

SWIFT

Higher-Order Function

Bill Kim(김정훈) | ibillkim@gmail.com

목차

Higher-Order Function?

map

filter

reduce

flatMap

compactMap

References

Higher-Order Function?

Higher-Order Function : 고차함수

- 1) 다른 함수를 전달인자로 받거나
- 2) 함수실행의 결과를 함수로 반환하는 함수

Swift에서는 대표적으로 아래의 고차함수 등을 제공합니다.

- map
- filter
- reduce
- flatMap
- compactMap

map

기존 데이터를 변형하여 새로운 컨테이너를 만들어 줍니다.
기존 데이터는 변형되지 않습니다.

map은 기존의 for-in 구문과 큰 차이가 없지만, map 사용시 다음과 같은 이점이 있습니다.

장점

코드의 간결성
재사용 용이
컴파일러 최적화 성능 좋음

map

for-in 구문 사용 시

```
let numArray = [1, 3, 5, 7, 9]
var multiArray = [Int]()
for num in numArray {
    multiArray.append(num * 2)
}

print(multiArray) // [2, 6, 10, 14, 18]
```

map 사용 시

```
print(multiArray) // [2, 6, 10, 14, 18]

let numArray = [1,3,5,7,9]

// 축약 미사용
let multiArray1 = numArray.map({ (number: Int) -> Int in
    return number * 2
})

// 축약 사용
let multiArray2 = numArray.map { $0 * 2 }

print(multiArray1) // [2, 6, 10, 14, 18]
print(multiArray2) // [2, 6, 10, 14, 18]
```

filter

컬렉션 내부의 데이터를 조건에 맞는 새로운 컬렉션으로 생성

```
let array = ["aaa", "bbb", "cc", "dddd", "e"]
```

```
// 축약 미사용
```

```
let newArray1 = array.filter({ (value: String) -> Bool in  
    return value.count == 3  
})
```

```
// 축약 사용
```

```
let newArray2 = array.filter { $0.count == 3 }
```

```
print(newArray1) // ["aaa", "bbb"]
```

```
print(newArray2) // ["aaa", "bbb"]
```

reduce

reduce는 데이터를 합쳐주기 위해 사용합니다.
기존 컨테이너에서 내부의 값들을 결합하여 새로운 값을 만듭니다.

```
let numberArray = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

// 축약 미사용
let sum1 = numberArray.reduce(0, { (first: Int, second: Int) -> Int in
    return first + second
})

// 축약 사용
let sum2 = numberArray.reduce(0) { $0 + $1 }

print(sum1) // 55
print(sum2) // 55
```

flatMap

flatten(평평하게 하다) + map(대응하다)가 합쳐진 의미
flatMap은 map과 다르게 아래의 3가지 기능을 가지고 있습니다.

- 1) non-nil인 결과들을 가지는 배열을 리턴(1차 배열에서만)
- 2) 주어진 Sequence내의 요소들을 하나의 배열로써 리턴(2차 배열을 1차 배열로)
- 3) 주어진 Optional이 not-nil인지 판단 후 unwrapping하여 closure 파라미터로 전달

flatMap

```
let array1 = [1, nil, 3, nil, 5, 6, 7]

// 축약 미사용
let flatMap1 = array1.flatMap() { (value: Int?) -> Int? in
    return value
}

print(flatMap1) // [1, 3, 5, 6, 7]

// 축약 사용
let flatMap2 = array1.flatMap() { $0 }

print(flatMap2) // [1, 3, 5, 6, 7]

// 2차원 배열

let array2: [[Int?]] = [[1, 2, 3], [nil, 5], [6, nil], [nil, nil]]
let flatMap3 = array2.flatMap { $0 }

print(flatMap3) // [Optional(1), Optional(2), Optional(3), nil, Optional(5),
Optional(6), nil, nil, nil]
```

compactMap

flatMap을 대체하기 위해서 Swift 4.1에서 나온 새로운 고차함수
1차원 배열에서 nil을 제거하고 옵셔널 바인딩을 하고 싶을 경우 사
용

