<You always have</pre>

time for Coding>

A Product of Viblo Platform

trong ngôn ngữ Swift (Part 2)

Đã đăng vào thg 2 12, 2018 12:41 PM - 2 phút đọc ◎ 1.2K ♀ 0

MỤC LỤC

CÁC TỔ CHỨC ĐƯỢC ĐỀ XUẤT Sun* Cyber Security Team

Keywords in Statements

000 Bài đăng này đã được cập nhật cách đây 3 năm kể từ khi nó được cập nhật lần cuối.

Avengers Group Sun* R&D Lab



+2

Tiếp tục phần phần 1: https://viblo.asia/p/ios-swift-tong-hop-tat-ca-tu-khoa-trong-ngon-

ngu-swift-part-1-E375zEAdlGW Trong phần này chúng ta sẽ tiếp tục với các từ khóa thường xuyên được sử dụng trong các câu lệnh.

} else {

//Party on

default:

default:

print(dataStore.findByLastName(name))

print("Box equals 0 or 1")

print("Box doesn't equal 0 or 1")

print("Box doesn't equal 0 or 1")

if: đảm bảo 1 hoặc nhiều điều kiện khi thực hiện 1 hoặc nhiều câu lệnh nào đó.

Keywords in Statements

if 1 > 2 { print("This will never execute")

if val > 1 {

print("val is greater than 1")

else: Nếu không phải thằng A thì là thằng B thôi.

```
print("val is not greater than 1")
return: thoát khỏi hàm luôn, và có thể trả về giá trị nào đó tuỳ chúng ta khai báo.
```

func doNothing() {

```
return //Immediately leaves the context
      let anInt = 0
      print("This never prints \(anInt)")
 func returnName() -> String? {
      return self.userName //Returns the value of userName
guard: đảm bảo điều kiện nào đó đúng, vừa có thể unwrap biến optional. Thật ra xài if
```

private func printRecordFromLastName(userLastName: String?) { guard let name = userLastName, userLastName != "Null" else { //Sorry Bill Null, find a new job return

let cũng được, nhưng xài guard đỡ phải lồng vào trong 1 scope nhìn rối mắt.

```
switch: xét và so sánh giá trị xem nó match với case nào, hoặc thích thì dùng if + else if
+ else.
 let box = 1
 switch box {
 case 0:
```

print("Box equals 0") fallthrough case 1:

```
case: một trường hợp trong câu lệnh switch.
 let box = 1
 switch box {
 case 0:
     print("Box equals 0")
 case 1:
     print("Box equals 1")
```

default : xét case từ trên xuống mà ko thấy cái nào dính thì bay vào default.

```
let box = 1
switch box {
case 0:
    print("Box equals 0")
case 1:
    print("Box equals 1") // In mỗi thẳg này ra thôi
default:
    print("Covers any scenario that doesn't get addressed above.")
```

let box = 0switch box {

fallthrough: xét trên xuống, nếu khớp 1 case nào đó vẫn tiếp tục xét tiếp 1 case liền kề

```
case 0:
    print(0) // In ra 0
   fallthrough
case 1:
    print(1) // In ra 1
case 2:
    print(2) // Không có được in ra
default:
    print("default")
```

print ("This prints 3 times")

for : giúp lặp các phần tử trong 1 sequence hoặc array hoặc các kí tự trong 1 String.

```
in : đi chung với for ở trên
```

for idx in 0...3 { if idx % 2 == 0 {

break: kết thúc chương trình trong loop, if hoặc switch

print ("This prints 3 times")

for _ in 0..<3 {

for _ in 0..<3 {

break

for idx in 0...3 {

do {

} while 1 > 2

Equatable {

Đã đăng ký Bản quyền

if idx % 2 == 0 {

```
continue: trong vòng lặp nếu gặp continue thì nó sẽ bỏ qua trường hợp đó (ở đây
trong câu lệnh if) rồi tiếp tục duyệt tiếp.
```

continue // Thoát khúc này thôi print("This code never fires on even numbers") // Vẫn in ra nhé

```
defer : sử dụng khi muốn đảm bảo thực thi 1 đoạn code nào đó ngay khi hàm kết thúc.
 func test() {
     defer {
         print("2") // thằng này print sau
     print("1") // thằng này print trước
```

try expression //statements } catch someError ex { //Handle error

loại generic type phải conform đúng protocol nào đó.

return "This things name is " + namedEntity.name

do: thực thi đoạn chương trình nào đó có khả năng xảy ra lỗi.

```
repeat: Y chang do while trong C/C++. Thực thi đoạn chương trình ít nhất 1 lần trước
khi lặp qua điều kiện
  repeat {
```

print("Always executes at least once before the condition is considered")

protocol Nameable { var name: String { get set }

func createdFormattedName<T: Nameable>(_ namedEntity: T) -> String where T:

// Chỉ có thực thể nào mà vừa conform Nameable và Equatable thì mới dùng (

where: dùng để ghép với for kiểm tra xem có đúng điều kiện không, hoặc dùng cho 1

```
for i in 0...3 where i % 2 == 0 {
      print(i) //Prints 0 and 2
while: loop điều kiện nào đó cho tới khi nó sai thì thôi.
 while foo != bar {
      print("Keeps going until the foo == bar")
```

Bài viết liên quan [iOS] [Swift] Tổng hợp tất cả Các Keyword cơ bản trong Top 10 Array (List) method trong dart nên biết từ khóa trong ngôn ngữ Swi... lập trình Swift

```
Nguyễn Quốc Tình
                           Tây Môn Xuý Tuyết
                           4 phút đọc
 5 phút đọc
                                                    2 phút đọc
 Bài viết khác từ Nguyễn Quốc Tình
                                                    Mở rộng ứng dụng iOS bằng
  (SwiftUI) GridStack - layout
                           Empty Strings in Swift
 lưới trong vài dòng code
                                                    module
 Nguyễn Quốc Tình
                           Nguyễn Quốc Tình
                                                    Nguyễn Quốc Tình
 4 phút đọc
                           2 phút đọc
                                                    13 phút đọc
```

trong Swift Nguyễn Quốc Tình 9 phút đọc

Sự hữu dụng của type alias

Trong swift: func là gì?,

closure là gì?

vietanh

4 phút đọc

7

000

Về chúng tôi Phản hồi Giúp đỡ FAQs RSS Điều khoản DMCA (1) PROTECTED © Viblo 2021

Bình luận Xem trước

C Tiếng Việt

