SWIFT

CLASS + HACKDAY

The set up

THE SETUP

Xcode6-Beta3

WHO AM I?

JP SIMARD

@SIMJP

REALM.IO

TODAY

CLASS

1. LEARN ABOUT SWIFT
2. BUILD AN APP
3. QUESTIONS

WHY SWIFT > OBJC?

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

Q: WHAT DOES IT LOOK LIKE?

TYPE SAFETY & INFERENCE

```
let anInt = 3
let aFloat = 0.1416
var pi = anInt + aFloat // Compile warning

pi = 3 + 0.1416
// Compiles: number literals are untyped
```

LIKE RUST & SCALA

CLOSURES

```
func backwards(s1: String, s2: String) -> Bool {
    return s1 > s2
}
sort(["b", "a"], backwards) // => ["a", "b"]
```

SWIFT CLOSURES 👉 OBJC BLOCKS

TUPLES

```
let http404Error = (404, "Not Found")
```

LIKE HASKELL & SCALA

SUPER-ENUMS*

*OK, NOT EXACTLY THE CORRECT TECHNICAL TERM

```
enum Suit {
    case Spades, Hearts, Diamonds, Clubs
    func simpleDescription() -> String {
        switch self {
        case .Spades:
            return "Spades"
        case .Hearts:
            return "Hearts"
        case .Diamonds:
            return "Diamonds"
        case .Clubs:
            return "Clubs"
        }
    }
}
```

FUNCTIONAL PROGRAMMING

```
let numbers = [1, 5, 3, 12, 2]
numbers.map {
    (number: Int) -> Int in
    return 3 * number
} // => [3, 15, 9, 36, 6]
numbers.filter {$0 % 2 == 0} // => [12, 2]
```

LIKE HASKELL, SCALA & MANY OTHERS

GENERICS

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

```swift
// Reimplement the Swift standard
// library's optional type
enum OptionalValue<T> {
    case None
    case Some(T)
}
var maybeInt: OptionalValue<Int> = .None
maybeInt = .Some(100)

// Specialized Array
var letters: [Array]
letters = ["a"]
```

Q: WHAT HAPPENED TO MY BELOVED

*

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>`, **etc.**

Q: WHAT HAPPENED TO MY BELOVED *?

C APIS ARE STILL USABLE

```
import Foundation
import Security

let secret = "Top Secret".dataUsingEncoding(NSUTF8StringEncoding)
let dict = [kSecClass as String: kSecClassGenericPassword,
    kSecAttrService as String: "MyService",
    kSecAttrAccount as String: "Some Account",
    kSecValueData as String: secret] as NSDictionary
let status = SecItemAdd(dict as CFDictionaryRef, nil)
```

JAZZY

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)

Free Apple Developer Account Required

LINKS (!)

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

LET'S BUILD A TO-DO APP!

THANK YOU!

Hackday().questions?.askThem!

```
Hackday().questions?.askThem!
```

JP SIMARD, @SIMJP, REALM.IO