SWIFT SCRIPTS: ZERO TO HERO 🦸‍♀️

FEDERICO ZANETELLO

★★★★★ FIVESTARS.BLOG · @ZNTFDR

WHY?

GETTING STARTED

hello

CREATING AN EXECUTABLE

```
$ mkdir hello
$ cd hello
$ swift package init --type executable
```

THE PACKAGE STRUCTURE

```
├── .gitignore
├── Package.swift
├── README.md
├── Sources
│   └── hello
│       └── main.swift
└── Tests
    ├── helloTests
    │   ├── helloTests.swift
    │   └── XCTestManifests.swift
    └── LinuxMain.swift
```

PACKAGE.SWIFT

```swift
// swift-tools-version:5.2

import PackageDescription

let package = Package(
    name: "hello",
    dependencies: [
    ],
    targets: [
        .target(
            name: "hello",
            dependencies: []),
        .testTarget(
            name: "helloTests",
            dependencies: ["hello"]),
    ]
)
```

THE PACKAGE STRUCTURE

```
├── .gitignore
├── Package.swift
├── README.md
├── Sources
│   └── hello
│       └── main.swift
└── Tests
    ├── helloTests
    │   ├── helloTests.swift
    │   └── XCTestManifests.swift
    └── LinuxMain.swift
```

MAIN.SWIFT

```swift
print("Hello, world!")
```

BUILD RUN TEST

```
$ swift build hello
$ swift run hello
$ swift test
```

BUILD RUN TEST

```
$ swift build hello
$ swift run hello
$ swift test
```

..OR USE XCODE

WE'RE DONE!

🙌

GUI -> TUI

LAUNCH ARGUMENTS

```swift
import Foundation

// The first argument is the script execution path.
let arguments = CommandLine.arguments.dropFirst()

if let name = arguments.first {
  print("Hello \(name)")
} else {
  exit(EXIT_FAILURE)
}
```

```
$ swift run hello Federico
> Hello Federico
```

INTERACTIVE

```swift
import Foundation

print("What's your name?")

guard
  let name = readLine(),
  !name.isEmpty else {
    exit(EXIT_FAILURE)
}

print("Hello \(name)")
```

```
$ swift run hello
> What's your name?
> Federico
> Hello Federico
```

ENVIRONMENT VARIABLES

```swift
import Foundation

let processInfo = ProcessInfo.processInfo
let environment = processInfo.environment

if let name = environment["MYNAME"] {
  print("Hello \(name)")
} else {
  exit(EXIT_FAILURE)
}
```

```
$ MYNAME=Federico swift run
> Hello Federico
```

PIPELINE MESSAGES

```swift
import Foundation

let standardInput: FileHandle = .standardInput
let data = standardInput.availableData

if let inputString = String(
  data: data,
  encoding: .utf8
) {
  print("Hello \(inputString)")
} else {
  exit(EXIT_FAILURE)
}
```

```
$ printf Federico | swift run hello
> Hello Federico
```

ARGUMENT PARSER

ADDING A DEPENDENCY

```swift
let package = Package(
    name: "hello",
    dependencies: [
        .package(url: "https://github.com/apple/swift-argument-parser.git",
                 from: "0.0.1"),
    ],
    targets: [
        .target(
            name: "hello",
            dependencies: [
                .product(name: "ArgumentParser",
                         package: "swift-argument-parser")
            ]
        ),
        .testTarget(
            name: "helloTests",
            dependencies: ["hello"]),
    ]
)
```

ARGUMENTPARSER

```swift
import ArgumentParser

struct Hello: ParsableCommand {
  @Argument(help: "Specify your name.")
  var name: String

  func run() throws {
    print("Hello \(name)")
  }
}

Hello.main()
```

```
$ swift run hello Federico
> Hello Federico

$ swift run hello
> Error: Missing expected argument '<name>'
> Usage: hello <name>

$ swift run hello --help
> USAGE: hello <name>
>
> OPTIONS:
>   <name>            Specify your name.
>   -h, --help        Show help information.
```

