

# **Taken For Granted**

**@basthomas**

**try! Swift NYC, 2018**

**More than eight years  
ago...**


initial swift test

Swift SVN r3

🔗 master

🔗 type-name-lookup-fail

⋮ autoclosure-dragon

 **lattner** committed on Jul 18, 2010

0 parents

commit 18844bc65229786b96b89a9fc7739c0fc897905e

Browse files

Showing 0 changed files with 0 additions and 0 deletions.

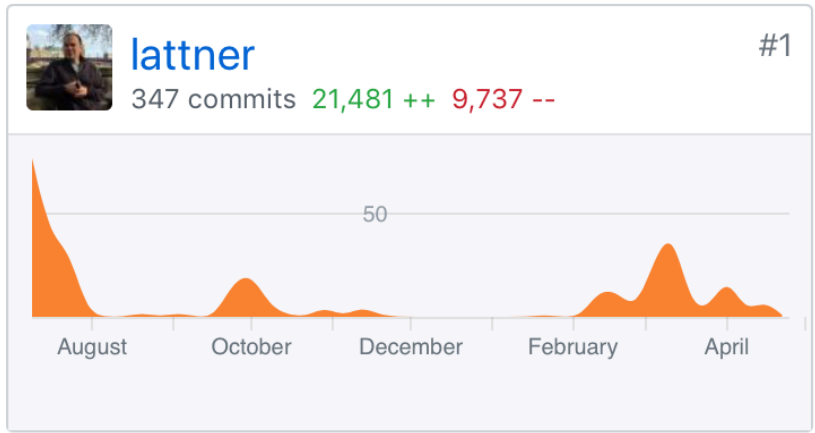
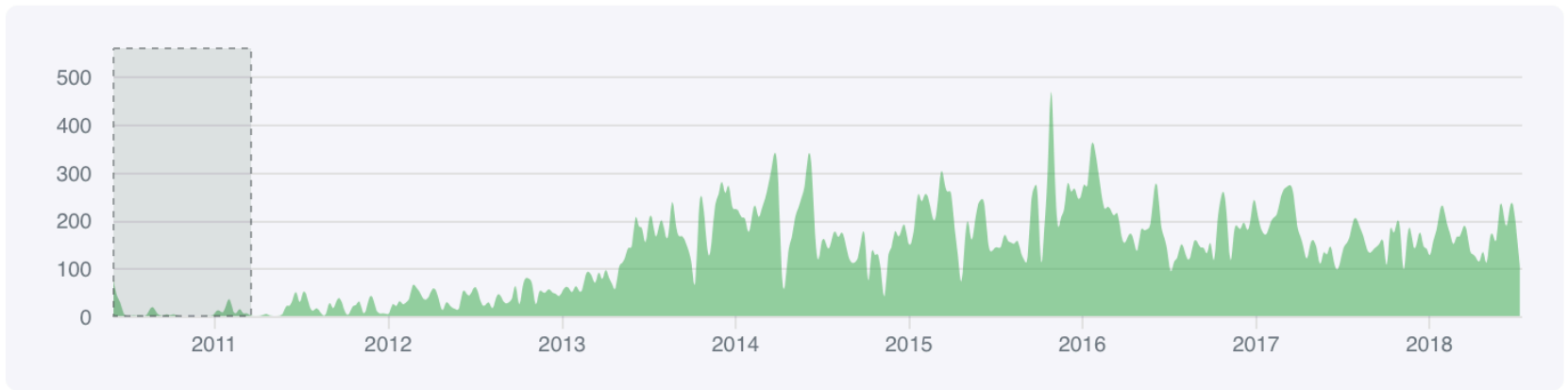
Unified Split

295 comments on commit 18844bc

Jul 18, 2010 – May 3, 2011

Contributions: Commits ▾

Contributions to master, excluding merge commits



# A history

# July 10, 2013

## 327 days until announcement

***nil is now more context aware***

```
// before, nil was an NSObject
```

```
var one = NSString(1)
```

```
one = nil as! NSString
```

```
// now, it picks up the object type
```

```
one = nil
```

