

생산공정 모니터링을 위한 데이터 사이언스 입문

홍익대학교
기계·시스템디자인공학과
Biomechanics Research Laboratory

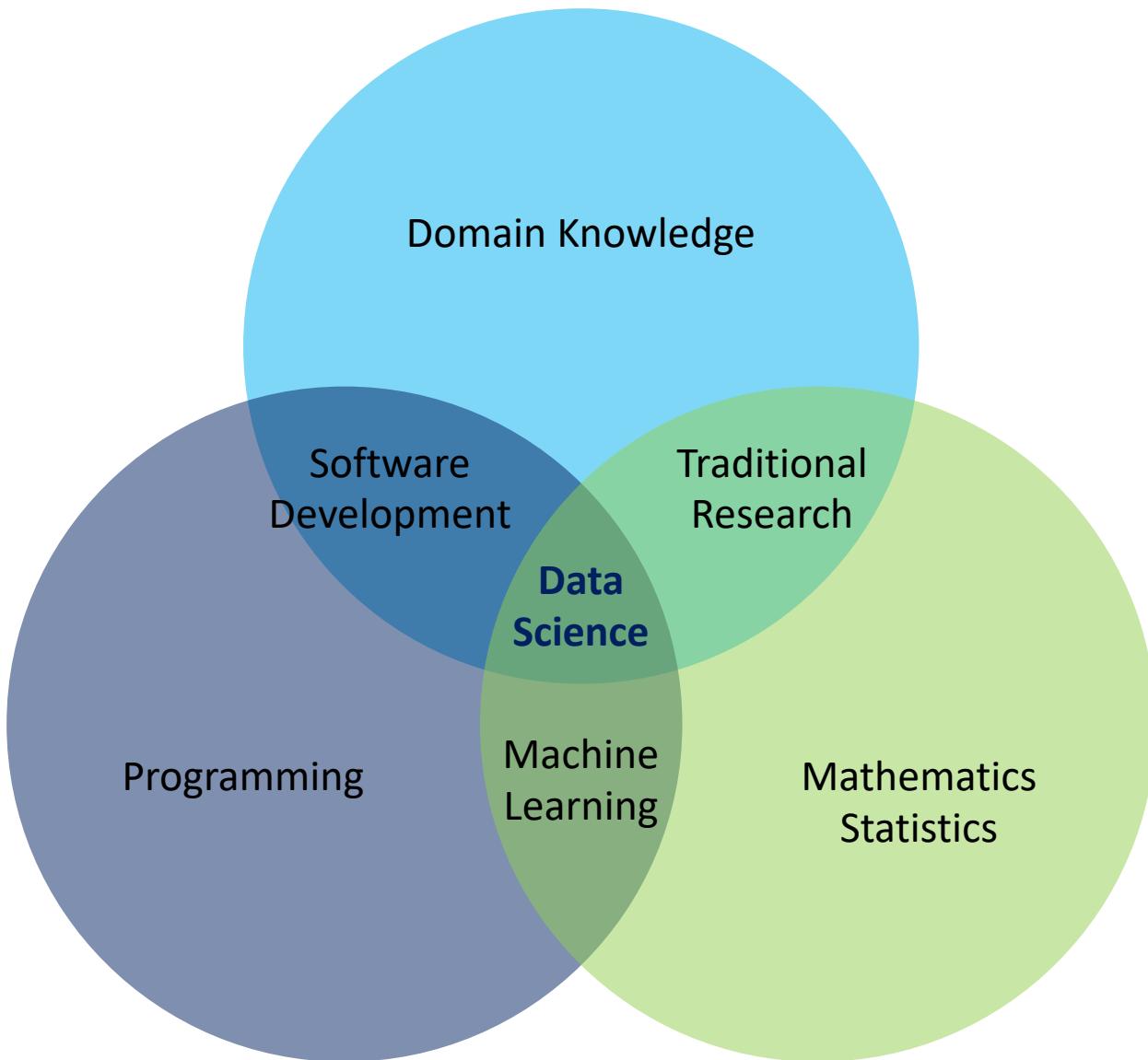
오 유 근 교수

강의 내용

- Day 1
 - Python review
 - Data acquisition
 - Website data scraping (사진, 일기예보, 실시간 주가)
 - IMU(Inertial Measurement Unit) data
- Day 2
 - Database programming (SQL)
 - Machine Learning
 - Supervised learning: regression, classification
 - (Optional) Unsupervised learning: clustering, PCA
- Day 3
 - OpenCV 활용한 ML
 - (Optional) Team project



