

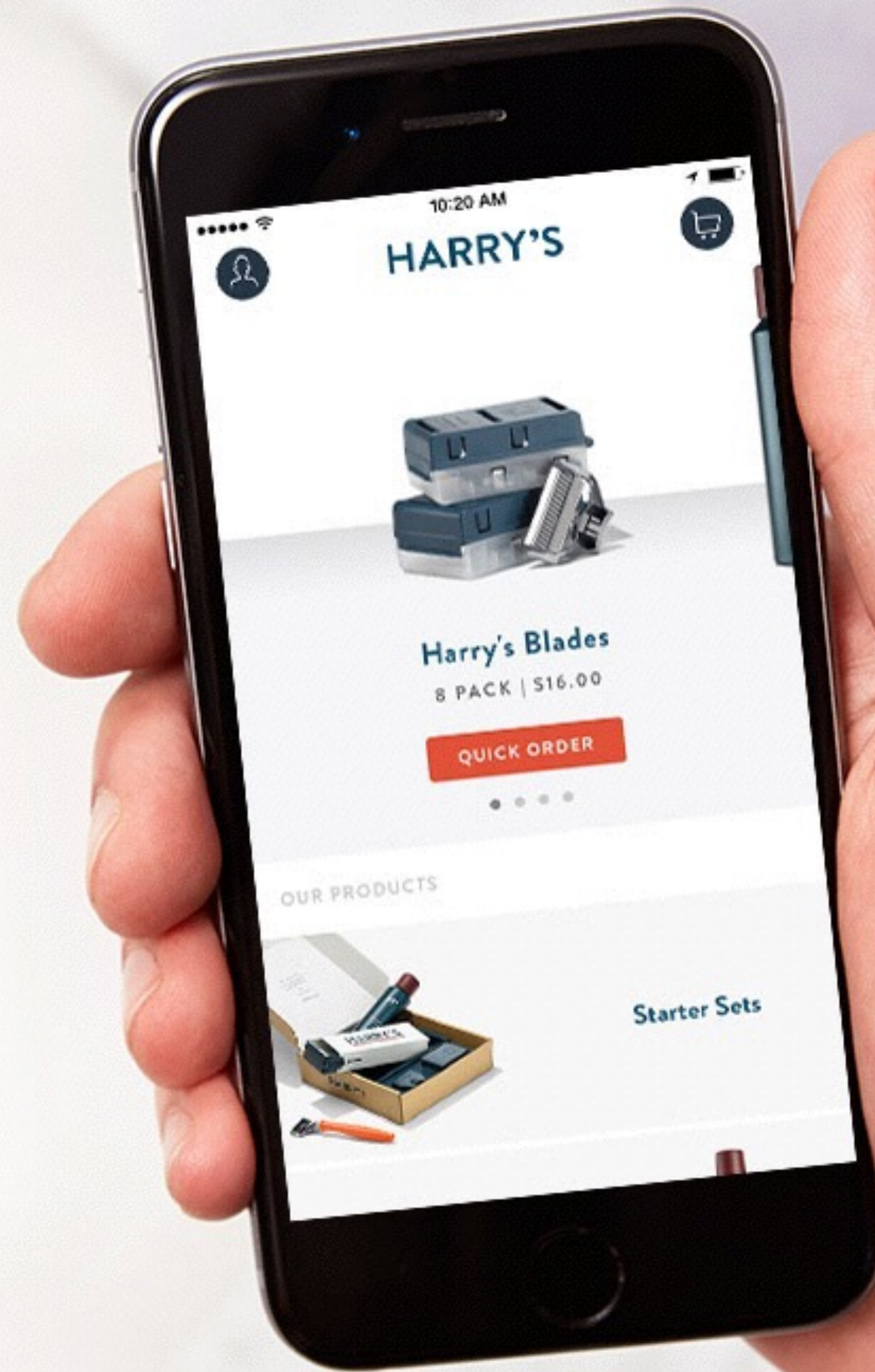
Writing Swift

by @neilkimmett

HARRY'S









Neil Kimmett 2:02 PM



aside: why is there `CGRect.zero` but no `UIEdgeInsets.zero`

2:03 ☆

whats that all about



Soroush Khanlou 2:03 PM

that's wack



Neil Kimmett 2:05 PM

can fix with

```
extension UIEdgeInsets {  
    static var zero: UIEdgeInsets {  
        return UIEdgeInsetsZero  
    }  
}
```

