

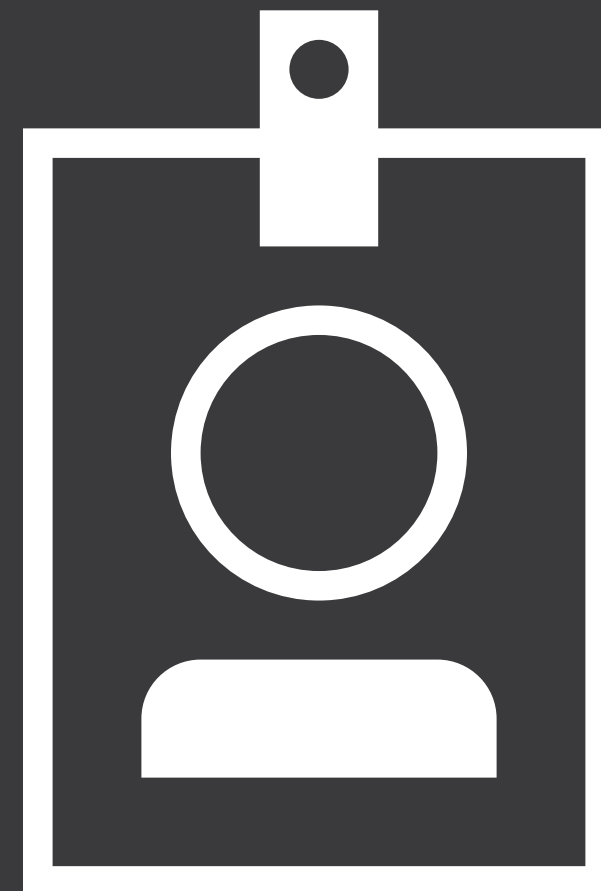


CLOUD **FOUNDRY**

SUMMIT 2016

swift workshop

Anton McConville





Anton McConville

Developer Advocate - IBM Bluemix

@antonmc

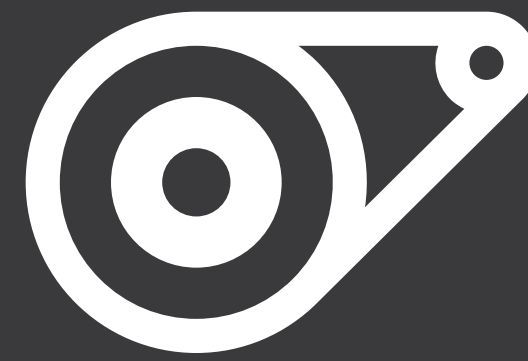
What's in store ...



why swift?



working with swift



**demo and
walkthrough**



future steps



why swift?

Swift. A modern programming language that is safe, fast, and interactive.

Swift is a powerful and intuitive programming language for iOS, OS X, tvOS, and watchOS. **Writing Swift code is interactive and fun, the syntax is concise yet expressive, and apps run lightning-fast.** Swift is ready for your next project — or addition into your current app — because Swift code works side-by-side with Objective-C.

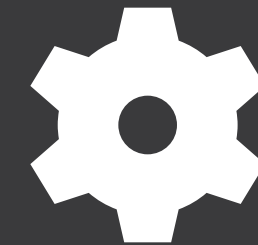
Full stack



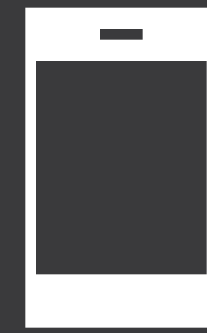
Full stack



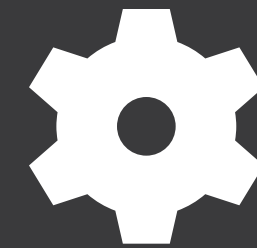
Compiled



Full stack



Compiled



Future friendly?





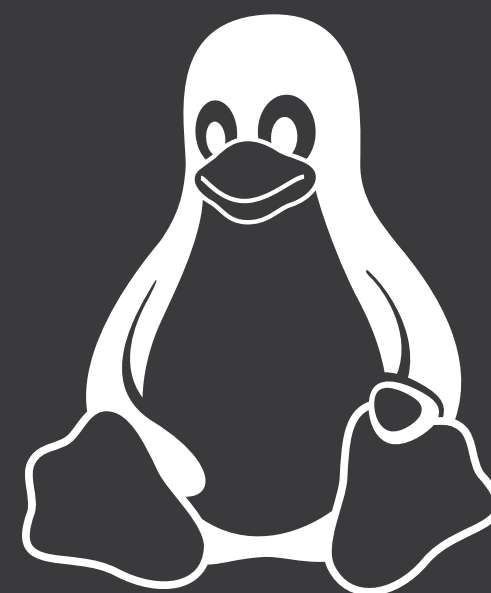
working with swift

where to get it ...



