

# SwiftCon China 2016

[www.swiftconchina.com](http://www.swiftconchina.com)



@HANKBAO

瀑布IM

---

逆向工程

从 OBJ-C 到 SWIFT

# 什么是逆向工程?

---

一种技术过程，即对一项目标产品进行逆向分析及研究，从而演绎并得出该产品的处理流程、组织结构、功能性能规格等设计要素，以制作出功能相近，但又不完全一样的产品。

— 维基百科

# 逆向工程的目标

---

- ▶ 学习

- ▶ 设计

- ▶ 实现

- ▶ 算法

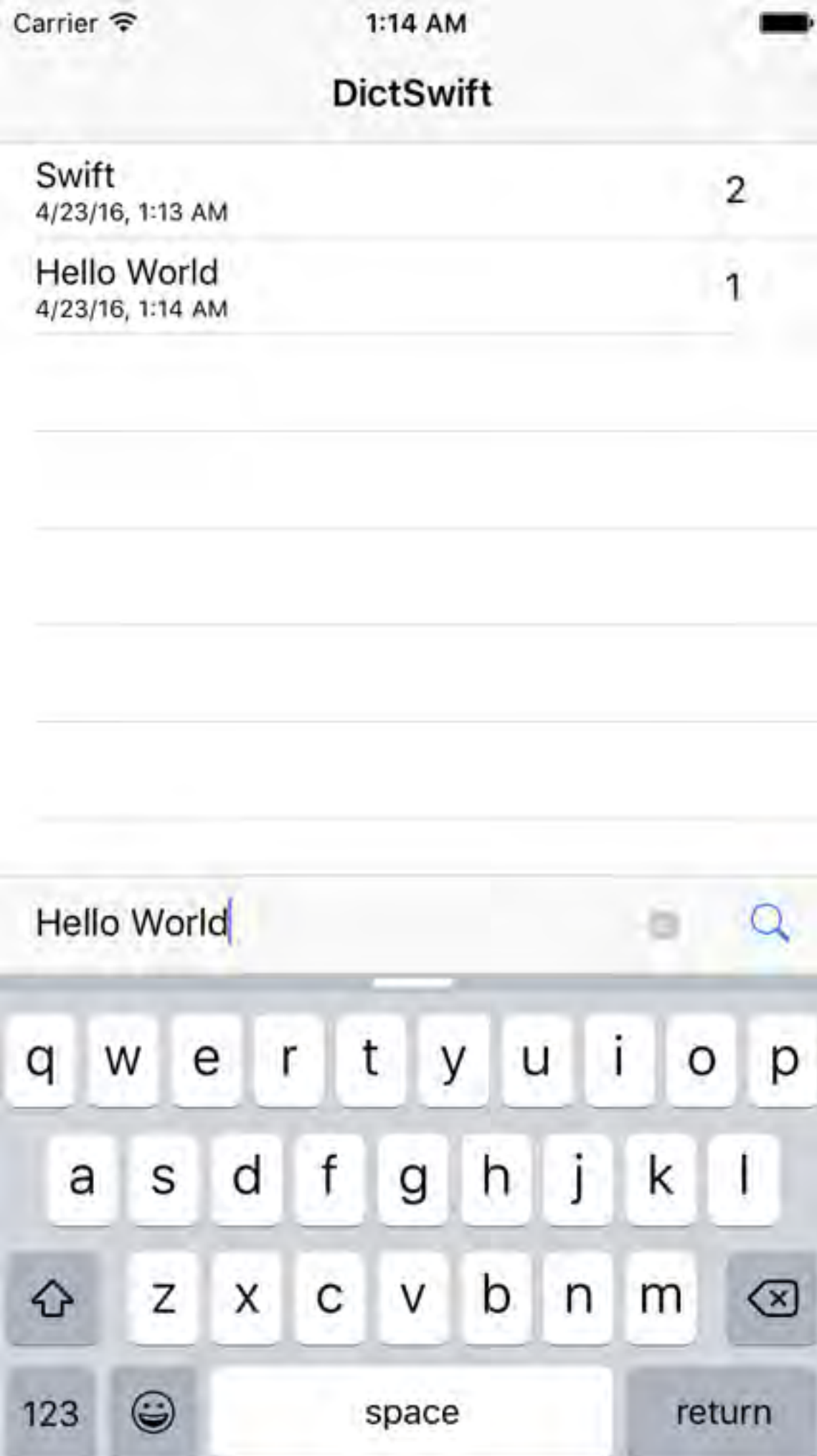
- ▶ 除错

- ▶ 扩展

- ▶ 插件

# APP 插件







# DEMO

```
11 @interface QueryRecord : NSObject
12
13 @property (nonatomic, copy, readonly) NSString *term;
14 @property (nonatomic, strong, readonly) NSDate *date;
15 @property (nonatomic, assign) NSUInteger queryCount;
16
17 - (instancetype)initWithTerm:(NSString *)term;
18
19 @end
```

|                                |   |
|--------------------------------|---|
|                                | <code>_OBJC_IVAR_\$_QueryRecord._term:</code>       |
| <code>00000000100008b68</code> | <code>dq 0x000000000000000008</code>                |
|                                | <code>_OBJC_IVAR_\$_QueryRecord._date:</code>       |
