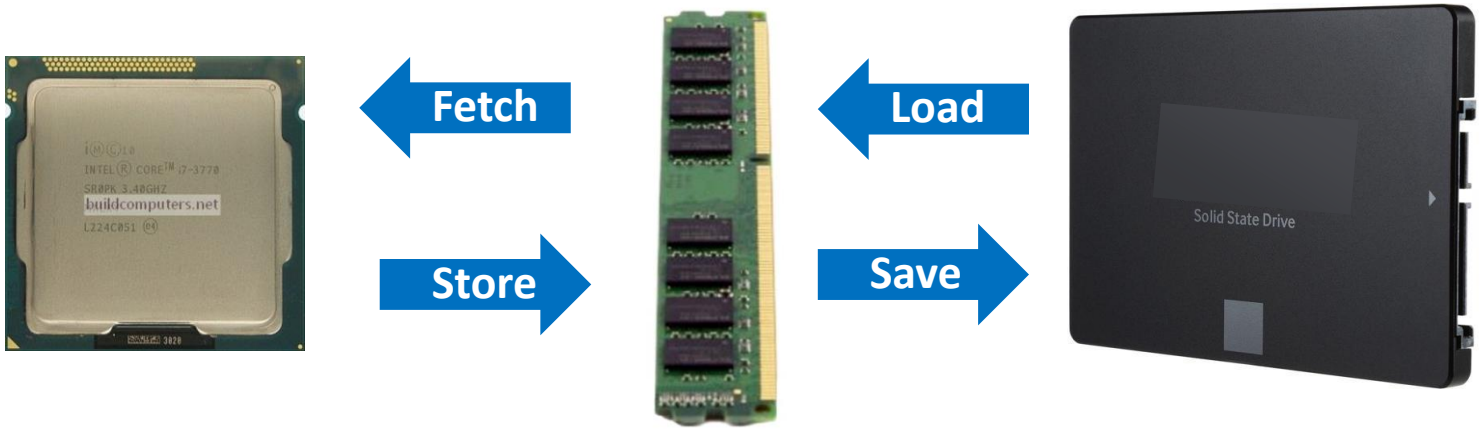
Text File Processing and Dictionary

Contents

- Text File
- Reading Text File
- Writing Text File
- Dictionary
- Dictionary vs. List
- Computational Problem

Text File

- CPU는 메모리의 명령어와 데이터를 가져와(Fetch) 연산을 수행하고 연산 결과를 메모리에 저장(Store)
- 메모리의 용량은 제한되어 있으므로 명령어와 데이터를 저장장치(HDD, SSD등)에 **파일(File)** 형태로 보관
- CPU에서 연산이 필요한 내용은 메모리에 적재(Load)하여 실행하고, 연산이 끝나면 다시 저장장치에 보관(Save)
- 프로그램과 데이터는 **파일** 형태로 영구적으로 보관이 가능



