머신러닝/딥러닝을 위한

파이썬(Python)

– function • lambda –

함수 – function

- 함수정의

  => 특정 기능을 수행하는 함수(function)는 파이썬에서 다음과 같이 정의함.

  def 함수이름 (입력1, 입력2,…):

```
def sum(x, y):
    s = x + y
    return s

result = sum(10, 20)
print(result)
```

```
30
```

- 함수 반환 값

  => 파이썬 함수는 한 개 이상의 return 값을 반환 할 수 있음.
  return 값은 콤마(,)로 분리하여 받거나 tuple 형태로 받을 수 있음

```
def multi_ret_func(x):
    return x+1, x+2, x+3    # return (x+1, x+2, x+3)

x = 100
y1, y2, y3 = multi_ret_func(x)  # (y1, y2, y3) = multi_ret_func(x)

print(y1, y2, y3)
```

```
101 102 103
```

함수 – function

- default parameter

  => 함수의 입력 파라미터에 기본 값을 지정하는 것을 말하며,

  => 이러한 디폴트 파라미터는, 함수가 호출되었을 경우 입력파라미터에 명시적인 값이 전달되지 않으면 기본으로 지정한 값을 사용하겠다는 의미

```
# 파이썬 default parameter

def print_name(name, count=2):

    for i in range(count):
        print("name ==", name)

print_name("DAVE")    # print_name("DAVE", 5)
```

```
name == DAVE
name == DAVE
```

- mutable / immutable parameter

  => 입력 파라미터가 mutable (list, dict, numpy 등) 데이터형인 경우는 와 원래의 데이터에 변형이 일어남

  => immutable (숫자, 문자, tuple 등) 은 원래의 데이터에 변형이 일어나짐 않음

```
# mutable,  immutable parameter

def mutable_immutable_func(int_x, input_list):

    int_x += 1
    input_list.append(100)

x = 1
test_list = [ 1, 2, 3 ]

mutable_immutable_func(x, test_list)

print("x ==", x, ", test_list ==", test_list)
```

```
x == 1 , test_list == [1, 2, 3, 100]
```

함수 – lambda

- 람다(lambda)는 한 줄로 함수를 작성하는 방법으로, 익명(anonymous)함수 또는 람다 표현식(lambda expression) 등으로 불림

- lambda는 다른 함수의 인수(parameter)로 넣을 때 주로 사용하며, 특히 머신러닝에서 미분을 계산하기 위해 필요한 수치 미분(numerical derivative)과 활성화 함수(activation function) 등을 표현할 때, 수학에서 함수 표현처럼 *f(x), f(x, y)* 등으로 사용되며, 마지막 표현으로 대체된다는 의미

C언어의 #define P printf 처럼 표현을 대체 해줌

함수명 = lambda 입력1, 입력2,... : 대체되는 표현식

```
# lambda 함수 선언

f = lambda x : x+100

for i in range(3):
    print(f(i))
```

```
100
101
102
```

f = lambda x : x+100

수학 함수표현 f(x) 를 x+100 으로 대체된다는 의미

함수 – lambda

```
# lambda 에서 입력값을 반드시 이용할 필요는 없음

def print_hello():

    print("hello python")

def test_lambda(s, t):

    print("input1 ==", s, ",  input2 ==", t)

s = 100    # 생각해보기 => s, t 선언 및 값 할당이 없으면 error
t = 200

fx = lambda x, y : test_lambda(s, t)    # fx(x,y) = test_lambda(s, t)
fy = lambda x, y : print_hello()        # fy(x,y) = print_hello()


fx(500, 1000)     # fx(500, 1000) = test_lambda(s, t)
fy(300, 600)      # fy(300, 600)  = print_hello()
```

```
input1 == 100 ,  input2 == 200
hello python
```

예제

- test_data = [[10, 20, 30], [1, 2, 3], [100, 200, 300]]

  xdata = [x[1:-1] for x in test_data]

  tdata = [x[-1] for x in test_data]

- 함수의 인자로 시작과 끝을 나타내는 숫자를 받아 시작부터 끝까지의 모든 정수값의 합을 반환하는 함수 add_start_to_end 를 작성하시오(시작값과 끝값을 포함).

- 함수의 인자로 문자열을 포함하는 리스트가 입력될 때 각 문자열의 첫 세 글자로만 구성된 리스트를 반환하는 함수 get_abbr 를 슬라이싱 기능을 이용하여 구현하시오.

  예를 들어, 함수의 입력으로 ['Seoul', 'Daegu', 'Kwangju', 'Jeju']가 입력될 때 함수의 반환값은 ['Seo', 'Dae', 'Kwa', 'Jej'] 임

- 입력값의 제곱을 리턴하는 함수를 일반적인 python 함수인 square 와 lambda 함수 f 로 구현하고, square(2) 와 f(2) 형태로 검증하시오