| <code>00000000100008b70</code> | <code>dq 0x000000000000000010</code>                |
|                                | <code>_OBJC_IVAR_\$_QueryRecord._queryCount:</code> |
| <code>00000000100008b78</code> | <code>dq 0x000000000000000018</code>                |

# DEMO: NON-FRAGILE LAYOUT

```
11 @implementation QueryRecord {
12     NSString* _term;
13 }
14
15 - (NSString *)term {
16     return self->_term;
17 }
18
```

-[QueryRecord term]:

```
00000000100002c6a    push    rbp                                ; 0
00000000100002c6b    mov     rbp, rsp
00000000100002c6e    mov     rax, qword [ds:_OBJC_IVAR_$_QueryRecord._term]
00000000100002c75    mov     rdi, qword [ds:rdi+rax]            ; a
00000000100002c79    pop     rbp
00000000100002c7a    jmp     imp___stubs__objc_retainAutoreleaseReturnValue
; endp
```



# X86\_64 调用约定

---

- ▶ arg1: \$rdi
- ▶ arg2: \$rsi
- ▶ arg3: \$rdx
- ▶ arg4: \$rcx
- ▶ arg5: \$r8
- ▶ arg6: \$r9
- ▶ ret1: \$rax
- ▶ ret2: \$rdx
- ▶ indirect: \$rax (== \$rdi)

