

02

Collection Type

Dart

컬렉션 타입 - List, Set, Map

리스트 타입

- list는 데이터를 여러 개 저장하고 인덱스값으로 데이터를 이용하는 컬렉션 타입
- 리스트를 선언하면서 초기화할 때는 대괄호([])를 이용

• 리스트 사용 예

```
main() {  
    List list1 = [10, 'hello', true];  
    list1[0] = 20;  
    list1[1] = 'world';  
    print('List : ${list1[0]}, ${list1[1]}, ${list1[2]}');  
}
```

▶ 실행 결과

```
List : [20, world, true]
```

컬렉션 타입 - List, Set, Map

리스트 타입

- 특정한 타입의 데이터만 저장하는 리스트를 선언할 때는 데이터 타입을 제네릭으로 명시

• 제네릭으로 타입을 지정한 예

```
main() {  
    List<int> list2 = [10, 20, 30];  
    list2[0] = 'hello'; // 오류  
}
```

- 데이터를 추가하거나 제거하려면 add()나 removeAt() 함수를 이용

• 리스트에 데이터 추가와 제거

```
main() {  
    List<int> list2 = [10, 20, 30];  
    print(list2);  
  
    list2.add(40);  
    list2.add(50);  
    print(list2);  
  
    list2.removeAt(0);  
    print(list2);  
}
```

컬렉션 타입 - List, Set, Map

리스트 타입

- filled(), generate()는 List 클래스에 선언된 생성자
- 데이터를 몇 개 저장할지 크기를 지정

```
• 리스트 크기 지정하기

main() {
    var list3 = List<int>.filled(3, 0);
    print(list3);    // [0, 0, 0]

    list3[0] = 10;
    list3[1] = 20;
    list3[2] = 30;
    print(list3);    // [10, 20, 30]

    list3.add(40);    // 런 타임 오류
}
```

컬렉션 타입 - List, Set, Map

리스트 타입

- 처음에 지정한 크기보다 많은 데이터를 저장할 수 있도록 허용하려면 `filled()` 생성자에 `growable` 매개변수를 `true`로 지정

```
• 리스트 확장 가능하게 하기

main() {
    var list3 = List<int>.filled(3, 0, growable: true);
    print(list3);

    list3[0] = 10;
    list3[1] = 20;
    list3[2] = 30;
    print(list3);

    list3.add(40);
    print(list3);
}
```

▶ 실행 결과

```
[0, 0, 0]
[10, 20, 30]
[10, 20, 30, 40]
```

컬렉션 타입 - List, Set, Map

리스트 타입

- 리스트를 초기화할 때 특정한 로직으로 구성된 데이터를 지정

• 특정한 로직으로 리스트 초기화하기

```
main() {  
    var list4 = List<int>.generate(3, (index) => index * 10, growable: true);  
    print(list4);  
}
```

▶ 실행 결과

[0, 10, 20]

컬렉션 타입 - List, Set, Map

집합 타입

- Set은 List와 마찬가지로 여러 건의 데이터를 저장하는 컬렉션 타입
- 리스트와 차이가 있다면 중복 데이터를 허용하지 않습니다.

• 집합 사용 예

```
main() {  
    Set<int> set1 = {10, 20, 10};  
    print(set1);  
    set1.add(30);  
    set1.add(40);  
    print(set1);  
  
    Set<int> set2 = Set();  
    set2.add(10);  
    set2.add(20);  
    print(set2);  
}
```

▶ 실행 결과

```
{10, 20}  
{10, 20, 30, 40}  
{10, 20}
```

컬렉션 타입 - List, Set, Map

맵 타입

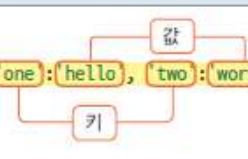
- Map은 여러 건의 데이터를 키와 값 형태로 저장하는 타입

• 맵 사용 예

```
main() {  
    Map<String, String> map1 = {"one": "hello", "two": "world"};  
  
    print(map1["one"]);  
    map1["one"] = "world";  
    print(map1["one"]);  
}
```

▶ 실행 결과

```
hello  
world
```



The diagram illustrates the internal structure of the Map. It shows two entries: one with the key 'one' and value 'hello', and another with the key 'two' and value 'world'. The keys are grouped under a box labeled '키' (Key), and the values are grouped under a box labeled '값' (Value).



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare