

**Lambda & MapReduce**

**Pythonic Code**

**최성철 교수**  
**Director of TEAMLAB**

01100  
00110

A 3D rendering of a white robot head, split vertically. The left side is dark grey, and the right side is white. The robot has large, circular, glowing eyes. Its right hand is raised, holding a blue binary code (01100, 00110) in its palm. The background is white.

**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

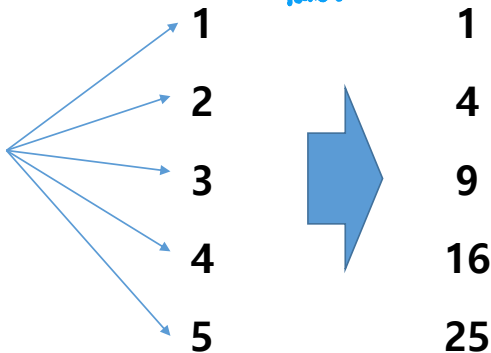
- Sequence 자료형 각 element에 동일한 function을 적용함

list나 tuple  
같은 거.

map(function\_name, list\_data)

element 하나하나에  
function이 적용.

lambda x: x \*\* 2



```
ex = [1,2,3,4,5]
```

```
f = lambda x: x ** 2
```

```
print(list(map(f, ex)))
```

list를 붙여서 결과값이 나옴.

```
f = lambda x, y: x + y
```

zip과 같은 효과.

```
print(list(map(f, ex, ex)))
```

[1, 2, 3, 4, 5]

map ( ) 의 첫 번째는 주소값.

# 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))
```



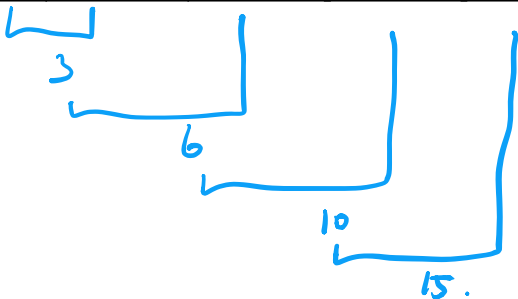
# Reduce function

- map function과 달리 list에 똑같은 함수를 적용해서 통합

```
from functools import reduce
```

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

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



각각의 element를  
적용시켜서 나가는 것.

factorial 같은 거에서 유용하게  
쓰인다.

---

# 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.**