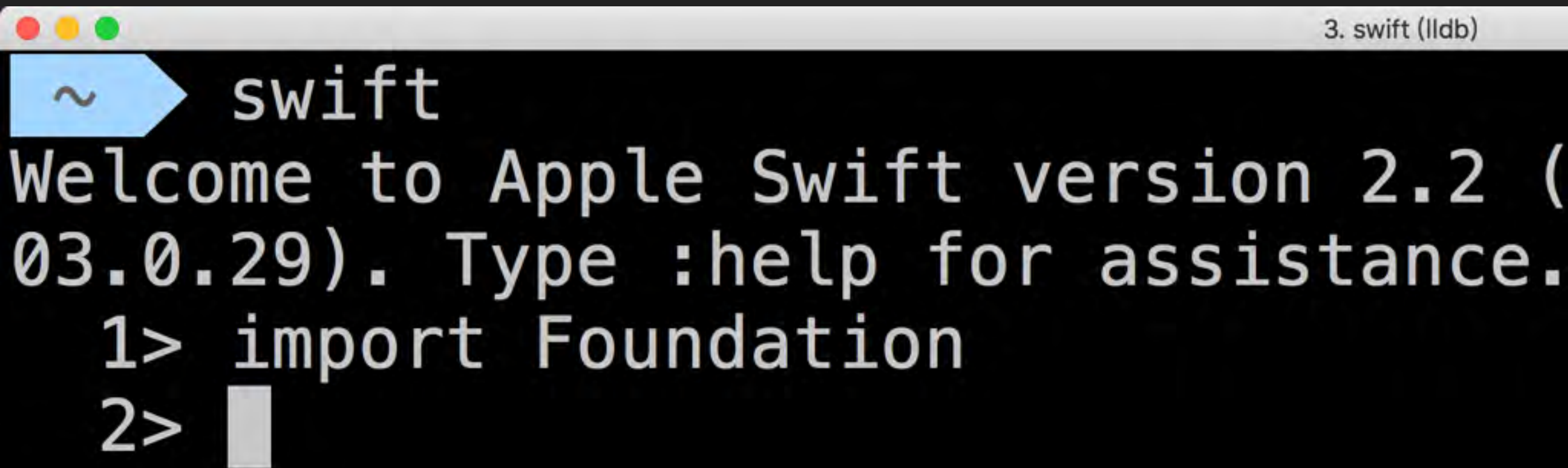
# OBJECTIVE-C 调用约定

---

- ▶ `arg1: $rdi -> self`
- ▶ `arg2: $rsi -> _cmd`
- ▶ `arg3: $rdx`
- ▶ `arg4: $rcx`
- ▶ `arg5: $r8`
- ▶ `arg6: $r9`
- ▶ `ret: $rax`

# SWIFT REPL

---

A screenshot of a macOS terminal window. The title bar at the top has three colored window control buttons (red, yellow, green) on the left and the text "3. swift (lldb)" on the right. The terminal content shows a blue prompt character "~" followed by the word "swift". Below this is a welcome message: "Welcome to Apple Swift version 2.2 (03.0.29). Type :help for assistance." The next line shows a prompt "1>" followed by the code "import Foundation". The final line shows a prompt "2>" followed by a white rectangular cursor.

```
3. swift (lldb)  
~ swift  
Welcome to Apple Swift version 2.2 (03.0.29). Type :help for assistance.  
1> import Foundation  
2> █
```

# DEMO: STRUCT

---

```
11 struct QueryRecord {  
12     let term: String  
13     let date: NSDate  
14     var queryCount: Int  
15  
16     init(term: String) {  
17         self.term = term  
18         date = NSDate()  
19         queryCount = 1  
20     }  
21 }
```

```
13> sizeof(QueryRecord)  
$R0: Int = 40  
14> sizeof(String)  
$R1: Int = 24  
15> sizeof(NSDate)  
$R2: Int = 8  
16> sizeof(Int)  
$R3: Int = 8
```



# DEMO

---

```
3> sizeof(String)
$R3: Int = 24
4> :type lookup String
struct String {
  init()
  init(_ _core: Swift._StringCore)
  var _core: Swift._StringCore
}
```

```
2> sizeof(_StringCore)
$R1: Int = 24
3> :type lookup _StringCore
struct _StringCore {
  var _baseAddress: Swift.COpaquePointer
  var _countAndFlags: Swift.UInt
  var _owner: AnyObject?
  init(baseAddress: Swift.COpaquePointer
```

# SWIFT-DEMANGLE

---

```
__TFV9DictSwift1 1 QueryRecordCfT4termSS_S0_  
__TFV9DictSwift1 1 QueryRecordg4termSS  
__TFV9DictSwift1 1 QueryRecordg4dateCSO6NSDate  
__TFV9DictSwift1 1 QueryRecordg10queryCountSi  
__TFV9DictSwift1 1 QueryRecords10queryCountSi  
__TFV9DictSwift1 1 QueryRecordm10queryCountSi
```

