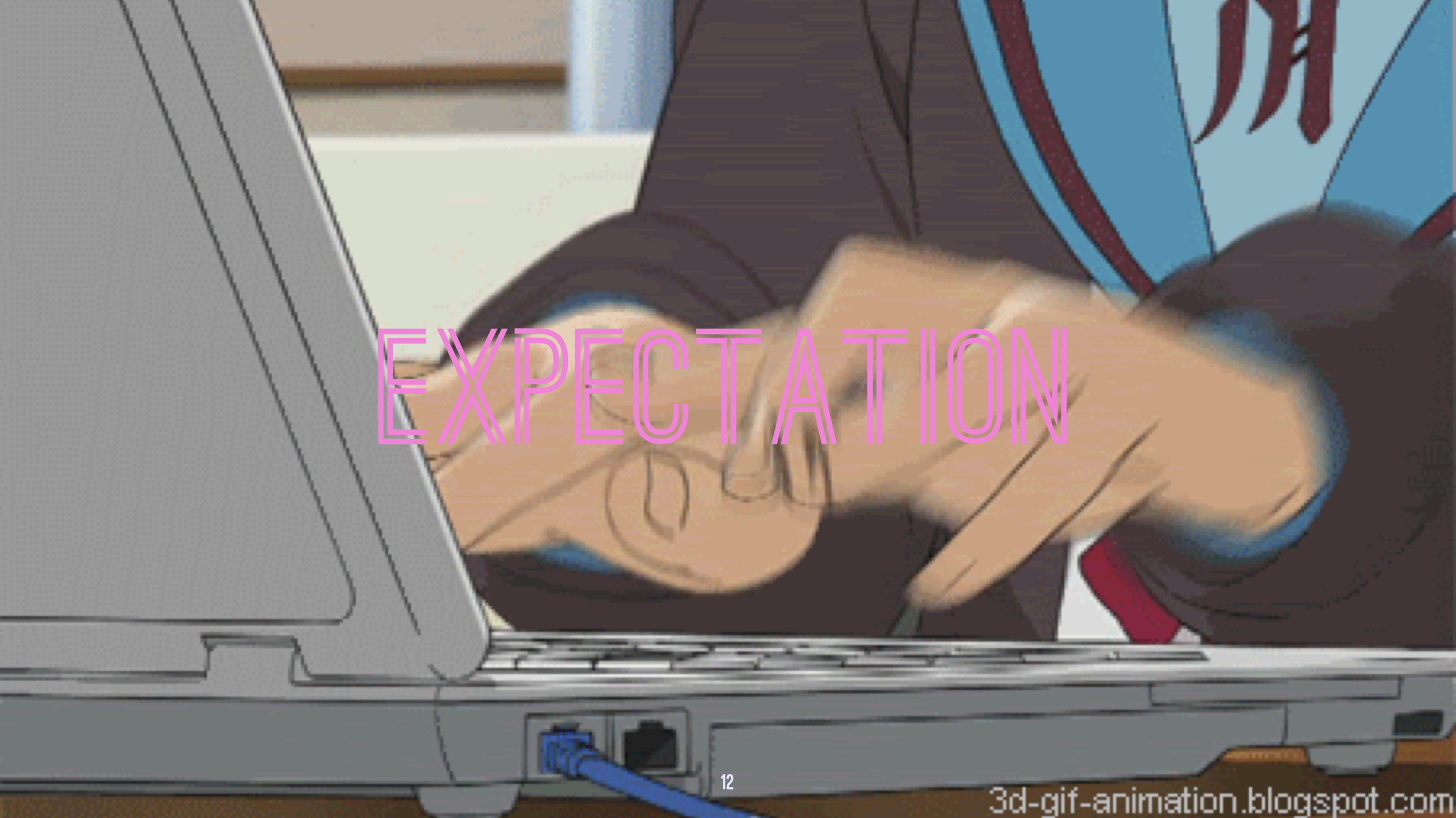
EXACTLY 3 PLATFORMS

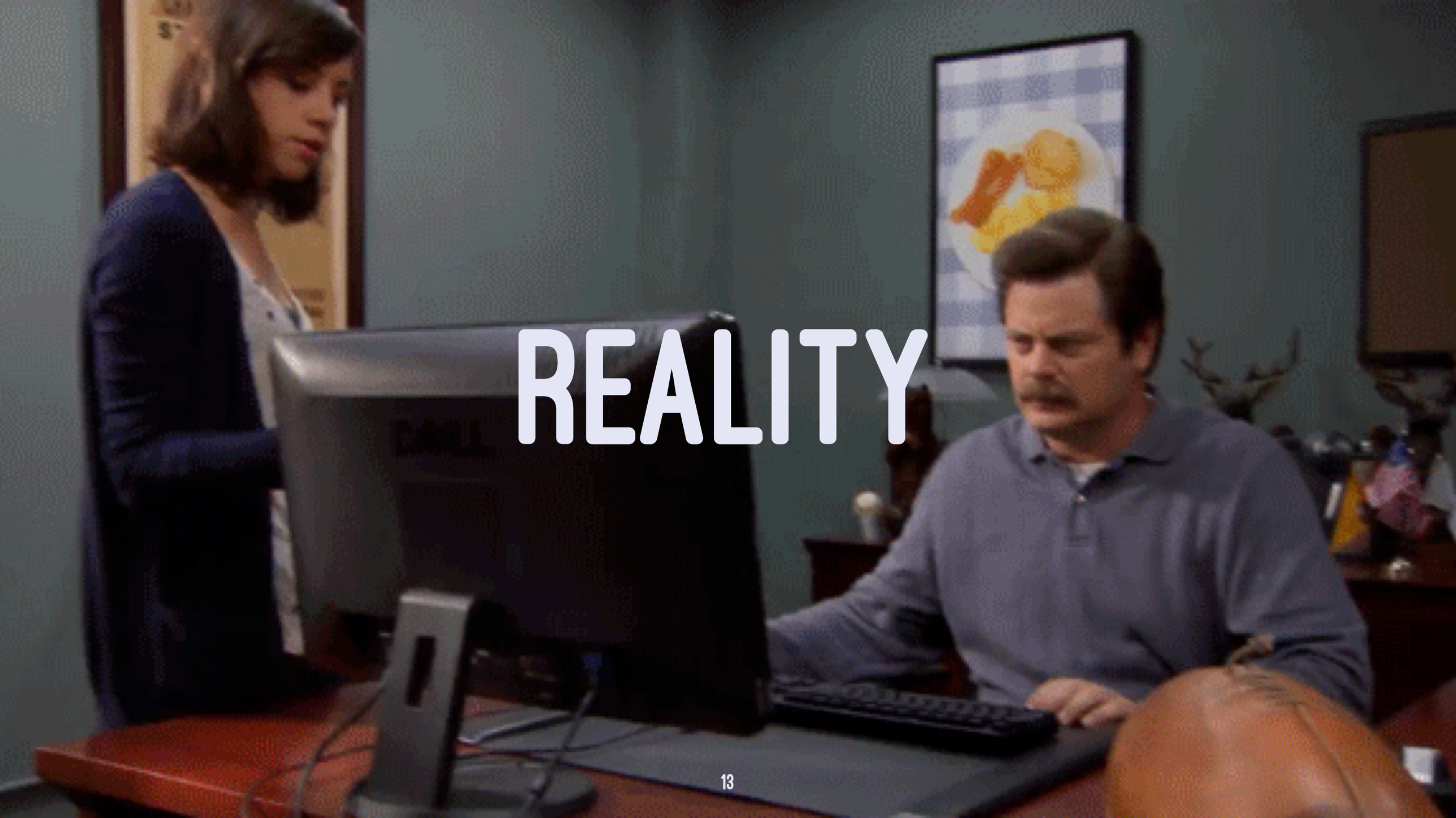
EXACTLY 3-ISH PLATFORMS

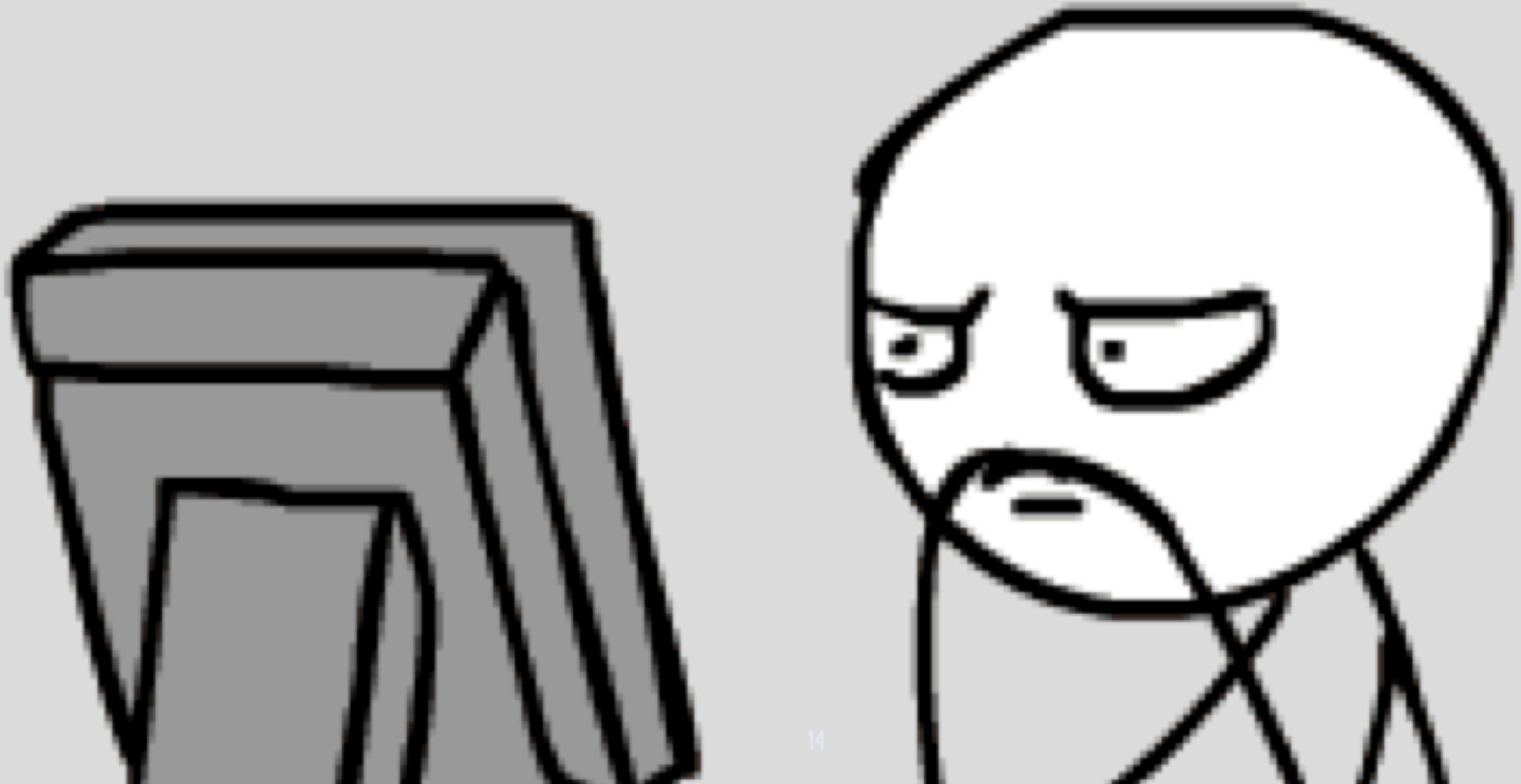

ForGIFs.com


EXPECTATION


EXPECTATION




REALITY



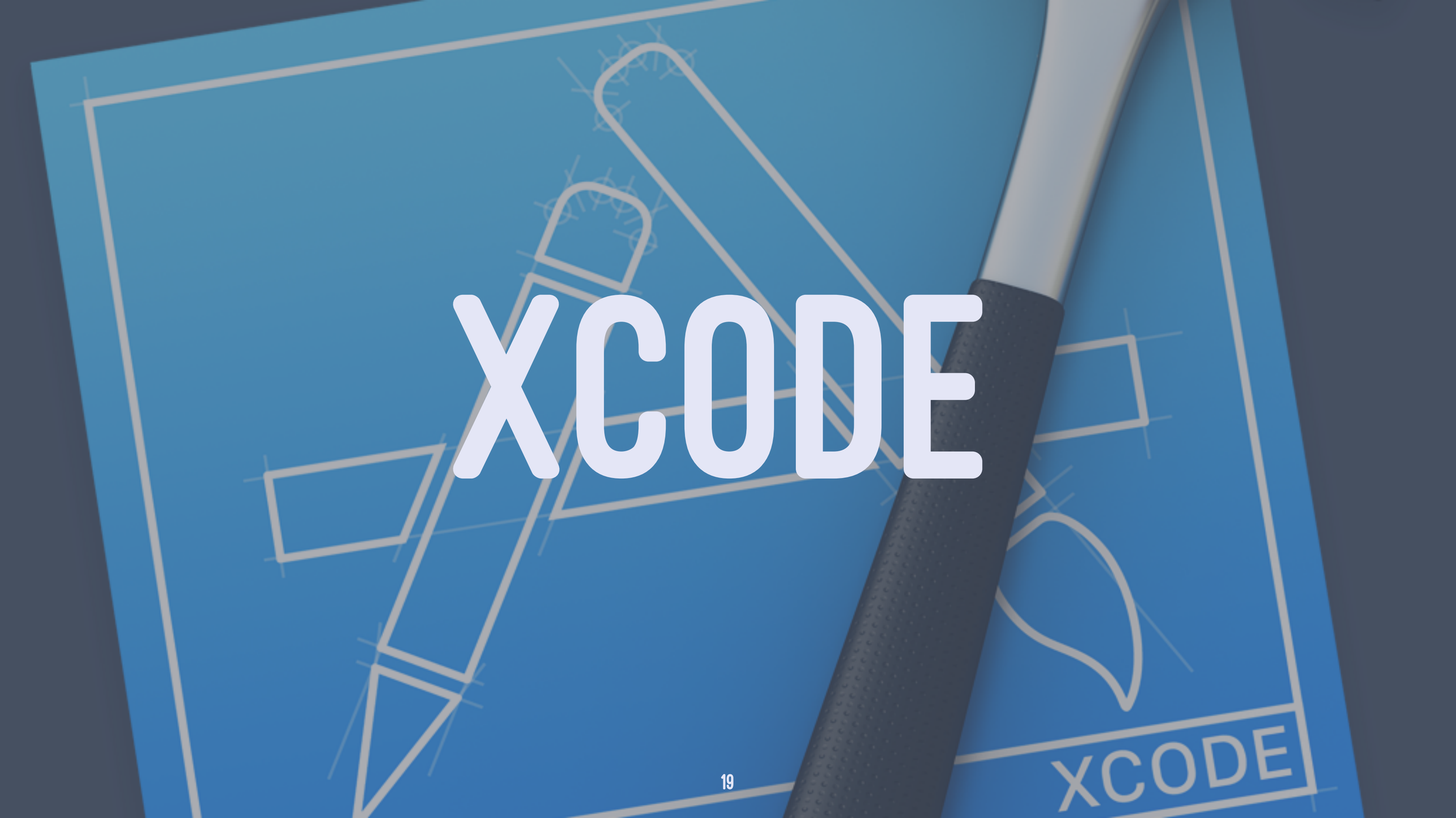

COMPUTER
MALFUNCTION

APPLE'S DOING A GREAT JOB

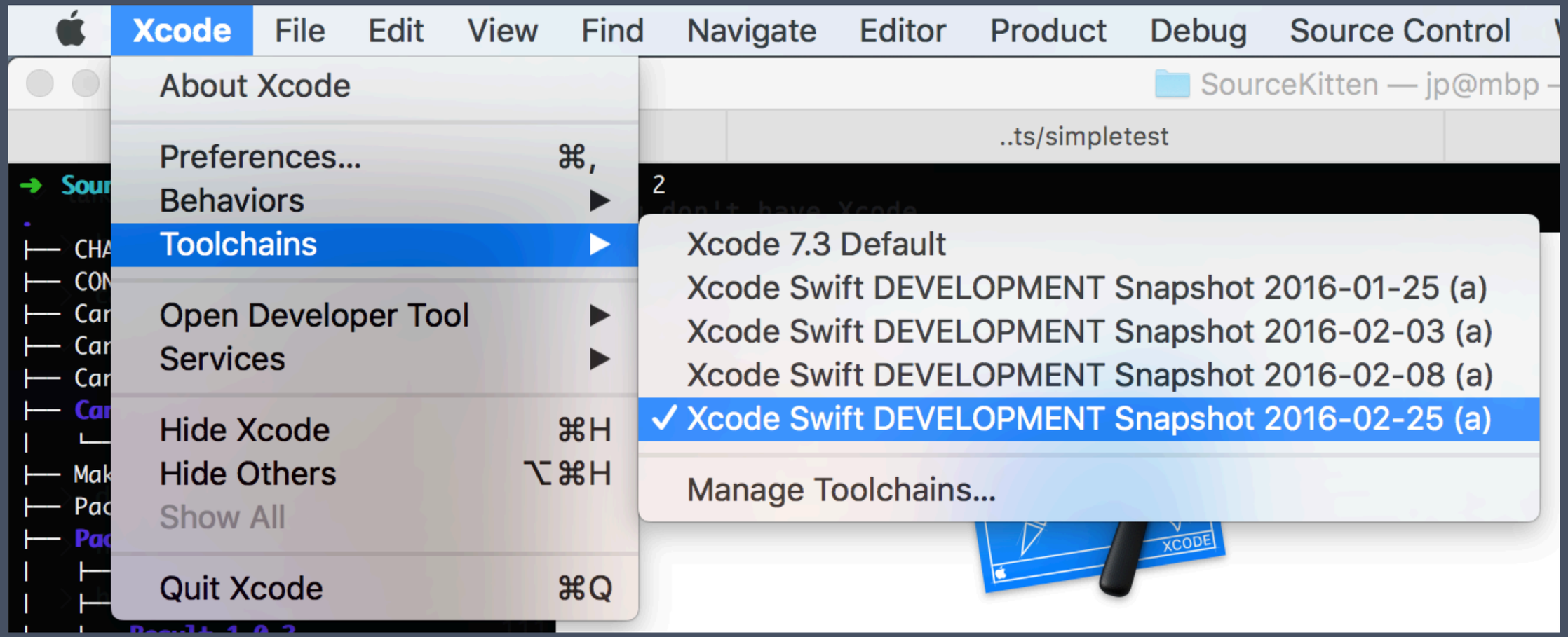
