An Empirical Study on the Usage of the Swift Programming Language

Marcel Oliveira, Gustavo Pinto, Felipe Ebert, Wesley Torres, Alexander Serebrenik, Fernando Castor
Objective-C

```objective-c
if (myDelegate != nil) {
    if ([myDelegate respondsToSelector:
        @selector(scrollViewDidScroll:)]) {
        [myDelegate scrollViewDidScroll: myScrollView]
    }
}
```

Swift

```swift
myDelegate?.scrollViewDidScroll?(myScrollView)
```
"Language that is **easy** and **fun** to use."

**Source:** TIOBE Programming Community Index. [http://www.tiobe.com/tiobe_index](http://www.tiobe.com/tiobe_index)
RQ1. What are the most common problems faced by Swift developers?

RQ2. Are developers having problems with the usage of Optionals?

RQ3. Are developers having problems with error handling in Swift?
RQ1. What are the most common problems faced by Swift developers?

RQ2. Are developers having problems with the usage of Optionals?

RQ3. Are developers having problems with error handling in Swift?
~59,000 Swift questions at the time of our query

Questions tagged "swift"

83,219 yesterday at night (1,408 this week).
12 Interviews

2 students, 3 instructors, 7 developers

11 also knew Objective-C

Average of 4 years of development experience
What are the most common problems faced by Swift developers?
“There isn’t much sense in learning Swift without learning and using the frameworks”

Problems: properly setting up layout constraints and correct customization/behavior of UI elements
"Swift is friendly to new programmers"

"Swift is easy and fun"

Only two interviewees said they had problems with the syntax

Problems: custom initializers, weak and strong references, generic types, closures
Testing and Errors: 10.2%

“the worst compiler I could ever imagine and that multiplied by a hundred”

Problems: error messages are not clear, the compiler is not stable
Xcode: 3.6%

“the version changes made some of the outdated code to stop working”

Problems: old code not compiling, Swift 2.0 converter is not reliable
Swift. A modern programming language that is safe, fast, and interactive.

Swift is a powerful and intuitive programming language for iOS, OS X, tvOS, and watchOS. Writing Swift code is interactive and fun, the syntax is concise yet expressive, and apps run lightning-fast. Swift is ready for your next project—or addition into your current app—because Swift code works side-by-side with Objective-C.
"I learned Swift without knowing Objective-C. But soon I had to use an API that reads barcodes, and it only had an Objective-C version."

**Problem:** the developers need to also know Objective-C
Are developers having problems with the usage of Optionals?
Usage of Optionals

1.451 Optional-related questions (8,5%)

353 were manually analyzed
Optional Types

42  42  
Int  Int?  Int?
Usage of Optionals

Problems: understanding the purpose of Optionals, doubts with the syntax, “?” vs. “!”
var regularOpt : String? = "This won't cause a problem."

var implicitlyUnwrappedOpt : String! = regularOpt?.lowercaseString

var alsoRegularOpt : String? = regularOpt!.uppercaseString

var idontKnowAnymore: String! = implicitlyUnwrappedOpt?.lowercaseString

if let ok = alsoRegularOpt?.uppercaseString as String? {
    print("This is getting really confusing")
}
var regularOpt : String? = "This won't cause a problem."

var implicitlyUnwrappedOpt : String! = regularOpt?.lowercaseString

var alsoRegularOpt : String? = regularOpt!.uppercaseString

var idontKnowAnymore: String! = implicitlyUnwrappedOpt?.lowercaseString

if let ok = alsoRegularOpt?.uppercaseString as String? {
    print("This is getting really confusing")
}
```swift
var regularOpt : String? = "This won't cause a problem."

var implicitlyUnwrappedOpt : String! = regularOpt?.lowercaseString

var alsoRegularOpt : String? = regularOpt!.uppercaseString
var idontKnowAnymore : String! = implicitlyUnwrappedOpt?.lowercaseString

if let ok = alsoRegularOpt?.uppercaseString as String? {
    print("This is getting really confusing")
}
```
```swift
var regularOpt : String? = "This won't cause a problem."

var implicitlyUnwrappedOpt : String! = regularOpt?.lowercaseString

var alsoRegularOpt : String? = regularOpt!.uppercaseString
var idontKnowAnymore: String! = implicitlyUnwrappedOpt?.lowercaseString

if let ok = alsoRegularOpt?.uppercaseString as String? {
    print("This is getting really confusing")
}
```
var regularOpt : String? = "This won't cause a problem."

var implicitlyUnwrappedOpt : String! = regularOpt?.lowercaseString

var alsoRegularOpt : String? = regularOpt!.uppercaseString
var idontKnowAnymore: String! = implicitlyUnwrappedOpt?.lowercaseString

if let ok = alsoRegularOpt?.uppercaseString as String? {
    print("This is getting really confusing")
}
An Empirical Study on the Usage of the Swift Programming Language

Marcel Oliveira, Gustavo Pinto, Felipe Ebert, Wesllley Torres, Alexander Serebrenik, Fernando Castor.

Contact: castor@cin.ufpe.br

Seems to be easy to understand and adopt
(most questions are about frameworks, not the language)

Still dependent on Objective-C.

The purpose and usage of optionals is still mysterious to some

It may be too early to use for production development.