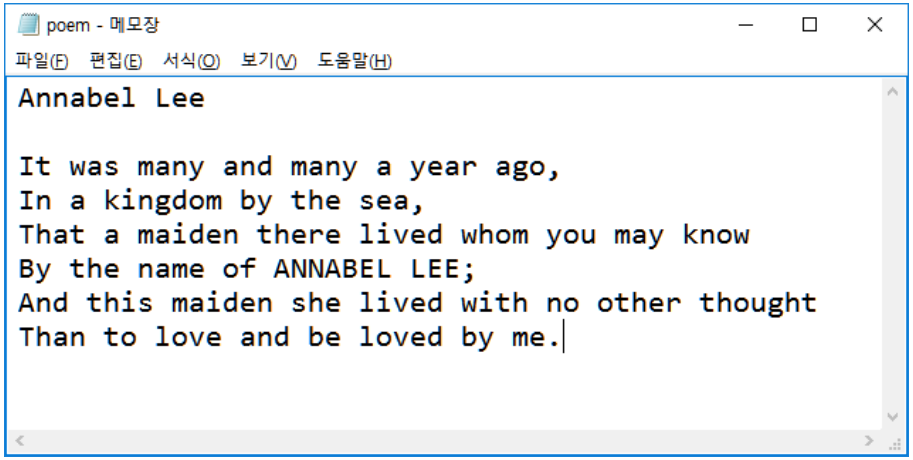
Text File

파일의 종류

- 텍스트 파일(Text File)
  - 사람이 읽을 수 있는 문자로 구성된 파일
  - 텍스트 파일의 예
    - ✓문서(txt), 웹(html) 및 프로그램 소스(py) 파일 등
- 이진 파일(Binary File)
  - 사람이 읽을 수 없는 형식의 데이터로 구성
  - 이진 파일의 예
    - ✓이미지(jpg), 사운드(mp3), 실행(exe) 및 한글(hwp) 파일 등
- 파일 사용의 필요성
  - 정보의 영구적인 저장
  - 대용량 데이터의 저장

Text File

- 텍스트 파일



poem.txt

- 이진 파일

explorer.exe

Reading Text File

- Python에서의 텍스트 파일 읽기

```
file = open('c:/work/file.txt')
```

open(filename): 인자(filename)로 지정한 파일을 열어 사용가능한 상태로 설정하고 파일 객체를 반환

```
print(file.read())
```

read(): 파일 객체의 전체 내용을 문자열로 반환

```
file.close()
```

close(): 파일 객체를 닫고 파일의 사용을 종료

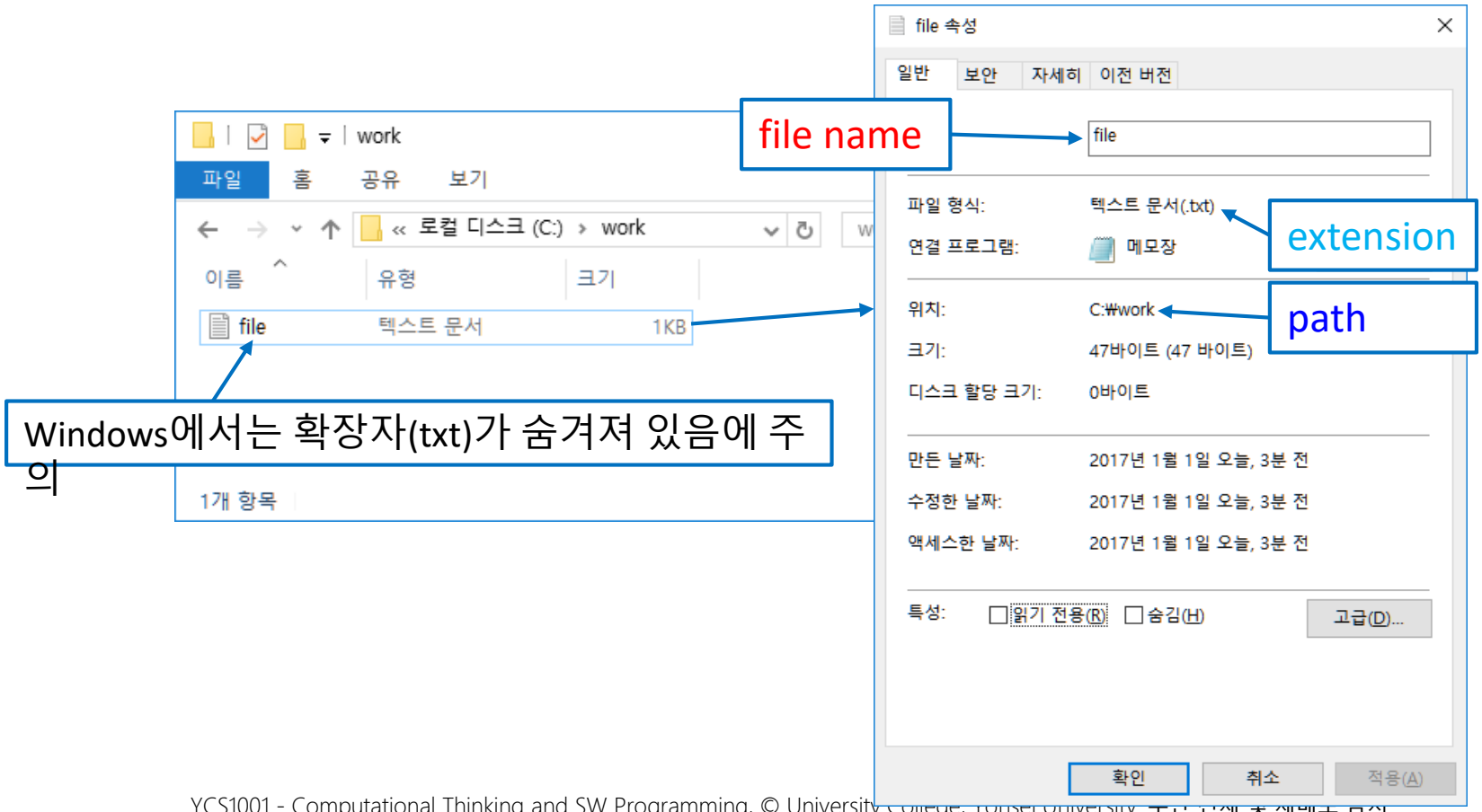
*파일 사용 후 꼭 파일을 닫습니다. 파일을 닫지 않으면 운영체제에서 해당 파일을 사용하지 못할 수 있습니다.

