



dot **Swift**

Tips and Tricks for Swift Dot Swift 2016

Adam Gask
March 2016

Overview

- Dependancy Management - left-pad (Gate?)
- Closures & Error Handling - Live Coding Demo
- Swift 2.2 - All the deprecation warnings
- Swift 3 & Roadmap - All the breaking changes
- Swift Stack
- Pizza! -  + 
- Dot Swift Recap

Dependancy Management

- left-pad (Gate?)



Dependancy Management



Dependancy Management

- TLDR:
 - Azer Koçulu
 - Naming issue with Azer's 'Kik' npm package
 - Legal issues were raised
 - 'Kik' module was transferred to Kik
 - left-pad was removed (along with 272 others)

And that's how JavaScript app development works in 2016. ®

Dependancy Management



A screenshot of a code editor showing two files: 'leftpad.js' and 'package.json'. The 'leftpad.js' file contains the following code:

```
1 module.exports = leftpad;
2 function leftpad (str, len, ch) {
3     str = String(str);
4     var i = -1;
5     if (!ch && ch !== 0) ch = ' ';
6     len = len - str.length;
7     while (++i < len) {
8         str = ch + str;
9     }
10    return str;
11 }
```

The 'package.json' file is partially visible at the top of the editor window.

Dependancy Management

- Cocoapods
- Carthage

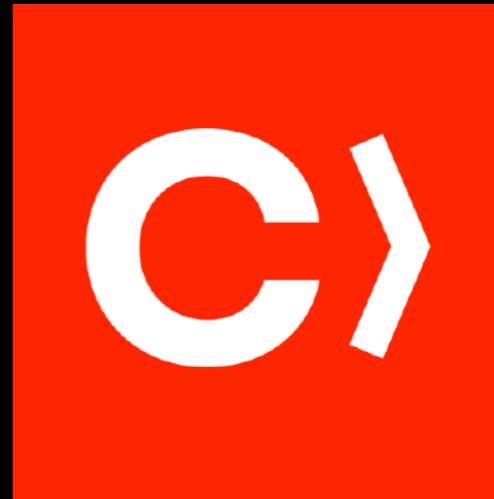


Dependancy Management

RANK	DESCRIPTION	STARS	OS	RELEASES
1	Font-Awesome by <i>FontAwesome</i> <i>The iconic font designed for Bootstrap. Contains only the official font files directly from Font Awesome.</i>	39498	iOS	2
2	react-native by <i>facebook</i> <i>Build high quality mobile apps using React.</i>	26919	iOS	31
3	AFNetworking by <i>AFNetworking</i> <i>A delightful iOS and OS X networking framework.</i>	23002	iOS + OSX	54
4	Alamofire by <i>Alamofire</i> <i>Elegant HTTP Networking in Swift</i>	14433	iOS + OSX	30
5	pop by <i>facebook</i> <i>Extensible animation framework for iOS and OS X.</i>	14226	iOS + OSX	10
6	SDWebImage by <i>rs</i> <i>Deprecated pod</i>	12861	iOS	2
7	ReactiveCocoa by <i>ReactiveCocoa</i> <i>A framework for composing and transforming streams of values.</i>	12528	iOS + OSX	68
8	GPUImage by <i>BradLarson</i> <i>OpenGL ES 2-based image and real-time camera filters for iOS.</i>	11454	iOS	9
9	MBProgressHUD by <i>jdg</i> <i>An iOS activity indicator view.</i>	10102	iOS	7

Dependancy Management

- Cocoapods
- Carthage
- Checking all dependancies into source control
- Just write all your own code... right? 😊



Closures & Error Handling

“Closures are self-contained blocks of functionality that can be passed around and used in your code. Closures in Swift are similar to blocks in C and Objective-C and to lambdas in other programming languages.”

The Swift Programming Language (Swift 2.1)

Closures & Error Handling

- Live Coding Demo

Closures & Error Handling

Really useful links:

- How Do I Declare a Closure in Swift? (NSFW URL, but content is fine)
- Swift closures cheatsheet
- Swift Closures—Everyday Gems
 - Part 1
 - Part 2

Swift 2.2

- Raft of non-breaking changes
- A gentle preparation for 3.0
- Linter warnings to help you on your way
- 2.2 is the time to fix your code, not 3.0
- Still bugs to fix - [JIRA](#) (Look for the label “StarterBug”)

Swift 2.2

Referencing the Objective-C selector of a method

- Accepted Proposal - [SE-0022](#)

```
class vc: UIViewController {  
  
    let button = UIButton(type: .System)  
  
    func addTarget() {  
  
        button.addTarget(self, action:  
#selector(vc.buttonTapped(_:)),  
forControlEvents: .TouchUpInside)  
    }  
}
```

