

# 인공지능 기반 설계 이론 및 사례 연구 4차) 딥러닝에 필요한 Python 문법

2020년 10월

김성신

기계시스템학부  
숙명여자대학교



# Contents

1. Python 특징
2. Python 기본 문법
  - 자료형
  - 제어문
  - Tools for Big Data
    - Numpy, Pandas, Matplotlib
  - Images processing
    - OpenCV
3. Python 실습 예제

# Python 특징

- Python은 1991년 프로그래머인 귀도 반 로섬이 개발한 인터프리터 언어
- 공동 작업과 유지 보수 용이
- 기업의 실무에 많이 사용되는 언어(예 : 구글 S/W 50%이상, Instagram, Dropbox)
- 머신러닝 및 딥러닝 라이브러리가 많아 AI 개발에 적합

# 자료형 – List

- List명 = [요소1, 요소2, 요소3, ...]  
    >>> a = [1, 2, 3]  
    >>> b = ['Life', 'is', 'too', 'short']
- List의 인덱싱(Indexing)과 슬라이싱(slicing)

```
>>> a = [1, 2, 3]
>>> a[0] + a[2]
4
>>> a[-1]
3
>>> del a[1]
>>> a
[1, 3]
```

```
>>> a = [1, 2, 3, 4, 5]
>>> a[0:2]
[1, 2]
>>> a = "12345"
>>> a[0:2]
'12'
```

# 자료형 – Tuple

- List는 그 값의 생성, 삭제, 수정이 가능하지만 Tuple은 그 값을 바꿀 수 없다.
- Tuple의 **indexing**과 **slicing**

```
>>> t1 = (1, 2, 'a', 'b')  
>>> del t1[0] -> 오류  
>>> t1[0] = 'c' -> 오류
```

```
>>> t1 = (1, 2, 'a', 'b')  
>>> t1[1:]  
(2, 'a', 'b')
```

# 자료형 – Dictionary

- {Key1:Value1, Key2:Value2, Key3:Value3, ...}

```
>>> dic = {'name':'pey', 'phone':'0119993323', 'birth': '1118'}
```

딕셔너리 dic의 정보

key	value
name	pey
phone	01199993323
birth	1118

- Dictionary 인덱싱

```
>>> a = {1: 'a'}  
>>> a[2] = 'b'  
>>> a  
{1: 'a', 2: 'b'}  
>>> del a[1]  
>>> a  
{1: 'a'}
```

- Key 사용해 Value 얻기

```
>>> grade = {'pey': 10, 'julliet': 99}  
>>> grade['pey']  
10  
>>> grade['julliet']  
99
```

# 제어문 – for문

- **for** 변수 **in** List(또는 Tuple, 문자열):  
수행할 문장1    수행할 문장2

```
>>> test_list = ['one', 'two', 'three']
>>> for i in test_list:
    print(i)
```

```
>>> sum = 0
>>> for i in range(1, 11):
    sum = sum + i
>>> print(sum)
55
```

```
>>> a = [1,2,3,4]
>>> result = [num * 3 for num in a]
>>> print(result)
[3, 6, 9, 12]
```

- 다양한 for문들

```
marks = [90, 25, 67, 45, 80]
for number in range(len(marks)):
    if marks[number] < 60:
        continue
    print("%d번 학생 합격입니다." %
          (number+1))
```

# 제어문 – for문

- for문 이용한 list 출력

```
a = [[1,2],[3,4]]  
b = [[5,6],[7,8]]  
def m(a_b):  
    for i in range(len(a_b)):  
        print(a_b[i])  
m(a)  
m(b)
```

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



# Class

## ■ Class 다루기

```
class Hello:
    def __init__(self, name):
        self.name = name # 인스턴스 값 초기화
    def greeting(self): # 인스턴스 메소드(함수)
        print( 'Hello, %s' % self.name)
g = Hello('홍길동') # Hello class의 인스턴스 생성
g.greeting()        # greeting 메소드(함수) 호출
```

Hello, 홍길동

- 대용량의 Data Analysis을 위해 만들어진 라이브러리
- 동일한 타입의 요소(elements) 저장
- 메모리 최적화 및 계산 속도 향상
- 처음부터 dimension 명확하게 정의

# Numpy

## ■ Array 선언

```
import numpy as np
```

```
x = np.array([1,2,3]) # 1차원 array 선언  
print(x)
```

```
[1 2 3]
```

```
x[0] = 4  
print(x)
```

```
[4 2 3]
```

```
x = np.array([[1,2,3],[4,5,6]]) # 2차원 array 선언  
print(x)
```

```
[[1 2 3]  
 [4 5 6]]
```

