

리스트 컴프리헨션(list comprehension)

리스트 내포, 리스트 내장, 리스트 축약, 리스트 해석



리스트 컴프리헨션

- 식으로 지정해서 생성된 것을 리스트로 구성하기 위한 기능
- 리스트 안에 식, for 반복문을 지정
- (장점) 여러 줄의 코드를 한 줄로 줄일 수 있음

[식 for 변수 in 리스트]
list(식 for 변수 in 리스트)

```
a = [i for i in range(10)]    # 0부터 9까지 숫자를 생성하여 리스트 생성  
b = list(i for i in range(10)) # 0부터 9까지 숫자를 생성하여 리스트 생성
```



리스트 컴프리헨션

숫자를 하나씩 꺼냄 숫자 10개 생성
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

i로 리스트 생성

`[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]` ← `[i for i in range(10)]`

0부터 9까지 숫자를 생성하면서 값에 5를 더하여 리스트 c 생성



(확장) 리스트 컴프리헨션

- 리스트 표현식에서 if 조건문 사용하기
- if 조건문은 for 반복문 뒤에 지정

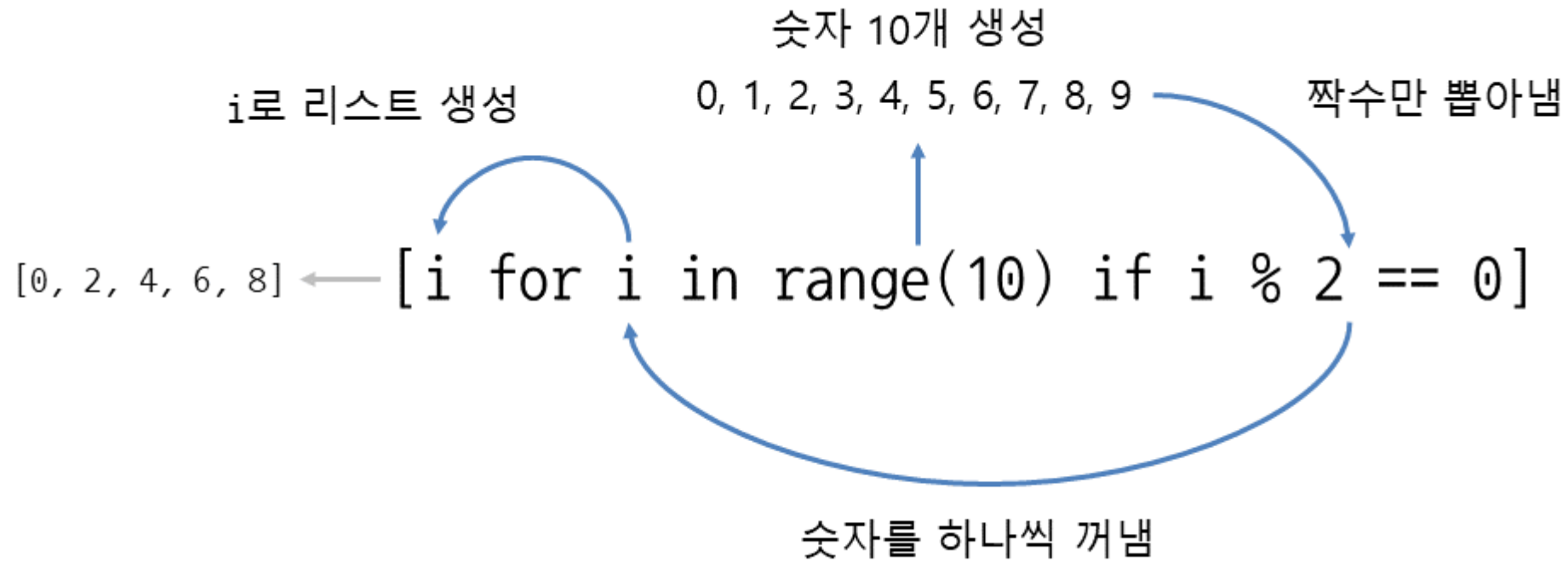
[식 for 변수 in 리스트 if 조건식]

list(식 for 변수 in 리스트 if 조건식)