Data Science

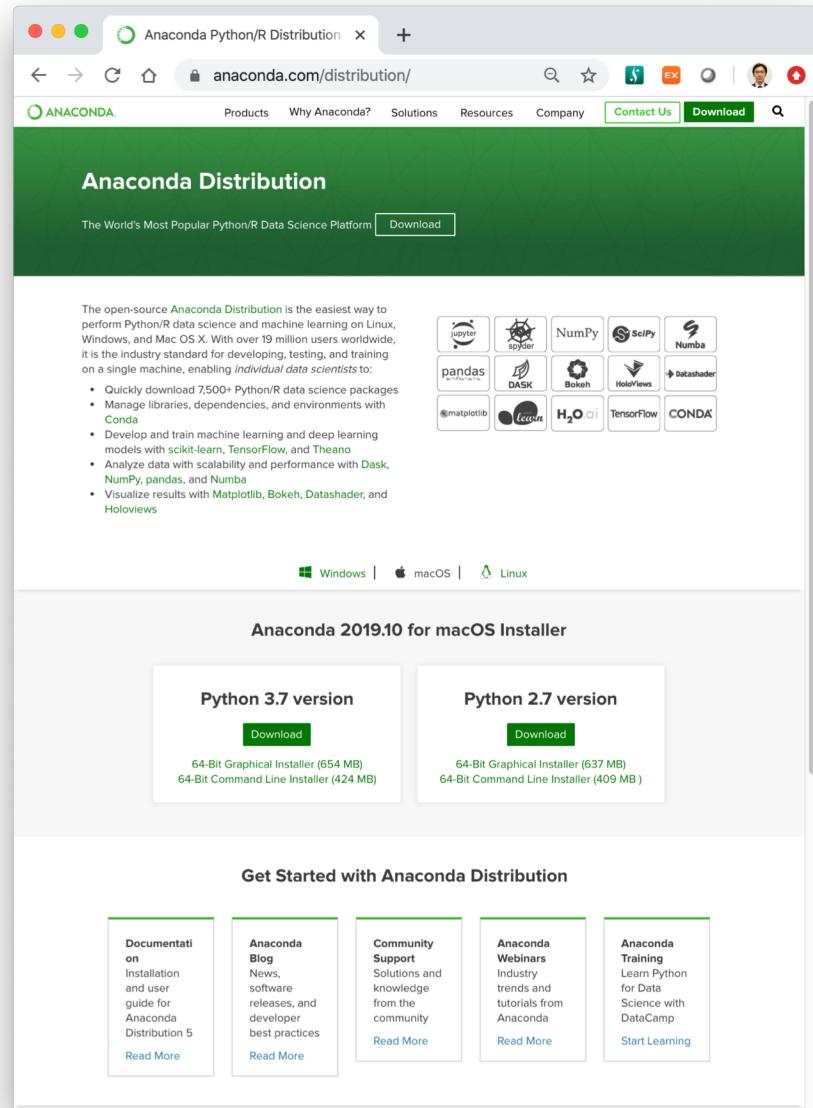


Ch1. Python Review



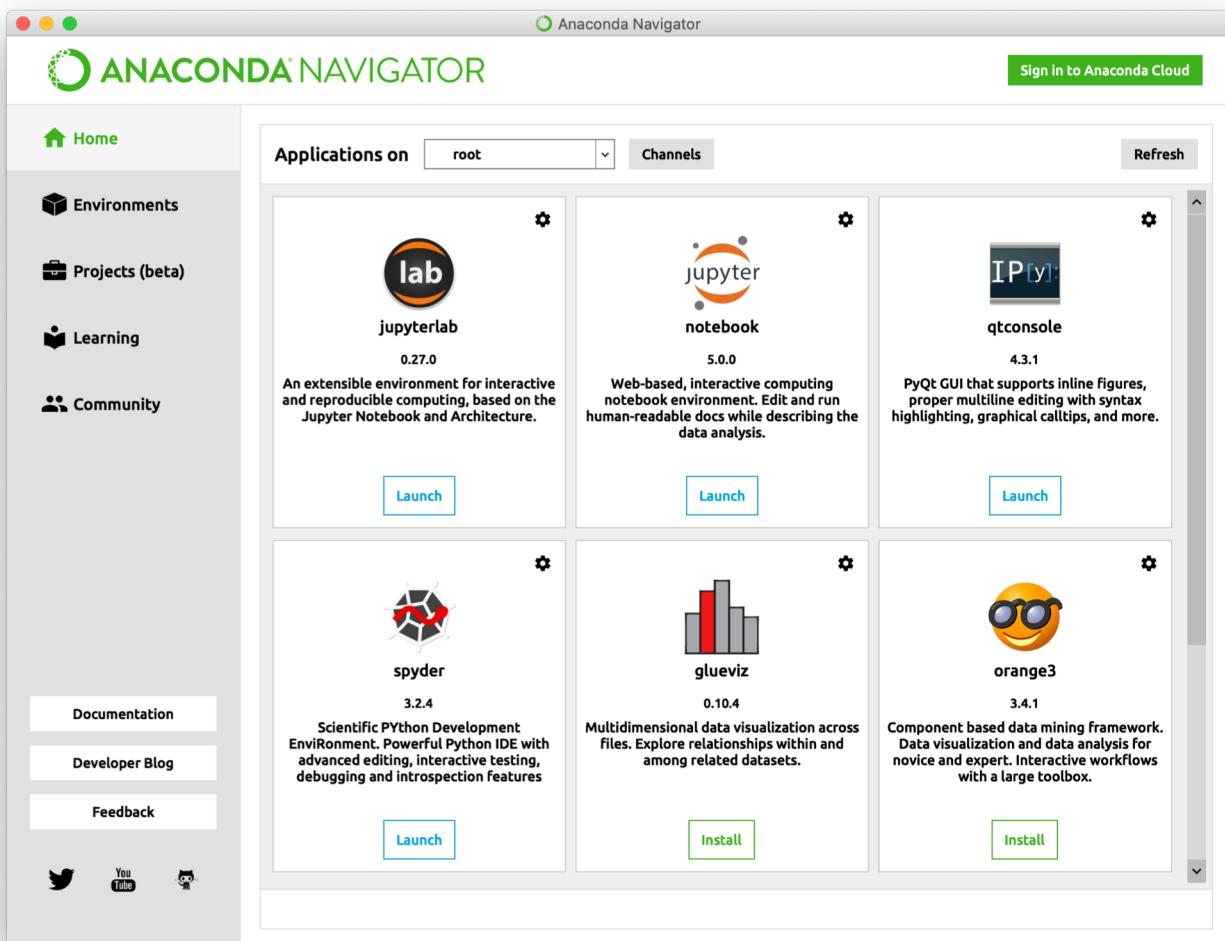
Anaconda 설치

- Anaconda
 - Python 3.7 version
 - 64 bit graphical installation
- 설치 메뉴얼
 - <https://docs.anaconda.com/anaconda/install/windows/#>



개발환경(IDE)

- Jupiter Notebook
- Spyder
- PyCharm
- ...



개발환경(IDE) - jupiter notebook



The screenshot shows a Jupyter Notebook window titled "Untitled1". The top bar includes standard OS controls, a tab for "Desktop/2020_01_27_DataScience", and a URL bar pointing to "localhost:8888/notebooks/Desktop/2020_01...". The menu bar offers File, Edit, View, Insert, Cell, Kernel, Widgets, and Help options. A "Trusted" button and a "Python 3" kernel selection are also present. Below the menu is a toolbar with various icons for file operations like saving, opening, and cell execution.

In [9]:

```
import math  
  
a = 1  
b = 2  
  
c = math.sqrt(a+b)  
print('c = ' + str(c))  
  
c = 1.7320508075688772
```

In [10]:

```
ruler1 = '1'  
ruler2 = ruler1 + '2' + ruler1  
  
print ('The value of ''ruler2'' is ' + ruler2)
```

The value of ruler2 is 121

개발환경(IDE) - Spyder



The screenshot displays the Spyder Python IDE interface. The main window title is "Spyder (Python 3.6)".

Script Editor: A red box highlights the code editor area where the script `day1_python_programming_01.py` is open. The code content is as follows:

```
1 import math
2
3 a = 1
4 b = 2
5
6 c = math.sqrt(a+b)
7 print('c = ' + str(c))
8
9 ruler1 = '1'
10 ruler2 = ruler1 + '2' + ruler1
11
12 print ('The value of "ruler2" is ' + ruler2)
```

Variable Explorer: A red box highlights the "Variable explorer" tab, which shows the following variable information:

Name	Type	Size	Value
a	int	1	1
b	int	1	2
c	float	1	1.7320508075688772
ruler1	str	1	1

IPython console: A red box highlights the "IPython console" tab, which shows the following session output:

```
Python 3.6.3 |Anaconda, Inc.| (default, Oct  6 2017,
12:04:38)
Type "copyright", "credits" or "license" for more
information.

IPython 6.1.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/YK0/Desktop/
2020_01_27_DataScience_Lecture/Pyton_Code/
day1_python_programming_01.py', wdir='/Users/YK0/Desktop/
2020_01_27_DataScience_Lecture/Pyton_Code')
c = 1.7320508075688772
The value of ruler2 is 121

In [2]:
```

At the bottom of the interface, status bars show: Permissions: RW, End-of-lines: LF, Encoding: UTF-8, Line: 1, Column: 1, Memory: 61 %.

