

7강

람다

## 7-1 람다 표현식

### ● 람다

- ✓ 람다 표현식(lambda expression)은 간단한 식 형태로 구성 되어 있다.
- ✓ 람다 표현식은 함수를 간편하게 작성할 수 있어서 다른 함수의 인수로 넣을 때 주로 사용한다.

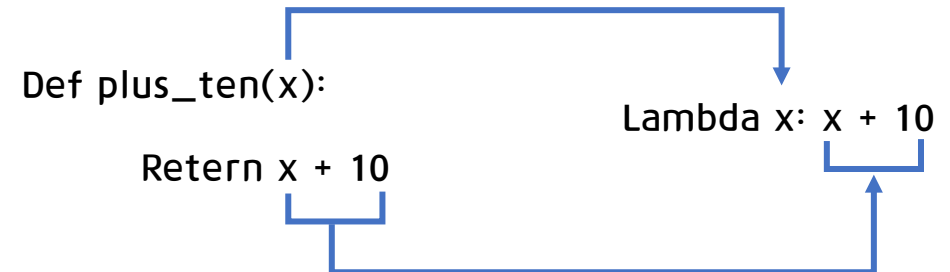
```
def plus_ten(x):  
    return x + 10  
  
print(plus_ten(10))
```

[결과]  
20

- ✓ 람다 표현식은 아래와 같이 lambda에 매개변수를 지정하고 : 뒤에 반환 값으로 사용할 식을 정의한다.

```
lf = lambda x : x+10  
print(lf(10))
```

[결과]  
20



- ✓ 람다 표현식의 다른 이름은 익명 함수(anonymous function)이다.

## 7-1 람다 표현식

### ● 람다

- ✓ 람다 표현식에서 반환 값 부분은 변수 없이, 식 한 줄로 표현해야 한다.
- ✓ 람다 표현식은 변수를 만들 수 없기 때문에 변수가 필요한 코드일 경우에는 def로 함수를 사용하는 것이 좋다.

```
print((lambda x: y=10 x + y)(10))
```

[결과]

SyntaxError: invalid syntax

- ✓ 바깥에 있는 변수는 람다 표현식에서 사용할 수 있다.

```
y = 20  
print((lambda x : x + y)(5))
```

[결과]

25

## 7-2 고위식

### ● 람다와 map()

- ✓ map(매개변수1, 매개변수2)의 매개변수1에 함수나 람다를 넣어 매개변수2에 전달되는 자료 요소 하나하나를 함수나 람다에 적용 할 수 있다.

```
def plus_ten(x):  
    return x + 10  
  
d1 = [5,2,8,9]  
data1 = list(map(plus_ten, d1))  
print(data1)  
data2 = list(map(lambda x : x+10, d1))  
print(data2)
```

[결과]

```
[15, 12, 18, 19]  
[15, 12, 18, 19]
```

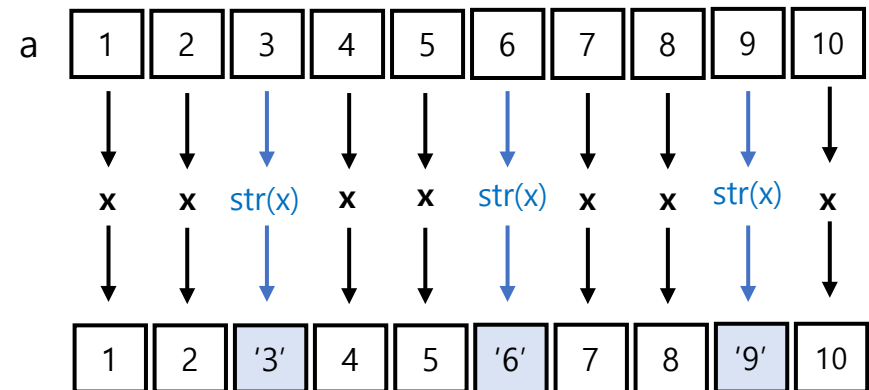
- ✓ 람다 표현식에도 조건부 표현식을 적용할 수 있다.

```
a = list(range(1,11))  
data1 = list(map(lambda x: str(x) if x % 3 == 0 else x, a))  
print(data1)
```

[결과]

```
[1, 2, '3', 4, 5, '6', 7, 8, '9', 10]
```

list(map(lambda x: str(x) if x % 3 == 0 else x, a))



## 7-2 고위식

### ● 멀티 객체 적용

- ✓ 람다 표현식의 매개변수 개수에 맞게 반복 가능한 객체도 콤마로 구분해서 넣어주면 된다.

```
data1 = [1,2,3,4,5]
data2 = [20,40,50,60,80]
print(list(map(lambda x,y: x * y, data1, data2)))
```

[결과]  
[20, 80, 150, 240, 400]

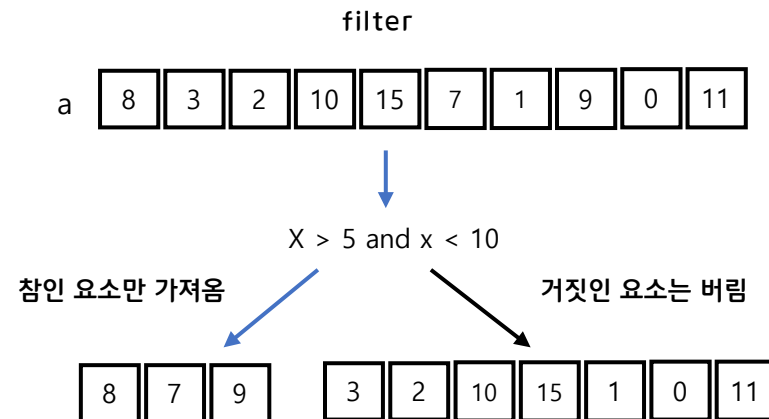
### ● filter() 적용

- ✓ filter는 반복 가능한 객체에서 특정 조건에 맞는 요소만 가져오는데, filter에 지정한 함수의 반환 값이 True일 때만 해당 요소를 가져온다.

```
def f(x):
    return x > 5 and x < 10

data = [8,3,2,10,15,7,1,9,0,11]
result = list(filter(f, data))
print(result)
```

[결과]  
[8, 7, 9]



## 7-2 고위식

### ● reduce() 적용

- ✓ reduce는 반복 가능한 객체의 각 요소를 지정된 함수로 처리한 뒤 이전 결과와 누적하여 생성된 값을 반환하는 함수이다.
- ✓ reduce 함수를 사용하기 위해서는 functools 모듈을 가져와야 한다.

```
from functools import reduce
```

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

```
data = [1,2,3,4,5]  
print(reduce(f, data))
```

[결과]

15

reduce

