HOW DO I EVEN Swift?

WHO AM I?

JP SIMARD

@SIMJP

REALM.IO

Realm β

```
// Using Realm Objects
let company = Company()
company.name = "Realm" // etc...
// Transactions
defaultRealm().transaction {
  realm.add(company)
}
// Querying objects
let companies = objects(Company)
companies[0]?.name // => Realm (generics)
let ftJohns = objects(Employee)
            .filter(.fullTime == true || .name == "John")
```

GITHUB.COM/REALM/REALM-COCOA

This talk is not about...

▶ **Optionals**
▶ **Type safety & inference**
▶ **Enums**
▶ **Tuples**
▶ **Generics**
▶ **Closures**

MONADS/MONOIDS

FUNCTORS/TENSORS

SKI COMBINATOR CALCULUS

FLUX CAPACITORS

GREAT SCOTT!

HOW DO I EVEN...

HOW DO I EVEN...
GET STARTED?

HOW DO I EVEN... KNOW WHEN TO USE IT?

$$\phi_{i+1} = \phi_i + \frac{2}{\Delta x} u_{i+1}$$

$$u^{n+1} - u^n$$

$$\begin{pmatrix} A & G \\ D & 0 \end{pmatrix} \begin{pmatrix} u^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} r^n \\ 0 \end{pmatrix} + \begin{pmatrix} bc_1 \\ bc_2 \end{pmatrix}$$

$$\nabla \cdot u = 0$$

$$\nabla^2 \phi = \nabla \cdot \dot{u}$$

$$\frac{p_{i+1} - 2p_i + p_{i-1}}{\Delta x^2}$$

$$A u^* = r^n + bc_1$$

$$A u^{n+1} + G$$

$$u_{solid} = u_{fluid} \qquad u^* = u^{n+1}$$

$$u_i$$
$$u_{i+1}$$
$$u_{i-1}$$

$$\nabla^2 p = \nabla (u \cdot \nabla u + \nu \nabla^2 u)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

$$\begin{pmatrix} A & \Delta t\, G \\ D & 0 \end{pmatrix} \begin{pmatrix} u^{n+1} \\ \end{pmatrix} = \begin{pmatrix} r^n + bc_1 \end{pmatrix}$$

$$\frac{u_0 - u_b}{\Delta x} + \frac{v_0 - v_b}{\Delta y} = 0$$

$$\frac{\partial p}{\partial n} = 0$$

$$\nabla^2 p = 0$$

$$u^* = u^{n+1}$$

WHEN TO USE SWIFT

▶ New apps

▶ Personal projects

▶ Scripts

▶ Bribe your boss to use it in production*

* I am not liable 😃

SWIFT IN PRODUCTION

▸ Large companies are doing it (twitter & Facebook)

▸ Contractors are doing it

▸ I'm doing it

▸ But be careful if you do it

HOW DO I EVEN...
MANAGE DEPENDENCIES?

INSTEAD OF THIS

COCOAPODS

THINK OF THIS

‹GIT SUBMODULES›

HOW DO I EVEN...
INTERACT WITH COCOA API'S?

Many API's have been "audited for optional conformance"

PORTING UIKIT CODE IS VERY STRAIGHTFORWARD

```swift
func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return items.count
}

func tableView(tableView: UITableView,
  cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell {
    var cell = tableView.dequeueReusableCellWithIdentifier("cell") as UITableViewCell

    cell.textLabel.text = items[indexPath.row]

    return cell
}
```

Though it's possible to replace [square brackets] with (parentheses) and call it Swift, it's by no means where it ends

```
21  func mallocfree() {
22      let ptr = malloc(42)                                                    "UnsafeMutablePointer(0x7FCCB960B1C
23      free(ptr)
24  }
25  mallocfree()
26
27  func mallocarray() {
28      let ptr = UnsafeMutablePointer<Int8>(malloc(42))                        "UnsafeMutablePointer(0x7FCCB9703CA
29      ptr[0] = 99                                                             99
30      ptr[0]                                                                  99
31      free(ptr)
32  }
33  mallocarray()
34
35  func realloctest() {
36      var ptr = UnsafeMutablePointer<Int8>(malloc(42))                        "UnsafeMutablePointer(0x7FCCB9703CA
37      ptr[0] = 99                                                             99
38      ptr = UnsafeMutablePointer<Int8>(realloc(ptr  3                         "UnsafeMutablePointer(0x7FCCB94311C0
39      ptr[0] // 99                                                            99
40      free(ptr)
41  }
42  realloctest()
43
44  func          memc                                                         0, 0, 0,   0, 0, 0, 0,    0, 0, 0,
45      var   fer    [Int   count   1024,  m  at   value                       0, 0, 0,   0, 0, 0, 0,    0, 0, 0,
46      get    an   &buff    UInt   uffer.c   t   ))
47      ut    uffe
48  }
49  h   nam
50
51  f    memc yte    {
52      var  l
53      var buf = [Int8](count: sizeofValue(val), repeatedValue: 0)             [0, 0, 0, 0, 0, 0, 0, 0]
54      memcpy(&buf, &val, UInt(buf.count))                                     "UnsafeMutablePointer(0x7FCCB9432540
55      buf                                                                     [42, 0, 0, 0, 0, 0, 0, 0]
56  }
57  memcpytest()
58
59  func mmaptest() {
60      let ptr = UnsafeMutablePointer<Int>(mmap(nil, UInt(getpagesize()), PROT_READ | PROT_WRITE, MAP_ANON | MAP_PRIVATE,   "UnsafeMutablePointer(0x1185FD000)"
            -1, 0))
61      ptr[0] = 3                                                              3
62      ptr[0] // 3                                                             3
63      munmap(ptr, UInt(getpagesize()))                                       0
64  }
65  mmaptest()
```

