



SWIFT

UNCHARTED TERRITORY



CHRIS LATIMER



CHRIS LATTNER



LLVM

CHRIS LATTNER



LLVM
SWIFT



CHRIS LATTNER



LLVM
SWIFT
HANDSOME



STUCK WITH ME

JP Simard

@simjp

realm.io

Realm^β

THIS TALK

1. Early, incomplete version of a language? ✓
2. Buggy, pre-release compiler, IDE, OS? ✓
3. Unreleased beta of presentation app? ✓
4. Presenter who doesn't fully understand the language? ✓

WHAT COULD POSSIBLY GO WRONG?

WHY SWIFT > OBJC?

- ▶ Type safety & inference
 - ▶ Closures
 - ▶ Tuples
- ▶ Super-Enums
- ▶ Functional programming
- ▶ Generics

**Q. WHAT DOES IT
LOOK LIKE?**

TYPE SAFETY & INFERENCE

```
1 let meaningOfLife = 42
2 // meaningOfLife is inferred to be of type Int
3
4 let pi = 3.14159
5 // pi is inferred to be of type Double
6
7 let anotherPi = 3 + 0.14159
8 // anotherPi is also inferred to be of type Double
9 // number literals have no explicit type so they can be mixed
10
11 let cantMix = meaningOfLife + pi           ! Could not find an overload for '+' that accepts the supplied arguments
12 // constant variables have an explicit type and can't be matched
13
14 let castMix = Double(meaningOfLife) + pi
15 // casting helps working with different types|
```

LIKE RUST & SCALA

CLOSURES

```
1 let names = ["Chris", "Alex", "Ewa", "Barry", "Daniella"]
2
3 func backwards(s1: String, s2: String) -> Bool {
4     return s1 > s2
5 }
6 var reversed = sort(names, backwards)
7 // OR //
8 reversed = sort(names, { (s1: String, s2: String) -> Bool in return s1 > s2 } )
9 // reversed is equal to ["Ewa", "Daniella", "Chris", "Barry", "Alex"]
```

SWIFT CLOSURES ➡ OBJC BLOCKS

TUPLES

```
1 let http404Error = (404, "Not Found")
2 // http404Error is of type (Int, String), and equals (404, "Not Found")"
```

LIKE HASKELL & SCALA

SUPER-ENUMS*

```
1 enum Suit {  
2     case Spades, Hearts, Diamonds, Clubs  
3     func simpleDescription() -> String {  
4         switch self {  
5             case .Spades:  
6                 return "spades"  
7             case .Hearts:  
8                 return "hearts"  
9             case .Diamonds:  
10                return "diamonds"  
11            case .Clubs:  
12                return "clubs"  
13        }  
14    }  
15}  
16 let hearts = Suit.Hearts  
17 let heartsDescription = hearts.simpleDescription()
```

*OK, NOT EXACTLY THE CORRECT TECHNICAL TERM

FUNCTIONAL PROGRAMMING

```
1 let numbers = [1, 5, 3, 12, 2] [1, 5, 3, 12, 2]
2
3 numbers.map({
4     number: Int) -> Int in
5     let result = 3 * number
6     return result
7 })
8
9 numbers.filter({$0 % 2 == 0}) (11 times)
```

LIKE HASKELL, SCALA & MANY OTHERS

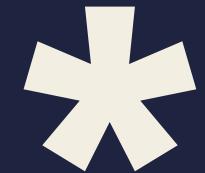
GENERICCS

```
1 // Reimplement the Swift standard library's
   optional type
2 enum OptionalValue<T> {
3     case None
4     case Some(T)
5 }
6 var possibleInteger: OptionalValue<Int> = .None
7 possibleInteger = .Some(100)
```

```
1 var shoppingList: String[] = ["Eggs", "Milk"]
2 // shoppingList has been initialized with two
   initial items
```

LIKE... UH... EVERY MODERN LANGUAGE!

Q: WHAT HAPPENED TO MY BELOVED



- ▶ concepts are still there: reference types and value types
- ▶ pointers still exist to interact with C APIs:

UnsafePointer<T>

- ▶ C APIs are still usable:

```
1 import Foundation
2 import Security
3
4 var secret: NSData = "Top Secret".dataUsingEncoding(NSUTF8StringEncoding,
5   allowLossyConversion: false)
6 let query = NSDictionary(objects: [kSecClassGenericPassword, "MyService", "Some account",
7   secret], forKeys: [kSecClass, kSecAttrService, kSecAttrAccount, kSecValueData])
8
9 let status = SecItemAdd(query as CFDictionaryRef, nil)
```

DEMO #1



**Q. HOW DOES IT
ALL WORK?**

A: it barely does 😊

```
tilViewController = segue!.destinationViewController  
| as String | as String SourceKitService  
| as String Terminated  
  
Editor functionality  
temporarily limited.  
You selected cell #0!
```

SERIOUSLY, HOW DOES IT WORK?

- ▶ Swift objects are actually Objective-C objects*
- ▶ *Without any methods or properties... strange!
- ▶ Just like C++, Swift methods are listed in a vtable
- ▶ Swift properties are ivars with Swift methods
 - ▶ ivars have no type encoding!!!

```
ivar_getTypeEncoding(); // always NULL
```

DEMO #2



A stack of cards showing assembly language and binary code. The cards are blue and white, with some text partially obscured by a grey overlay.

| Binary | Assembly |
|----------------|----------|
| 01101010110100 | |
| 11010101011001 | |
| 01010100101100 | |
| 10101001011001 | |
| C1101001010101 | XOR |
| | D1010101 |
| | MOV |
| | D0100100 |
| | PUSH RA |
| | D0100110 |
| | PUSH RBX |
| | D0010101 |
| | MOV RCX, |

XCODE & TOOLS INTEGRATION

- ▶ Clang knows absolutely nothing about Swift
- ▶ Swift compiler talks to clang through XPC

| | |
|-------------------|---------|
| sourcekitd-repl | 226 KB |
| sourcekitd-test | 425 KB |
| strings | 115 KB |
| strip | 183 KB |
| swift | 33.6 MB |
| swift-demangle | 431 KB |
| swift-ide-test | 16.3 MB |
| swift-stdlib-tool | 38 KB |

/Applications/Xcode6-Beta.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/
usr/bin

DEMO #3



JAZZY



[GITHUB.COM/REALM/JAZZY](https://github.com/realm/jazzy)

A SOULFUL WAY TO GENERATE DOCS FOR SWIFT & OBJECTIVE-C

LINKS ()

- ▶ Official Swift website
- ▶ The Swift Programming Language Book
- ▶ WWDC Videos
- ▶ WWDC Sample Code
- ▶ Xcode 6 (and other resources)

Apple Developer Account Required

LINKS (!

- ▶ This talk: github.com/jpsim/talks
- ▶ Jay Freeman's AltConf talk: debugging your (Swift) apps with `cycrypt`
- ▶ ObjC/Swift doc generator: github.com/realm/jazzy
- ▶ Evan Swick: Inside Swift
- ▶ Swift on StackOverflow

THANK YOU!

Meetup().questions?.askThem!

JP SIMARD, @SIMJP, REALM.IO