EARLY DAYS

DEVELOPING ALL FROM ONE PLACE

- XCODE
- XCODE+TOOLCHAIN
- DOCKER+CLI+EDITOR


XCODE
XCODE

XCODE+TOOLCHAIN



DOCKER+CLI+EDITOR

```
$ brew install docker docker-machine
$ docker-machine create --driver virtualbox default
$ eval $(docker-machine env default)
$ docker pull swiftdocker/swift
$ docker run -it -v `pwd`:/project swiftdocker/swift bash
---
root@445ab1838149 $ cd /project
root@445ab1838149 $ swift build & swift test
```

SWIFT PACKAGE MANAGER

THE SWIFT PACKAGE MANAGER IS A TOOL FOR MANAGING THE DISTRIBUTION OF SWIFT CODE. IT'S INTEGRATED WITH THE SWIFT BUILD SYSTEM TO AUTOMATE THE PROCESS OF DOWNLOADING, COMPILING, AND LINKING DEPENDENCIES.

USE SPM EVEN FOR SMALL OR PRIVATE PROJECTS!

THINGS YOU'D EXPECT TO WORK

- `dynamic` KEYWORD
- CASTING
- FOUNDATION
- GRAND CENTRAL DISPATCH
- AUTO-IMPORTING OF FRAMEWORKS

CASTING HAS NEVER BEEN SO DIFFICULT…

```
public func materialize<T>(@autoclosure f: () throws -> T) -> Result<T, NSError> {
    do {
      return .Success(try f())
-   } catch {
-     return .Failure(error as NSError)
+   } catch let error as NSError {
+     return .Failure(error)
    }
 }
```

FROM PORTING RESULT TO LINUX:
HTTPS://GITHUB.COM/ANTITYPICAL/RESULT/PULL/135

#IFS EVERYWHERE

```
#if SWIFT_PACKAGE
import SomeModuleOtherwiseAvailable
#endif

#if os(Linux)
// some arcane hack
#else
// something more reasonable
#endif
```