SWIFTTOOLSSUPPORT'S

TSCUTILITY & TSCBASIC

ADDING A DEPENDENCY 2

```
let package = Package(
    name: "hello",
    dependencies: [
        .package(url: "https://github.com/apple/swift-tools-support-core.git",
                 from: "0.0.1"),
    ],
    targets: [
        .target(
            name: "hello",
            dependencies: [
                .product(name: "SwiftToolsSupport",
                         package: "swift-tools-support-core")
            ]
        ),
        .testTarget(
            name: "helloTests",
            dependencies: ["hello"]),
    ]
)
```

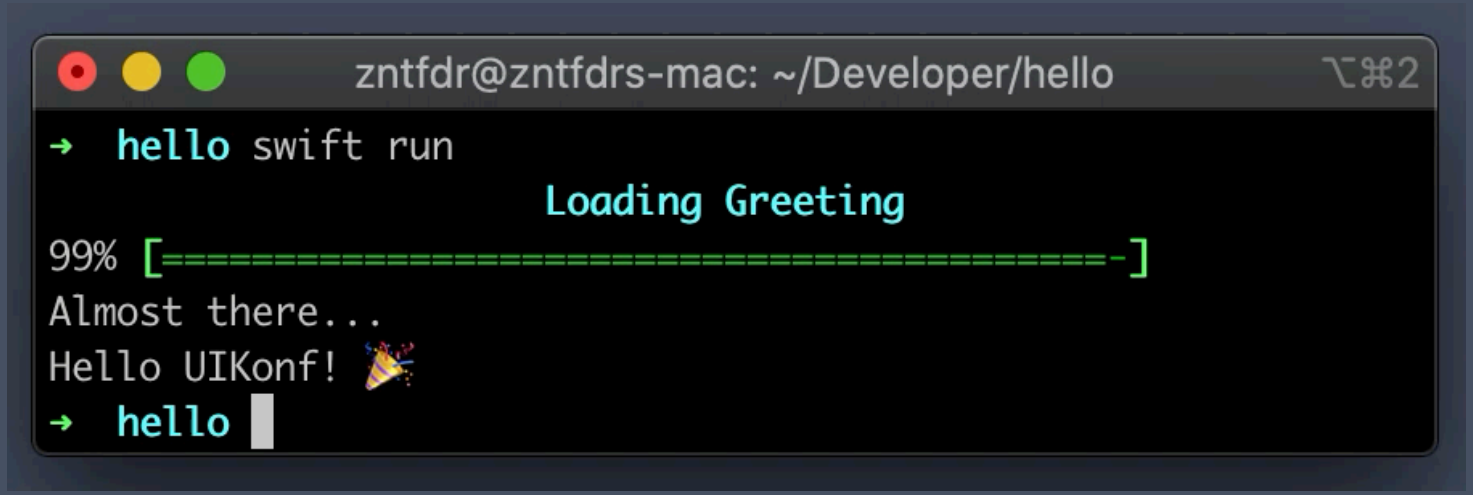
PROGRESS STATE

```swift
import Foundation
import TSCBasic
import TSCUtility

let animation = PercentProgressAnimation(
  stream: stdoutStream,
  header: "Loading Greeting")

for i in 0..<100 {
  let second: Double = 1_000_000
  usleep(UInt32(second * 0.05))
  animation.update(step: i,
                   total: 100,
                   text: "Almost there...")
}

animation.complete(success: true)
print("Hello UIKonf! 🎉")
```



```
➜  hello swift run
                 Loading Greeting
99% [=============================================-]
Almost there...
Hello UIKonf! 🎉
➜  hello
```

RELEASE

USE THE SCRIPT ANYWHERE

```
$ swift build -c release
$ cp .build/release/hello /usr/local/bin/hello

$ hello
```

WE'RE DONE!

*REALLY 🥳

LINKS

GITHUB.COM/APPLE/SWIFT-PACKAGE-MANAGER
GITHUB.COM/APPLE/SWIFT-ARGUMENT-PARSER
GITHUB.COM/APPLE/SWIFT-TOOLS-SUPPORT-CORE
FIVESTARS.BLOG/ULTIMATE-GUIDE-SWIFT-EXECUTABLES.HTML
GITHUB.COM/ZNTFDR/TALKS

SWIFT SCRIPTS: ZERO TO HERO 🦸🏼‍♀️

FEDERICO ZANETELLO

★★★★★ FIVESTARS.BLOG · @ZNTFDR