# July 17, 2013

## 320 days until announcement

### switch *on more than integers*

```
var coordinate: (Int, Int) = (1, 1)
switch(coordinate) {
case (0, 0):
    println("origin")
case (_, 0):
    println("on the x-axis")
case (var x, var y) where x == y:
    println("on the y = x diagonal")
case (-10..10, -10..10):
    println("close to the origin")
}
```

# July 31, 2013

## 306 days until announcement

### *Underscores in literals*

```
var billion = 1_000_000_000
```

```
var hex = 0x7FFF_FFFF_FFFF_FFFF
```

# August 14, 2013

## 292 days until announcement

weak *and* unowned *variables*

```
var [weak] delegate = NSObject()  
var [unowned] parent: Parent
```



# August 28, 2013

## 278 days until announcement

this *is now* self

```
func mutate() {  
    this.id = 123  
}
```

```
func mutate() {  
    self.id = 123  
}
```

# August 28, 2013

## 278 days until announcement

### *Unions*

```
union HTMLTag {  
    case A(href: String)  
    case IMG(src: String, alt: String)  
    case BR  
}
```

```
var img: HTMLTag = .IMG(src: "https://www.tryswift.co/events/2018/nyc/",  
                        alt: "A conference about Swift in New York City")
```

# September 4, 2013

## 271 days until announcement

### *No implicit construction*

```
// before  
var number: Int // fails, needs to be set explicitly
```

```
// now  
var number: Int  
if someBoolean {  
    number = 1  
} else {  
    number = 0  
}
```

# September 17, 2013

## 258 days until announcement

### *Constructor changes*

```
class Conference: NSObject {  
    init withYear(i: Int) place(s: String) {  
        super.init()  
    }  
}
```

```
Y(withYear: 2018, place: "New York City")
```

# September 17, 2013

## 258 days until announcement

### *Generic unions with single payload*

```
union State<DataType> {  
    case Loading  
    case Data(DataType)  
    case Empty  
}
```

# September 24, 2013

## 251 days until announcement

### Name changes & *Optionals*

```
enum Optional<T> {  
    case Some(T), None  
}
```

```
// error: 'weak' variable should have  
// optional type 'NSObject?'  
[weak] var object: NSObject
```

# October 2, 2013

## 243 days until announcement

### byref *to* inout, *optional chaining*

```
func changing(x: [inout] Int) {  
    x = 10  
}  
var x = 0  
changing(x: &x) // `var x` is now 10  
  
// `?` now permitted as a postfix operator  
// we all know what that means...  
(x ? y : z)
```

# October 2, 2013

## 243 days until announcement

byref **to** inout, ***optional chaining***

```
x as T // returns `Optional<T>`
```

```
textView.text?.count
```

```
var (_, reason) = httpStatus
```



# October 9, 2013

## 236 days until announcement

### *Enums with raw types*

```
enum TrySwift: String {  
    case NYC = "New York City"  
    case Tokyo = "Tokyo"  
    case Bangalore = "Bangalore"  
}
```

# October 23, 2013

## 222 days until announcement

### *Rewriting* Vector<T>

```
// error: Missing return in a function expected to return 'Int'
```

```
func randomInt() -> Int { }
```

```
var numbers: Array<Int> = [1, 2, 3]
```

```
var rect = Point(x: 0, y: 0)
```

```
switch rect {
```

```
case Rect(x: 0, y: 0):
```

```
    println("That is not soo big")
```

```
default:
```

```
    println("Not today")
```

```
}
```

# November 6, 2013

## 208 days until announcement

### *Swift is now Ruby*

```
def make_beverage  
    "Coffee"  
end
```

```
def makeBeverage() -> String {  
    return "Coffee"  
}
```

# November 13, 2013

## 201 days until announcement

### *Enum bridging & static properties*

```
typedef NSInteger, UITableViewCellStyle) {  
    UITableViewCellStyleDefault, UITableViewCellStyleValue1,  
    UITableViewCellStyleValue2, UITableViewCellStyleSubtitle  
};
```

```
enum UITableViewCellStyle: Int {  
    case Default, Value1, Value2, Subtitle  
}
```

```
struct Foo { static var bar: Int = 0 }  
println(Foo.bar)
```