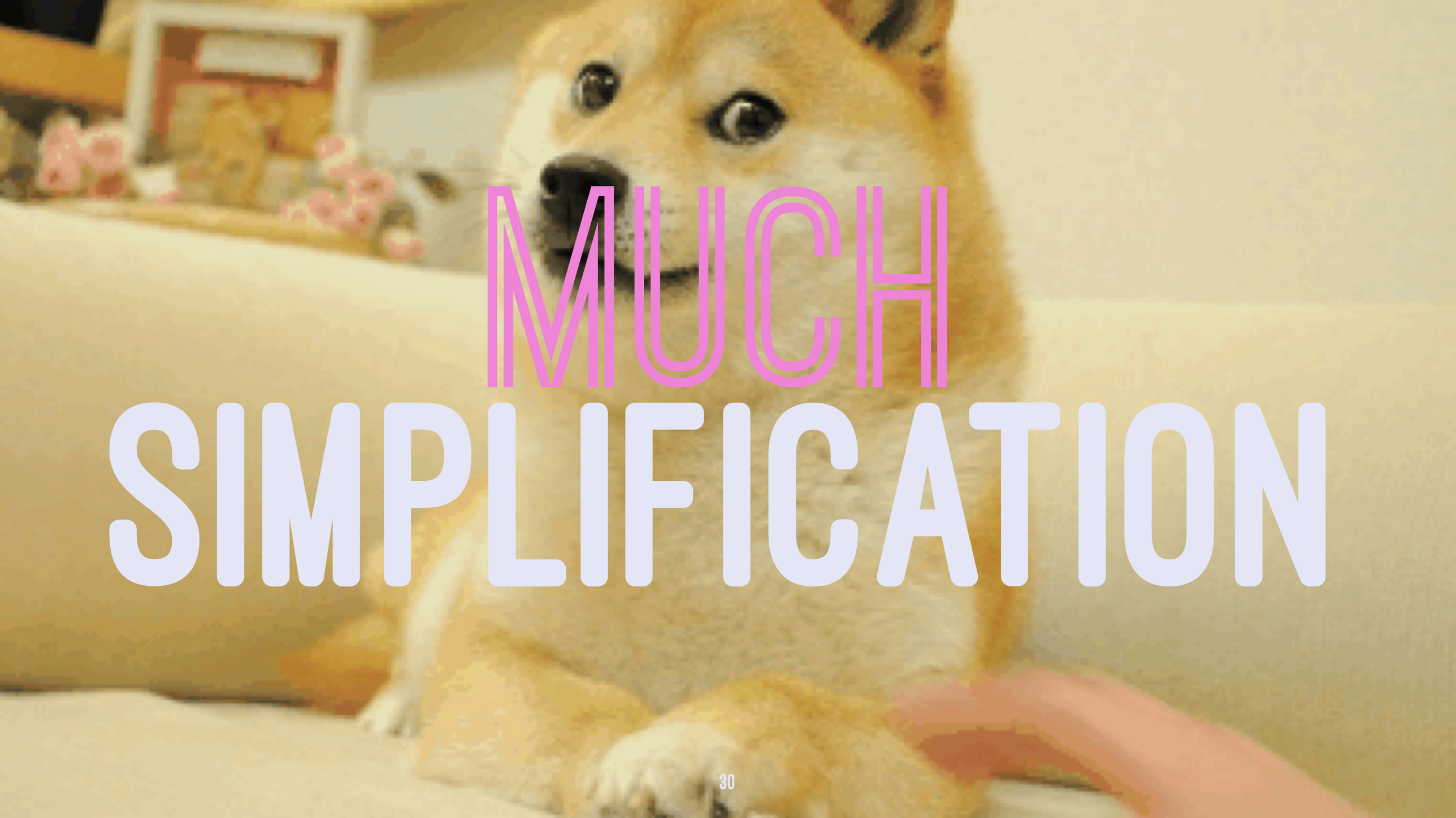
TESTING

👀 REDISCOVERING XCTEST 👀

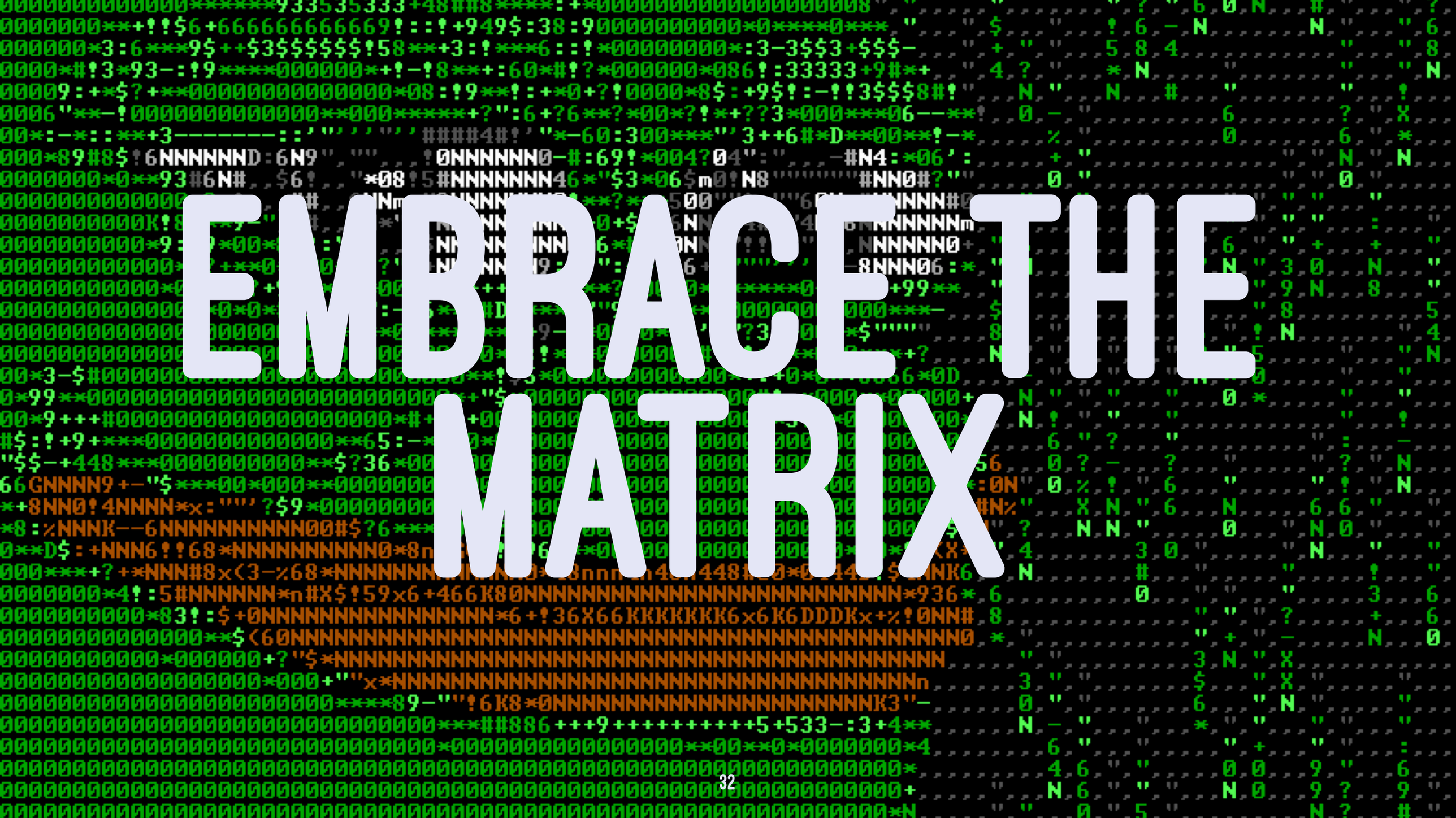
```
import PackageDescription

let package = Package(
  name: "MyPackage",
  targets: [
    Target(name: "MyPackage"),
    Target(name: "MyPackageTests", // build tests as a regular target...
      dependencies: [.Target(name: "MyPackage")]), // ...that depend on the main one
  ],
  dependencies: [
#if !os(Linux) // XCTest is distributed with Swift releases on Linux
    .Package(
      // no version tags at apple/swift-corelibs-xctest, so fork it
      url: "https://github.com/username/swift-corelibs-xctest.git",
      majorVersion: 0
    ),
#endif
  ]
)
```