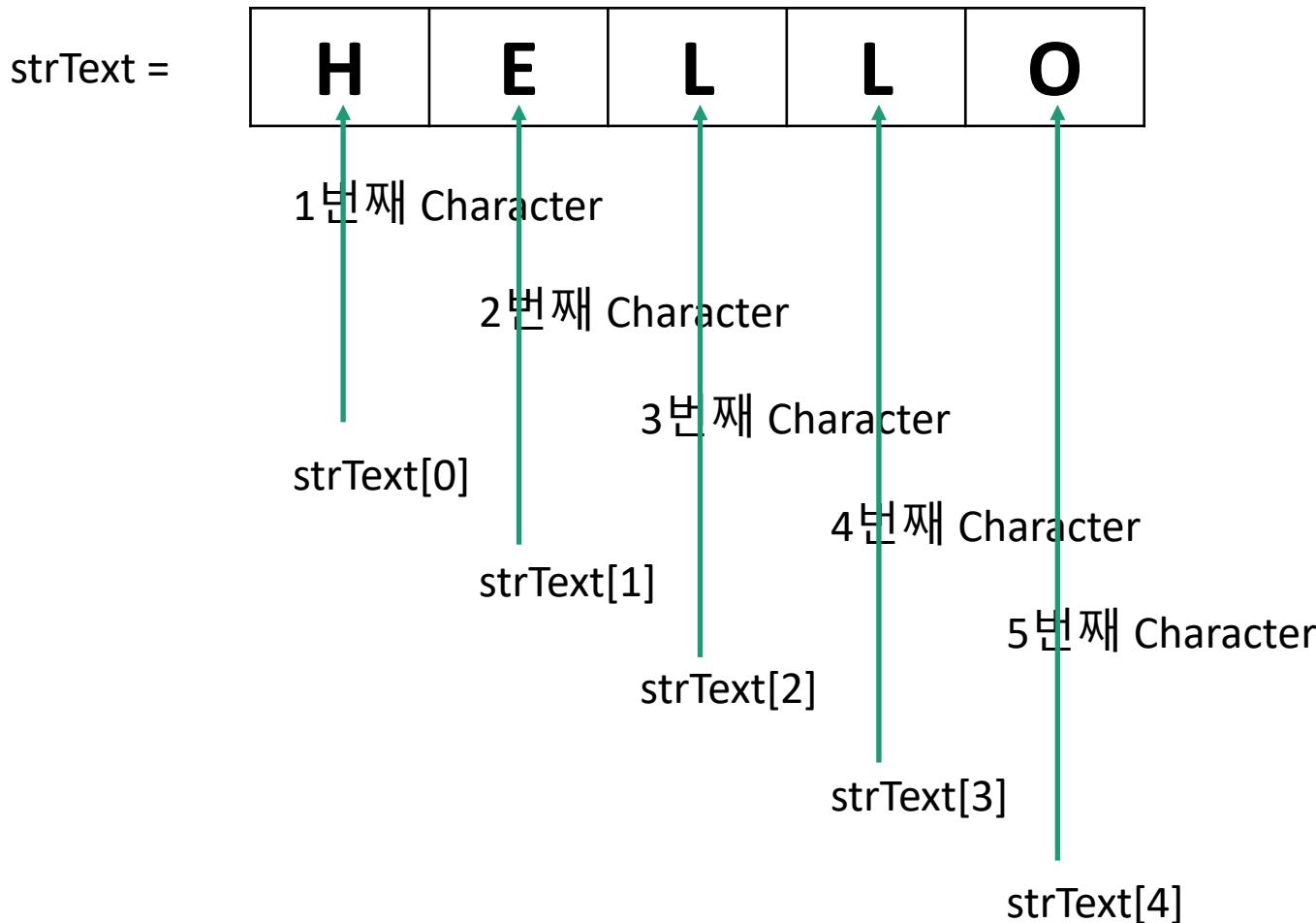
Basic Syntax

- 변수 Type
 - String - 문자
 - Integer - 정수
 - Float – 실수
 - Boolean – True/False
 - Complex – 복소수
 - Type 확인 – type()
- Back slash (\)
 - 여러 줄에 걸쳐서 써야할 때
- Semicolon (;)
 - 한 줄에 여러 구문을 쓸 때
- 들여쓰기 (Indentation)
 - Block으로 구분해야 할 때
 - 예: For loop, while loop 등
- 주석처리 (Comment)
 - # 혹은 """ ~~~ """

<https://www.tutorialsteacher.com/python/python-basic-syntax>

```
1 # Basic Syntax
2
3 # 변수 Type - string, integer, float, boolean, complex number
4 # type(***) 함수를 이용하여 각 변수의 type을 확인한다.
5 strMessage1 = "Hello World"; print(strMessage1); print(type(strMessage1))
6 intNum = 1234; print(intNum); print(type(int))
7 fltNum = 1234.5678; print(fltNum); print(type(fltNum))
8 isTrue = True; print(isTrue); print(type(isTrue))
9 cmpNum = 1 + 2j; print(cmpNum); print(type(cmpNum))
10
11 # Multiple lines - back slash (\)
12 strMessage2 = 'Python is an easy to learn, powerful programming \
13 language. It has efficient high-level data structures \
14 and a simple but effective approach to object-oriented \
15 programming. Python's elegant syntax and dynamic typing, \
16 together with its interpreted nature, make it an ideal \
17 language for scripting and rapid application development \
18 in many areas on most platforms.'
19
20 # 한 줄에 여러 문장을 쓸 때 - semicolon (;)
21 strMessage1 = "Hello World"; intNum = 1234; fltNum = 1234.5678
22
23 # 들여쓰기 (Indentation)
24 j = 0
25 for i in range(1, 4):
26     print('i = ', i)
27     j = j + 1
28     print('>> j = ', j)
29
30 j = 0
31 for i in range(1, 4):
32     print('i = ', i)
33
34     j = j + 1
35 print('>> j = ', j)
36
37 # 주석 처리 (comment/uncomment), ctrl + 1
38 #print('# Case 1')
39 #j = 0
40 #for i in range(1, 4):
41 #    print('i = ', i)
42 #    j = j + 1
43 #
44 #    print('>> j = ', j)
45
46 """
47 print('# Case 2')
48 j = 0
49 for i in range(1, 4):
50     print('i = ', i)
51
52     j = j + 1
53 print('>> j = ', j)
54 """
```

String



String vs. List vs. Tuple

• String

- strText = 'Hello'
- strText[0] = ?

• List

- []
- Mutable

• Tuple

- ()
- Immutable

```
1 # Variable Type
2
3 # Sequence Type: 1) String, 2) List, 3) Tuple
4 strName1 = 'James'
5 strName2 = "James"
6
7 print('strName1 = ', strName1);      print(type(strName1))
8 print('strName2 = ', strName2);      print(type(strName2))
9
10 strName_1st = strName1[0];          print(strName_1st)
11 strName_2nd = strName1[1];          print(strName_2nd)
12 strName_3rd = strName1[2];          print(strName_3rd)
13 strName_4th = strName1[3];          print(strName_4th)
14 strName_5th = strName1[4];          print(strName_5th)
15
16 strName3 = strName1[0] + strName1[1] + strName1[2] + strName1[3] + strName1[4]
17 print(strName3)
18
19 # List [] vs. Tuple ()
20 lstNumber = [1, 2, 3, 4, 5]
21 print(type(lstNumber))
22 print('The first element of ''lstNumber'' is ' + str(lstNumber[0]))
23
24 lstString = ['abc', 'def', 'ghi', 'jkl', 'mno']
25 print(type(lstString))
26 print(lstString[1])
27 print(lstString[2][2])
28
29 tplNumber = (1, 2, 3, 4, 5)
30 print(type(tplNumber))
31 print('The second element of ''tplNumber'' is ' + str(tplNumber[1]))
32
33 tplString = ('abc', 'def', 'ghi', 'jkl', 'mno')
34 print(type(tplString))
35 print(tplString[1])
36 print(tplString[2][2])
37
38 lstNumber[1] = lstNumber[1] * 10;      print(lstNumber)
39 tplNumber[1] = tplNumber[1] * 10;      print(tplNumber)
40
```

<https://www.tutorialsteacher.com/python/python-string>
<https://www.tutorialsteacher.com/python/python-list>
<https://www.tutorialsteacher.com/python/python-tuple>

Set vs. Dictionary

• Dictionary

- {}
- dict = {key1:value1,
key2:value2,
...,
keyN:valueN}

• Set

- {}
- set = {value1,
value2,
...,
valueN}
- 중복 허용 X

```
day1_python_programming_04.py*
1 # Dictionary vs. Set
2 # Dictionary
3 capitals = {"USA": "Washington", "France": "Paris", "India": "New Delhi"}
4 print(capitals.get('France'))
5 print(capitals.get('Paris'))
6
7 capitals['USA'] = 'Washington, D.C.'
8 print(capitals.get('USA'))
9
10 del capitals['India']
11
12 for key in capitals:
13     print("Key = " + key + ", Value = " + capitals[key])
14
15 print(capitals.keys())
16 print(capitals.values())
17
18 capitals.update({"India": "New Delhi"})
19 for key in capitals:
20     print("Key = " + key + ", Value = " + capitals[key])
21
22
23 dict1 = {"Fruit": ["Mango", "Banana"], "Colour": ["Blue", "Red"]}
24 print(dict1.get('Fruit'))           # ['Mango', 'Banana']
25 print(dict1.get('Fruit')[0])       # Mango
26 print(dict1.get('Fruit')[1])       # Banana
27
28 # Set
29 setStudent = {1, "Bill", 75.50}
30 print(setStudent)
31
32 setNumbers = {1, 2, 2, 3, 4, 4, 5, 5}
33 print(setNumbers)
```