단 2차원 배열을 1차원 배열로 flatten하게 만들때는 여전히
flatMap을 사용

compactMap

```
let array1 = [1, nil, 3, nil, 5, 6, 7]
let array2: [[Int?]] = [[1, 2, 3], [nil, 5], [6, nil], [nil, nil]]

let compactMap1 = array1.compactMap { $0 } // 1차원 배열에 대해서 nil 요소 제거
let compactMap2 = array2.compactMap { $0 } // 2차원 배열에 대해서는 여전히 nil도 포함된다.
let compactMap3 = array2.flatMap { $0 }.compactMap { $0 } // 2차원 배열을 flatMap으로
1차원으로 변경 후 nil 요소 제거

print(compactMap1) // [1, 3, 5, 6, 7]
print(compactMap2) // [[Optional(1), Optional(2), Optional(3)], [nil,
Optional(5)], [Optional(6), nil], [nil, nil]]
print(compactMap3) // [1, 2, 3, 5, 6]
```

References

- [1] The Swift Language Guide : <https://jusung.gitbook.io/the-swift-language-guide/>
- [2] 고차함수 : https://yagom.github.io/swift_basic/contents/22_higher_order_function/
- [3] Swift 고차 함수 - Map, Filter, Reduce : <https://oaksong.github.io/2018/01/20/higher-order-functions/>
- [4] Swift의 클로저 및 고차 함수 이해하기 : <https://academy.realm.io/kr/posts/closure-and-higher-order-functions-of-swift/>
- [5] 03. 고차함수 활용 : <https://edu.goorm.io/learn/lecture/1141/야곰의-스위프트-프로그래밍/lesson/43408/고차함수>

References

[6] Swift - 클로저 & 고차함수 : <https://medium.com/@ggaa96/swift-study-1-클로저-고차함수-abe199f22ad8>

[7] Swift :: 고차함수 -Map, Filter, Reduce 알아보기 : <https://shark-sea.kr/entry/Swift-고차함수-Map-Filter-Reduce-알아보기>

[8] 스위프트 고차함수 swift higher-order function : <https://studyhard24.tistory.com/88>

[9] Swift 클로저 및 고차 함수 (CLOSURE & HIGHER-ORDER-FUNCTION) : <https://kanghoon.tistory.com/1>

[10] 고차함수 : <https://blog.yagom.net/565/>

References

- [11] Swift의 고차함수 : <https://jinunthing.tistory.com/54>
- [12] Swift 고차함수 사용법 : <https://usinuniverse.bitbucket.io/blog/high-order-function.html>
- [13] swift 기본문법 - 고차 함수(higher order function) : https://www.zehye.kr/swift/2020/01/16/19swift_grammar23/
- [14] Swift - 고차함수(Youtube) : <https://www.youtube.com/watch?v=HmabXrK2tRo>
- [15] [부스트코스] iOS 프로그래밍을 위한 스위프트 기초 : https://www.edwith.org/boostcamp_ios/lecture/11285

References

- [16] [Swift] 고차함수(2) - map, flatMap, compactMap : [https://jinshine.github.io/2018/12/14/Swift/22.고차함수\(2\)%20-%20map,%20flatMap,%20compactMap/](https://jinshine.github.io/2018/12/14/Swift/22.고차함수(2)%20-%20map,%20flatMap,%20compactMap/)
- [17] Swift4.1) flatMap -> compactMap : <https://zeddios.tistory.com/448>
- [18] flatMap() 정리하기. map(), compactMap() 과의 차이 : <https://ontheswift.tistory.com/22>
- [19] swift. map, flatMap, compactMap : <https://mrgamza.tistory.com/606>
- [20] Map, FlatMap and CompactMap : <https://www.swiftbysundell.com/basics/map-flatmap-and-compactmap/>

Thank you!