

2강

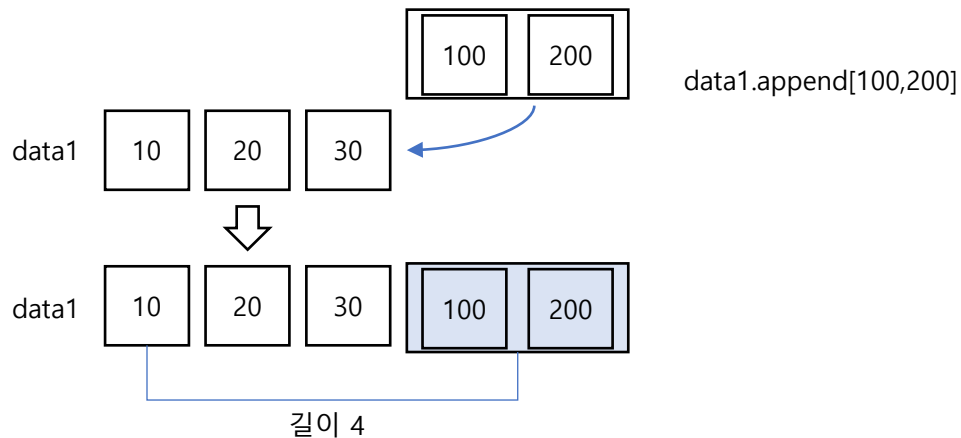
지능형 자료구조(리스트)

2-1 리스트 조작

● 리스트 안에 리스트 추가

```
data1 = [10,20,30]
data1.append([100,200])
print(data1)
print(len(data1))
```

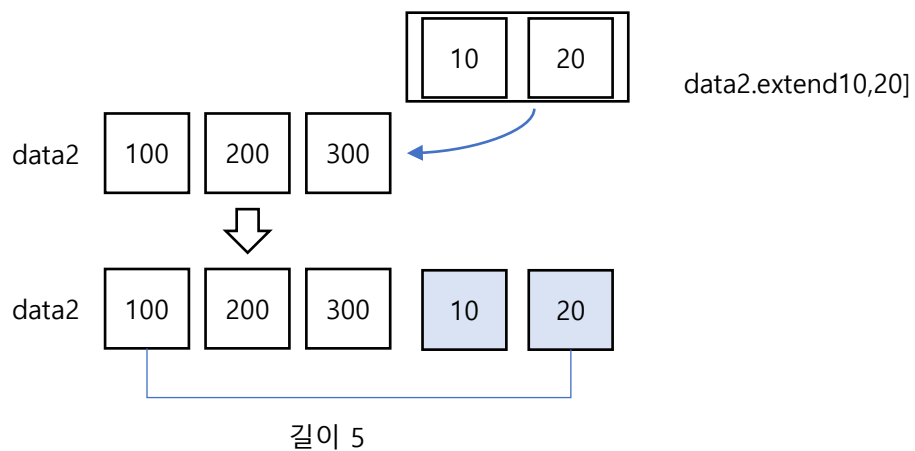
[결과]
[10, 20, 30, [100, 200]]
4



● 리스트 확장

```
data2 = [100,200,300]
data2.extend([10,20])
print(data2)
print(len(data2))
```

[결과]
[100, 200, 300, 10, 20]
5



2-1 리스트 조작

● 리스트 요소 삭제

- ✓ pop()는 마지막 요소 및 인덱스 요소를 삭제한다.

```
data3 = [10,20,30,40]  
print(data3.pop())  
print(data3)
```

[결과]

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

```
print(data3.pop(1))  
print(data3)
```

