

Ch3. Swift Standard Library

Dictionary

key-value 형태

순서를 보장하지 않음(순서는 존재하지만, 매번 동일한 순서를 보장하지 않음)

조회 => Constant Time

삽입 => Constant Time

Key type이 'Hashable'이어야 함

Hashable??

<https://zeddios.tistory.com/498> / <https://baked-corn.tistory.com/123>

Dictionary - 순서를 보장하지 않는다

```
var scores: [String: Int] = ["Eric": 9, "Mark": 12, "Wayne": 1]
```

```
scores["Andrew"] = 0
```

```
["Eric": 9, "Mark": 12, "Andrew": 0, "Wayne": 1]
```

Hashable

(Swift) A type that provides an integer hash value

정수 hash value를 제공하는 타입 => 그 자체로 유일하게 표현이 가능해야 함

<https://bcho.tistory.com/1072>

Generic

상황에 따라 다른 타입의 데이터를 사용하고 싶을 때 사용함

타입 파라미터 **T**를 이용

중복을 피하고 의도를 명확하게 표현하고, 추상적인 방법으로 코드를 작성할 수 있다.

<https://zeddios.tistory.com/226>

<https://baked-corn.tistory.com/133>

<http://minsone.github.io/mac/ios/swift-generics-summary>

Array

일정한 순서가 있는 자료구조

랜덤 액세스가 가능함 => 조회가 Constant Time

(끝)삽입 => Constant Time or N Time

중간 삽입 => N Time

=> 일반적인 배열과 다소 다름

이유? <https://zeddios.tistory.com/117>

Array - 순서가 보장되는 구조

```
let people = ["Brian", "Stanley", "Ringo"]
```

```
people[0] // "Brian"  
people[1] // "Stanley"  
people[2] // "Ringo"
```

Set

중복값이 없는 컬렉션

Set - 중복값을 알아서 걸러냄 / Set이 유리한 프로그래밍

```
var bag: Set<String> = ["Candy", "Juice", "Gummy"]
bag.insert("Candy")
print(bag) // prints ["Candy", "Juice", "Gummy"]
```

```
let values: [String] = [...]
var bag: Set<String> = []
for value in values {
    if bag.contains(value) {
        // bag already has it, therefore it is a duplicate
    }
    bag.insert(value)
}
```