but still

cmon Apple



Soroush Khanlou 2:05 PM

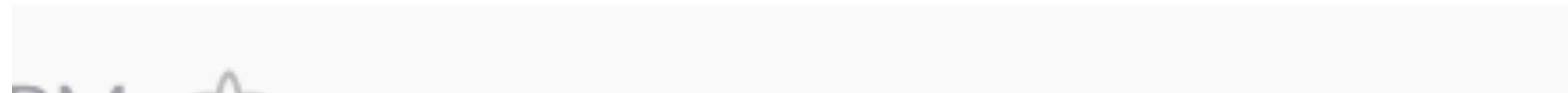
yeah this one's on apple

if you're that kind of person you could make a radar

2:05 PM

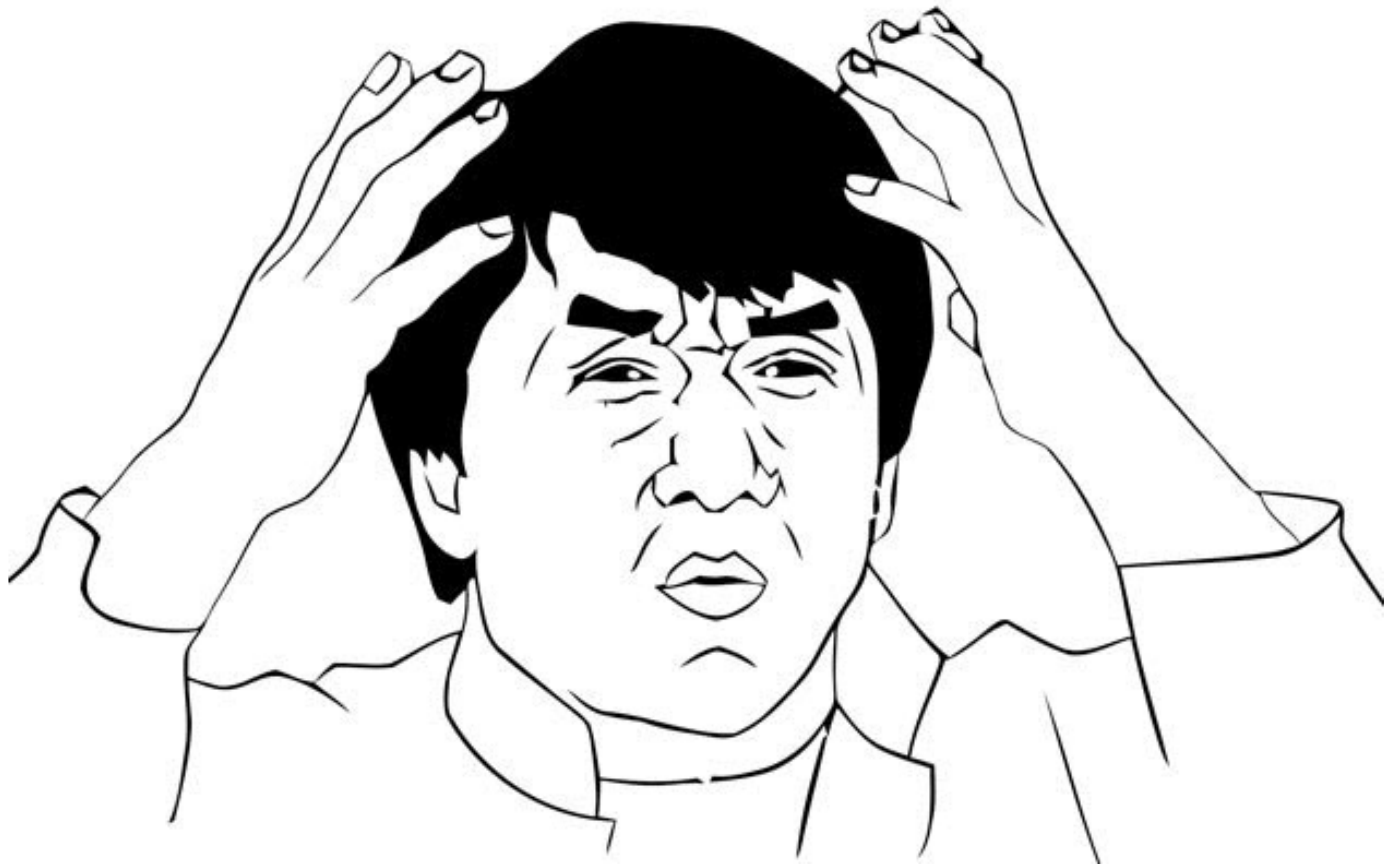
Apple

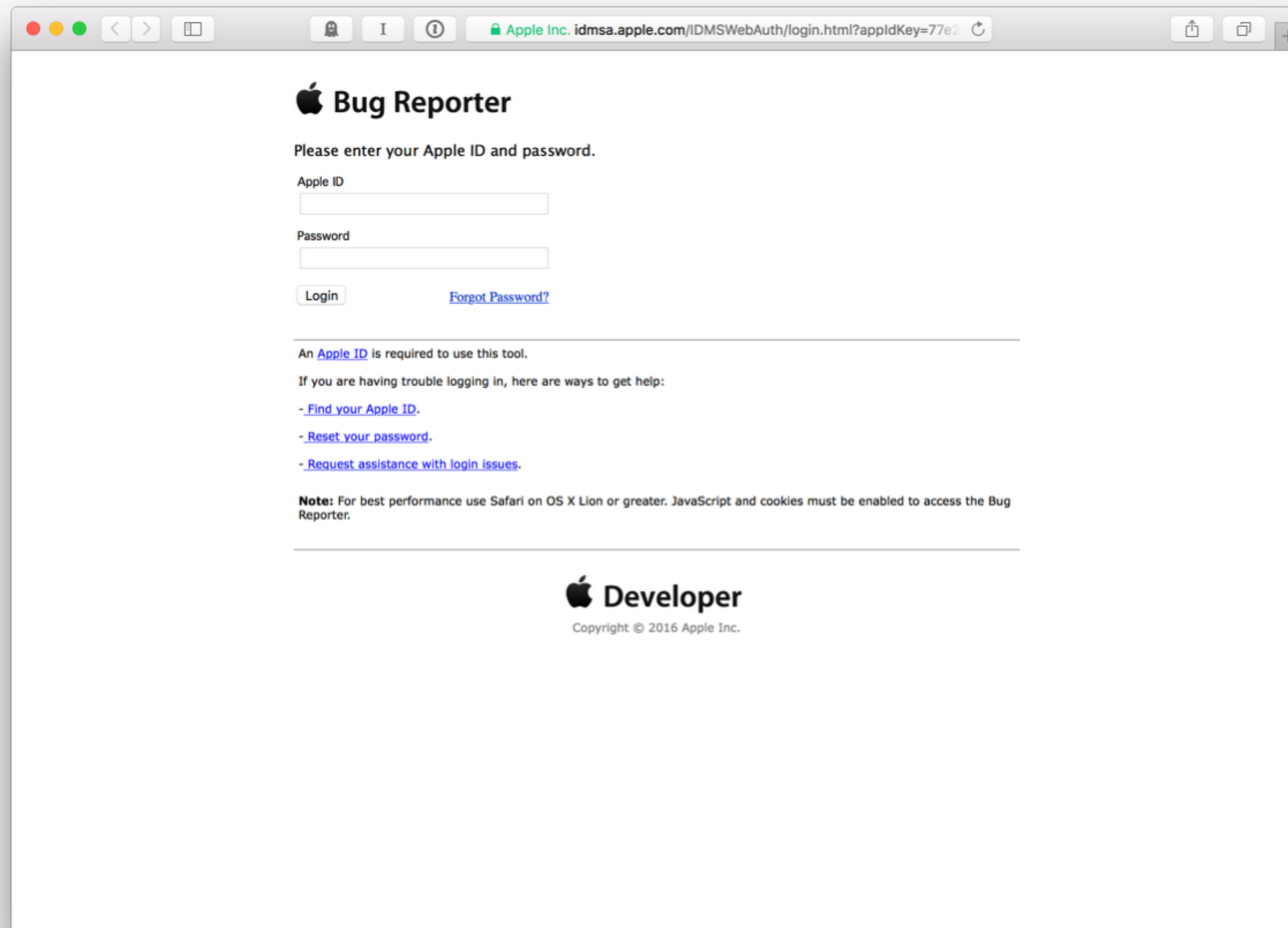
of person you could make a radar



u could make a radar

a radar

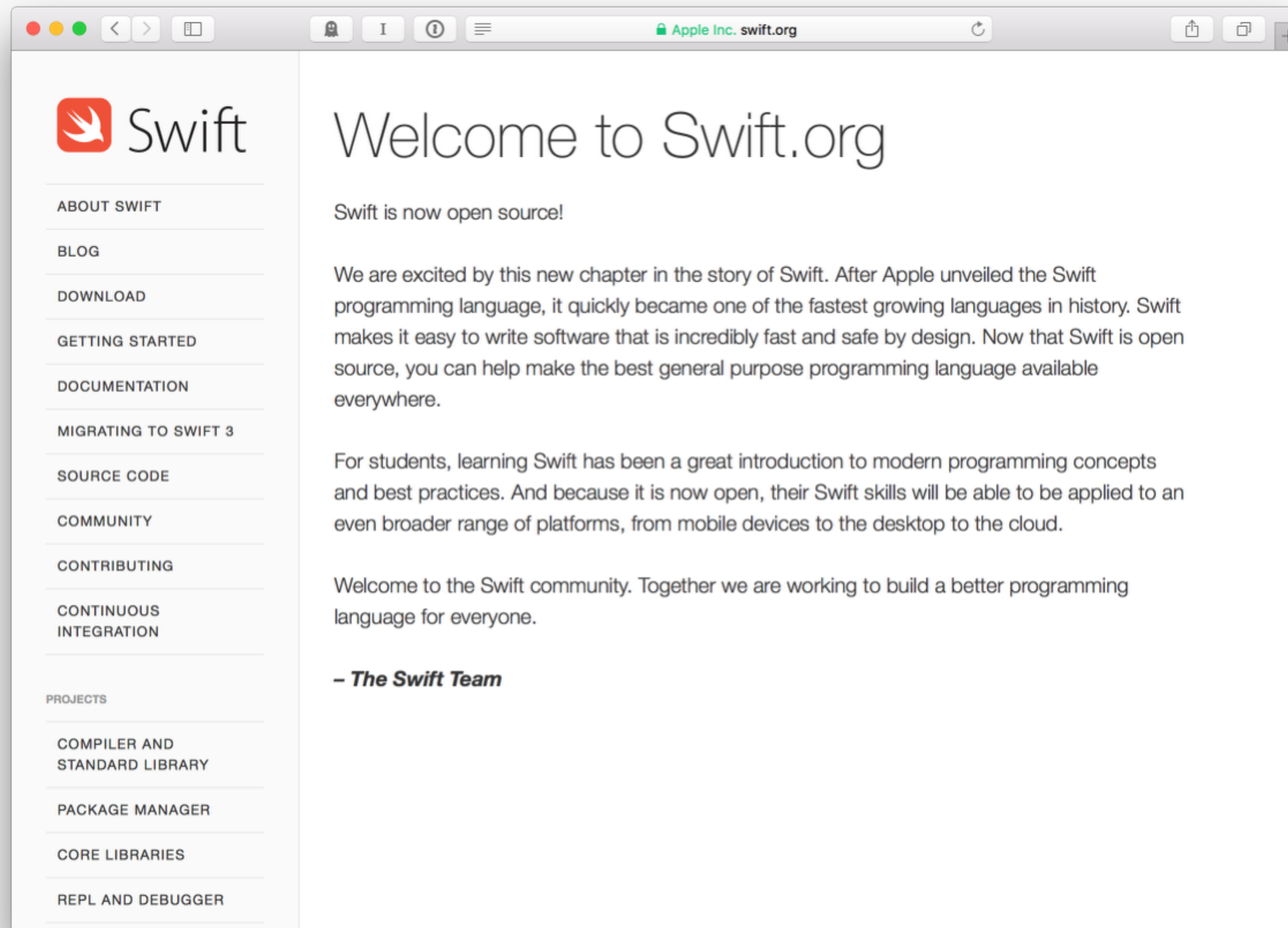




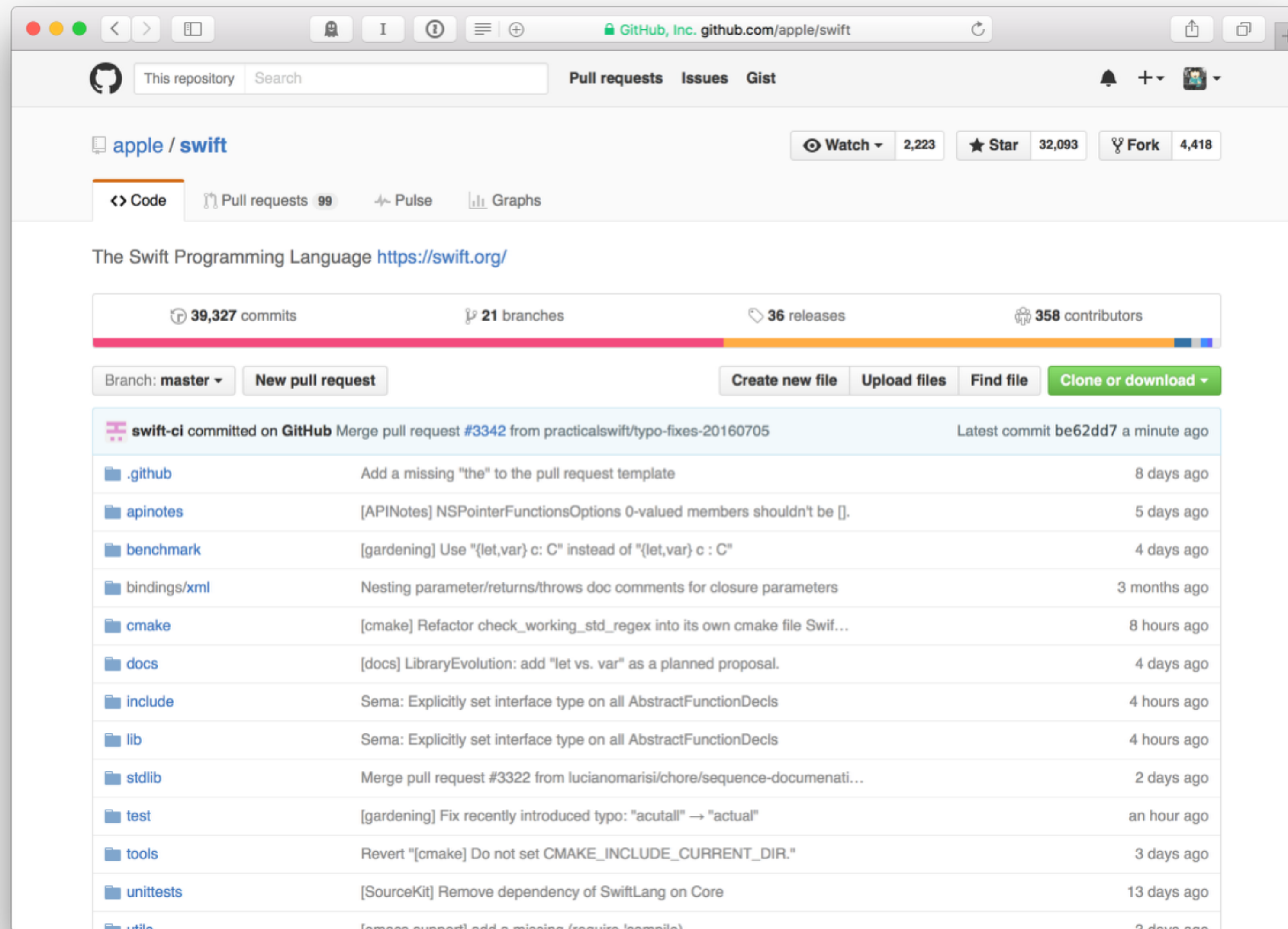
<http://bugreport.apple.com/>

A close-up of Homer Simpson from the animated series 'The Simpsons'. He is shown from the chest up, with his head tilted back and his right hand resting on his chin in a classic 'thinking' pose. His eyes are wide open, and his expression is one of intense concentration or realization. He is wearing a white collared shirt. The background is a simple, stylized interior with a pink wall and a purple door frame. A small, rectangular object, possibly a light switch or a small cabinet, is visible on the wall to the right.

THAT GIVES ME AN IDEA.



<https://swift.org>



<https://github.com/apple/swift>



swift / stdlib / public / SDK / CoreGraphics / **CoreGraphics.swift**

swift / stdlib / public / SDK / CoreGraphics / CoreGraphics.swift

```
12
13 @_exported import CoreGraphics
14 import Darwin
15
16 //===-----
17 // CGGeometry
18 //===-----
19
20 public extension CGPoint {
21     static var zero: CGPoint {
22         @_transparent // @fragile
23         get { return CGPoint(x: 0, y: 0) }
24     }
25
26     @_transparent // @fragile
27     init(x: Int, y: Int) {
28         self.init(x: CGFloat(x), y: CGFloat(y))
29     }
30
31     @_transparent // @fragile
32     init(x: Double, y: Double) {
33         self.init(x: CGFloat(x), y: CGFloat(y))
34     }
35
36     @available(*, unavailable, renamed="zero")
37     static var zeroPoint: CGPoint {
38         fatalError("can't retrieve unavailable property")
39     }
40 }
41
```

swift / stdlib / public / SDK / UIKit / **UIKit.swift**