Swift 2.2

Naming Functions with Argument Labels

- Accepted Proposal - [SE-0021](#)

```
extension MyModel {

    func insertData(data: NSData, at index: Int) {}

    func insertData(data: NSData, aboveData
siblingData: NSData) {}

    func insertData(data: NSData, belowData
siblingData: NSData) {}

}
```

Swift 2.2

Modernizing Swift's Debugging Identifiers

- Accepted Proposal - [SE-0028](#)

`__FILE__` => `#file`

`__LINE__` => `#line`

`__COLUMN__` => `#column`

`__FUNCTION__` => `#function` (Added during review)

`__DSO_HANDLE__` => `#dsohandle`

Swift 2.2

Allow (most) keywords as argument labels

- Accepted Proposal - [SE-0001](#)

Allow the use of all keywords except `inout`, `var`, and `let` as argument labels.

```
for i in 1.stride(through: 9, by: 2) {  
    print(i)  
}  
  
func find(data: NSData, in array: [NSData]) {  
}
```

Swift 2.2

Tuple comparison operators

- Accepted Proposal - SE-0015

```
let singer = ("Taylor", "Swift")
```

```
let alien = ("Justin", "Bieber")
```

```
if singer == alien {
```

```
    print("Matching tuples!")
```

```
} else {
```

```
    print("Non-matching tuples!")
```

```
}
```

Swift 2.2

Arrays and other slice types now have `removeFirst()`

This complements the existing `removeLast()`

```
var array = Array(1...10)  
  
array.removeFirst()  
  
for number in array {  
  
    print("\((number) green bottles")  
  
}
```

Swift 2.2

- Remove the ++ and -- operators - SE-0004
- Removing var from Function Parameters -
SE-0003
- Remove C-style for-loops with conditions and incrementers

Say goodbye to...

```
for(var a = 0; a <10 ; a++)
```

Swift 3.0 & Roadmap

What we know is coming:

- Stable ABI - “guarantee a level of binary compatibility moving forward”
- Complete generics - an array of Equatable elements is Equatable
- Focus and refine the language

Swift 3.0 & Roadmap

Better Translation of Objective-C APIs Into Swift

- Accepted Proposal - [SE-0005](#)

```
let content =  
listItemView.text.stringByTrimming...  
CharactersInSet(NSCharacterSet.whitespaceAnd  
NewlineCharacterSet())
```

```
let content =  
listItemView.text.trimming(.whitespace...  
AndNewlines)
```

Swift 3.0 & Roadmap

Apply API Guidelines to the Standard Library

- Accepted Proposal - SE-0006

-public protocol SequenceType { ... } =>
+public protocol Sequence { ... }

sort() => sorted(), sortInPlace() => sort()

reverse() => reversed()

enumerate() => enumerated()

Swift 3.0 & Roadmap

What we know is out of scope:

- Full source compatibility - there will be breaking changes
- Concurrency
- C++ Interoperability
- Major new library functionality

Did you know?



Possible full stack in Swift?

Swift Stack

Server Options

- Swift Server IO
- Perfect
- Kitura



Maybe...