`a = [i for i in range(10) if i % 2 == 0]` # 0~9 숫자 중 2의 배수인 숫자로 리스트 생성



(확장) 리스트 컴프리헨션



(확장) 리스트 컴프리헨션

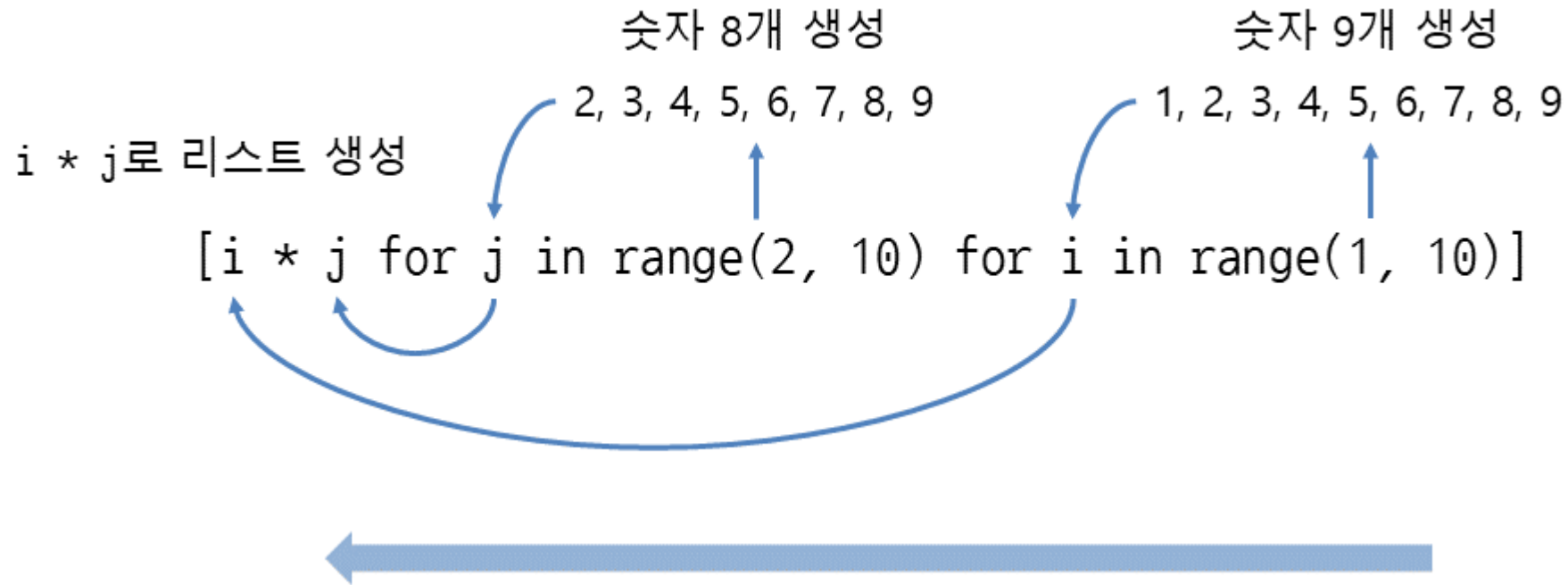
- for 반복문과 if 조건문을 여러 번 사용하기
- for가 여러 개일 때 처리 순서는 뒤에서 앞으로 처리됨

```
a = [i * j for j in range(2, 10) for i in range(1, 10)]
```

```
a = [i * j for j in range(2, 10)  
      for i in range(1, 10)]
```



(확장) 리스트 컴프리헨션



(확장) 리스트 컴프리헨션

```
# matrix를 리스트 하나로 표현하기
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
flat = [x for row in matrix for x in row]
print(flat)
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
# matrix의 각 행(셀)의 제곱을 구하기
squared = [[x**2 for x in row] for row in matrix]
print(squared)
[[1, 4, 9], [16, 25, 36], [49, 64, 81]]
```