swift / stdlib / public / SDK / UIKit / UIKit.swift

```
11 // -----
12
13 import Foundation
14 @_exported import UIKit
15
16
17 //===-----
18 // Equatable types.
19 //===-----
20
21 @_transparent // @fragile
22 @warn_unused_result
23 public func == (lhs: UIEdgeInsets, rhs: UIEdgeInsets) -> Bool {
24     return lhs.top == rhs.top &&
25           lhs.left == rhs.left &&
26           lhs.bottom == rhs.bottom &&
27           lhs.right == rhs.right
28 }
29
30 extension UIEdgeInsets : Equatable {}
31
32 @_transparent // @fragile
33 @warn_unused_result
34 public func == (lhs: UIOffset, rhs: UIOffset) -> Bool {
35     return lhs.horizontal == rhs.horizontal &&
36           lhs.vertical == rhs.vertical
37 }
38
39 extension UIOffset : Equatable {}
40
```

```
13  import Foundation
14  @_exported import UIKit
15
16  +//===-----
17  +// UIGeometry
18  +//===-----
19  +
20  +public extension UIEdgeInsets {
21  +    static var zero: UIEdgeInsets {
22  +        @_transparent // @fragile
23  +        get { return UIEdgeInsets(top: 0.0, left: 0.0, bottom: 0.0, right: 0.0) }
24  +    }
25  +}
26  +
27  +public extension UIOffset {
28  +    static var zero: UIOffset {
29  +        @_transparent // @fragile
30  +        get { return UIOffset(horizontal: 0.0, vertical: 0.0) }
31  +    }
32  +}
```