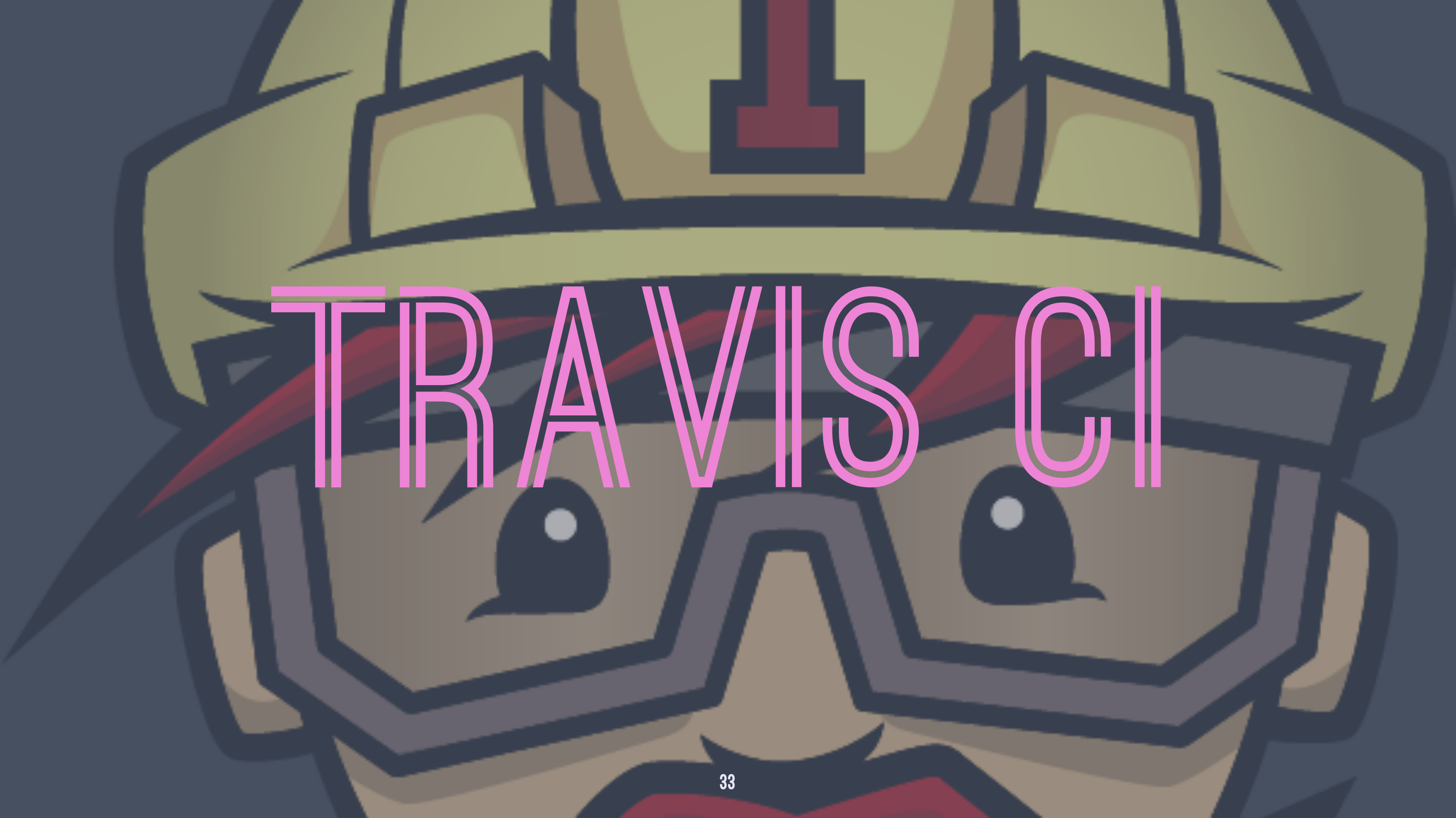
SPM TESTING

```
Package.swift # can be empty in simplest configuration
Sources/
  MyPackage/
    file.swift
Tests/
  LinuxMain.swift # needs `@testable import MyPackagetest` & `XCTMain()`
  MyPackage/ # *must* be named after package being tested
    test.swift # can `import XCTest` and `@testable import MyPackage`
```


MUCH
SIMPLIFICATION

CONTINUOUS INTEGRATION


EMBRACE THE
MATRIX


TRAVIS CI

TRAVIS CONFIGURATION

```
matrix:
  include:
    - env: JOB=OSX_Xcode
    - env: JOB=OSX_SPM
    - env: JOB=Linux
```

TRAVIS XCODE

```
script: xcodebuild test
env: JOB=OSX_Xcode
os: osx
osx_image: xcode7.2
language: objective-c
before_install: pod install / carthage update / etc.
```

TRAVIS OS X SPM

```
script:
  - swift build
  - .build/Debug/MyUnitTests
env: JOB=OSX_SPM
os: osx
osx_image: xcode7.2
language: objective-c
before_install:
  - export SWIFT_VERSION=swift-DEVELOPMENT-SNAPSHOT-2016-02-25-a
  - curl -O https://swift.org/builds/development/xcode/$(SWIFT_VERSION)/$(SWIFT_VERSION)-osx.pkg
  - sudo installer -pkg $(SWIFT_VERSION)-osx.pkg -target /
  - export PATH=/Library/Developer/Toolchains/$(SWIFT_VERSION).xctoolchain/usr/bin:"${PATH}"
```

TRAVIS LINUX

```
script:
  - swift build
  - .build/Debug/MyUnitTests
env: JOB=Linux
dist: trusty
sudo: required
language: generic
before_install:
  - DIR="$(pwd)"
  - cd ..
  - export SWIFT_VERSION=swift-DEVELOPMENT-SNAPSHOT-2016-02-25-a
  - wget https://swift.org/builds/development/ubuntu1404/$SWIFT_VERSION/$SWIFT_VERSION-ubuntu14.04.tar.gz
  - tar xzf $SWIFT_VERSION-ubuntu14.04.tar.gz
  - export PATH="${PWD}/${SWIFT_VERSION}-ubuntu14.04/usr/bin:${PATH}"
  - cd "$DIR"
```

EARLY DAYS

どうもありがとう

THANK YOU!

JP SIMARD - @SIMJP - TRY! SWIFT - TOKYO - MARCH 2, 2016