

SWIFT NIGHT

20 APRIL 2016

- > SWIFT LANGUAGE @STIJN (30 MIN)
- > USING PLAYGROUND'S WITH SCRIPTING @JELLE (30 MIN)
 - > SWIFT AND OPEN SOURCE @HANS (15 MIN)
- > SWIFT AND BACKEND DEVELOPMENT @DYLAN @HANNES (20 MIN)

**BASED ON A PRESENTATION OF @SRBAKER @NSSCOTLAND
2014**

PROCEDURAL / OBJECT ORIENTED – 00 / FUNCTIONAL

> TYPE SYSTEMS

AS GOES WITH PATHS. THEY THEND TO
MIX. BUT THEY WHERE NEVER INTENDED
TO DO SO...

- @SRBAKER

BIND Code TO hardware \equiv TYPE

WHAT IS TYPE?

MORE LATER ...

LETS JUGGLE SOME TYPES AROUND...

PROCEDURAL

DATA (TYPE) MOVES AND CHANGES

MANY OO PROGRAMMING IS ACTUALLY PROCEDURAL
-- @SRBAKER

EXAMPLES OF PURE PROCEDURAL

C / GO / FORTRAN / PASCAL

00

OO MAKES CODE UNDERSTANDABLE BY
INCAPSULATING THE MOVING PARTS.



– MICHEAL FEATHERS

OBJECTIVE-C / PYTHON

MESSAGING?



GROW?

SO WHAT?

```
class Foo {  
    let webService = Service ()  
  
    func do (completion: ()->()) {  
        webService.do {  
            //code async  
            completion()  
        }  
    }  
}
```

// write unit test

```
class FooTests {  
    func testFoo () {  
        let foo = Foo()  
        foo.do {  
            Assert ...  
            //This will be async  
        }  
    }  
}
```

FUNCTIONAL

FP MAKES CODE UNDERSTANDABLE BY
MINIMIZING THE MOVING PARTS.



– MICHEAL FEATHERS

00 -> INCAPSULATING
FP -> MINIMIZING



MOVING PARTS



-- MICHEAL FEATHERS



RECAP



CHECKING

static compile time Run time Dynamic

STATIC

```
var i = 1  
i = "One" // Error
```

DYNAMIC

CLASS INSTANTIATION CAN BE DATA DRIVEN

```
NSString *className = [self getClassNameFromJSON];
```

```
//This can only be checked at run time. The JSON file might change.  
A* a = [[NSClassFromString(className) alloc] init];
```

TYPE INFERENCE

```
var s = "A string"  
var s2:String  
s2 = "Another string"
```

MORE INFO

MESSAGING IN SWIFT

NOT YET, OR MIGHT NEVER BE ...

MORE INFO

SWIFT IS AN IMPURE LANGUAGE THAT
IS BUILT INTENTIONALLY THAT WAY.

– ANONYMOUS



.NET TO SWIFT -> WILL THERE BE ANYTHING ELSE?

- > SWIFT LANGUAGE 
- > USING PLAYGROUND'S WITH SCRIPTING @JELLE (30 MIN)
 - > SWIFT AND OPEN SOURCE @HANS (15 MIN)
- > SWIFT AND BACKEND DEVELOPMENT @DYLAN @HANNES (20 MIN)

PLAYGROUND

PLAYGROUND WHUT?

INTERACTIVE SWIFT CODING ENVIRONMENT

PLAYGROUND DEMO

SCRIPTING

SCRIPTING WHUT?

YOU CAN JUST RUN SIMPLE COMMAND LINE SCRIPTS WRITTING IN SWIFT.

SCRIPTING SETUP

DOWNLOAD AND INSTALL THE SWIFT DEVELOPMENT
SNAPSHOT¹.

¹ [HTTPS://SWIFT.ORG/DOWNLOAD/](https://swift.org/download/)

SCRIPTING SETUP

ADD `export TOOLCHAINS=swift` TO
YOUR `.bash_profile`.

SCRIPTING SETUP

RUN THE FOLLOWING COMMAND FROM YOUR CLI.

```
swift build --version
```

SCRIPTING HOW?

CREATE AN EMPTY FILE WITH THE SWIFT HASHBANG AT THE TOP.

```
#!/usr/bin/swift  
print("We 💕 Swift!")
```

SCRIPTING HOW?

GIVE THE FILE EXECUTION RIGHTS.

```
chmod +x someScript.swift
```

SCRIPTING GO!

LET'S RUN THE SCRIPT.

```
swift someScript.swift  
./someScript.swift
```

SCRIPTING DEMO

REPL

REPL WHUT?

READ EVAL PRINT LOOP

REPL DEMO

- > SWIFT LANGUAGE 
- > USING PLAYGROUND'S WITH SCRIPTING 
- > SWIFT AND OPEN SOURCE @HANS (15 MIN)
- > SWIFT AND BACKEND DEVELOPMENT @DYLAN @HANNES (20 MIN)

DEC 3, 2015



SWIFT OPEN SOURCE



[HTTPS://GITHUB.COM/APPLE/SWIFT/COMMIT/
18844BC65229786B96B89A9FC7739C0FC897905E](https://github.com/apple/swift/commit/18844BC65229786B96B89A9FC7739C0FC897905E)

COMPILER ≠ DEBUGGER

- > LLVM, CLANG
- > LLDB + REPL

STANDARD LIBRARY

- **FUNDAMENTAL DATA TYPES** (INT, DOUBLE, ...)
- **COLLECTIONS** (ARRAY, DICTIONARY, ...)
- **PROTOCOLS & FUNCTIONS**
- **WRITTEN IN 'SWIFT'**