[결과]

```
20  
[10, 30]
```

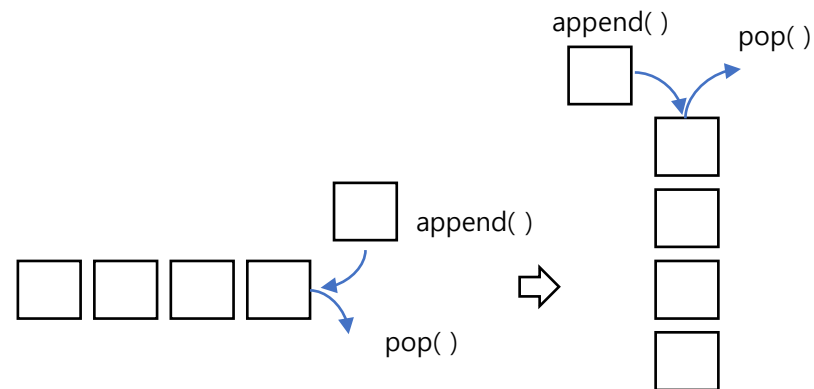


그림1 pop()를 이용한 스택

2-2 리스트 할당과 복사

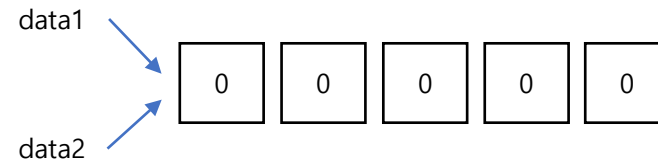
● 리스트 할당

- ✓ 리스트를 아래와 같이 생성 후 다른 변수에 할당한다.
- ✓ `data1 = data2`로 할당하면 두 변수는 같은 리스트를 지정하고 있다.

```
data1 = [0,0,0,0,0]  
data2 = data1  
print(data1 is data2)
```

[결과]

True



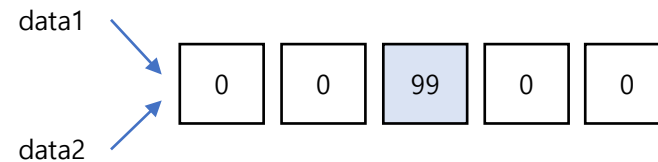
- ✓ `data1`와 `data2`는 같은 리스트이므로 `data2`를 변경하면 `data1`에도 반영이 된다.

```
data2[2] = 99  
print(data2)  
print(data1)
```

[결과]

[0, 0, 99, 0, 0]

[0, 0, 99, 0, 0]



2-2 리스트 할당과 복사

● 리스트 복사

- ✓ 리스트를 복사하기 위해서는 `copy()`를 이용한다.
- ✓ `data3`와 `data4`는 각각 다른 리스트이므로 `data3`의 값을 변경해도 `data4`의 값은 변하지 않는다.

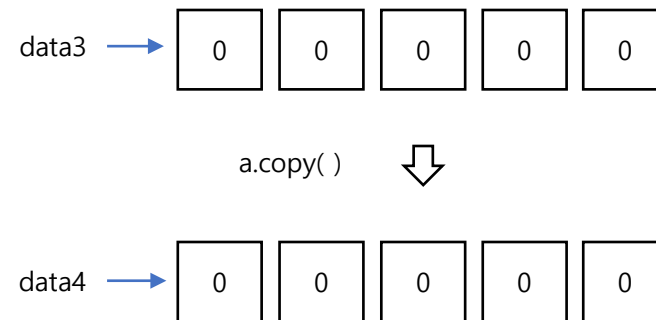
```
data3 = [0,0,0,0,0]
data4 = data3.copy()
print(data3 is data4)
data3[2] = 99
print(data3)
print(data4)
```

[결과]

False

[0, 0, 99, 0, 0]

[0, 0, 0, 0, 0]



2-3 리스트 컴프리헨션

● 리스트 컴프리헨션 사용

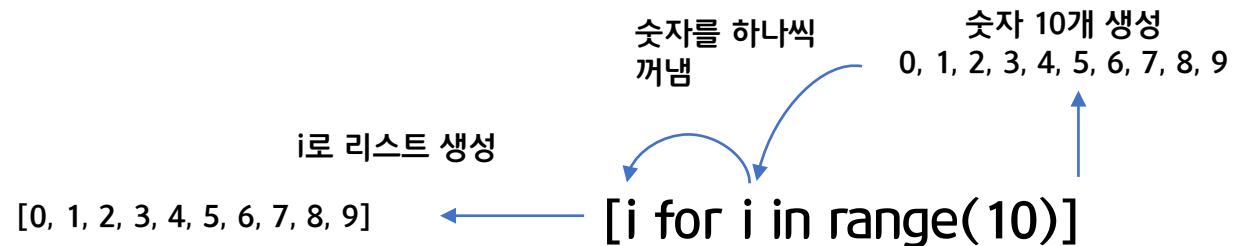
- ✓ 컴프리헨션은 이해, 이해력, 포용, 포용력, 포함, 압축 등의 뜻을 가지고 있다.
- ✓ 파이썬에서는 컴프리헨션을 내포, 내장, 축약 등으로 쓰인다.