\$ xcrun swift-demangle

\_\_TFV9DictSwift1 1 QueryRecordg4termSS

\_\_TFV9DictSwift1 1 QueryRecordg4termSS --->

DictSwift.QueryRecord.term.getter : Swift.String



# DEMO

```
                                ; DictSwift.QueryRecord.term.getter : Swift.String
__TFV9DictSwift11QueryRecordg4termSS:
000000001000031a0      push      rbp
000000001000031a1      mov       rbp, rsp
000000001000031a4      push      r15
000000001000031a6      push      r14
000000001000031a8      push      rbx
000000001000031a9      push      rax
000000001000031aa      mov       r15, qword [ds:rdi]
000000001000031ad      mov       r14, qword [ds:rdi+8]
000000001000031b1      mov       rbx, qword [ds:rdi+0x10]
000000001000031b5      mov       rdi, rbx                                ; argument "
000000001000031b8      call     imp___stubs__swift_unknownRetain
000000001000031bd      mov       rax, r15
000000001000031c0      mov       rdx, r14
000000001000031c3      mov       rcx, rbx
000000001000031c6      add       rsp, 0x8
000000001000031ca      pop       rbx
000000001000031cb      pop       r14
000000001000031cd      pop       r15
000000001000031cf      pop       rbp
000000001000031d0      ret
                                ; endp
```

# SWIFT NATIVE 调用约定

---

- ▶ arg1: \$rdi
- ▶ arg2: \$rsi
- ▶ arg3: \$rdx
- ▶ arg4: \$rcx
- ▶ arg5: \$r8
- ▶ arg6: \$r9
- ▶ ret1: \$rax
- ▶ ret2: \$rdx
- ▶ ret3: \$rcx
- ▶ indirect: \$rax (== \$rdi)



# DEMO

---

```
11 @protocol QueryURLConvertible <NSObject>
12
13 @property (nonatomic, strong, readonly) NSURL* zt_queryURL;
14
15 @end
16
13 @interface NSString (Query) <QueryURLConvertible>
14 @end
15
```

```
11 protocol QueryURLConvertible {
12     var zt_queryURL: NSURL? { get }
13 }
14
15 extension String: QueryURLConvertible {
16     var zt_queryURL: NSURL? {
```

# DEMO

```
114 - (void)showTerm:(id <QueryURLConvertible>)convertible {
115     NSURL *url = [convertible zt_queryURL];
116
117     UIViewController *libViewController
118         = [[SFSafariViewController alloc] initWithURL:url
119            entersReaderIfAvailable:YES];
120     [self presentViewController:libViewController
121        animated:YES completion:NULL];
122 }
```

```
-[ViewController showTerm:]
0000000100001e7f      push     rbp                                ; Objective C Implementation
0000000100001e80      mov      rbp, rsp
0000000100001e83      push     r15
0000000100001e85      push     r14
0000000100001e87      push     r12
0000000100001e89      push     rbx
0000000100001e8a      mov      r14, rdi
0000000100001e8d      mov      rsi, qword [ds:0x100008980]          ; @selector(zt_queryURL),
0000000100001e94      mov      r12, qword [ds:imp___got__objc_msgSend]
0000000100001e9b      mov      rdi, rdx                            ; argument "instance" for
0000000100001e9e      call     r12                                ; _objc_msgSend
0000000100001ea1      mov      rdi, rax                            ; argument "instance" for
```



# DEMO

---

```
11> var qurl: QueryURLConvertible = ""  
qurl: String = ""  
12> sizeof(QueryURLConvertible)  
$R0: Int = 40  
13> sizeofValue(qurl)  
$R1: Int = 40
```

# DEMO

---

```
18> func addrOf<T>(inout v: T) {  
19.     withUnsafePointer(&v) { print($0) }  
20. }  
21> addrOf(&curl)  
0x00000001004fcf40  
  
22> :x/5xg 0x00000001004fcf40  
0x1004fcf40: 0x0000000101c00800 0x0000000000000000  
0x1004fcf50: 0x0000000000000000 0x00000001002724c8  
0x1004fcf60: 0x00000001004fc180
```



