

20. List Comprehension (리스트 컴프리헨션)

- `list_variable = [x for x in iterable]` 과 같이 list 내에 for loop 표현식을 사용하여 쉽게 새로운 list 를 생성할 수 있다.
- list comprehension 을 사용하면 for loop 과 map, filter 함수를 대체할 수 있다.
- list comprehension 내에서 if ~ else ~ 를 사용할 수 있다.

In [1]:

```
1 nums = [1, 2, 3, 4]
2
3 squares = [n **2 for n in nums]
4
5 print(squares)
```

[1, 4, 9, 16]

In [2]:

```
1 strs = ['hello', 'and', 'goodbye']
2
3 shouting = [s.upper() + '!!!' for s in strs]
4
5 print(shouting)
```

['HELLO!!!', 'AND!!!', 'GOODBYE!!!']

In [3]:

```
1 nums = [2, 8, 1, 6]
2
3 small = [n for n in nums if n <= 2]
4
5 small
```

Out[3]:

[2, 1]

In [4]:

```
1 # a 가 포함된 과일명을 대문자로 바꾸기
2 fruits = ['apple', 'banana', 'cherry', 'lemon']
3
4 [fruit.upper() for fruit in fruits if 'a' in fruit]
```

Out[4]:

['APPLE', 'BANANA']

In [5]:

```
1 # 1-50 까지의 홀수 list 작성
2 # using for loop & 조건문
3 result = []
4 for i in range(1, 51):
5     if i % 2 == 1:
6         result.append(i)
7 print(result)
```

[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]

In [6]:

```
1 # using list comprehension
2 a = [x for x in range(1, 51) if x % 2 == 1]
3 print(a)
```

[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]

list comprehension 은 빠르다

In [7]:

```
1 %timeit [x for x in range(1, 51) if x % 2 == 1]
```

3.85 μ s \pm 1.05 μ s per loop (mean \pm std. dev. of 7 runs, 100000 loops each)

In [8]:

```
1 %%timeit
2 result = []
3 for i in range(1, 51):
4     if i % 2 == 1:
5         result.append(i)
```

4.41 μ s \pm 504 ns per loop (mean \pm std. dev. of 7 runs, 100000 loops each)

map, filter 함수를 대체하는 list comprehension

In [9]:

```
1 # map + lambda
2
3 list(map(lambda x: x*x, [1,2,3,4,5]))
```

Out[9]:

[1, 4, 9, 16, 25]

In [10]:

```
1 [x*x for x in [1,2,3,4,5]]
```

Out[10]:

```
[1, 4, 9, 16, 25]
```

In [11]:

```
1 # filter + lambda
2
3 list(filter(lambda x: x > 5, [1,2,3,4,5,6,7,8,9]))
```

Out[11]:

```
[6, 7, 8, 9]
```

In [12]:

```
1 [x for x in [1,2,3,4,5,6,7,8,9] if x > 5]
```

Out[12]:

```
[6, 7, 8, 9]
```

list comprehension 내의 ifelse

In [13]:

```
1 [x if x > 5 else x * 2 for x in [1,2,3,4,5,6,7,8,9]]
```

Out[13]:

```
[2, 4, 6, 8, 10, 6, 7, 8, 9]
```

연습문제

alist = [('x', 4), ('s', 5), ('a', 4), ('t', 5), ('k', 4), ('w', 5)] 와 같이 tuple element 로 구성된 list 에서 tuple 의 첫번째 element 가 'a' 인 tuple 이 alist 의 몇번째 element 인지 구하라. 단, for loop 과 비교문을 사용하지 않고 list comprehension 을 사용.

-(Hint) enumerate : list 를 index, element tuple 로 반환

In [21]:

```
1 alist = ['a', 'b', 'c', 'd']
2 list(enumerate(alist))
```

Out[21]:

```
[(0, 'a'), (1, 'b'), (2, 'c'), (3, 'd')]
```

In [20]:

```
1 alist = [('x', 4), ('s', 5), ('a', 4), ('t', 5), ('k', 4), ('w', 5), ('a', 4)]
2
3 # CODE HERE
```