Core Data
Storage



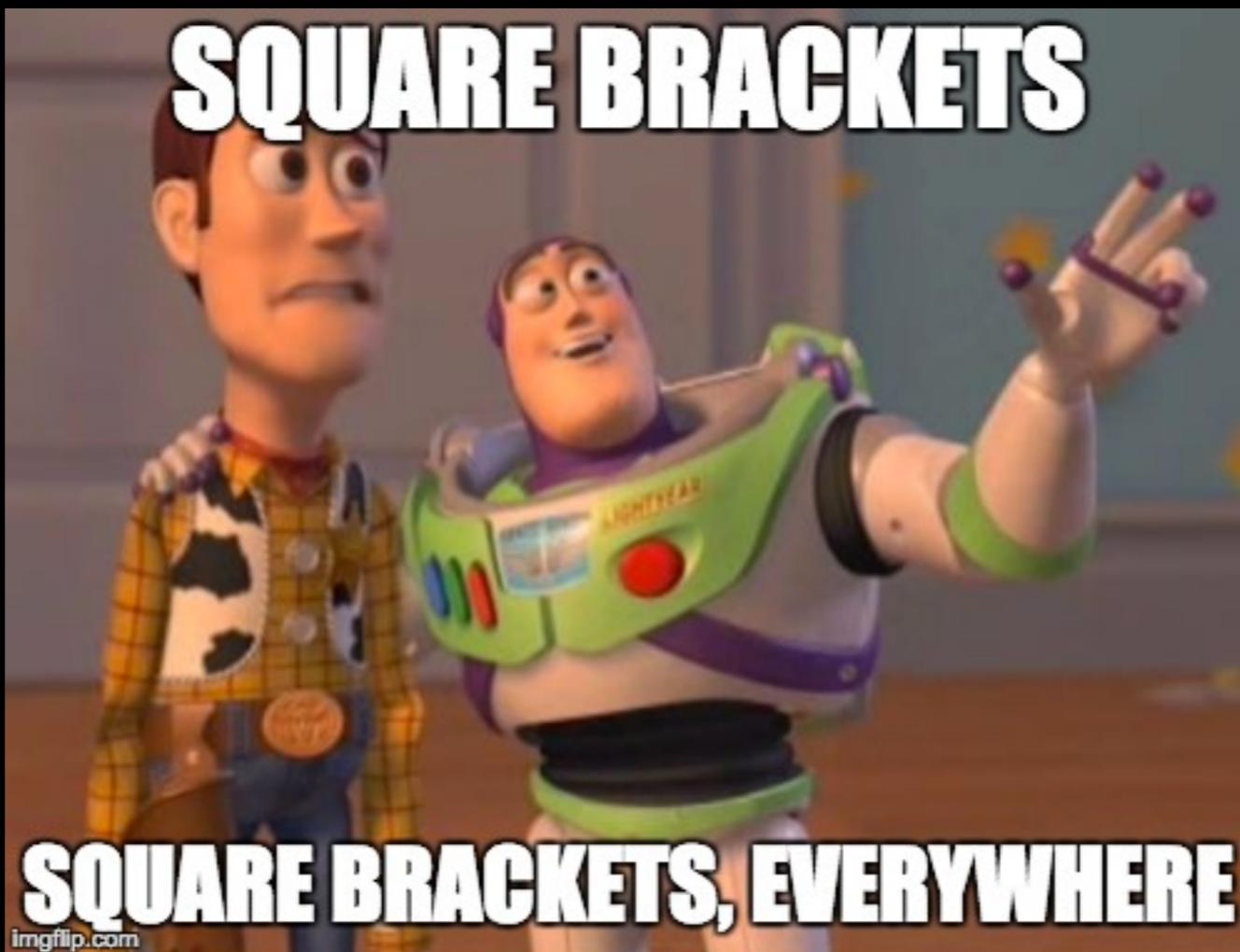
Swift HTTP
Server



UICollectionView
Web UI



Could be worse....





Dot Swift 2016

Paris - 29th January 2016

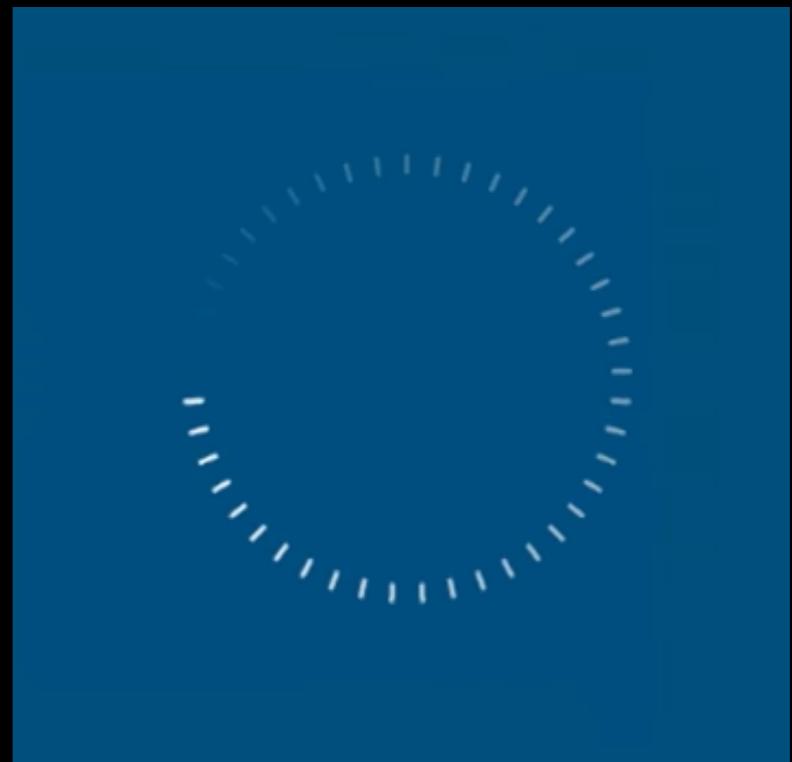
Dot Swift 2016

TLDR:

- Annual European Swift conference
- Packed schedule of 12 talks
- Ranging from UI, Live Coding & “The Bigger Picture”
- Two sessions of 4 lightning speakers
- Each talk had a structured Q&A after

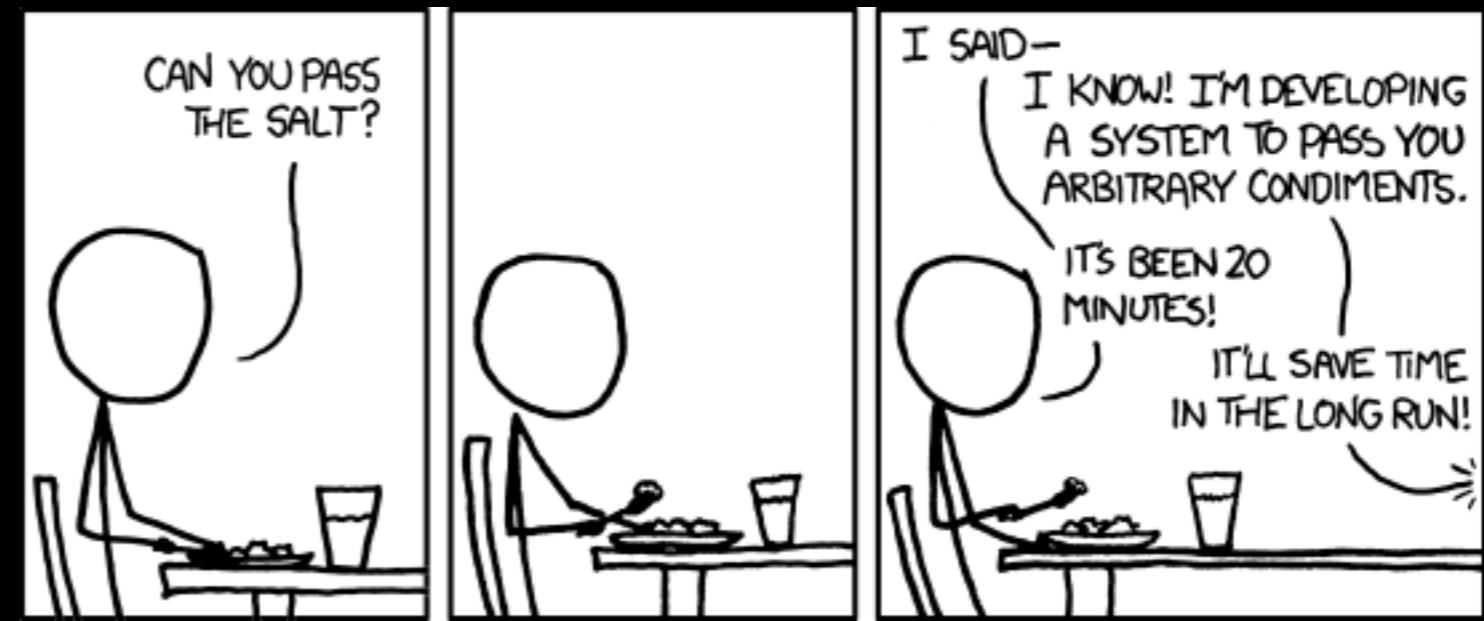
The Talks

- Creating an animation with CAReplicatorLayer
 - Showing that animations can be short and succinct
 - Talk
 - Alessandro Ludovici



The Talks

- The Empathetic Developer
 - Probably most thought provoking talk
 - Talk
 - Michael May



The Talks

- Tiny Networking in Swift

- Easily the best Live

Coding Demo

- Talk
 - Chris Eidhof



The Talks

- Other notable talks:
 - Swift [and the] Evolution - TJ Usiyan
 - CloudKit: The Forgotten Cloud - Paul Ardeleanu
 - iOS UIs built as Lego bricks - Victor Wang
 - Beyond JSON in Swift - Maxim Zaks
 - Going Swift and Beyond - Ayaka Nonaka

Featured Talk

- Localization is hard
 - Lots of takeaways popular & obscure
 - Talk
 - Roy Marmelstein



And Finally

- Protocol Oriented Programming
 - Best use of cartoon characters
 - Talk
 - Grégoire Lhotellier



Thank you for your time

Adam Gask
 @_AJ9