## ■ zeros, ones

```
x = np.zeros((2,3))  
print(x)  
x = np.ones((2,3))  
print(x)
```

```
[[0. 0. 0.]  
 [0. 0. 0.]  
 [[1. 1. 1.]  
 [1. 1. 1.]
```

## ■ 덧셈, 곱셈, 행렬 곱

```
a = np.array([[1,2],[3,4]])  
b = np.array([[1,2],[3,4]])  
print(np.add(a,b))  
print(np.multiply(a,b))  
print(np.matmul(a,b))
```

```
[[2 4]  
 [6 8]]  
[[ 1  4]  
 [ 9 16]]  
[[ 7 10]  
 [15 22]]
```

# Numpy

## ■ 행 추출

```
a = np.array([[1,2,3],[4,5,6]])  
print(a)  
b = a[:2, 1:3]  
print(b)
```

```
[[1 2 3]  
 [4 5 6]]  
[[2 3]  
 [5 6]]
```

## ■ Random(Uniform , Gaussian)

```
x = np.random.random((2,2))  
print(x)  
x = np.random.randn(1,10)  
print(x)
```

```
[[0.38883585 0.35963249]  
 [0.47636737 0.10456658]]  
[[-0.99631136 -1.09328154  0.27602173  2.00358939 -0.65399985  0.66027  
 -0.12826222  0.87535427 -0.7317103 -0.98683368]]
```

## ■ Data 타입 선언

```
a = np.array([1,2])  
b = np.array([1,2], dtype=np.int32)  
c = np.array([1,2], dtype=np.float32)  
  
print(a,b,c)
```

```
[1 2] [1 2] [1. 2.]
```

참고 : <http://www.numpy.org/>

# Pandas

- Data Analysis 라이브러리
- 행과 열로 된 데이터 객체를 다루며, 대용량의 데이터 처리하는데 편리한 툴

## DataFrame

```
import pandas as pd

df = pd.read_csv( 'gapminder.csv', index_col = 'Unnamed: 0' )
df.head()
```

	year	continent	country	income	life_exp	population
0	2014	asia	Philippines	6598.0	70.7	100102249.0
1	2014	americas	Paraguay	8038.0	74.3	6552584.0
2	2014	asia	Palau	14078.0	NaN	21094.0
3	2014	asia	Pakistan	4619.0	65.6	185546257.0
4	2014	americas	St.-Pierre-et-Miquelon	NaN	NaN	6277.0

```
df.loc[3]
```

```
year                2014
continent           asia
country            Pakistan
income             4619
life_exp           65.6
population  1.85546e+08
Name: 3, dtype: object
```

```
df.loc[2, 'country' ]
```

```
'Palau'
```

참고 : <https://pandas.pydata.org/>  
<https://github.com/tyoon10/T-academy>

# Pandas

## Conditional Selection

```
df[ df['income'] > 50000 ]
```

```
# multiple conditions : & (and) , | (or)  
df[ ( df['income'] > 50000 ) & ( df['life_exp'] > 80 ) ]
```

## Setting and resetting indices

```
df.set_index('year')
```

```
df.dropna().reset_index()
```

## Missing Data

```
df['income'].fillna( value = 0 )
```

참고 : <https://github.com/tyoon10/T-academy>

# Matplotlib

- Data Visualization
- 그래프 또는 이미지 그릴 때 사용하는 2D, 3D Plotting Package

```
import matplotlib.pyplot as plt
```

```
%matplotlib inline
```

```
import numpy as np  
x = np.linspace(0,5,11)  
y = x ** 2
```

x

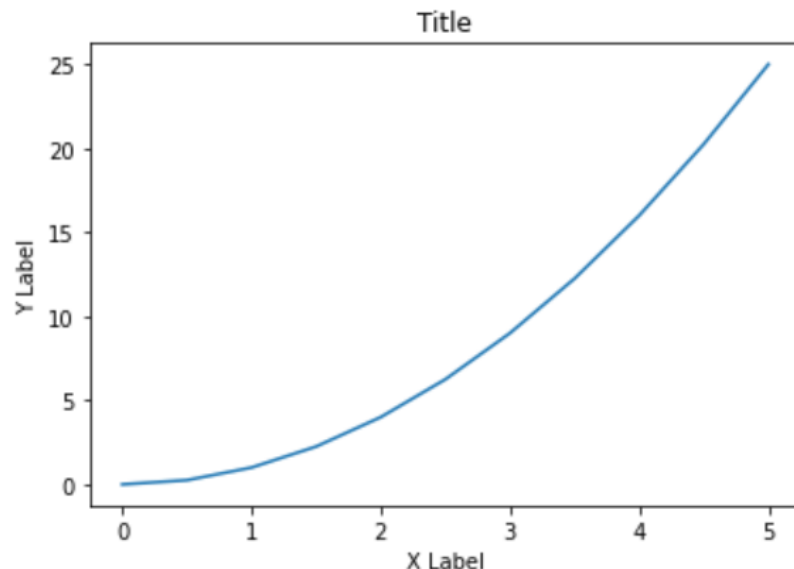
```
array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
```

y

```
array([ 0. ,  0.25,  1. ,  2.25,  4. ,  6.25,  9. , 12.25, 16.  
       20.25, 25. ])
```