swift.org

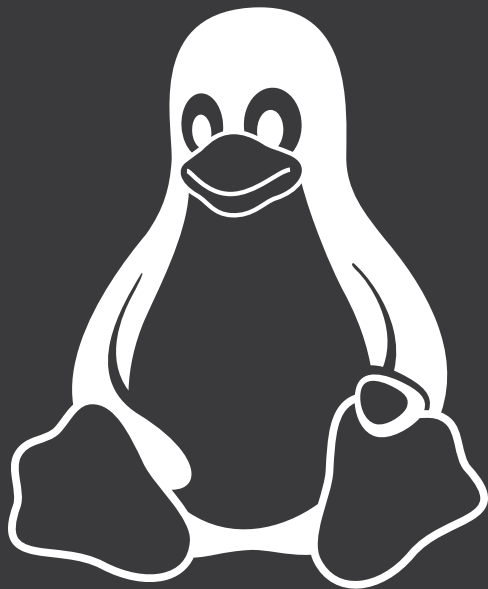
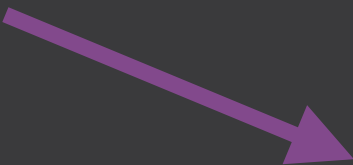
compatibility



linux

compatibility

early days still



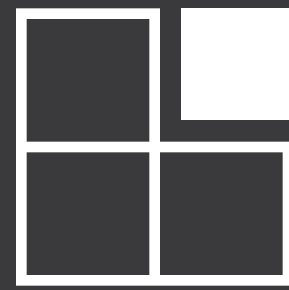
linux



OS X



package manager



package manager

Swift organizes code into **modules**. Each module specifies a namespace and enforces access controls on which parts of that code can be used outside of the module.

A program may have all of its code in a single module, or it may import other modules as **dependencies**.

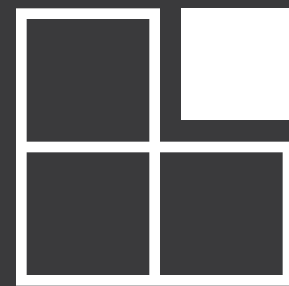


packages

A **package** consists of **Swift source files** and **a manifest file**.

The manifest file, called `Package.swift`, defines the package's name and its contents using the `PackageDescription` module.

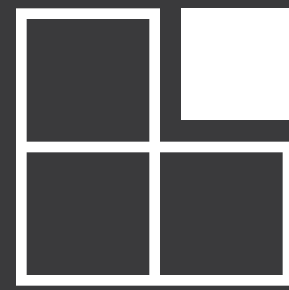
A **package has one or more targets**. Each target specifies a product and may declare one or more dependencies.



package.swift

```
import PackageDescription
```

```
let package = Package(  
    name: "project",  
    targets: [],  
    dependencies: [  
        .Package(url: "https://github.com/apple/example-package.git",  
            majorVersion: 1),  
    ]  
)
```



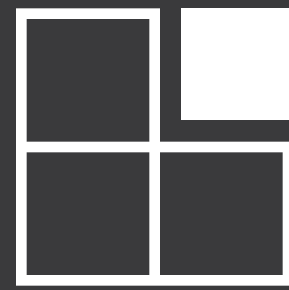
structure

```
project
├── Sources
│   ├── main.swift
│   ├── another.swift
│   └── stuff.swift
└── Package.swift
```



build

> swift build



build

project

└── .build

└── debug

├── project.a

├── project.o

├── project.swiftdoc

└── project.swiftmodule



run

```
> .build/debug/Server
```



**demo and
walkthrough**



future steps

Kitura

web framework

Kitura

Swift 3

evolution

Kitura

Swift 3

Windows?

more platforms

Bluemix & Swift



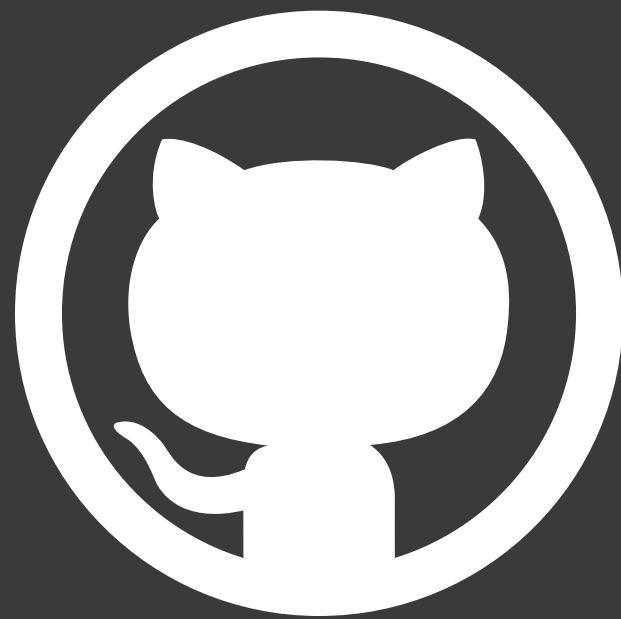
<http://www.ibm.com/cloud-computing/bluemix/swift/>

open swift packages



<https://swiftpkgs.ng.bluemix.net/>

This workshop



<https://github.com/antonmc/swift-workshop>

bluemix.net

swift.org



thanks :)