# November 20, 2013

## 194 days until announcement

### *Swift is now Swift again & improved compiler support*

```
func makeBeverage() -> String {  
    return "Coffee"  
    println("Tea!") // Code after 'return' will never be executed  
}
```

# December 11, 2013

## 173 days until announcement

### *Binary operator precedence & property initialization checks*

```
@interface CustomObject: NSObject  
+ (id) sharedInstance;  
@end
```

```
// `AnyObject`, before `DynamicLookup`  
CustomObject.sharedInstance
```

# December 11, 2013

## 173 days until announcement

### *Binary operator precedence & property initialization checks*

- Multiplicative:  $*$ ,  $/$ ,  $\%$ ,  $\&$
- Additive:  $+$ ,  $-$ ,  $|$ ,  $^$
- Comparative:  $==$ ,  $!=$ ,  $<$ ,  $<=$ ,  $>=$ ,  $>$
- Conjunctive:  $\&\&$
- Disjunctive:  $||$

# December 11, 2013

## 173 days until announcement

### *Binary operator precedence & property initialization checks*

```
struct TryConference: Conference {  
    var officeHours: Bool  
  
    init() {  
        super.init()  
        // error: property `self.officeHours` not initialized at `super.init` call  
    }  
}
```



**December 25, 2013** 

**159 days until announcement**

Array.map

```
let array = [1, 2, 3, 4]  
array.map(String.init) // ["1", "2", "3", "4"]
```

# January 8, 2014

## 145 days until announcement

### static *is now* type & init *delegation*

```
class A { }
class B : A {
    static var bar = "Bar"
    var title: String

    init() {
        self.init(withTitle: "My Title")
    }

    init withTitle(title: String) {
        self.title = title
        super.init()
    }
}
```

# January 22, 2014

## 131 days until announcement

### *Immutable properties and value types*

```
struct Coffee {  
    let bean: Bean  
}
```

# January 29, 2014

## 124 days until announcement

### *Property observing*, `OptionSet` *improvements*

```
var isEnabled: Bool {  
    willSet(value):  
        println("Changing from \(isEnabled) to \(value)")  
    didSet: self.needsDisplay = true  
}
```

# February 5, 2014

## 117 days until announcement

### *Conditional variable binding & executables*

```
if let value = optValue {  
    println(value)  
}
```

```
# creates an executable  
$ swift main.swift
```

# February 12, 2014

## 110 days until announcement

***New message send syntax, let is now val & range changes***

```
var mutable = 1  
val immutable = 2
```

```
1...3 // 1, 2
```

```
1..3 // 1, 2, 3
```

# February 19, 2014

## 103 days until announcement

type *is now* static *and* class

```
struct Foo {  
    class func bar() {}  
    static var baz: Int = 1  
}
```

# February 26, 2014

## 96 days until announcement

override *attribute required* & `let`, *compiler directives*

```
class X {  
    func foo() { }  
}
```

```
class Y: X {  
    let z = 0  
    override func foo() { }  
}
```



# February 26, 2014

## 96 days until announcement

override *attribute required* & let, *compiler directives*

```
#if os(OSX)
import AppKit
#else
import UIKit
#endif
```

# March 5, 2014

## 89 days until announcement

### *Designated and convenience initializers, implicitly conforming*

```
struct TryConference {  
    let city: String  
    let year: Int  
  
    init(city: String, year: Int) {  
        self.city = city  
        self.year = year  
    }  
  
    init(city: String) {  
        self.init(city: city, year: 2018)  
    }  
}
```

# March 5, 2014

## 89 days until announcement

***Designated and convenience initializers, implicitly conforming***

```
enum DeviceType { // Conforms to `Equatable`, `Hashable`  
    case Phone, Pad  
}
```

```
DeviceType.Phone == DeviceType.Pad  
let dict: Dictionary<DeviceType, String> = [.Phone: "iPhone"]
```