즉 코드를 통해 축약된 형태로 자료구조를 생성하는 것이 컴프리헨션이다.

```
data1 = []  
for i in range(10):  
    data1.append(i)  
print(data1)  
  
data2 = [i for i in range(10)]  
print(data2)
```

[결과]

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```



2-3 리스트 컴프리헨션

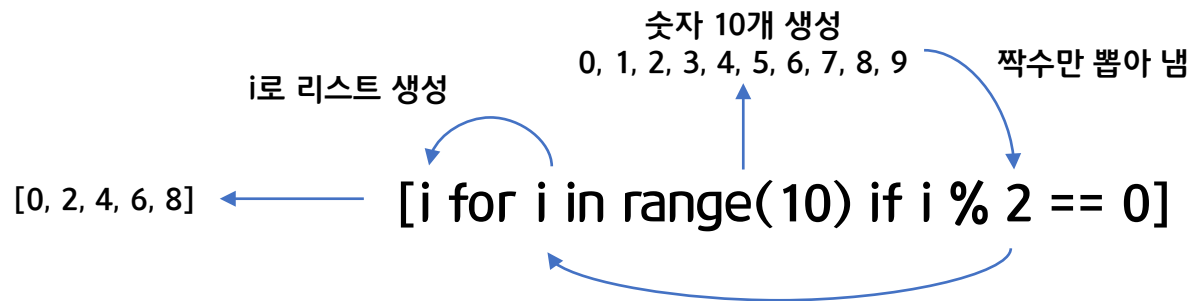
● 컴프리헨션에 if 조건문 사용

- ✓ 컴프리헨션을 통해 리스트를 구성할 때 if문을 이용할 수 있다.
- ✓ for 반복문 뒤에 if 조건문을 지정하면 숫자를 생성한 뒤 if문에서 특정 숫자만 뽑아내서 리스트를 생성할 수 있다.

```
data3 = []  
for i in range(10):  
    if i % 2 == 0:  
        data3.append(i)  
print(data3)  
  
data4 = [i for i in range(10) if i % 2 == 0]  
print(data4)
```

[결과]

[0, 2, 4, 6, 8]
[0, 2, 4, 6, 8]

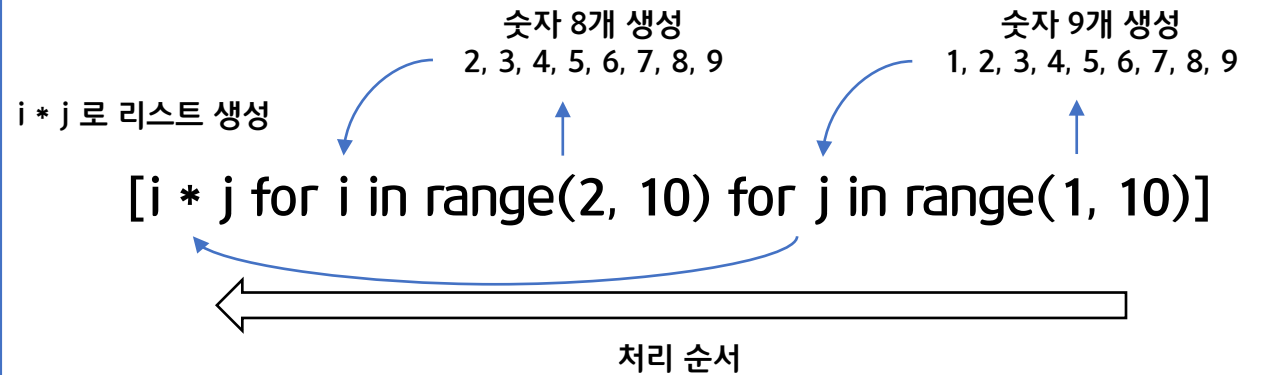


2-3 리스트 컴프리헨션

● 컴프리헨션에 중복 for문 사용

- ✓ 컴프리헨션을 통해 리스트를 구성할 때 중복 for문을 이용할 수 있다.
- ✓ 중복 for문의 처리 순서는 아래와 같다.

```
data5 = []  
for i in range(2,10):  
    for j in range(1,10):  
        data5.append(i*j)  
print(data5)  
  
data6 = [i * j for i in range(2,10) for j in range(1,10)]  
print(data6)
```



[결과]

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 3, 6, 9, 12, 15, 18, 21, 24, 27, 4, 8, 12, 16, 20, 24, 28, 32, 36, 5, 10, 15, 20, 25, 30, 35, 40, 45, 6, 12, 18, 24, 30, 36, 42, 48, 54, 7, 14, 21, 28, 35, 42, 49, 56, 63, 8, 16, 24, 32, 40, 48, 56, 64, 72, 9, 18, 27, 36, 45, 54, 63, 72, 81]  
[2, 4, 6, 8, 10, 12, 14, 16, 18, 3, 6, 9, 12, 15, 18, 21, 24, 27, 4, 8, 12, 16, 20, 24, 28, 32, 36, 5, 10, 15, 20, 25, 30, 35, 40, 45, 6, 12, 18, 24, 30, 36, 42, 48, 54, 7, 14, 21, 28, 35, 42, 49, 56, 63, 8, 16, 24, 32, 40, 48, 56, 64, 72, 9, 18, 27, 36, 45, 54, 63, 72, 81]
```