12.Python 람다 함수

강사 강태광

1. Python 함수

1] **람다 함수** 정의

-일반적인 함수를 한 줄의 문(Statement)으로 정의할 수 있는 새로운 함수 정의 리터럴

-함수 몸체에는 식(expression)만이 올 수 있다.

- 대부분의 경우 함수 이름을 정의하지 않으면서 일회성으로 활용할 함수를 정의할 때 활용

예시 1(인수가 두 개 있는 람다 함수를 지니는 변수 지정 및 함수 호출)

```
g = lambda x, y: x + y
print ("g(1, 2)->", g(1, 2))
# 기본 인수를 지니는 람다 함수 정의
incr = lambda x, inc = 1: x + inc
print ("incr(10)->", incr(10))  # inc 기본 인수 값으로 1 사용
print ("incr(10, 5)->", incr(10, 5))
```

결과

```
------------------------------------------------------------
g(1, 2)-> 3
incr(10)-> 11
incr(10, 5)-> 15
```

예시 2(가변 인수를 지니는 람다 함수 정의)

```
vargs = lambda x, *args: args   # *args → 가변 인수
print (vargs(1,2,3,4,5)     )    # (1, 2, 3, 4, 5) → 1 은 x, 나머지는 튜플형태로 args에 할당됨
```

결과

```
------------------------------------------------------------
g(1, 2)-> 3
incr(10)-> 11
incr(10, 5)-> 15
```

1. Python 함수

예시 3(더하기, 빼기, 곱하기, 나누기에 해당하는 람다 함수 리스트 정의)
-인덱스 0은 첫 번째 인자 / 인덱스 2는 세 번째 인자로 곱셈 수행 / 리스트의 원소에 람다함수가 들어가고, 검색도 가능

```
# 리스트의 원소가 람다함수로 들어감
func = [lambda x, y: x + y, lambda x, y: x - y, lambda x, y: x * y, lambda x, y: x / y]
def menu():
    print ("0. add" )
    print ("1. sub" )
    print ("2. mul" )
    print ("3. div" )
    print ("4. quit" )
    return input('Select menu:')

while 1:                          # while 1 → 무한 루프
    sel = int(menu())
    print("sel->",sel)
    print("len(func)->",len(func))
    if sel < 0 or sel > len(func):  # 0보다 작은 값이거나 4보다 큰 값(-1, -2, -3, 5, 6, 7, 8)은 수행 X
        continue
    if sel == len(func):
        break                     # 중간에 break 있어, 조건 미 충족 시 무한루프를 빠져나옴
    x = int(input('First operand:') )
    y = int(input('Second operand:') )
    print ('Result =', func[sel](x,y))
```

결과

```
------------------------------------------------------------
0. add
1. sub
2. mul
3. div
4. quit
Select menu:2
sel-> 2
len(func)-> 4
First operand:20
Second operand:5
Result = 100
```

1. Python 함수

2] map 내장 함수 정의

- seq 시퀀스 자료형이 지닌 각 원소값들에 대해 function에 적용한 결과를 동일
- 시퀀스 자료형으로 반환

예시 1

```
def f(x):
    return x * x
X = [1, 2, 3, 4, 5]
Y = map(f, X)      # map(함수, 시퀀스 자료형), 시퀀스 자료형의 첫 번째 원소를 함수 수행
# 시퀀스 자료형의 유형에 따라 반환되는 유형 결정 , map 함수는 쌍을 지어주는 것 , x 원소가 5개이면, 수행 결과의 원소도 5개
print (Y)
```

결과

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

예시 2(**map 내장 함수를 사용하지 않을 때 코드 비교**)

```
def f2(x):
    return x * x
X = [1, 2, 3, 4, 5]
Y = []
for x in X:
    y = f2(x)
    Y.append(y)   # append(y) → 하나씩 x를 함수에 대응 후 결과 값을 추가함
print ("Y->",Y)
```

결과

```
------------------------------------------------------------
Y-> [1, 4, 9, 16, 25]
```