

## Chapter9 – Pythonic code2

### Lambda

-함수 이름 없이, 함수처럼 쓸 수 있는 익명함수

General function	Lambda function
<pre>def f(x, y):     return x + y  print(f(1, 4))</pre>	<pre>f = lambda x, y: x + y print(f(1, 4))</pre>

-f으로 정의하지 않아도 사용가능 ex) (lambda x,y : x+y)(1, 10)

### Map & Reduce

#### Map function

-Sequence 자료형 각 element에 동일한 function을 적용함

```
ex = [1, 2, 3, 4, 5]  
f = lambda x, y: x + y  
map(function_name, list_data) print(list(map(f, ex, ex))) [2, 4, 6, 8, 10]
```

-두 개 이상의 list에도 적용 가능함, if filter도 사용가능

```
ex = [1,2,3,4,5]  
print(list(map(lambda x: x+x, ex)))  
print((map(lambda x: x+x, ex)))  
  
f = lambda x: x ** 2  
print(map(f, ex))  
for i in map(f, ex):  
    print(i)
```

두 번째 줄과 같이 list를 붙여줘야 list를 사용 할 수 있다.

세 번째 줄은 generator상태,(실행시점의 값을 생성, 메모리를 효율적으로 사용)

#### Reduce function

-map function과 달리 list에 똑같은 함수를 적용해서 통합

```
from functools import reduce  
print(reduce(lambda x, y: x+y, [1, 2, 3, 4, 5]))
```

1	2	3	4	5
---	---	---	---	---

결과값은 15

```
def factorial(n):
    return reduce(
        lambda x, y: x*y, range(1, n+1))

factorial(5)
```

팩토리얼 함수를 사용할때도 참고

결론 : lambda나 reduce는 코드의 직관성이 떨어져서 python3에서 사용을 권장하지 않음

## Asterisk

-, 단순 곱셈, 제곱연산, 가변 인자 활용 등 다양하게 사용된다.

<pre>def asterisk_test(a, *args):     print(a, args)     print(type(args))  asterisk_test(1,2,3,4,5,6)</pre>	<pre>def asterisk_test(a, **kargs):     print(a, kargs)     print(type(kargs))  asterisk_test(1, b=2, c=3, d=4, e=5, f=6)</pre>
<pre>1 (2, 3, 4, 5, 6) &lt;class 'tuple'&gt;</pre>	<pre>1 {'b': 2, 'c': 3, 'd': 4, 'e': 5, 'f': 6} &lt;class 'dict'&gt;</pre>

## Asterisk – unpacking a container

-tuple, dict 등 자료형에 들어가 있는 값을 unpacking

<pre>def asterisk_test(a, *args):     print(a, args)     print(type(args))  asterisk_test(1, *(2, 3, 4, 5, 6))</pre>	<pre>def asterisk_test(a, *args):     print(a, args)     print(type(args))  asterisk_test(1, (2, 3, 4, 5, 6))</pre>
<pre>1 (2, 3, 4, 5, 6) &lt;class 'tuple'&gt;</pre>	<pre>1 ((2, 3, 4, 5, 6),) &lt;class 'tuple'&gt;</pre>

```
def asterisk_test(a, args):
    print(a, *args)
    print(type(args))

asterisk_test(1, (2,3,4,5,6))
```

```
1 2 3 4 5 6
<class 'tuple'>
```

\* 를 print가 있는 구문에 적어서 unpacking하는 과정

```
a, b, c = ([1, 2], [3, 4], [5, 6])
print(a, b, c) [1, 2] [3, 4] [5, 6]
```

```
data = ([1, 2], [3, 4], [5, 6])
print(*data) [1, 2] [3, 4] [5, 6]
```

\*를 두번 사용하면(\*\*) dict type도 처리가 가능하다.