Reading Text File

- 파일명(filename)의 표현방법

  파일명은 'path' + 'file name' + '.' + 'extension' 형식의 문자열로 표현

  예) 'c:/work/file.txt'



Reading Text File

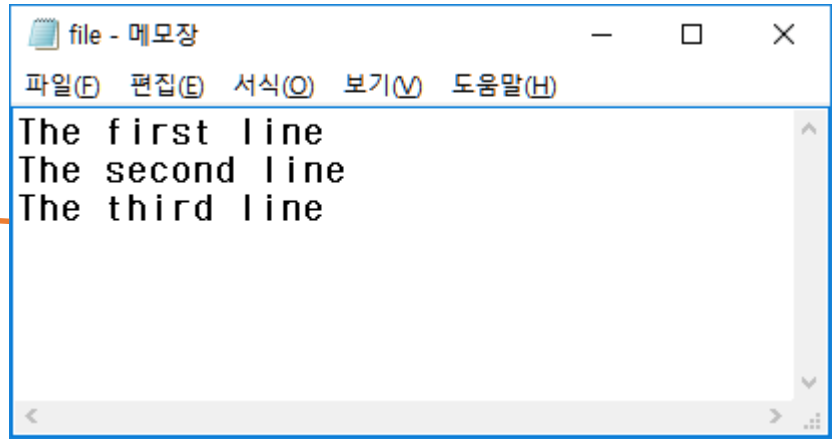
- 예1) 텍스트 파일 읽기 및 내용 출력

```
filename = 'c:/work/file.txt'
inputFile = open(filename)
content = inputFile.read()
inputFile.close()

print(content)
```

출력결과

```
The first line
The second line
The third line
>>>
```



Reading Text File

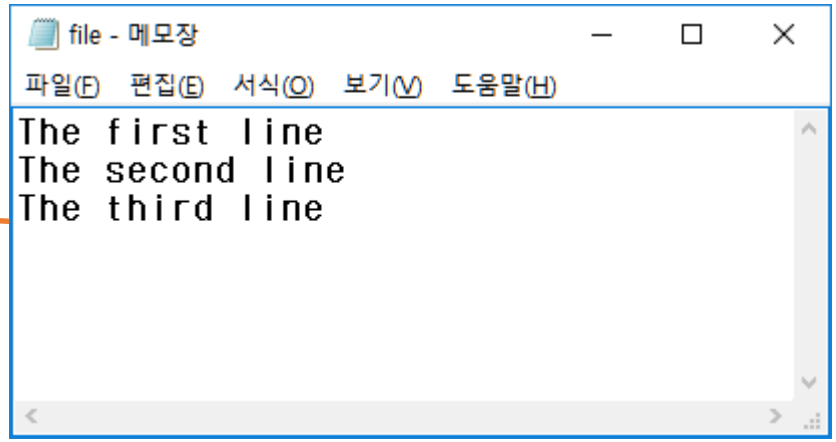
- 예2) 텍스트 파일을 줄 단위로 읽기

```
filename = 'c:/work/file.txt'
inputFile = open(filename)
for line in inputFile:
    print(line, end='')
inputFile.close()
```