CORE LIBRARIES

- HIGHER LEVEL FUNCTIONALITY (NETWORKING, FILE SYSTEM, THREADS, NOTIFICATIONS, ...)
 - FOUNDATION (NS...), XCTEST, LIBDISPATCH
 - RE-IMPLEMENTED IN SWIFT (~~OBJC RUNTIME~~)
 - NOT YET READY FOR PRODUCTION

PACKAGE MANAGER

- > CFR RUBY GEM, COCOAPODS POD
- > WORK IN PROGRESS (SWIFT 3.0)



~~OPEN SOURCE~~ ~~OPEN SOURCE~~

> ~~XCODE~~

> ~~UIKit, AppKit, ...~~

> ~~CoreGraphics, CoreAnimation, ...~~

EVOLUTION

- SWIFT 1.X (WWDC14)
- SWIFT 2.0 (WWDC15)
- SWIFT 2.2 (CURRENT, MARCH 2016)
 - SWIFT 3.0

SUPPORTED PLATFORMS

- APPLE PLATFORMS
- LINUX (UBUNTU, X86_64)

**SOURCE COMPATIBILITY ACROSS MULTIPLE PLATFORMS, BUT
IMPLEMENTATION MECHANISMS MAY DIFFER**

ANDROID

- [HTTPS://GITHUB.COM/SWIFTANDROID](https://github.com/swiftandroid)
- [HTTPS://GITHUB.COM/APPLE/SWIFT/PULL/1442](https://github.com/apple/swift/pull/1442)

ANDROID

- STANDARD LIB COMPILES TO ANDROID ARMV7
 - 'RUN SWIFT CODE ON ANDROID DEVICE'
 - HACKS/DISABLED STUFF
- ~~WRITE ANDROID APPLICATIONS IN SWIFT~~

ANDROID

**'GOOGLE IS CONSIDERING MAKING SWIFT A 'FIRST CLASS'
LANGUAGE FOR ANDROID'**

- THE NEXT WEB

WINDOWS

- [HTTPS://GITHUB.COM/TINYSUN212/SWIFT-WINDOWS](https://github.com/tinysun212/swift-windows)
- [HTTPS://GITHUB.COM/APPLE/SWIFT/PULL/1108](https://github.com/apple/swift/pull/1108)

WINDOWS

- > CYGWIN
- > STANDARD LIBRARY
- > ~~REPL. DEBUGGER.~~

WEB

- > [HTTPS://GITHUB.COM/PERFECTLYSOFT/PERFECT](https://github.com/perfectlysoft/perfect)
 - > [HTTPS://GITHUB.COM/QUTHEORY/VAPOR](https://github.com/qutheory/vapor)
 - > [HTTPS://GITHUB.COM/IBM-SWIFT/KITURA](https://github.com/ibm-swift/kitura)
 - > [HTTPS://GITHUB.COM/NECOLT/SWIFTON](https://github.com/necolt/swifton)
- > [HTTPS://GITHUB.COM/ELLIOTTMINNS/BLACKFISH](https://github.com/elliottminns/blackfish)

ARMV6/ARMV7

> [HTTPS://BUGS.SWIFT.ORG/BROWSE/SR-40](https://bugs.swift.org/browse/SR-40)

RASPBERRY PI

- > RASPI MODEL B REVISION 2.0
 - > RASPBIAN
- > INSTALLED CLANG (DEPENDENCY FOR THE COMPILER)
 - > PRE-COMPILED ARMV6 LIB

RASPBERRY PI

> [HTTPS://GITHUB.COM/URAIMO/SWIFTYGPIO](https://github.com/URAIMO/SWIFTYGPIO)



DEMO



- > SWIFT LANGUAGE 
- > USING PLAYGROUND'S WITH SCRIPTING 
- > SWIFT AND OPEN SOURCE 
- > SWIFT AND BACKEND DEVELOPMENT @DYLAN @HANNES (20 MIN)

VAPOR

AN ELEGANT WEB FRAMEWORK FOR SWIFT

SWIFT-DEVELOPMENT-SNAPSHOT

[HTTPS://SWIFT.ORG/DOWNLOAD/#SNAPSHOTS](https://swift.org/download/#snapshots)

WHY VAPOR

- > WAT DOE VAPOR?
- > VAPOR HOST VOOR WEBSITE
 - > GESCHREVEN IN SWIFT

PACKAGE

```
import PackageDescription
```

```
let package = Package(  
    name: "Welcome",  
    dependencies: [  
        .Package(url: "https://github.com/qutheory/vapor.git", majorVersion: 0)  
    ]  
)
```

SERVE JSON HIGHSCORE

SERVER MAKEN

```
let application = Application()
application.get("highscore") { request in
    return Json(["name": "dylan", "score": "120"])
}
application.start()
```

MAAK SERVER ACTIEF

```
swift build
.build/debug/Welcome
```

VIEW DATA

URL OPENEN IN WEBVIEW




WAT KAN / KAN (NOG) NIET

- > SERVE JSON DATA
- > DATABANK NIET GELUKT `Fluent`
- > GEEN `Cloud` code



DEMO



- > SWIFT LANGUAGE 
- > USING PLAYGROUND'S WITH SCRIPTING 
- > SWIFT AND OPEN SOURCE 
- > SWIFT AND BACKEND DEVELOPMENT 