33





swift / test / 1_stdlib / UIKit.swift

```
50 var inset1 = UIEdgeInsets(top: 1.0, left: 2.0, bottom: 3.0, right: 4.0)
51 var inset2 = UIEdgeInsets(top: 1.0, left: 2.0, bottom: 3.1, right: 4.0)
52 print("inset1 == inset1: \(inset1 == inset1)")
53 print("inset1 != inset1: \(inset1 != inset1)")
54 print("inset1 == inset2: \(inset1 == inset2)")
55 // CHECK: inset1 == inset1: true
56 // CHECK: inset1 != inset1: false
57 // CHECK: inset1 == inset2: false
58
```

swift / test / 1_stdlib / UIKit.swift

```
68 +
69 +var inset0 = UIEdgeInsets(top: 0.0, left: 0.0, bottom: 0.0, right: 0.0)
70 +var insetDot0 = UIEdgeInsets.zero
71 +print("inset0 == insetDot0: \(inset0 == insetDot0)")
72 +// CHECK: inset0 == insetDot0: true
73 +
74 +var offset0 = UIOffset(horizontal: 0.0, vertical: 0.0)
75 +var offsetDot0 = UIOffset.zero
76 +print("offset0 == offsetDot0: \(offset0 == offsetDot0)")
77 +// CHECK: offset0 == offsetDot0: true
```


The screenshot shows a GitHub pull request interface. At the top, the title is "Add .zero convenience static vars for UIEdgeInsets and UIOffset #1323". Below the title, it says "Merged" and "jrose-apple merged 3 commits into apple:master from neilkimmett:master on Mar 25". There are tabs for "Conversation" (9), "Commits" (3), and "Files changed" (2). A green bar indicates "+27 -0" changes. The main content area shows a comment from neilkimmett dated Feb 16, discussing the use of UIEdgeInsetsZero and the logical placement of static variables. Below this is a commit entry for "Add .zero convenience static vars for UIEdgeInsets and UIOffset" with commit hash c4d5ccb. Further down, there are two comments from jrose-apple, both dated Feb 16, discussing the need for internal approval at Apple and the choice between print and StdlibUnitest for tests. The right sidebar contains sections for "Labels" (None yet), "Milestone" (No milestone), "Assignees" (jrose-apple), "4 participants" (with avatars), and "Notifications" (Unsubscribe button).

GitHub, Inc. github.com/apple/swift/pull/1323

Add .zero convenience static vars for UIEdgeInsets and UIOffset #1323

Merged jrose-apple merged 3 commits into apple:master from neilkimmett:master on Mar 25

Conversation 9 Commits 3 Files changed 2 +27 -0

neilkimmett commented on Feb 16

Using UIEdgeInsetsZero instead of UIEdgeInsets.zero was annoying.
Copied the style from '/stdlib/public/SDK/CoreGraphics/CoreGraphics.swift'

I think this is the logical place to put all the bits. Was unsure whether to use the `print(...) ... // CHECK:` style tests or just an `assert` so I did both.

1

Add .zero convenience static vars for UIEdgeInsets and UIOffset ... c4d5ccb

jrose-apple self-assigned this on Feb 16

jrose-apple commented on Feb 16 Apple member

Changes to the overlays have to go through the framework teams at Apple. I'll talk to them internally and report back to you.

jrose-apple commented on Feb 16 Apple member

For the tests, I would just stick to `print` for now, since that's what the rest of the file is doing. The real right answer is to convert the test to use `StdlibUnitest`, but that can be done separately.

Labels
None yet

Milestone
No milestone

Assignees
jrose-apple

4 participants

Notifications
Unsubscribe
You're receiving notifications because you authored the thread.

<https://github.com/apple/swift/pull/1323>



jrose-apple commented on Mar 21

Apple member



Okay, UIKit has approved this change! (Sorry for the turnaround time.) I'll trigger the tests again.



jrose-apple commented on Mar 21

Apple member



@swift-ci Please test



jrose-apple merged commit **d9596ab** into `apple:master` on Mar 25

2 checks passed

[View details](#)

[Revert](#)



jrose-apple commented on Mar 25

Apple member



Thanks, Neil!



```
let scrollView = UIScrollView()  
scrollView.contentInset = UIEdgeInsets.zero
```

UIEdgeInsets zero

Thanks!

from @neilkimmett

<http://kimmett.me/2016/04/19/ownership-or-my-first-swift-pr-got-merged.html>