# DEMO

---

```
22> :ima lookup -a 0x0000000101c00800
    Address: $__lldb_expr11[0x0000000101c00800] ($__lldb_expr11.
    __cstring + 0)
    Summary:
22> :ima lookup -a 0x00000001002724c8
    Address: libswiftCore.dylib[0x00000000002674c8] (libswiftCo
re.dylib.__DATA.__const + 81656)
    Summary: libswiftCore.dylib`type metadata for Swift.String
22> :x/xg 0x00000001004fc180
0x1004fc180: 0x00000001004fb8e0
22> :x/i 0x00000001004fb8e0
    0x1004fb8e0: 55  pushq  %rbp
22> :ima lookup -a 0x00000001004fb8e0
    Address: $__lldb_expr7[0x00000001004fb8e0] ($__lldb_expr7.
    _text + 160)
    Summary: $__lldb_expr7`protocol witness for __lldb_expr_4.Q
ueryURLConvertible.zt_queryURL.getter : Swift.Optional<__ObjC.NSU
RL> in conformance Swift.String : __lldb_expr_4.QueryURLConvertib
le in __lldb_expr_6 at repl6.swift
```



# DEMO

```
90 private func showTerm(convertible: QueryURLConvertible) {
91     guard let url = convertible.zt_queryURL else { return }
92
93     let libViewController =
94         SFSafariViewController(URL: url, entersReaderIfAvailable: true)
95     presentViewController(libViewController, animated: true, completion: nil)
96 }
97
```

|                   |      |  |                  |
|-------------------|------|--|------------------|
| 00000000100004c09 | mov  | rax, qword [ss:rbp+var_50]                               | ; XREF=__TTSf4g  |
| 00000000100004c0d | mov  | qword [ss:rbp+var_40], rax                               |                  |
| 00000000100004c11 | mov  | rax, qword [ss:rbp+var_48]                               |                  |
| 00000000100004c15 | mov  | qword [ss:rbp+var_38], rax                               |                  |
| 00000000100004c19 | mov  | rdi, qword [ss:rbp+var_60]                               | ; argument "ins" |
| 00000000100004c1d | mov  | qword [ss:rbp+var_30], rdi                               |                  |
|                   |      | ; protocol witness table for                             |                  |
|                   |      | ; Swift.String : DictSwift.QueryURLConvertible           |                  |
| 00000000100004c21 | mov  | rbx, qword [ds:__TWPSS9DictSwift19QueryURLConvertibleS_] |                  |
| 00000000100004c28 | call | imp___stubs__swift_unknownRetain                         |                  |
| 00000000100004c2d | mov  | rsi, qword [ds:imp___got___TMSS]                         |                  |
| 00000000100004c34 | lea  | rdi, qword [ss:rbp+var_40]                               | ; argument #1 f  |
| 00000000100004c38 | call | rbx  | ; __TTWSS9DictS  |
| 00000000100004c3a | mov  | r14, rax   |                  |
| 00000000100004c3d | test | r14, r14   |                  |
| 00000000100004c40 | je   | 0x100004c9b  |                  |



# DEMO

```
12 class ViewController: UITableViewController {  
13  
14     private var toolbar: UIToolbar!  
15     private weak var textField: UITextField!  
16  
17     private var dataSource: DataSource!
```

```
98     private func queryTerm(term: String) {  
99         textField.text = nil  
100  
101         let (indexPath, isNewTerm) = dataSource.addTerm(term)  
102         if (isNewTerm) {  
103             tableView.insertRowsAtIndexPaths([indexPath], withRowAnimation: .Bottom)  
104         } else {  
105             let to = NSIndexPath(forRow: 0, inSection: 0)  
106             dataSource.moveTermAtIndexPath(indexPath, toIndexPath: to)  
107             tableView.moveRowAtIndexPath(indexPath, toIndexPath: to)  
108             tableView.reloadRowsAtIndexPaths([to], withRowAnimation: .Automatic)  
109         }  
110  
111         showTerm(term)  
112     }
```

# DEMO



纳尼? !





