파이썬 프로그래밍

람다 함수

한국기술교육대학교
온라인평생교육원

▣ 람다(lambda) 함수 정의

1. 람다 함수 정의 예

•람다(lambda) 함수 (or 축약 함수): https://wikidocs.net/64
- 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
- 구문(syntax)
  ◦lambda 콤마로 구분된 인수들: 식(expression)

•인수가 한 개 있는 람다 함수

```
f = lambda x: x + 1
print f(1)
```

```
2
```

▪ lamda x: x + 1 → 하나의 statement
▪콜론(:) 뒤에는 expression(식)만 올 수 있음
▪f 식별자로 람다함수를 가리키게 할 수 있음 → 람다함수도 하나의 객체

▣ 람다(lambda) 함수 정의

1. 람다 함수 정의 예

•인수가 두 개 있는 람다 함수를 지니는 변수 지정 및 함수 호출

```
g = lambda x, y: x + y
print g(1, 2)
```

```
3
```

▪(1,2) → (x,y)

•기본 인수를 지니는 람다 함수 정의

```
incr = lambda x, inc = 1: x + inc
print incr(10) #inc 기본 인수 값으로 1 사용
print incr(10, 5)
```

```
11
15
```

▪람다함수도 기본 인수를 가질 수 있음

▪inc = 1 → 기본 인수

▣ 람다(lambda) 함수 정의

1. 람다 함수 정의 예

•가변 인수를 지니는 람다 함수 정의

```
vargs = lambda x, *args: args
print vargs(1,2,3,4,5)
```

```
(2, 3, 4, 5)
```

▪가변 인수도 가질 수 있음
▪*args → 가변 인수
▪(1, 2, 3, 4, 5) → 1은 x, 나머지는 튜플형태로 args에 할당됨
▪args 변수가 리턴되는 것

▣ 람다(lambda) 함수 정의

2. 람다 함수 사용하기

```
def f1(x):
    return x*x + 3*x - 10

def f2(x):
    return x*x*x

def g(func):
    return [func(x) for x in range(-10, 10)]

print g(f1)
print g(f2)
```

```
[60, 44, 30, 18, 8, 0, -6, -10, -12, -12, -10, -6, 0, 8, 18, 30, 44, 60, 78, 98]
[-1000, -729, -512, -343, -216, -125, -64, -27, -8, -1, 0, 1, 8, 27, 64, 125, 216, 343,
512, 729]
```

▪람다 함수가 쓰이고 있지 않은 예제

▪함수 g 호출하는데 인자로 함수가 들어감 (함수도 객체이기 때문)

▪range(-10, 10) → [-10, -9, -8, ……, 7, 8, 9]

▪-10 * -10 + 3 * -10 -10 = 60

▪9 * 9 + 3 * 9 -9 = 98

▪TRUE

▪9 * 9 * 9 = 729

▣ 람다(lambda) 함수 정의

2. 람다 함수 사용하기

```
def g(func):
    return [func(x) for x in range(-10, 10)]

print g(lambda x: x*x + 3*x - 10)
print g(lambda x: x*x*x)
```

```
[60, 44, 30, 18, 8, 0, -6, -10, -12, -12, -10, -6, 0, 8, 18, 30, 44, 60, 78, 98]
[-1000, -729, -512, -343, -216, -125, -64, -27, -8, -1, 0, 1, 8, 27, 64, 125, 216, 343,
512, 729]
```

- 같은 예제를 람다함수로 간단히 수행 가능
- func에 한 줄짜리 람다 함수가 그대로 들어감
- 정상적인 함수를 미리 구현 X → 함수를 호출하는 순간 함수를 정의

▣ 람다(lambda) 함수 정의

2. 람다 함수 사용하기

•람다 함수를 사용하는 코드 예제

```
# 더하기, 빼기, 곱하기, 나누기에 해당하는 람다 함수 리스트 정의
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:
    sel = menu()
    if sel < 0 or sel > len(func):
        continue
    if sel == len(func):
        break
    x = input('First operand:')
    y = input('Second operand:')
    print 'Result =', func[sel](x,y)
```

```
0. add
1. sub
2. mul
3. div
4. quit
Select menu : 0
First operand : 10
Second operand : 20
Result = 30
```

▣ 람다(lambda) 함수 정의

2. 람다 함수 사용하기

```
0. add
1. sub
2. mul
3. div
4. quit
Select menu : 1
First operand : 20
Second operand : 10
Result = 10

0. add
1. sub
2. mul
3. div
4. quit
Select menu : 2
First operand : 5
Second operand : 7
Result = 35

0. add
1. sub
2. mul
3. div
4. quit
Select menu : 3
First operand : 10
Second operand : 2
Result = 5

0. add
1. sub
2. mul
3. div
4. quit
Select menu : 4
```

▣ 람다(lambda) 함수 정의

2. 람다 함수 사용하기

- ▪리스트의 원소가 람다함수로 들어감
- ▪while 1 → 무한 루프
- ▪중간에 break 있어, 조건 미 충족 시 무한루프를 빠져나옴
- ▪0보다 작은 값이거나 4보다 큰 값(-1, -2, -3, 5, 6, 7, 8)은 수행 X
- ▪func의 길이가 4라서 4를 넣으면 break가 걸려 끝날 수 있음
- ▪select menu : 0 → sel에는 0이 들어가 있음
- ▪인덱스 0은 첫 번째 인자
- ▪인덱스 2는 세 번째 인자로 곱셈 수행
- ▪리스트의 원소에 람다함수가 들어가고, 검색도 가능