Lambda & MapReduce

Pythonic Code

최성철 교수
Director of TEAMLAB

Lambda

Lambda

- 함수 이름 없이, 함수처럼 쓸 수 있는 익명함수
- 수학의 람다 대수에서 유래함

General function

```
def f(x, y):
    return x + y

print(f(1, 4))
```

Lambda function

```
f = lambda x, y: x + y
print(f(1, 4))
```

Lambda

- Python 3부터는 권장하지는 않으나 여전히 많이 쓰임

```
f = lambda x, y: x + y
print(f(1, 4))
```

```
f = lambda x: x ** 2
print(f(3))
```

```
f = lambda x: x / 2
print(f(3))
```

```
print((lambda x: x +1)(5))
```

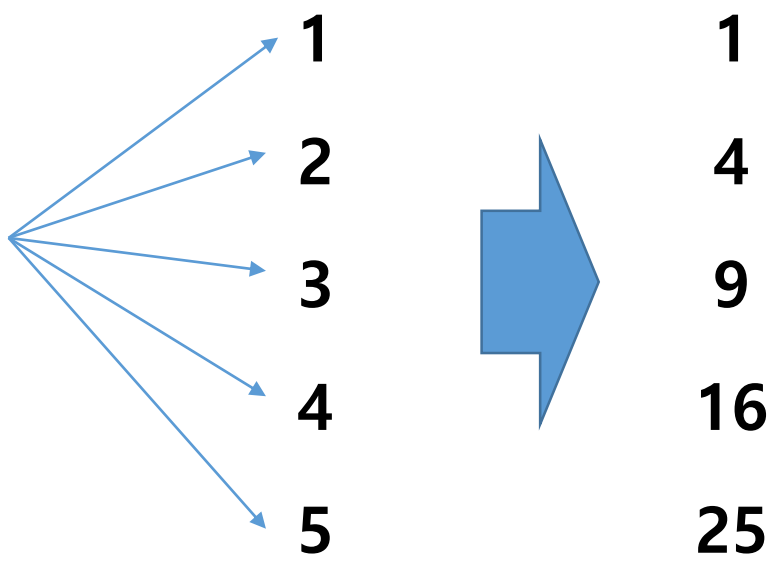
Map & Reduce

Map function

- Sequence 자료형 각 element에 동일한 function을 적용함

map(function_name, list_data)

lambda x: x ** 2




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


f = lambda x, y: x + y
print(list(map(f, ex, ex)))
```

Map function

- 두 개 이상의 list에도 적용 가능함, if filter도 사용가능

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

```
list(
  map(
  lambda x: x ** 2 if x % 2 == 0
  else x,
  ex)
)
```

Map function

- python3 는 iteration을 생성 → list을 붙여줘야 list 사용가능
- 실행시점의 값을 생성, 메모리 효율적

```
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)
```

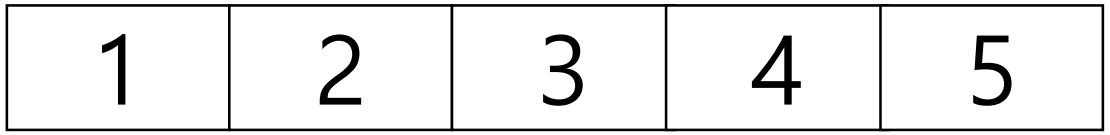
```
result = map(f, ex)
print(next(result))
```

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]))
```



Summary

- Lambda, map, reduce는 간단한 코드로 다양한 기능을 제공

- 그러나 코드의 직관성이 떨어져서 lambda나 reduce는 python3에서 사용을 권장하지 않음

- Legacy library나 다양한 머신러닝 코드에서 여전히 사용중

Human knowledge belongs to the world.