# DEMO

98  
99  
100

```
private func queryTerm(term: String) {  
    textField.text = nil  
}
```

```
                                ; function signature specialization  
                                ; <Arg[0] = Owned To Guaranteed and Exploded>  
                                ; DictSwift.ViewController.(queryTerm) (Swift.String) -> ()  
                                ;  
                                ; {rdi, rsi, rdx} = term: String  
                                ; rcx = self  
__TTSf4gs_n____TFC9DictSwift14ViewControllerP33_A554EBA85D7A3574C09A68073201E51  
00000000100004840      push    rbp                                ; XREF=-[_TtC9DictS  
00000000100004841      mov     rbp, rsp  
00000000100004844      push    r15  
00000000100004846      push    r14  
00000000100004848      push    r13  
0000000010000484a      push    r12  
0000000010000484c      push    rbx  
0000000010000484d      sub     rsp, 0x48  
00000000100004851      mov     r13, rcx  
00000000100004854      mov     r12, rdx  
00000000100004857      mov     qword [ss:rbp+var_48], rsi  
0000000010000485b      mov     qword [ss:rbp+var_50], rdi  
                                ; direct field offset for  
                                ; DictSwift.ViewController.(textField) : weak __ObjC.UITextF  
0000000010000485f      mov     rdi, qword [ds:__TWvdvC9DictSwift14ViewControllerP33_A554EBA85D  
00000000100004866      add     rdi, r13  
00000000100004869      call    imp___stubs__swift_unknownWeakLoadStrong  
0000000010000486e      mov     rbx, rax  
00000000100004871      test    rbx, rbx  
00000000100004874      je      0x100004cba
```

00000000100004cba

ud2



# DEMO

```
101 let (indexPath, isNewTerm) = dataSource.addTerm(term)
102 if (isNewTerm) {
```

```
                                ; direct field offset for
                                ; DictSwift.ViewController(dataSource) : DictSwift.DataSource!
0000000100004893 mov r14, qword [ds:__TWvdvC9DictSwift14ViewControllerP33_A554EBA85D7A357
000000010000489a mov rbx, qword [ds:r13+r14]
000000010000489f test rbx, rbx
00000001000048a2 je 0x100004cba
-----
00000001000048a8 mov rax, qword [ds:imp__got__swift_isaMask]
00000001000048af mov rax, qword [ds:rax]
00000001000048b2 and rax, qword [ds:rbx]
00000001000048b5 mov r15, qword [ds:rax+0x70]
00000001000048b9 mov rdi, rbx ; argument "instance" for
00000001000048bc call imp__stubs__objc_retain
00000001000048c1 mov rdi, r12 ; argument "instance" for
00000001000048c4 call imp__stubs__swift_unknownRetain
00000001000048c9 mov rdi, qword [ss:rbp+var_50]
                                ; mov qword [ss:rbp+var_48], rsi
                                ; {var_50, var_48, r12} == term
00000001000048cd mov rsi, qword [ss:rbp+var_48]
00000001000048d1 mov rdx, r12
00000001000048d4 mov qword [ss:rbp+var_60], r12
00000001000048d8 mov rcx, rbx
00000001000048db call r15
00000001000048de mov qword [ss:rbp+var_58], rax
00000001000048e2 mov r15b, dl
00000001000048e5 mov rdi, rbx ; argument "instance" for
00000001000048e8 call imp__stubs__objc_release
00000001000048ed test r15b, 0x1
00000001000048f1 je 0x100004a21
```



# 逆向工程理论基础

---

- ▶ C / C++ / Objective-C / Swift
- ▶ Assembly (x86, x86\_64, arm / thumb, arm64)
- ▶ 平台 ABI / 语言特定 ABI
- ▶ 编译器优化
- ▶ 操作系统

# 逆向工程方法和工具

---

## ▶ 静态分析

- ▶ Hopper Disassembler
- ▶ IDA Pro
- ▶ otool
- ▶ class-dump

## ▶ 动态调试

- ▶ lldb / gdb
- ▶ F-Script
- ▶ cycrypt

- ▶ Wikipedia
- ▶ System V Application Binary Interface (AMD64)
- ▶ Procedure Call Standard for the ARM 64-bit Architecture
- ▶ iOS ABI Function Call Guide
- ▶ The Swift ABI
- ▶ The Swift Calling Convention
- ▶ Friday Q&A



## 相关资源

---

- ▶ <https://github.com/apple/swift>
- ▶ <https://github.com/hankbao/DictObjc>
- ▶ <https://github.com/hankbao/DictSwift>

THANKS

Q & A