참고: <https://github.com/tyoon10/T-academy/blob/master/180329%20%5BT%20academy%5D%204.%20matplotlib.ipynb>

```
plt.plot(x, y)  
plt.xlabel('X Label')  
plt.ylabel('Y Label')  
plt.title('Title')
```

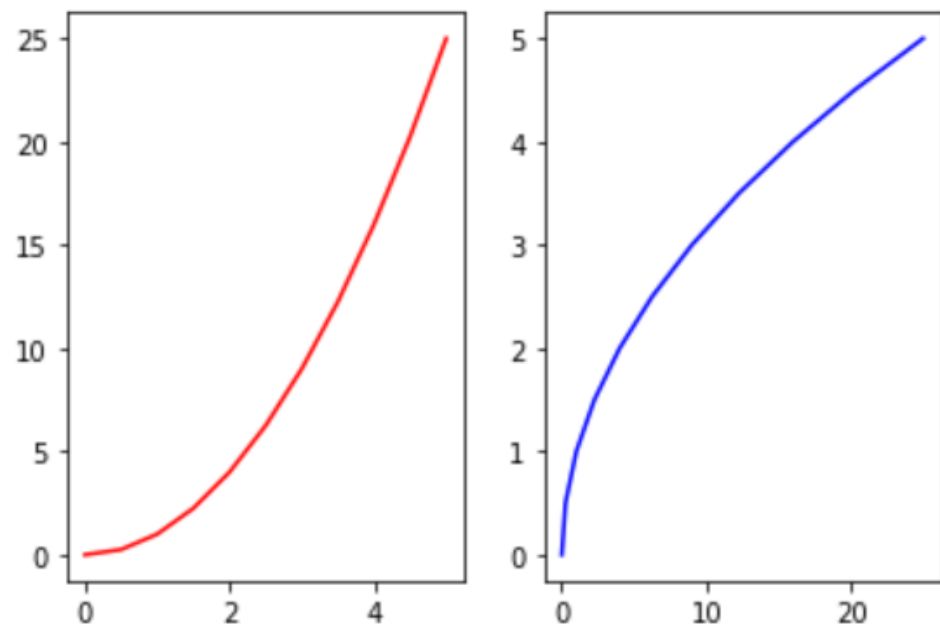


# Matplotlib

```
plt.subplot(1,2,1)  
plt.plot(x,y, 'r')
```

```
plt.subplot(1,2,2)  
plt.plot(y,x, 'b')
```

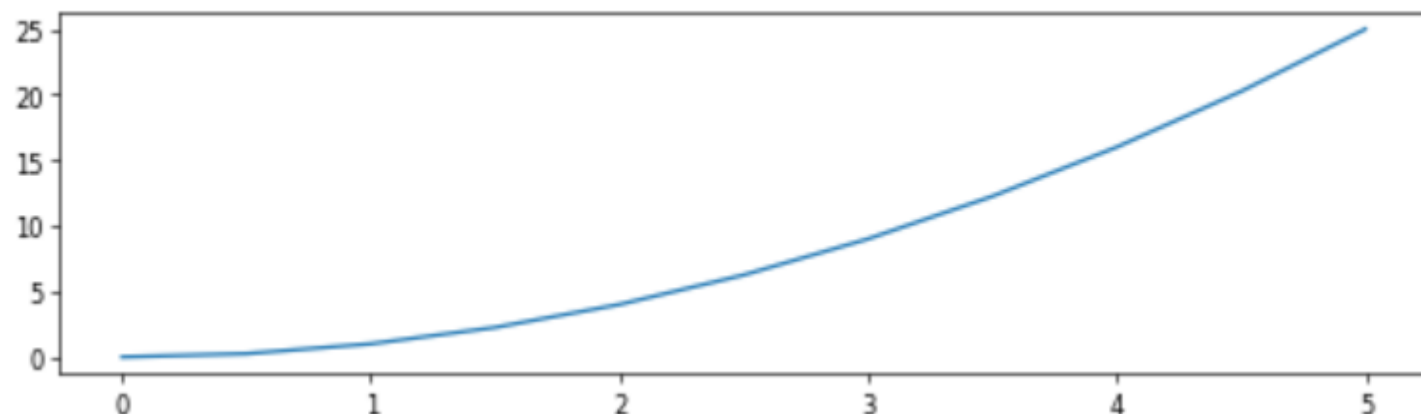
[<matplotlib.lines.Line2D at 0x7f203641f4a8>]



```
fig = plt.figure(figsize = (8,2), dpi = 60)
```

```
ax = fig.add_axes([0,0,1,1])  
ax.plot(x,y)
```

[<matplotlib.lines.Line2D at 0x7f2033aa1e10>]



참고: <https://github.com/tyoon10/T-academy/blob/master/180329%20%5BT%20academy%5D%204.%20matplotlib.ipynb>



# Matplotlib

- Images visualization

```
import matplotlib.pyplot as plt
from tensorflow.python.keras.datasets import mnist

(X_train, y_train), (X_test, y_test) = mnist.load_data()

# 테스트 데이터의 첫 10장을 표시합니다
for i in range(10):
    plt.subplot(1, 10, i+1)
    plt.imshow(X_test[i].reshape((28,28)), "gray")
plt.show()
```

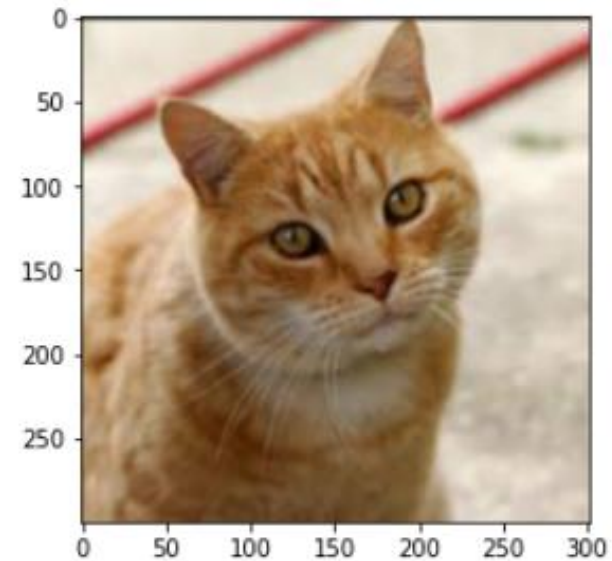


- 이미지, 영상처리, Object Detection 등의 기능을 제공하는 라이브러리
- Images processing

```
import matplotlib.pyplot as plt
import cv2 as cv

# 이미지 읽기
img = cv.imread('./cat.png')

# 이미지 화면에 표시
plt.imshow(img)
```



## ■ Images processing

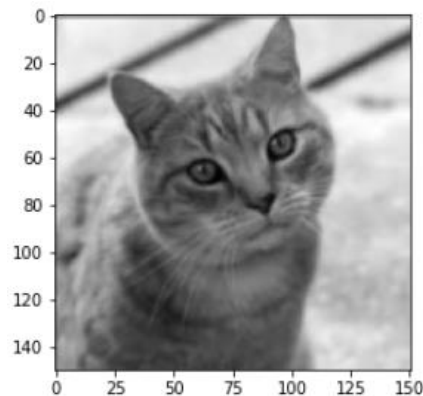
```
# 컬러 이미지를 그레이스케일로 변환
img_gray = cv2.cvtColor(img_color, cv2.COLOR_BGR2GRAY)

# 이미지 사이즈 변경
img_gray_reduced = cv2.resize(img_gray, None, fx=0.5, fy=0.5, interpolation=cv2.INTER_AREA) # 가로, 세로 모두 반으로 줄이기

# 이미지 보여주기
plt.imshow(img_gray_reduced, cmap='gray') # color라는 이름의 윈도우 안에 img_color라는 이미지를 보여줌

# 이미지 저장
cv2.imwrite('result.jpg', img_gray_reduced) #img_gray_reduced를 result.jpg 이미지 파일로 저장

True
```



# References

- [1] 점프투파이썬 <https://wikidocs.net/book/1>
- [2] T 아카데미 <https://github.com/tyoon10/T-academy>
- [3] 모두를 위한 딥러닝 Lab  
<https://github.com/deeplearningzerotoall/TensorFlow>
- [4] SeongShin.K's Blog <https://kss0222.github.io/>

# Lab 실습

[http://www.smartdesignlab.org/dl\\_hmc.html](http://www.smartdesignlab.org/dl_hmc.html)

## 4차) 파이썬의 기초

-이론: PDF

-실습: [TensorFlow 설치와 파이썬 연습](#)

현대자동차 버추얼이노베이션리서치랩

# What Questions Do You Have?

[kss@sm.ac.kr](mailto:kss@sm.ac.kr)