HOW DO I EVEN...
INTERACT WITH C?

```swift
var aString = "Portland"
withUnsafePointer(&aString) { (arg: UnsafePointer<String>) in
    println("Hello " + arg.memory) // => Hello Portland
}
```

DON'T

Wrap C++ in Objective-C++ or C to call it from Swift

CloudKit Framework Reference > CKAsset Class Reference

Search iOS Developer Library
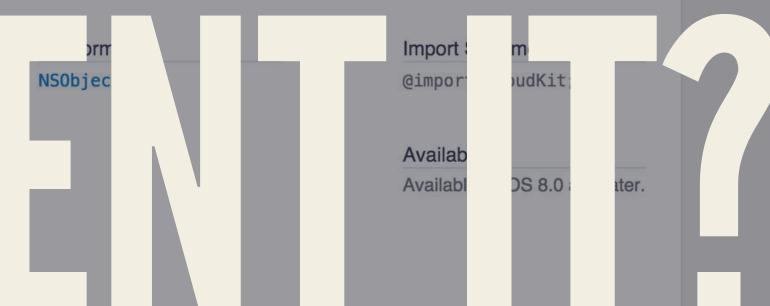
Language: Swift Obj-C Both   On This Page ⌄   Options ⌄

CKAsset

A CKAsset object represents a large file associated with a record. Use asset objects to incorporate external files—such as image, sound, video, text, and binary data files—into your app's records. You can also use assets in places where you want to pass in a file that's more than a few kilobytes in size. To associate an asset with a record, assign it as the value of one of the record's fields. **More...**

HOW DO I EVEN... DOCUMENT IT?

Inheritance   NSObject

NSObject

CKAsset

Import S...

@import CloudKit;

Availab...

Available... OS 8.0 a... ater.

Initializing an Asset

– initWithFileURL:

Getting the URL of the Asset

fileURL

Feedback

JAZZY ♩♫

GITHUB.COM/REALM/JAZZY

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

```objc
//
//  JAZMusician.h
//  JazzyApp
//

#import <Foundation/Foundation.h>

/**
 JAZMusician models, you guessed it... Jazz Musicians!
 From Ellington to Marsalis, this class has you covered.
 */
@interface JAZMusician : NSObject

/**
 The name of the musician. i.e. "John Coltrane"
 */
@property (nonatomic, readonly) NSString  *name;

/**
 The year the musician was born. i.e. 1926
 */
@property (nonatomic, readonly) NSUInteger birthyear;

/**
 Initialize a JAZMusician.
 Don't forget to have a name and a birthyear.

 @warning Jazz can be addicting.
 Please be careful out there.

 @param name      The name of the musician.
 @param birthyear The year the musician was born.

 @return          An initialized JAZMusician instance.
 */
- (instancetype)initWithName:(NSString *)name birthyear:(NSUInteger)birthyear;

@end
```

Language: Swift Obj-C **Both**

JAZMusician

JAZMusician models, you guessed it... Jazz Musicians! From Ellington to Marsalis, this class has you covered.

Inheritance	Conforms To	Import Statement
└○ JAZMusician	NSObject	@import Foundation;

Methods

– initWithName:birthyear:

Initialize a JAZMusician. Don't forget to have a name and a birthyear.

Declaration

```swift
SWIFT

init(name name: String!,
birthyear birthyear: Int)
```

```objc
OBJECTIVE-C

- (instancetype)initWithName:(NSString *)name birthyear:(NSUInteger)birthyear;
```

Parameters

name	The name of the musician.

HOW DO I EVEN...

DO EVIL THINGS WITH IT?

HOW DO I EVEN...
CALL PRIVATE FUNCTIONS?

nm -a libswiftCore.dylib

_swift_stdlib_NSObject_isEqual
_swift_stdlib_NSStringHasPrefixNFD
_swift_stdlib_NSStringHasSuffixNFD
_swift_stdlib_NSStringNFDHashValue
_swift_stdlib_atomicCompareExchangeStrongPtr
_swift_stdlib_compareNSStringDeterministicUnicodeCollation
_swift_stdlib_conformsToProtocol
_swift_stdlib_demangleName
_swift_stdlib_dynamicCastToExistential1
_swift_stdlib_dynamicCastToExistential1Unconditional
_swift_stdlib_getTypeName

```swift
@asmname("swift_demangleSimpleClass")
func demangleSimpleClass(mangledName: UnsafePointer<Int8>,
    inout moduleName: Int,
    inout className: Int) -> Bool
```

LINKS (🍎)

▸ <u>Official Swift blog</u>

▸ <u>The Swift Programming Language Book</u>

▸ <u>WWDC Videos</u>

▸ <u>WWDC Sample Code</u>

▸ <u>Xcode 6</u> (and other resources)

Free Apple Developer Account Required

LINKS (!🍎)

▸ **This talk: github.com/jpsim/talks**

▸ **Other Swift talks: realm.io/news**

▸ **Airspeed Velocity: airspeedvelocity.net**

▸ **ObjC/Swift doc generator: github.com/realm/jazzy**

▸ **Russ Bishop: Horrible things in Swift**

▸ **Mike Ash's Swift/C Playground from NSSpain: Zip Download**

LINKS (!)

▸ nshipster.com

▸ objc.io

▸ Functional Programming in Swift

THANK YOU!

Meetup().questions?.askThem!!
JP SIMARD, @SIMJP, REALM.IO