

# 파이썬 프로그래밍 기초 - 2

최도진

# 목차

01 파이썬 시작하기

02 변수와 객체

03 자료형과 연산자

04 조건문과 반복문

05 함수

06 파일 처리

07 데이터 분석을 위한 주요 라이브러리

# 05. 함수

## ■ 사용자 정의 함수

- 함수 정의

```
>>> def sum1(a, b): [Enter]
        x = a + b [Enter]
        return x [Enter]
[Enter]
>>> def sum2(*args): [Enter]
        x = 0 [Enter]
        for i in args: [Enter]
            x += i [Enter]
        return x [Enter]
[Enter]
```

- 함수 호출

```
>>> a = 5
>>> b = 3
>>> sum1(a, b)
8
>>> sum1(3, 5)
8
>>> sum2(1, 2, 3, 4, 5)
15
>>> sum2(2, 3.5, 10)
15.5
```

# 05. 함수

## ■ 내장 함수

- 함수 종류

abs(x)  
all(iterable\_x)  
any(iterable\_x)  
chr(x)  
ord(c)  
dir(x)  
divmod(a, b)  
oct(x)  
hex(x)  
id(object)  
int(x)  
str(x)  
list(x)

- 파이썬 코드

```
>>> abs(-3.5)
3.5
>>> all([1, 2, 3, 4])
True
>>> all([4, -2, 0.0, 4])
False
>>> any([1, 2, 3, 4])
True
>>> any([4, -2, 0.0, 4])
True
>>> chr(97)
'a'
>>> chr(48)
'0'
>>> ord('a')
97
>>> ord('0')
48
>>> dir([1, 2, 3])
>>> dir({'1': 'a'})
>>> dir(1)
```

```
>>> divmod(7, 3)
(2, 1)
>>> divmod(1.3, 0.2)
(6.0, 0.09999999999999998)
>>> oct(8)
'0o10'
>>> oct(234)
'0o352'
>>> hex(16)
'0x10'
>>> hex(234)
'0xea'
>>> a = 3
>>> id(a)
1728080976
>>> int('3')
3
>>> str(3)
'3'
>>> list("Python")
['P', 'y', 't', 'h', 'o', 'n']
>>> list((1, 2, 3))
[1, 2, 3]
```

# 05. 함수

## ■ 내장 함수

- 함수 종류

tuple(x)  
type(x)  
Lambda  
max (iterable\_x)  
min(iterable\_x)  
pow(x, y)  
input()  
range(x)  
len(s)  
sorted(iterable\_x)

- 파이썬 코드

```
>>> tuple("Python")
('P', 'y', 't', 'h', 'o', 'n')
>>> tuple([1, 2, 3])
(1, 2, 3)
>>> type("abc")
<class 'str'>
>>> type(a)
<class 'int'>
>>> sum = lambda a,b: a+b
>>> sum
<function <lambda> at 0x000002C826BABA0>
>>> sum(3, 5)
8
>>> max([1, 4, 2, 8, 6])
8
>>> max("Python")
'y'
>>> min([1, 4, 2, 8, 6])
1
>>> min("Python")
'P'
```

```
>>> pow(2, 4)
16
>>> c = input()
21 [Enter]
>>> c
'21'
>>> c = input("정수를 입력하세요: ")
정수를 입력하세요: 21 [Enter]
>>> c
'21'
>>> range(5)
range(0, 5)
>>> list(range(5))
[0, 1, 2, 3, 4]
>>> list(range(5, 10))
[5, 6, 7, 8, 9]
>>> list(range(5, 10, 2))
[5, 7, 9]
>>> len('Python')
6
>>> sorted([3, 0, 2, 1])
[0, 1, 2, 3]
>>> sorted('Python')
['P', 'h', 'n', 'o', 't', 'y']
```

# 05. 함수

## ■ 모듈과 패키지

- 파이썬 코드

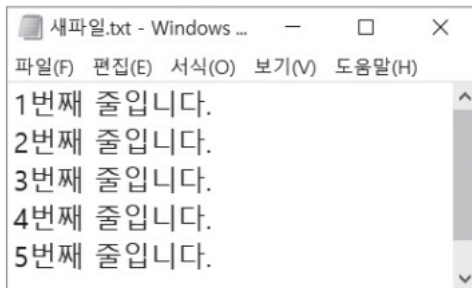
```
>>> Request('http://www.hanb.co.kr')
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    Request('http://www.hanb.co.kr')
NameError: name 'Request' is not defined
>>> import urllib.request
>>> urllib.request.Request('http://www.hanb.co.kr')
<urllib.request.Request object at 0x000001E5E0AE8390>
>>> import pandas
>>> pandas.DataFrame()
Empty DataFrame
Columns: []
Index: []
>>> from datetime import datetime
>>> datetime.now()
datetime.datetime(2018, 7, 25, 15, 42, 53, 119540)
```

# 06. 파일 처리

## ■ 파일 사용 모드

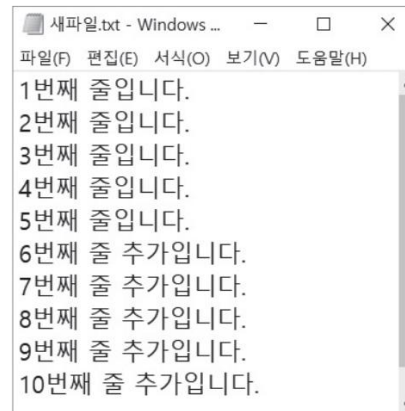
- 쓰기 모드

```
>>> f = open("D:/새파일.txt", 'w')
>>> f
<_io.TextIOWrapper name='D:/새파일.txt' mode='w'
encoding='cp949'>
>>> f.close()
>>> f = open("D:/새파일.txt", 'w')
>>> for i in range(1, 6):
    data = "%d번째 줄입니다. \n"% i [Enter]
    f.write(data) [Enter]
    [Enter]
>>> f.close()
```



- 추가 모드

```
>>> f= open("D:/새파일.txt", 'a')
>>> for i in range (6, 11):
    data = "%d번째 줄 추가입니다. \n"% i [Enter]
    f.write(data) [Enter]
    [Enter]
>>> f.close()
```



## 06. 파일 처리

### ■ 파일 사용 모드

- 읽기 모드

```
>>> f= open("D:/새파일.txt", 'r')
>>> line = f.readline()
>>> print(line)
1번째 줄입니다.
>>> while True:
    line = f.readline() [Enter]
    if not line: break [Enter]
    print(line) [Enter]
    [Enter]
2번째 줄입니다.
3번째 줄입니다.
4번째 줄입니다.
5번째 줄입니다.
6번째 줄 추가입니다.
7번째 줄 추가입니다.
8번째 줄 추가입니다.
9번째 줄 추가입니다.
10번째 줄 추가입니다.
>>> f.close()
```

```
>>> f= open("D:/새파일.txt", 'r')
>>> lines = f.readlines()
>>> print(lines)
['1번째 줄입니다. \n', '2번째 줄입니다. \n', '3번째 줄
입니다. \n', '4번째 줄입니다. \n', '5번째 줄입니다.
\n', '6번째 줄 추가입니다. \n', '7번째 줄 추가입니다.
\n', '8번째 줄 추가입니다. \n', '9번째 줄 추가입니다.
\n', '10번째 줄 추가입니다. \n']
>>> for line in lines:
    print(line) [Enter]
    [Enter]
1번째 줄입니다.
2번째 줄입니다.
3번째 줄입니다.
4번째 줄입니다.
5번째 줄입니다.
6번째 줄 추가입니다.
7번째 줄 추가입니다.
8번째 줄 추가입니다.
9번째 줄 추가입니다.
10번째 줄 추가입니다.
>>> f.close()
```



# 06. 파일 처리

## ■ 파일 사용 모드

- 읽기 모드

```
>>> f= open("D:/새파일.txt", 'r')
>>> data = f.read()
>>> data
'1번째 줄입니다. \n2번째 줄입니다. \n3번째 줄입니다.
\n4번째 줄입니다. \n5번째 줄입니다. \n6번째 줄 추가
입니다. \n7번째 줄 추가입니다. \n8번째 줄 추가입니다.
\n9번째 줄 추가입니다. \n10번째 줄 추가입니다. \n'
>>> f.close()
>>> with open("D:/새파일.txt", 'w') as f:
    f.write("Now is better than never.") [Enter]
    [Enter]
>>> data = f.read()
Traceback (most recent call last):
  File "<pyshell#95>", line 1, in <module>
    data = f.read()
ValueError: I/O operation on closed file.
```

# 07. 데이터 분석을 위한 주요 라이브러리

## ■ numpy

- 파이썬 코드

```
>>> import numpy as np
>>> np.__version__
'1.18.3'

>>> ar1 = np.array([1, 2, 3, 4, 5])
>>> ar1
array([1, 2, 3, 4, 5])
>>> type(ar1)
<class 'numpy.ndarray'>
>>> ar2 = np.array([[10, 20, 30], [40, 50, 60]])
>>> ar2
array([[10, 20, 30],
       [40, 50, 60]])
>>> ar3 = np.arange(1, 11, 2)
>>> ar3
array([1, 3, 5, 7, 9])
>>> ar4 = np.array([1, 2, 3, 4, 5, 6]).reshape((3, 2))
>>> ar4
array([[1, 2],
       [3, 4],
       [5, 6]])
>>> ar5 = np.zeros((2, 3))
>>> ar5
array([[0., 0., 0.],
       [0., 0., 0.]])
```

```
>>> ar6 = ar2[0:2, 0:2]
>>> ar6
array([[10, 20],
       [40, 50]])
>>> ar7 = ar2[0, :]
>>> ar7
array([10, 20, 30])
>>> ar8 = ar1 + 10
>>> ar8
array([11, 12, 13, 14, 15])
>>> ar1 + ar8
array([12, 14, 16, 18, 20])
>>> ar8 - ar1
array([10, 10, 10, 10, 10])
>>> ar1 * 2
array([ 2, 4, 6, 8, 10])
>>> ar1 / 2
array([0.5, 1. , 1.5, 2. , 2.5])
>>> ar9 = np.dot(ar2, ar4)
>>> ar9
array([[220, 280],
       [490, 640]])
```

# 07. 데이터 분석을 위한 주요 라이브러리

## ■ pandas

- Series 자료형

```
>>> import pandas as pd
>>> pd.__version__
'1.0.3'
>>> data1 = [10, 20, 30, 40, 50]
>>> data1
[10, 20, 30, 40, 50]
>>> data2 = ['1반', '2반', '3반', '4반', '5반']
>>> data2
['1반', '2반', '3반', '4반', '5반']
>>> sr1 = pd.Series(data1)
>>> sr1
0    10
1    20
2    30
3    40
4    50
dtype: int64
>>> sr2 = pd.Series(data2)
>>> sr2
0    1반
1    2반
2    3반
3    4반
4    5반
dtype: object
```

```
>>> sr3 = pd.Series([101, 102, 103, 104, 105])
>>> sr3
0    101
1    102
2    103
3    104
4    105
dtype: int64
>>> sr4 = pd.Series(['월', '화', '수', '목', '금'])
>>> sr4
0    월
1    화
2    수
3    목
4    금
dtype: object
```

# 07. 데이터 분석을 위한 주요 라이브러리

## ■ pandas

### • Series 자료형

```
>>> sr5 = pd.Series(data1, index = [1000, 1001, 1002, 1003, 1004])
>>> sr5
1000    10
1001    20
1002    30
1003    40
1004    50
dtype: int64
>>> sr6 = pd.Series(data1, index = data2)
>>> sr6
1반    10
2반    20
3반    30
4반    40
5반    50
dtype: int64
>>> sr7 = pd.Series(data2, index = data1)
>>> sr7
10    1반
20    2반
30    3반
40    4반
50    5반
dtype: object
```

```
>>> sr8 = pd.Series(data2, index = sr4)
>>> sr8
월    1반
화    2반
수    3반
목    4반
금    5반
dtype: object
>>> sr8[2]
'3반'
>>> sr8['수']
'3반'
>>> sr8[-1]
'5반'
>>> sr8[0:4]
월    1반
화    2반
수    3반
목    4반
dtype: object
>>> sr8.index
Index(['월', '화', '수', '목', '금'], dtype = 'object')
>>> sr8.values
array(['1반', '2반', '3반', '4반', '5반'], dtype = object)
```

# 07. 데이터 분석을 위한 주요 라이브러리

## ■ pandas

- Series 자료형

```
>>> sr1 + sr3
```

```
0    111
```

```
1    122
```

```
2    133
```

```
3    144
```

```
4    155
```

```
dtype: int64
```

```
>>> sr4 + sr2
```

```
0    월1반
```

```
1    화2반
```

```
2    수3반
```

```
3    목4반
```

```
4    금5반
```

```
dtype: object
```

# 07. 데이터 분석을 위한 주요 라이브러리

## ■ pandas

### • DataFrame 자료형

```
>>> data_dic = {
    'year': [2018, 2019, 2020],
    'sales': [350, 480, 1099]
}
>>> data_dic
{'year': [2018, 2019, 2020], 'sales': [350, 380, 1099]}
>>> df1 = pd.DataFrame(data_dic)
>>> df1
   year  sales
0  2018   350
1  2019   380
2  2020  1099
>>> df2 = pd.DataFrame([[89.2, 92.5, 90.8], [92.8, 89.9, 95.2]],
index = ['중간고사', '기말고사'], columns = data2[0:3])
>>> df2
      1반    2반    3반
중간고사 89.2  92.5  90.8
기말고사 92.8  89.9  95.2
>>> data_df = [['20201101', 'Hong', '90', '95'], ['20201102',
'Kim', '93', '94'], ['20201103', 'Lee', '87', '97']]
>>> df3 = pd.DataFrame(data_df)
>>> df3
   0      1      2      3
0 20201101 Hong   90   95
1 20201102 Kim   93   94
2 20201103 Lee   87   97
```

```
>>> df3.columns = ['학번', '이름', '중간고사', '기말고사']
>>> df3
   학번   이름  중간고사  기말고사
0 20201101 Hong      90      95
1 20201102 Kim      93      94
2 20201103 Lee      87      97
>>> df3.head(2)
   학번   이름  중간고사  기말고사
0 20201101 Hong      90      95
1 20201102 Kim      93      94
>>> df3.tail(2)
   학번   이름  중간고사  기말고사
1 20201102 Kim      93      94
2 20201103 Lee      87      97
>>> df3['이름']
0 Hong
1 Kim
2 Lee
Name: 이름, dtype: object
```

## 07. 데이터 분석을 위한 주요 라이브러리

### ■ pandas

- DataFrame 자료형

```
>>> df3.to_csv('C:/Users/kmj/My_Python/score.csv', header = 'False')
```

```
>>> df4 = pd.read_csv('C:/Users/kmj/My_Python/score.csv', encoding='utf-8', index_col=0, engine='python')
```

```
>>> df4
```

	학번	이름	중간고사	기말고사
0	20201101	Hong	90	95
1	20201102	Kim	93	94
2	20201103	Lee	87	97

# 07. 데이터 분석을 위한 주요 라이브러리

## ■ matplotlib

### • 라인플롯 차트 그리기

#### 1. 데이터 준비

```
>>> x = [2016, 2017, 2018, 2019, 2020]
>>> y = [350, 410, 520, 695, 543]
```

#### 2. x축과 y축 데이터를 지정하여 라인플롯 생성

```
>>> plt.plot(x, y)
[<matplotlib.lines.Line2D object at 0x0000015DB82D58C8>]
```

#### 3. 차트 제목 설정

```
>>> plt.title('Annual sales')
Text(0.5, 1.0, 'Annual sales')
```

#### 4. x축 레이블 설정

```
>>> plt.xlabel('years')
Text(0.5, 0, 'years')
```

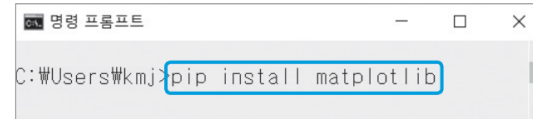
#### 5. y축 레이블 설정

```
>>> plt.ylabel('sales')
Text(0, 0.5, 'sales')
```

#### 6. 라인플롯 표시

```
>>> plt.show()
```

### • импорт



```
>>> import matplotlib
matplotlib 버전 확인 >>> matplotlib.__version__
'3.2.1'
```

```
pyplot 모듈 импорт하기 >>> import matplotlib.pyplot as plt
```

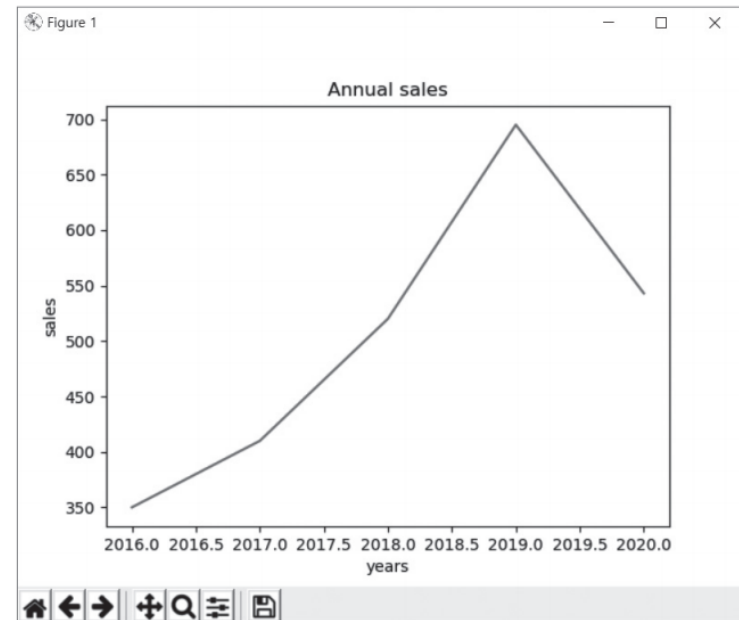


그림 4-7 라인플롯 차트



# 07. 데이터 분석을 위한 주요 라이브러리

## ■ matplotlib

### • 바차트 차트 그리기

#### 1. 데이터 준비

```
>>> y1 = [350, 410, 520, 695]
>>> y2 = [200, 250, 385, 350]
>>> x = range(len(y1))
```

#### 2. x축과 y축 데이터를 지정하여 라인플롯 생성

```
>>> plt.bar(x, y1, width = 0.7, color = "blue")
<BarContainer object of 4 artists>
>>> plt.bar(x, y2, width = 0.7, color = "red",
bottom = y1)
<BarContainer object of 4 artists>
```

#### 3. 차트 제목 설정

```
>>> plt.title('Quarterly sales')
Text(0.5, 1.0, 'Quarterly sales')
```

#### 4. x축 레이블 설정

```
>>> plt.xlabel('Quarters')
Text(0.5, 0, 'Quarters')
```

#### 5. y축 레이블 설정

```
>>> plt.ylabel('sales')
Text(0, 0.5, 'sales')
```

#### 6. 눈금 이름 리스트 생성

```
>>> xLabel = ['first', 'second', 'third', 'fourth']
```

#### 7. 바 차트의 x축 눈금 이름 설정

```
>>> plt.xticks(x, xLabel, fontsize = 10)
([<matplotlib.axis.XTick object at
0x0000015DB5722B48>, <matplotlib.axis.XTick
object at 0x0000015DB5722B08>, <matplotlib.
axis.XTick object at 0x0000015DB82E2688>,
<matplotlib.axis.XTick object at
0x0000015DB60C5188>], [Text(0, 0, 'first'),
Text(0, 0, 'second'), Text(0, 0, 'third'), Text(0,
0, 'fourth')])
```

#### 8. 범례 설정

```
>>> plt.legend(['chairs', 'desks'])
<matplotlib.legend.Legend object at
0x0000020F2BBA0908>
```

#### 9. 바 차트 표시

```
>>> plt.show()
```

## 07. 데이터 분석을 위한 주요 라이브러리

### ■ matplotlib

- 바차트 차트 그리기

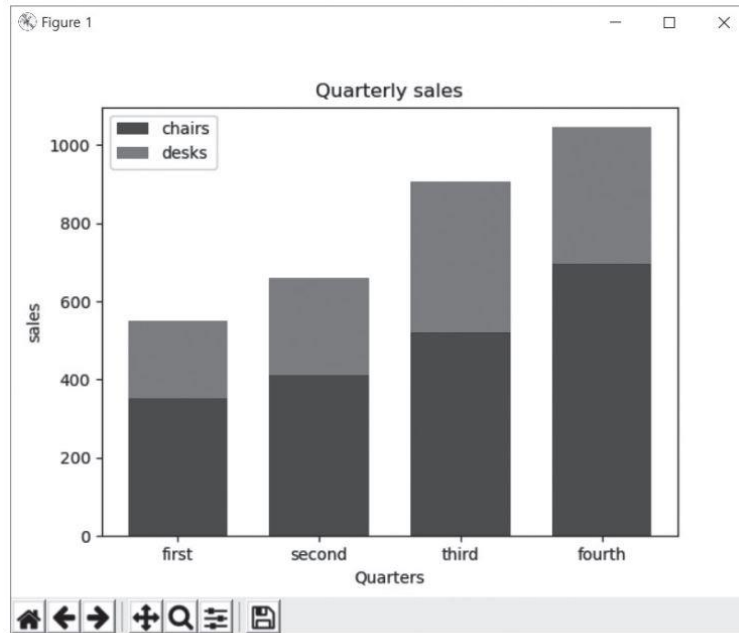


그림 4-8 바 차트

