

Trần Xuân Thiện @tran.xuan.thien Theo dõi

Đã đăng vào thg 9 19, 2018 4:25 PM - 2 phút đọc

🕒 768 🗂 0 📄 1

MỤC LỤC

Khái niệm:

Syntax:

Options in Subscript:

Kết:

CÁC TỔ CHỨC ĐƯỢC ĐỀ XUẤT



Sun* Cyber Security Team

174

10

258

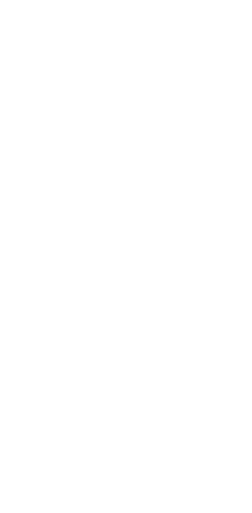


Avengers Group

55

35

117



Sun* R&D Lab

43

80

266

Subscripts in swift

+1

⋮

! Bài đăng này đã được cập nhật cách đây 2 năm 2 kể từ khi nó được cập nhật lần cuối.

Khái niệm:

Subscript được sử dụng để truy cập thông tin của một collection, sequence và một list trong Classes, Structures và Enumerations. Đặc biệt những Subscript có thể lưu trữ và truy xuất các giá trị bằng index mà không sử dụng một method riêng biệt. Đây là một lối tắt rất hiệu quả và ngắn gọn trong lập trình.

Cách thức truy cập: Chỉ cần truyền index vào "A[index]" là OK.

Ví dụ:

1. subscripting an array:

```
let array = [1, 2, 3, 5, 8, 13]
print(array[0])
print(array[1..<4])
print(array[1...4])
```

Result:

```
1
[2, 3, 5]
[2, 3, 5, 8]
```

Ở trên, chúng ta có thể truyền 1 index hoặc một dãy index (1..<4, 1...4) vào.

2. subscripting a dictionary:

```
var dictionary = ["male": "I am a male"]
print(dictionary["male"])
```

Result:

```
Optional("I am a male")
```

Syntax:

```
subscript (<parameters>) -> <return type> {
    get {
        // used for subscript value declarations
    }
    set(newValue) {
        // definitions are written here
    }
}
```

Danh sách **parameters** có thể lấy nhiều giá trị.

Method **get** là **required** (bắt buộc).

Method **setter** là **optional** (không bắt buộc)

Tham số **newValue** trong method **set** được khai báo hoàn toàn. Bạn có thể thay đổi nó thành bất kỳ thứ gì bạn muốn, điều này sẽ cải thiện khả năng đọc code.

Ví dụ:

```
class DayOfWeek {
    private var days = ["Sunday", "Monday", "Tuesday", "Wednesday",
    subscript(index: Int) -> String {
        get {
            return days[index]
        }
        set(newValue) {
            self.days[index] = newValue
        }
    }
    var p = DayOfWeek()
    print(p[0])
}
```

```
p[0] = "Monday"
print(p[0])
```

Result:

```
Sunday
Monday
```

```
struct subexample {
    let decrementer: Int
    subscript(index: Int) -> Int {
        return decrementer / index
    }
}
let division = subexample(decrementer: 100)

print("The number is divisible by \(division[9]) times")
print("The number is divisible by \(division[2]) times")
```

Result:

```
The number is divisible by 11 times
The number is divisible by 50 times
```

Options in Subscript:

Subscript có thể lấy bất kỳ số lượng tham số đầu vào và những tham số đầu vào này có thể là bất kỳ kiểu dữ liệu nào. Subscript cũng có thể trả về bất kỳ kiểu dữ liệu nào. Subscript có thể sử dụng variable parameters và variadic parameters, nhưng không thể sử dụng in-out parameters hoặc cung cấp giá trị mặc định cho tham số.

Một class hoặc struct có thể cung cấp nhiều subscript để thực thi khi nó cần. Vì những subscript thích hợp được sử dụng sẽ suy luận dựa trên kiểu của giá trị hoặc giá trị được chứa trong giỏ subscript tại điểm mà subscript sử dụng. Sự định nghĩa nhiều subscript gọi là subscript overloading.

Subscript phổ biến nhất là một tham số đơn, bạn cũng có thể định nghĩa một subscript với nhiều tham số nếu nó thích hợp cho kiểu của bạn. Ví dụ về subscript 2 tham số:

```
struct Matrix {
    let rows: Int, columns: Int
    var valuesArray: [Double]
    init(rows: Int, columns: Int) {
        self.rows = rows
        self.columns = columns
        valuesArray = Array(repeating: 0.0, count: rows * columns)
    }
    subscript(row: Int, column: Int) -> Double {
        get {
            return valuesArray[(row * columns) + column]
        }
        set {
            valuesArray[(row * columns) + column] = newValue
        }
    }
    var mat = Matrix(rows: 3, columns: 3)
```

```
mat[0,0] = 1.0
mat[0,1] = 2.0
mat[1,0] = 4.0
mat[1,1] = 5.0
```

```
print("\(mat[0,0])")
print("\(mat[0,1])")
print("\(mat[1,0])")
print("\(mat[1,1])")
```

Result:

```
1.0
2.0
4.0
5.0
```

Kết:

Trên đây mình đã giới thiệu về Subscript trong Swift. Hy vọng nó giúp ích cho mọi người. Thanks. 😊

Đã đăng ký Bản quyền

⋮

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...