SWIFT
Subscripts

Bill Kim(김정훈) | ibillkim@gmail.com

목차

Subscripts

Subscripts : 콜렉션, 리스트, 시퀀스 등 집합의 특정 멤버 엘리먼트에 간단하게 접근할 수 있도록 해주는 문법

서브스크립트를 이용하면 간편하게 추가적인 메소드 없이 특정 값을 할당하거나 가져올 수 있게된다.

즉 내가 원하는 값을 쉽게 찾기 위해 정의하는 문법을 서브스크립트 (예 : array[1] 과 같이, array에도 이미 내부에 서브스크립트가 구현되어 있음)

Subscript Syntax

기본적인 서브스크립트 문법은 아래 코드와 같이 subscript 지시어와 함께 get, set을 정의하여 사용할 수 있습니다.

```
struct TimesTable {
    let multiplier: Int

    subscript(index: Int) -> Int {
        set(value){ // 생략 가능하며 newValue로 전달인자를 사용할 수 있습니다.
            print(value)
        }
        get{
            return index*3
        }
    }
}

var threeTimesTable = TimesTable(multiplier: 3)

print("six times three is \(threeTimesTable[6])")
// "six times three is 18" 출력, 자동적으로 get을 통하여 6번째 엘리먼트 값을 3배로 하여 가져온다.

threeTimesTable[2] = 1 // 2번째 엘리먼트 값을 1로 설정(set)
```

Subscript Usage

```
// numberOfLegs값은 타입 추론에 의해 [String: Int]형을 갖는다.
var numberOfLegs = ["spider": 8, "ant": 6, "cat": 4]

numberOfLegs["bird"] = 2

print(numberOfLegs["ant"]!) // 6, Forced Unwrapping으로 실제 값을 출력하도록 한다.
```

```
class MovieList { // 영화 리스트 클래스.
private var tracks = ["The Godfather", "The Dark Night", "Superman"]
    subscript(index: Int) -> String {
        get {
            return self.tracks[index]
        }
        set{
            self.tracks[index] = newValue
        }
    }
}

var movieList = MovieList() // 클래스 인스턴스 만들고
print("영화 리스트에서 두 번째 영화는 \(movieList[1])") // The Dark Night 출력됨.
```

Subscript Options

서브스크립트는 아래의 옵션 사항을 제공합니다.

- 입력 인자는 어떠한 값이든 취할 수 있음
- 입력 인자는 어떤 타입이든 가능함
- 서브스크립트는 어떠한 타입도 반환 가능
- 변수 인자와 가변 인자 사용이 가능
- in-out 인자는 사용할 수 없음(주소값 파라미터 사용 안됨)
- 적합하고 추론 가능한 경우 다중 인자를 사용할 수 있음

Subscript Options

서브스크립트 다중 인자 예시

```
struct Matrix {
let rows: Int, columns: Int
var grid: [Double]

   init(rows: Int, columns: Int) {
        self.rows = rows
        self.columns = columns
        grid = Array(repeating: 0.0, count: rows * columns)
   }

   func indexIsValidForRow(row: Int, column: Int) -> Bool {
        return row >= 0 && row < rows && column >= 0 && column < columns
   }

   subscript(row: Int, column: Int) -> Double {
      get {
          assert(indexIsValidForRow(row: row, column: column), "Index out of range")
          return grid[(row * columns) + column]
       }
       set {
          assert(indexIsValidForRow(row: row, column: column), "Index out of range")
          grid[(row * columns) + column] = newValue
       }
   }
}

var matrix = Matrix(rows: 2, columns: 2)

matrix[0, 1] = 1.5
matrix[1, 0] = 3.2

print(matrix[0, 1])
print(matrix[1, 0])
print(matrix[2, 2]) // Assetion!!! 서브스크립트 접근이 행렬을 벗어날 경우 assert를 발생함
```

References

[1] [스위프트 : 기초] 서브스크립트 : https://the-brain-of-sic2.tistory.com/37

[2] [Swift]Subscripts 정리 : http://minsone.github.io/mac/ios/swift-subscripts-summary

[3] Swift - 서브스크립트 : https://penguin-story.tistory.com/40

[4] Swift4 subscript(서브스크립트) : http://blog.naver.com/PostView.nhn?blogId=hjleesm&logNo=221349109702

[5] 서브스크립트(Subscripts) : https://kka7.tistory.com/118

References

[6] [Swift 3] 서브스크립트 (Subscript) : https://beankhan.tistory.com/163

[7] 서브스크립트 (Subscripts) : https://jusung.gitbook.io/the-swift-language-guide/language-guide/12-subscripts

[8] Swift - 서브스크립트(Subscripts) : http://seorenn.blogspot.com/2014/06/swift-subscripts.html

[9] [Swift] 서브스크립트(Subscript) : https://jinnify.tistory.com/37

[10] [swift] Subscript : https://zetal.tistory.com/entry/swift-기초문법-18-서브스크립트Subscript

Thank you!