# March 19, 2014

## 75 days until announcement

### *Defaults*

```
let stringArray: String[] = []  
let someDictionary: Dictionary<String, String> = [:]
```

# April 2, 2014

## 61 days until announcement

### *Changed call syntax, @final*

```
NSColor colorWithRed(r) green(g) blue(b) alpha(a)
```

```
UIColor.init withRed(r) green(g) blue(b) alpha(a)
```

```
NSColor colorWithRed(r, green: g, blue: b, alpha: a)
```

```
UIColor(withRed: r, green:g, blue:b, alpha: a)
```

```
@final class HammerTime { }
```

# April 9, 2014

## 54 days until announcement

### *Dictionary Elements are tuples*

```
let dict = ["one": 1, "two": 2]
for (k, v) in dict {
    println(k, v)
}
```

# May 7, 2014

## 26 days until announcement

### *Floating point ranges and convenience init changes*

1.0...2.0

```
class Foo {  
    init(x: Int) {}  
    init() -> Self { self.init(42) }  
}
```

```
class Foo {  
    init(x: Int) {}  
    convenience init() { self.init(42) }  
}
```

# May 9, 2014

## 20 days until announcement

### *Keywords in functions*

```
func mixColorWithRed(red: Float, green: Float, blue: Float) { }  
// error: missing argument labels 'green:blue:' in call  
mixColorWithRed(r, g, b)
```

```
func mixColor(`red: Float, green: Float, blue: Float) { /* ... */ }  
func mixColorGuess(red: Float, _ green: Float, _ blue: Float) { /* ... */ }  
mixColor(red: r, green: g, blue: b)  
mixColorGuess(r, g, b)
```



# May 13, 2014

## 16 days until announcement

### *Dictionary subscriptions returning optionals, capture lists*

```
dict["one"] // Optional(1)
dict["one"] = nil
dict["one"] // nil
```

```
completion { [weak self] in }
```

```
UIColor(withRed: r)
UIColor(red: r)
```

# May 19, 2014

## 10 days until announcement

### *Closures for functional functions, init over factory methods*

```
numbers.filter { $0 % 2 == 0 }
```

```
let tens = filter(0..50) { $0 % 10 == 0 }
```

```
let tenX = map(tens) { X($0) } // Lazy!
```

```
1...3 // [1, 2, 3]
```

```
1..2 // [1, 2]
```

**Beta 1 was shipped!**

# June 23, 2014

## 84 days until release

### *Range readability, sorting*

```
1..<2 // [1, 2]
```

```
var numbers = [1, 3, 2]
```

```
let sortedNumbers = numbers.sorted()
```

```
numbers.sort() // numbers is now [1, 2, 3]
```

# July 3, 2014

## 74 days until release

### *Array & Dictionary syntactic sugar, escape sequence improvements*

```
let numbers: [Int] // instead of `Int[]`  
let dict: [String: Int] // instead of `Dictionary<String, Int>`  
  
println("\u{123456}") // was `println("\u123456")`
```

# July 21, 2014

## 56 days until release

### *Access Control*

Oh yes...

- public
- internal (default)
- private

# August 4, 2014

## 42 days until release

### *nil coalescing, optional chaining mutation*

```
let empty: [Int] = []  
empty.first ?? 0 // 0
```

```
var sequences = ["fibonacci": [1, 1, 2, 3, 4]]  
sequences["fibonacci"]?[4]++ // Increments element 4 of key "fibonacci"
```

# September 15, 2014

## Release day 🎉

### *Failable inits*

```
enum Foo: Int { case A = 0, B = 1, C = 2 }  
let foo = Foo(rawValue: 2)! // formerly 'Foo.fromRaw(2)!'  
println(foo.rawValue) // formerly 'foo.toRaw()'
```



# Swift 2

- defer
- @available & #available
- Protocol extensions
- Protocol default implementations
- throw, catch
- @testable
- guard

# Swift 3

- Swift open source
- @escaping
- Never
- fileprivate
- id as Any
- Renaming, renaming, renaming...

# Swift 4

- Keypaths
- Conditional Conformance
- Codable

# Swift 5

- ABI Stability
- String ergonomics

# Thanks!!

**@basthomas**