

3강

지능형 자료구조(튜플)

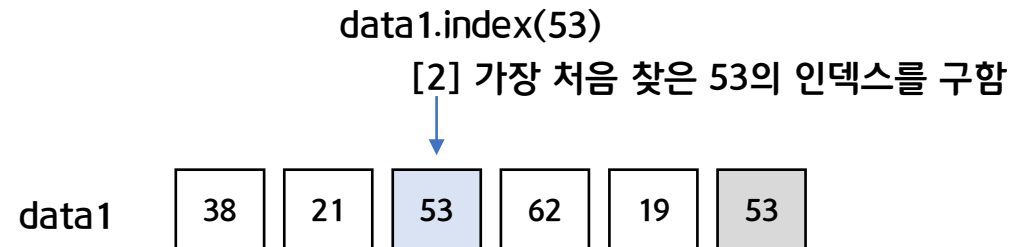
3-1 튜플 조작

● 튜플 함수

- ✓ 튜플은 리스트와는 달리 내용을 변경할 수 없다.
- ✓ 리스트와 같이 내용을 변경하는 append 같은 메서드는 사용할 수 없고, 요소의 정보를 구하는 메서드만 사용할 수 있다.
- ✓ Index()는 매개 변수 값이 위치한 인덱스를 리턴한다. 같은 값이 여러 개일 경우 처음 찾은 인덱스를 반환한다.

```
data1 = 38,21,53,62,19,53  
print(type(data1))  
print(data1.index(53))
```

[결과]
<class 'tuple'>
2



3-1 튜플 조작

● 튜플 함수

- ✓ count()는 매개 변수 값의 개수를 리턴한다.

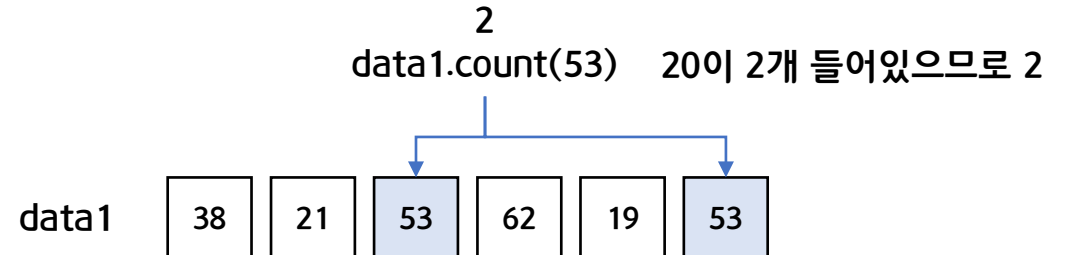
```
data1 = 38,21,53,62,19,53  
print(type(data1))  
print(data1.count(53))
```

[결과]
2

- ✓ 튜플 요소 출력

```
data3 = 38,21,53,62,19  
for i in data3:  
    print(i, end=' ')
```

[결과]
38 21 53 62 19



3-2 튜플 컴프리헨션과 map

● 튜플 컴프리헨션 사용

- ✓ 파이썬에서는 리스트와 마찬가지로 튜플을 생성할 때도 컴프리헨션 문법을 지원한다.
- ✓ 튜플을 사용할 때는 ()를 사용하지 않고 tuple()을 이용해야 한다.
- ✓ tuple을 사용하지 않으면 generator 객체를 생성한다.

```
data1 = tuple(i for i in range(10) if i%2==0)
print(data1)
data2 = (i for i in range(10) if i%2==0)
print(data2)
```

[결과]

(0, 2, 4, 6, 8)

<generator object <genexpr> at 0x0000025DDF476E48>

3-2 튜플 컴프리헨션과 map

● map()

✓ map()는 자료구조의 각 요소를 지정된 함수로 처리해준다.

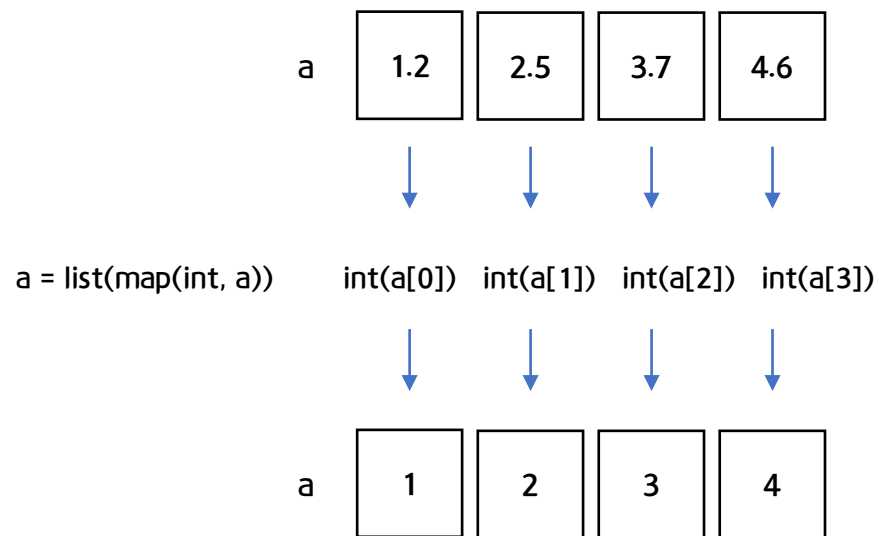
```
data1 = [1.2, 2.5, 3.7, 4.6]
for i in range(len(data1)):
    data1[i] = int(data1[i])
print(data1)

data2 = [1.2, 2.5, 3.7, 4.6]
data2 = list(map(int, data2))
print(data2)
```

[결과]

[1, 2, 3, 4]

[1, 2, 3, 4]



3-2 튜플 컴프리헨션과 map

● map()

```
data3 = input().split()  
print(data3)
```

[결과]
300 500 <입력>
['300', '500']

✓ map() 사용하면 쉽게 변수로 구성할 수 있다.

```
data4 = list(map(int, input().split()))  
print(data4)
```

[결과]
300 500 <입력>
[300, 500]

✓ map()가 반환하는 맵 객체는 언패킹(unpacking)이 가능하다.

```
data5, data6 = list(map(int, input().split()))  
print(data5, data6)
```

[결과]
300 500 <입력>
300 500