비교 연산자 (Comparison Operator)

Operator	Description	Example
>	True if the left operand is higher than the right one	<code>>>> x>y False</code>
<	True if the left operand is lower than right one	<code>>>> x<y True</code>
==	True if the operands are equal	<code>>>> x==y False</code>
!=	True if the operands are not equal	<code>>>> x!=y True</code>
>=	True if the left operand is higher than or equal to the right one	<code>>>> x>=y False</code>
<=	True if the left operand is lower than or equal to the right one	<code>>>> x<=y True</code>

논리 연산자 (Logical Operator)

Operator	Description	Example
and	True if both are true	>>> x and y False
or	True if at least one is true	>>> x or y True
not	Returns True if an expression evaluates to false and vice-versa	>>> not x True

제어문

- If 구문

- If [boolean expression]:
[statements]
- elif [boolean expression]:
[statements]
- elif [boolean expression]:
[statements]
- elif [boolean expression]:
[statements]
- else:
[statements]

```
day1_python_programming_05.py | day1_python_programming_05_1.py
```

```
1 # if 구문
2
3 price = int(input("Enter Price: "))
4 qty = int(input("Enter Quantity: "))
5 amt = price * qty
6
7 if amt > 1000:
8     print('10% discount is applicable')
9     discount = amt * 10 / 100
10    amt = amt - discount
11
12 print("Amount payable: ", amt)
13
14
```

```
day1_python_programming_05.py | day1_python_programming_05_1.py
```

```
1 # if 구문
2
3 def isDiscountable(inpArg_Amount):
4     if inpArg_Amount > 1000:
5         rtnIsDiscountable = True
6         rtnFinalPrice = inpArg_Amount - inpArg_Amount * 0.1
7     else:
8         rtnIsDiscountable = False
9         rtnFinalPrice = inpArg_Amount
10
11    return rtnIsDiscountable, rtnFinalPrice
12
13 price = int(input("Enter Price: "))
14 qty = int(input("Enter Quantity: "))
15 amt = price * qty
16
17 [isDiscountable, amt] = isDiscountable(amt)
18 if isDiscountable:
19     print('10% discount is applicable.')
20 else:
21     print('10% discount is not applicable.')
22
23 print("Amount payable: ", amt)
24
```



반복문 – While Loop

- While 구문

- While [boolean expression]:

- statement1

- statement2

- ...

- statementN

- else:

- statement

- break

- continue

- pass

- 무한루프?

```
day1_python_programming_06.py
1 # While Loop
2
3 num = 0
4 count = 0
5 sum = 0
6
7 while num >= 0:
8     num=int(input("enter any number .. -1 to exit\n"))
9
10    if num >= 0:
11        count = count + 1 #this counts number of inputs
12        sum = sum + num # this adds input number cumulatively.
13
14        if count > 10:
15            break
16
17 avg = sum / count
18 print ("numbers input:",count, "average:",avg)
19
```



반복문 – For Loop

- For Loop 구문
 - For x in sequence:
 - statement1
 - statement2
 - ...
 - statement

- break
- continue
- pass

```
1 # For Loop
2
3 myList=[10, 20, 30, 40, 50]
4 for i in myList:
5     print(i)
6
7 # 1부터 10까지의 합
8 tmpSum = 0
9 for i in range(1, 11):
10    tmpSum = tmpSum + i
11    print(tmpSum)
12
13 # dictionary
14 dictVar = { 1:100, 2:200, 3:300 }
15 for k,v in dictVar.items():
16     print("key=" + str(k) + ", value=" + str(v))
17
18
```

Function

- Function 정의

- Def function_name(inpArg1, inpArg2, ... inpArgN):
 statement1
 statement2

 return outArg1, outArg2, ... outArgM

```
day1_python_programming_08.py
1 # Function
2
3 def testResult(m1,m2,m3):
4     ttl=m1+m2+m3
5     percent=ttl/3
6     if percent>=50:
7         print ("Result: Pass")
8     else:
9         print ("Result: Fail")
10    return
11
12 p=int(input("Enter your marks in physics: "))
13 c=int(input("Enter your marks in chemistry: "))
14 m=int(input("Enter your marks in maths: "))
15
16 testResult(p,c,m)
```

By Value vs. By Reference

- 함수에 인자(input argument)를 전달하는 방식
 - By value: 인자의 값 자체를 넘김
 - By reference: 인자에 해당하는 변수의 주소를 넘김
- Python은 항상 인자를 by reference 방식으로 넘김

```
# By Reference
def myFunction(input_arg):
    print('Value received:', input_arg, 'id:', id(input_arg))

    input_arg = input_arg * 10
    print('Value multiplied:', input_arg, 'id:', id(input_arg))
    return

x = 10
print('Value before being passed:', x, 'id:', id(x))
myFunction(x)
print('Value after being passed:', x, 'id:', id(x))
```

변수의 범위

- Global variable
 - 함수 내/외부에서 모두 접근/사용 가능함
- Local variable
 - 함수 내에서만 접근/사용 가능함

The image shows a code editor with two tabs open, both titled "day1_python_programming_10.py". The left tab shows the original code, and the right tab shows the modified code with comments added.

```
# Scope of Variables
def SayHello():
    global userName

    userName = 'Steve'
    print("userName after modified = ", userName)

    return

userName = 'John'      # global variable
print("userName before called = ", userName)
SayHello()
print("userName after called = ", userName)
```

```
# Scope of Variables
def SayHello():

    userName = 'Steve'
    print("userName after modified = ", userName)

    return

userName = 'John'      # global variable
print("userName before called = ", userName)
SayHello()
print("userName after called = ", userName)
```

Module

- Import module_name as renamed_name
 - Ex: import math as cal
- From ... import ...
 - Ex: from myFunctions import sum
 - Ex: from myFunctions import sum, average, power
 - Ex: from myFunctions import *

```
1 # Module
2 #
3 import myFunctions
4
5 x = 1
6 y = 2
7
8 z = myFunctions.sum(x,y);      print('sum: ', z)
9 z = myFunctions.average(x,y);  print('average: ', z)
10 z = myFunctions.power(x,y);   print('power: ', z)
11
--
```

```
1 # Module
2 #
3 from myFunctions import sum, average, power
4
5 x = 1
6 y = 2
7
8 z = sum(x,y);      print('sum: ', z)
9 z = average(x,y); print('average: ', z)
10 z = power(x,y);  print('power: ', z)
11
--
```

Class

- 구성

- Constructor: 생성자 (initialization)
- Instance attributes: 속성
- Methods: 동작

```
class person:  
    count=0 #class attribute  
    def __init__(self): #constructor  
        self.name="unknown" #instance attribute  
        self.age=0 #instance attribute  
    def displayInfo(self): #method  
        print(self.name, self.age)  
  
    def display_all_attributes(self):  
        print('Name: ', self.name)  
        print('Age: ', self.age)  
        print('Power level: ', self.energy_level)  
  
person1 = person() # instance 생성  
person1.name = 'James' # 속성(attribute) 정의  
person1.display_all_attributes() # method (action) 실행  
person1.power_down()  
  
person2 = person() # instance 생성
```

[실습] Class

```
1 # Class 예제
2
3 class person:
4     def __init__(self, name="Guest", age=25, energy_level=100):
5         self.name = name
6         self.age = age
7         self.energy_level = energy_level
8
9     def power_up(self):
10        self.energy_level = self.energy_level + 5
11
12    def power_down(self):
13        self.energy_level = self.energy_level - 5
14
15    def display_all_attributes(self):
16        print('Name: ', self.name)
17        print('Age: ', self.age)
18        print('Power level: ', self.energy_level)
19
20
21 person1 = person()                      # instance 생성
22 person1.name = 'James'                   # 속성(attribute) 정의
23
24 person1.display_all_attributes()        # method (action) 실행
25 person1.power_down()
26
27
28 person2 = person()                      # instance 생성
29
```

Modules

- Useful modules
 - Numpy: array
 - Matplotlib: plot
 - Scikit-Learn: ML
 - OpenCV: image processing
 - Pandas: data structure
 - TensorFlow: ML/DL framework
 - Keras: ML/DL framework
 - PyTorch: ML/DL framework

Ch2. Data Acquisition



홍익대학교
HONGIK UNIVERSITY

기계시스템디자인공학과
Biomechanics Research Laboratory

Scraping & Crawling

- Scraping / Crawling
 - 웹 사이트에 있는 특정 정보를 추출하는 작업
- [실습 1] 홍익대학교 홈페이지에서 로고 그림 읽고 저장
 - import urllib.request
 - urllib.request.urlretrieve(url, save_name)
- [실습 2] 홍익대학교 홈페이지에서 로고 그림 읽고 저장
 - urllib.request.urlopen(url).read()
- [실습 3] 기상청 싸이트에서 기상정보 가져오기 1
- [실습 4] 기상청 싸이트에서 기상정보 가져오기 2
 - BeautifulSoup 4

[실습 1] – 홍익대학교 홈페이지에서 로고 파일 읽고 저장하기

```
② [x] day1_python_programming_12.py [x] day1_python_programming_13.py  
1 # Scraping & Crawling  
2  
3 # 예제: 홍익대학교 홈페이지에서 학교 로고 읽어와 저장하기  
4  
5 import urllib.request  
6  
7 strUrl = 'http://www.hongik.ac.kr/front/images/local/header_logo.png'  
8 strFileName = 'hongik_logo.png'  
9  
10 # Case #1  
11 urllib.request.urlretrieve(strUrl, strFileName)  
12 print('저장 완료')  
13  
14 # Case #2  
15 rtn = urllib.request.urlretrieve(strUrl, strFileName)  
16 print(rtn)  
17 print(type(rtn))  
18 print('저장 완료')  
19  
20 # Case #3 - Error Handling (try/except)  
21 try:  
22     rtn = urllib.request.urlretrieve(strUrl, strFileName)  
23     print(rtn)  
24     print('저장 완료')  
25 except:  
26     print('Not found')  
27
```



[실습 2] – 홍익대학교 홈페이지에서 로고 파일 읽고 저장하기

```
↳ [x] day1_python_programming_13.py
1 # Scraping & Crawling
2
3 # 예제: 홍익대학교 홈페이지에서 학교 로고 읽어와 저장하기
4
5 import urllib.request
6
7 strUrl = 'http://www.hongik.ac.kr/front/images/local/header_logo.png'
8 strFileName = 'hongik_logo.png'
9
10 # 해당 이미지를 읽고 메모리에 로딩
11 imgLogo = urllib.request.urlopen(strUrl).read()
12 #print(imgLogo)
13
14 # 파일로 저장
15 with open(strFileName, mode='wb') as f:
16     f.write(imgLogo)
17     print('파일 저장 완료')
```

[실습 3] 기상청 사이트에서 기상정보 가져오기 1

```
day1_python_programming_14.py
1 # Scraping & Crawling
2
3 # [실습 3] 기상청 사이트에서 기상정보 가져오기
4
5 import urllib.request
6 import urllib.parse
7
8 API = 'http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp'
9
10 locID = {'stnId':'109'}      # 서울/경기도 109
11
12 params = urllib.parse.urlencode(locID)
13 url = API + "?" + params
14 print(url)
15
16 rtnData = urllib.request.urlopen(url).read()
17 rtnText = rtnData.decode("utf-8")
18 print(rtnText)
19
```



[실습 4] 기상청 사이트에서 기상정보 가져오기 2

```
day1_python_programming_15.py
1 # Scraping & Crawling
2
3 # [실습 4] 기상청 사이트에서 기상정보 가져오기
4
5 import urllib.request as req
6 import urllib.parse
7 from bs4 import BeautifulSoup
8
9 API = 'http://www.kma.go.kr/weather/forecast/mid-term-rss3.jsp'
10
11 locID = {'stnId':'109'}      # 서울/경기도 109
12 # 전국 108, 서울/경기도 109, 강원도 105, 충청북도 131, 충청남도 133
13 # 전라북도 146, 전라남도 156, 경상북도 143, 경상남도 159, 제주특별자치도 184
14
15 params = urllib.parse.urlencode(locID)
16 url = API + "?" + params
17 print(url)
18
19 source = req.urlopen(url)
20 #print(rtnData_Raw)
21 #
22 source = BeautifulSoup(source, "html.parser")
23 #print(rtnData_BS)
24 #print(source.prettify())
25
26 tmpTitle = source.find('title').string
27 print(tmpTitle)
28
29 tmpProvince = source.find('province').string
30 print(tmpProvince)
31
32 lstDate = source.find_all('tmef')
33 #print(len(lstDate))
34
35 for i in range(1, len(lstDate)):
36     tmpDate = source.find_all('tmef')[i-1].text
37     tmpWeather = source.find_all('wf')[i-1].text
38
39     print(tmpProvince, ', ', tmpDate, ', ', tmpWeather)
40
```

Quiz 1

- 기상청 싸이트에서 아래 요구조건을 만족시키면서 일기예보를 조회하시오.
 - 지역명을 입력으로 받는다 (input 이용)
 - 해당 지역에 대한 지역코드를 검색한다. (if 문 이용)
단, 해당 지역에 대한 지역코드가 정의되어 있지 않는 경우, '해당 지역코드 미정의'를 출력한다.
 - 함수를 사용한다.

Scraping & Crawling

신세계 : 네이버 금융

finance.naver.com/item/sise.nhn?code=004170

NAVER 금융 종목명·펀드명·환율명·원자재명 입력 통합검색 오유근 57

금융 홍 국내증시 해외증시 시장지표 펀드 리서치 뉴스 MY

신세계 004170 코스피 2020.02.04 기준(장마감) 즐시관 기업개요

273,000 전일 269,000 고가 280,000 (상한가 349,500) 거래량 91,740 시가 268,500 저가 263,500 (하한가 188,500) 거래대금 25,074 백만

종합정보 시세 차트 | 투자자별 매매동향 | 뉴스공시 | 종목분석 | 종목토론실 | 전자공시 | 공매도현황

주요 시세

	현재가	전일대비	등락률(%)	거래량	거래대금(백만)	액면가
현재가	273,000	▲ 4,000	+1.49%	273,500	25,074	5,000원
전일대비				273,000		25,074
등락률(%)				269,000		25,074
거래량				91,740		25,074
거래대금(백만)				268,500		25,074
액면가				263,500		25,074
상한가	349,500			347,000		
하한가	188,500			187,000		
PER	11.25	EPS	24,274			
52주 최고	345,500	52주 최저	213,500			
시가총액	26,877억원	상장주식수	9,845,181			
외국인현재	2,993천주	자본금	49,225백만			

종가 (20분 지연) *5단계 * 10단계

	매도잔량	매도호가	매수호가	매수잔량
매도잔량	1,236	275,500		
52주최고	52주 최저	345,500	213,500	
PER EPS(FnGuide)		11.25배	24,274원	
PER EPS(KRX)		11.66배	23,415원	
추정PER EPS		N/A	N/A	
PBR BPS(FnGuide)		0.77배	352,681원	
배당수익률		N/A		
동일업종 PER		23.46배		
동일업종 등락률			+0.22%	

273,000 251
272,500 255
272,000 747
271,500 1,158
271,000 982

최근 조회된 종목이 없습니다.

하이투자증권 (비대면 신규/휴면-2.28) 주식 신용/대출 6개월간 2.99% 여 %

Elements Console Sources Network Performance Memory Application Styles Computed Event Listeners

```

<script type="text/javascript" src="/js/myfinance/MyFinance.Util.js?z"></script>
<script type="text/javascript" src="/js/myfinance/MyFinance.MyStock.ItemAddlayer.js?z"></script>
<!-- //종목 summary -->
<!-- 공통메뉴 영역 -->
<h3 class="blind">시세</h3>
<!-- Sub Content -->
<div class="section inner_sub">
<div style="float:left; width:381px; margin-right:9px;">
<h4 class="tline2"></h4>
<div summary="주요시세 정보에 관한 표입니다." cellspacing="0" class="type2_type_tax">
<caption>주요시세</caption>
<tbody>
<tr>
<th class="title">현재가</th>
<td class="num">
<strong class="tah p11" id="nowVal">273,000<strong> == $0
</td>
<th class="title">매도호가</th>
<td class="num">
<strong class="tah p11" id="nowVal">273,000<strong> == $0
</td>
</tr>
<tr>
<th class="title">매수호가</th>
<td class="num">
<strong class="tah p11" id="nowVal">273,000<strong> == $0
</td>
<th class="title">매수잔량</th>
<td class="num">1,236</td>
</tr>
<tr>
<th class="title">52주최고</th>
<td class="num">>345,500</td>
<th class="title">52주 최저</th>
<td class="num">>213,500</td>
</tr>
<tr>
<th class="title">PER | EPS(FnGuide)">
<td class="num">>11.25배</td>
<th class="title">PER | EPS(KRX)">
<td class="num">>11.66배</td>
</tr>
<tr>
<th class="title">추정PER | EPS">
<td class="num">>N/A</td>
<th class="title">PBR | BPS(FnGuide)">
<td class="num">>0.77배</td>
</tr>
<tr>
<th class="title">배당수익률">
<td class="num">>N/A</td>
<th class="title">동일업종 PER">
<td class="num">>23.46배</td>
</tr>
<tr>
<th class="title">동일업종 등락률">
<td class="num">>+0.22%</td>
</tr>



```

참고문헌

- 파이썬을 이용한 머신러닝, 딥러닝 실전개발 입문, 위키북스
- 파이썬을 활용한 금융공학 레시피, 김용환, 한빛미디어

Scraping & Crawling

- Web site에서 stock price 추출하기
 - [실습 5] BeautifulSoup 활용
 - [실습 6] Timer를 이용하여 매 1초마다 추출
 - [실습 7] CSV 파일에 저장
 - [실습 8] 주가 추이 그래프 그리기

[실습 5] BeautifulSoup 활용

```
day1_python_programming_15.py day1_python_programming_16.py day1_python_programming_17.py

1
2 # [실습 5] 주가 조회하기
3
4 import bs4
5 from urllib.request import urlopen
6
7 idx_stock = '004170'
8 naver_index = 'https://finance.naver.com/item/sise.nhn?code=' + idx_stock
9
10 source = urlopen(naver_index).read()
11 source = bs4.BeautifulSoup(source, 'lxml')
12 #print(source.prettify())
13
14 price = source.find_all('td', class_='num')[0]
15 print(price)
16
17 tmpStrong = source.find_all('strong', class_='tah p11')[0]
18 #tmpStrong = source.find_all('strong')[0]
19 tmpValue = tmpStrong.text
20 print(tmpValue)
21
```

[실습 6] Timer를 이용하여 매 1초마다 추출

```
1 # [실습 6] 주기 조회하기 - Timer
2
3
4 from threading import Timer
5 from time import sleep
6 import bs4
7 from urllib.request import urlopen
8
9 # Timer 출처
10# https://stackoverflow.com/questions/3393612/run-certain-code-every-n-seconds
11class RepeatedTimer(object):
12    def __init__(self, interval, function, *args, **kwargs):
13        self._timer      = None
14        self.interval   = interval
15        self.function   = function
16        self.args       = args
17        self.kwargs     = kwargs
18        self.is_running = False
19        self.start()
20
21    def _run(self):
22        self.is_running = False
23        self.start()
24        self.function(*self.args, **self.kwargs)
25
26    def start(self):
27        if not self.is_running:
28            self._timer = Timer(self.interval, self._run)
29            self._timer.start()
30            self.is_running = True
31
32    def stop(self):
33        self._timer.cancel()
34        self.is_running = False
```

Timer Class

<https://stackoverflow.com/questions/3393612/run-certain-code-every-n-seconds>



[실습 6] Timer를 이용하여 매 1초마다 추출

```
36 def getCurrentStockPrice(idx_stock):
37     naver_index = 'https://finance.naver.com/item/sise.nhn?code=' + idx_stock
38     source = urlopen(naver_index).read()
39     source = bs4.BeautifulSoup(source, 'lxml')
40
41     curPrice = source.find_all('strong', class_='tah p11')[0].text
42     print(curPrice)
43
44     return curPrice
45
46 print('Starting...')
47 idx_stock = '004170'
48 rt = RepeatedTimer(1, getCurrentStockPrice, idx_stock)
49 try:
50     sleep(10)
51 finally:
52     rt.stop()
```

[실습 7] CSV 파일에 저장

```
35 def getCurrentStockPrice(idx_stock):
36     naver_index = 'https://finance.naver.com/item/sise.nhn?code=' + idx_stock
37     source = urlopen(naver_index).read()
38     source = bs4.BeautifulSoup(source, 'lxml')
39
40     curPrice1 = source.find_all('em', class_='no_up')[0]
41     #     print(curPrice1)
42
43     curPrice2 = curPrice1.find_all('span')[0].text
44     #     print(curPrice2)
45
46     #     curPrice = int(curPrice2.replace(',', ''))
47     curPrice = curPrice2.replace(',', '')
48
49     tmpTime = dt.datetime.now()
50     #     curTime = str(tmpTime.year) + '-' + str(tmpTime.month) + '-' + str(tmpTime.day) \
51     #             + ' ' \
52     #             + str(tmpTime.hour) + ':' + str(tmpTime.minute) + ':' + str(tmpTime.second)
53     curTime = tmpTime.strftime('%Y-%m-%d %H:%M:%S')
54
55     print(idx_stock, ',', curTime, ',', curPrice)
56
57     return curPrice
58
59 print('Starting...')
60 idx_stock = '004170'
61 rt = RepeatedTimer(1, getCurrentStockPrice, idx_stock)
62 try:
63     sleep(10)
64 finally:
65     rt.stop()
66
67
68
```



[실습 8] 주가 추이 그래프 그리기

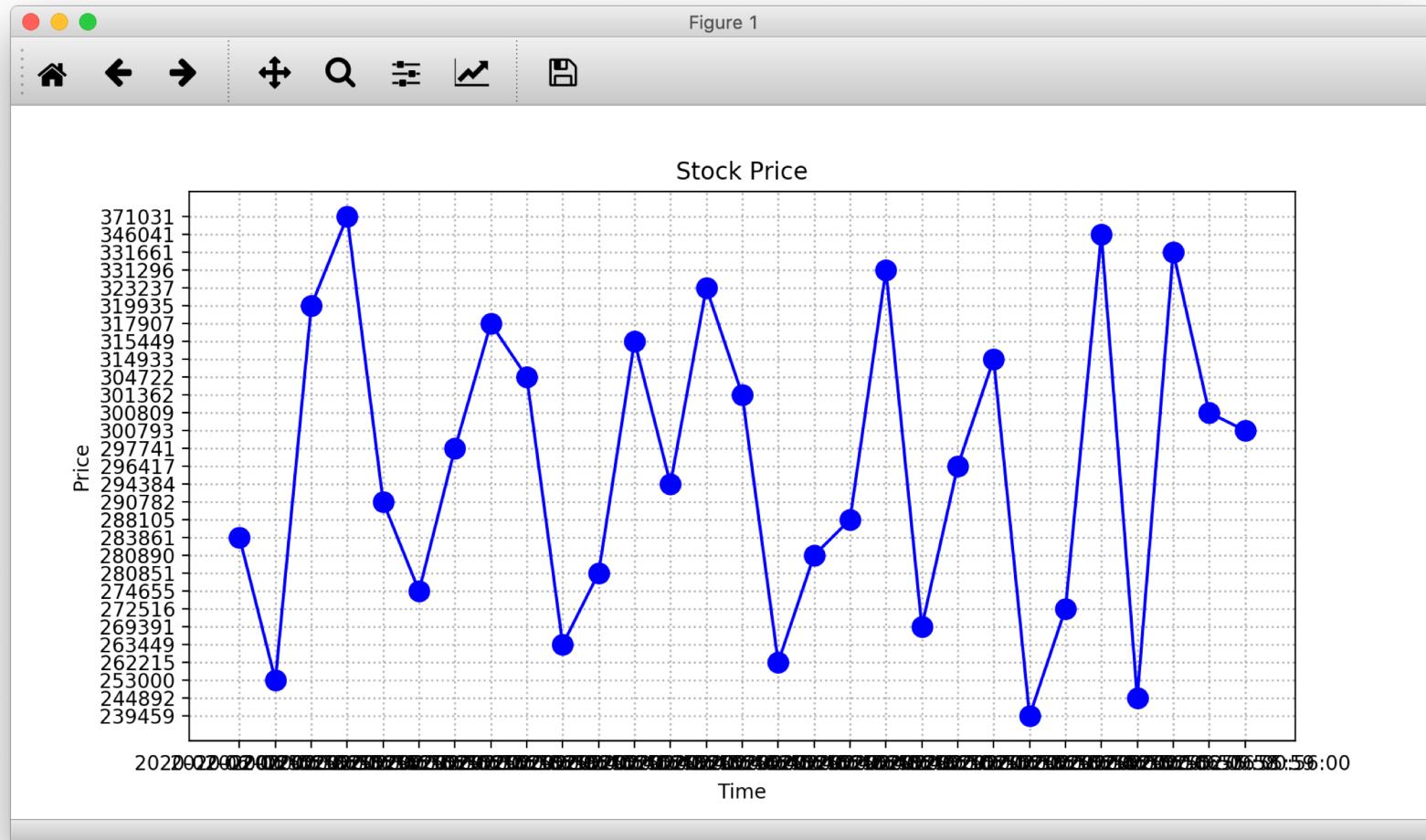
```
① [x] day1_python_programming_17.py [x] day1_python_programming_18.py [x] untitled20.py [x] day1_python_programming_19.py*
1 # [실습 7] 주가 조회하기 - Timer
2 from threading import Timer
3 from time import sleep
4 import bs4
5 from urllib.request import urlopen
6 import datetime as dt
7 import random as rnd
8 import math
9 import matplotlib.pyplot as plt
10 import pandas as pd
11
12 # Timer Class 생략
13
14 def getCurrentStockPrice(idx_stock):
15     naver_index = 'https://finance.naver.com/item/sise.nhn?code=' + idx_stock
16     source = urlopen(naver_index).read()
17     source = bs4.BeautifulSoup(source, 'lxml')
18
19     curPrice1 = source.find_all('em', class_='no_up')[0]
20     curPrice2 = curPrice1.find_all('span')[0].text
21     curPrice = curPrice2.replace(',', '')
22     curPrice = float(curPrice2.replace(',', ''))
23
24     tmpTime = dt.datetime.now()
25     curTime = tmpTime.strftime('%Y-%m-%d %H:%M:%S')
26
27     rndCurPrice = math.ceil(curPrice * rnd.gauss(1, 0.1))
28     print(idx_stock, ',', curTime, ',', rndCurPrice)
29
30     historicalData[curTime] = str(rndCurPrice)
31
32 return historicalData
33
```



[실습 8] 주가 추이 그래프 그리기

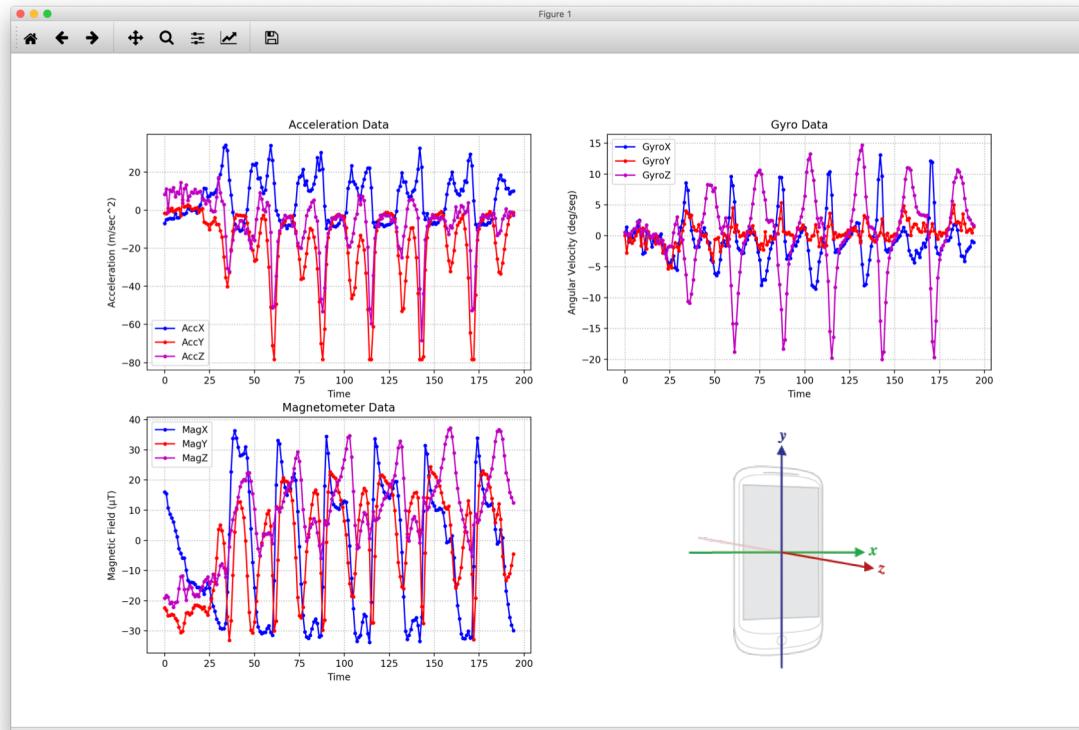
```
61 print('Starting... ')
62 idx_stock = '004170'
63
64 historicalData = dict()
65 rt = RepeatedTimer(1, getCurrentStockPrice, idx_stock)
66 try:
67     sleep(30)
68 finally:
69     rt.stop()
70
71 tmpData4Plot = {'신세계': historicalData}
72 dfHistoricalData = pd.DataFrame(tmpData4Plot)
73 dfHistoricalData.to_csv('004170_Stock_Price.csv')          # CSV 저장
74
75 plt.figure(1, figsize = (10, 5))
76 plt.plot(dfHistoricalData, color='b', linestyle='-', marker='o', markersize=10)
77 plt.legend(loc=0)
78 plt.grid(True, color='0.7', linestyle=':', linewidth=1)
79 plt.xlabel('Time')
80 plt.ylabel('Price')
81 plt.title('Stock Price')
82 plt.show()
```

[실습 8] 주가 추이 그래프 그리기



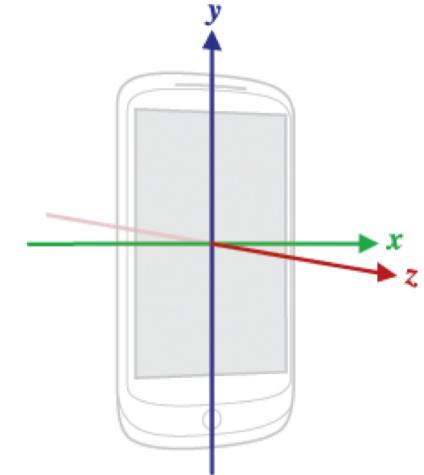
[실습 9] IMU 데이터 실시간 취합

- IMU (Inertial Measurement Unit)
 - 3-axis accelerometer: x-, y-, z-direction acceleration
 - 3-axis gyroscope: x-, y-, z-direction angular velocity
 - 3-axis magnetometer: x-, y-, z-direction magnet field



[실습 9] IMU 데이터 실시간 취합

- 준비 작업
 - Andorid cellphone에 App 설치
 - IP address 확인
 - i. Windows: command prompt에서 ipconfig
 - ii. Mac OS: command prompt에서 ifconfig |grep inet
- getCurrentIMUData() 완성하기
 - 아래 sample code 참고



```
import socket, traceback
```

```
host = ""  
port = 5555  
  
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)  
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)  
s.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)  
s.bind((host, port))
```

```
while 1:  
    try:  
        message, address = s.recvfrom(8192)  
        print (message)  
    except (KeyboardInterrupt, SystemExit):  
        raise  
    except:  
        traceback.print_exc()
```

[실습 9] IMU 데이터 실시간 취합

- First try!

```
import socket, traceback
```

```
host = "  
port = 5555
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)  
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)  
s.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)  
s.bind((host, port))
```

```
while 1:  
    try:  
        message, address = s.recvfrom(8192)  
        print (message)  
    except (KeyboardInterrupt, SystemExit):  
        raise  
    except:  
        traceback.print_exc()
```

[실습 9] IMU 데이터 실시간 취합

```
1 # [실습 9] Timer를 이용하여 IMU Data 취합하기
2 import socket
3 from threading import Timer
4 from time import sleep
5 #import datetime as dt
6 import matplotlib.pyplot as plt
7 import pandas as pd
8 import numpy as np
9
10 # Timer 출처
11 # https://stackoverflow.com/questions/3393612/run-certain-code-every-n-seconds
12 class RepeatedTimer(object):
13     def __init__(self, interval, function, *args, **kwargs):
14         self._timer      = None
15         self.interval   = interval
16         self.function   = function
17         self.args       = args
18         self.kwargs     = kwargs
19         self.is_running = False
20         self.start()
21
22     def _run(self):
23         self.is_running = False
24         self.start()
25         self.function(*self.args, **self.kwargs)
26
27     def start(self):
28         if not self.is_running:
29             self._timer = Timer(self.interval, self._run)
30             self._timer.start()
31             self.is_running = True
32
33     def stop(self):
34         self._timer.cancel()
35         self.is_running = False
```



[실습 9] IMU 데이터 실시간 취합

```
37 def getCurrentIMUData():
38     global imuData
39     global intCnt
40
41 #     try:
42 #         message, address = s.recvfrom(8192)
43 #         curTime = dt.datetime.now()
44
45     strData = message.decode("utf-8")      # convert a byte type to a string
46     strData = strData.split(',')          # convert a string to a list
47
48     if len(strData) >= 13:
49         imuData = np.append(imuData, \
50                             np.array([[intCnt, intCnt, \
51                                         float(strData[2]), float(strData[3]), float(strData[4]) ], \
52                                         [float(strData[6]), float(strData[7]), float(strData[8]) ], \
53                                         [float(strData[10]), float(strData[11]), float(strData[12])]]), axis=0)
54         intCnt += 1
55
56 #         print(imuData)
57 #     except (KeyboardInterrupt, SystemExit):
58 #         raise
59 #     except:
60 #         traceback.print_exc()
61
62     return
```

[실습 9] IMU 데이터 실시간 취합

```
64 print('Starting... ')
65
66 host = '192.168.35.11'
67 port = 5555
68
69 s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
70 s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
71 s.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)
72 s.bind((host, port))
73
74 intCnt = 0
75 imuData = np.empty((0,11), float)
76
77 rt = RepeatedTimer(0.1, getCurrentIMUData)
78 try:
79     sleep(20)
80 finally:
81     rt.stop()
82
83
84 imuIdx = imuData[:,0]
85 accX = imuData[:,2];    accY = imuData[:,3];    accZ = imuData[:,4]
86 gyrX = imuData[:,5];    gyrY = imuData[:,6];    gyrZ = imuData[:,7]
87 magX = imuData[:,8];    magY = imuData[:,9];    magZ = imuData[:,10]
88
89 fig1 = plt.figure(1, figsize = (10, 10))
90 #-----
91 ax1 = fig1.add_subplot(2,2,1)
92 ax1.plot(imuIdx, accX, color='b', linestyle='-', marker='o', markersize=3, label='AccX')
93 ax1.plot(imuIdx, accY, color='r', linestyle='-', marker='o', markersize=3, label='AccY')
94 ax1.plot(imuIdx, accZ, color='m', linestyle='-', marker='o', markersize=3, label='AccZ')
95
96 ax1.legend(loc=0)
97 ax1.grid(True, color='0.7', linestyle=':', linewidth=1)
98 ax1.set_xlabel('Time')
99 ax1.set_ylabel('Acceleration (m/sec^2)')
100 ax1.set_title('Acceleration Data')
```



[실습 9] IMU 데이터 실시간 취합

```
102 ax2 = fig1.add_subplot(2,2,2)
103 ax2.plot(imuIdx, gyrX, color='b', linestyle='-', marker='o', markersize=3, label='GyroX')
104 ax2.plot(imuIdx, gyrY, color='r', linestyle='-', marker='o', markersize=3, label='GyroY')
105 ax2.plot(imuIdx, gyrZ, color='m', linestyle='-', marker='o', markersize=3, label='GyroZ')
106
107 ax2.legend(loc=0)
108 ax2.grid(True, color='0.7', linestyle=':', linewidth=1)
109 ax2.set_xlabel('Time')
110 ax2.set_ylabel('Angular Velocity (deg/seg)')
111 ax2.set_title('Gyro Data')
112 #
113 ax3 = fig1.add_subplot(2,2,3)
114 ax3.plot(imuIdx, magX, color='b', linestyle='-', marker='o', markersize=3, label='MagX')
115 ax3.plot(imuIdx, magY, color='r', linestyle='-', marker='o', markersize=3, label='MagY')
116 ax3.plot(imuIdx, magZ, color='m', linestyle='-', marker='o', markersize=3, label='MagZ')
117
118 ax3.legend(loc=0)
119 ax3.grid(True, color='0.7', linestyle=':', linewidth=1)
120 ax3.set_xlabel('Time')
121 ax3.set_ylabel('Magnetic Field ( $\mu$ T)')
122 ax3.set_title('Magnetometer Data')
123 #
124 fig1.show()
125
126
127 dfImuData = pd.DataFrame(imuData)
128 dfImuData.to_csv('IMU_Data_Period_0_1s.csv')
```