

SWIFT Tuple

Bill Kim(김정훈) | ibillkim@gmail.com

목차

Tuple?

Tuple Basic Usage

Naming Index

Tuple Type

Function Type

Why tuples are good?

References

Tuple?

사전적 의미

튜플(Tuple)이란 유한 개의 사물의 순서있는 열거

Swift에서의 튜플은 다양한 값(데이터)의 묶음이다.

튜플의 구성 요소들은 서로 다른 타입이 가능하며 개수도 사용하고 싶은 만큼 사용이 가능하다.

Tuple Basic Usage

튜플의 기본 사용은 () 안에 다양한 데이터 값을 넣어주면 된다.

```
var tuple:(String, Int, Bool) = ("Bill", 100, true)
var simpleTuple = ("Joyce", 200, false) // 추론을 통한 데이터 타입을 생략

print(tuple.0) // Bill
print(simpleTuple.0) // Joyce

var (name, index, isMan) = tuple // 튜플의 값들에 변수나 상수에도 넣을 수 있다.
print("이름 : \(name)") // 이름 : Bill

var tupleArr = [(1, "Hello, world!", true) , (2, "Hello, world!", false)]

// 튜플 배열에 대해서 아래와 같이 loop를 돌 수 있습니다.

for index in tupleArr {
    print(index.0) // 1 2
    print(index.1) // "Hello, world!" "Hello, world!"
    print(index.2) // true false
}
```

Naming Index

튜플의 각 엘리먼트(원소)에는 이름을 줄 수도 있습니다.

해당 이름을 통해서 기존에 멤버 접근 시 숫자로 접근하던 부분을 해당 이름으로 접근할 수 있습니다.(숫자 인덱스처럼 사용)

```
var namedTuple = (name: "Bill", age: 30, likes : ["Swift", "iOS"])  
print(namedTuple.name) // Bill  
print(namedTuple.age) // 30  
print(namedTuple.likes) // ["Swift","iOS"]
```

```
namedTuple.name = "Joyce" // name을 다른 값으로 변경 가능  
print(namedTuple.name) // Joyce
```

Tuple Type

튜플은 기본 Swift내에 **명명된 데이터 타입(Named Type)**외에도 새롭게 선언한 **튜플 타입**도 저장할 수도 있습니다.

```
// 입력하는 데이터 타입에는 제한이 없으며 튜플 타입도 저장이 가능하다.  
var anotherTuple = (1, (tuple))  
print(anotherTuple.0) // 1  
print(anotherTuple.1.0) // Bill
```

Function Type

튜플은 아래와 같이 함수 타입(Function Type) 또한 저장 가능합니다.

```
func a() -> Int { return 1 }  
func b() -> String { return "Bill" }  
func c() -> Bool { return false }
```

```
var functionTuple = (a(), b(), c())
```

```
print(functionTuple.0) // 1  
print(functionTuple.1) // Bill  
print(functionTuple.2) // false
```

Why tuples are good?

튜플을 사용하면 아래와 같은 이점을 얻을 수 있습니다.

1. 다양한 데이터 타입을 담는 배열을 만들 수 있다.

타입 제한없이 다양한 데이터를 담는 배열을 가질 수 있다.

2. 구조체의 대체가 가능하다.

기존 구조체보다 훨씬 간단한 형태를 통해 구조체처럼 사용할 수 있다.

3. 멀티 리턴 함수를 만들 수 있다.

함수에서 사용 시 하나 이상의 값을 리턴하는 함수를 만들 수 있다.

```
func plusAndMinus(a: Int, b: Int) -> (Int, Int) {  
    return (a + b, a - b)  
}
```

```
let (plusResult, minusResult) = plusAndMinus(a: 1, b: 2)
```

```
print(plusResult) // 3  
print(minusResult) // -1
```


References

[1] Swift) tuple : <https://zeddios.tistory.com/238>

[2] 튜플의 고급 활용과 모범 사례 : <https://outofbedlam.github.io/swift/2016/04/02/TupleBestPractice/>

[3] Swift: Tuple : <https://medium.com/swift-programming/swift-tuple-328aecff50e7>

[4] What is a tuple? : <https://www.hackingwithswift.com/example-code/language/what-is-a-tuple>

[5] [Swift 공부] 튜플(Tuple)이란? : <https://noahlogs.tistory.com/13>

References

[6] Swift - 튜플(Tuple) : <http://seorenn.blogspot.com/2014/06/swift-tuple.html>

[7] 주간 Swift #1: Set, Tuple, Array를 이해하기 : <https://academy.realm.io/kr/posts/swiftweekly1/>

[8] Swift - Tuples : https://www.tutorialspoint.com/swift/swift_tuples.htm

[9] [Swift] Tuple : <http://blog.davepang.com/post/329>

[10] Tuples In Swift Explained : <https://learnappmaking.com/tuples-how-to-swift/>

Thank you!