텍스트 파일 내용에 \n이 포함되어 있으므로 print()의 줄 바꿈 기능을 사용하지 않음

출력결과

```
The first line
The second line
The third line
>>>
```



Writing Text File

- 텍스트 파일 쓰기

```
file = open('c:/work/newfile.txt', 'w')
```

open(filename, 'w'): 지정한 파일을 쓰기 모드로 열어 객체 반환

```
file.write('text')
```

write(string): 인자로 넘겨준 string을 파일에 쓰기

```
file.close()
```

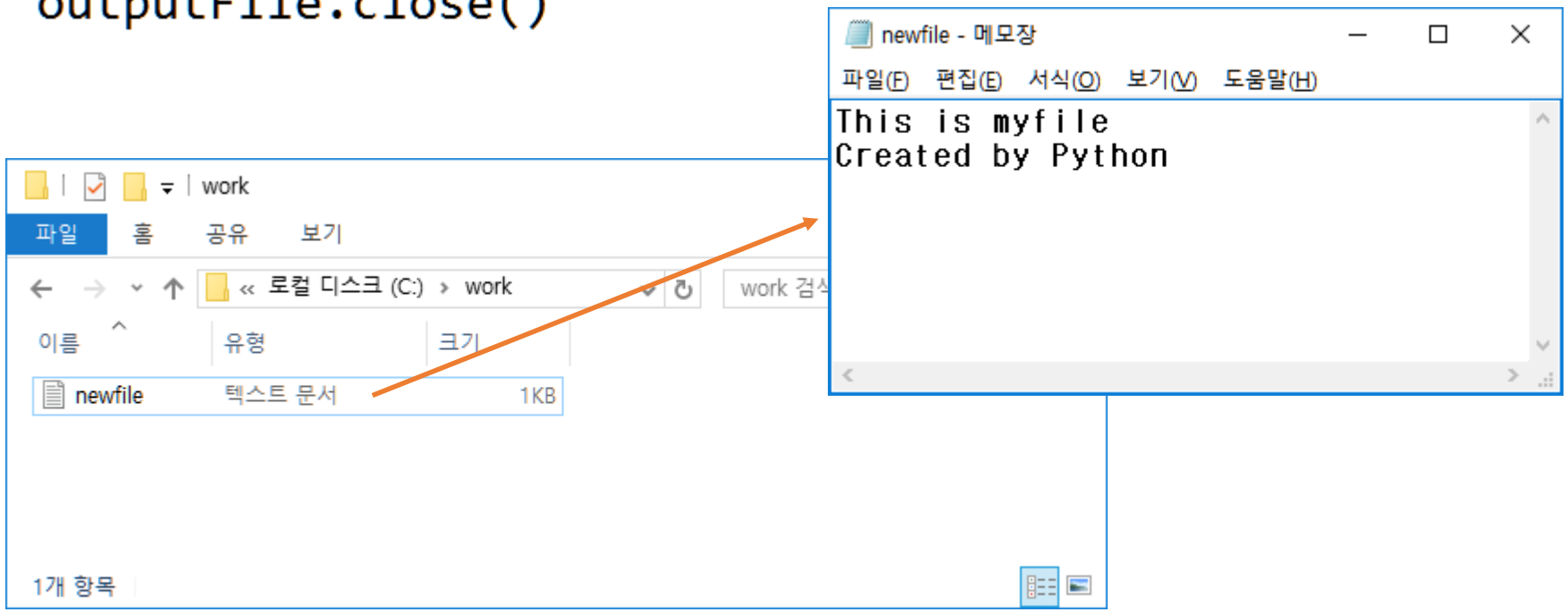
close(): 파일을 닫아 파일의 사용을 종료

*파일 사용 후 꼭 파일을 닫습니다. 파일을 닫지 않으면 write()로 작성한 내용이 파일에
제대로 반영되지 않을 수 있습니다

Writing Text File

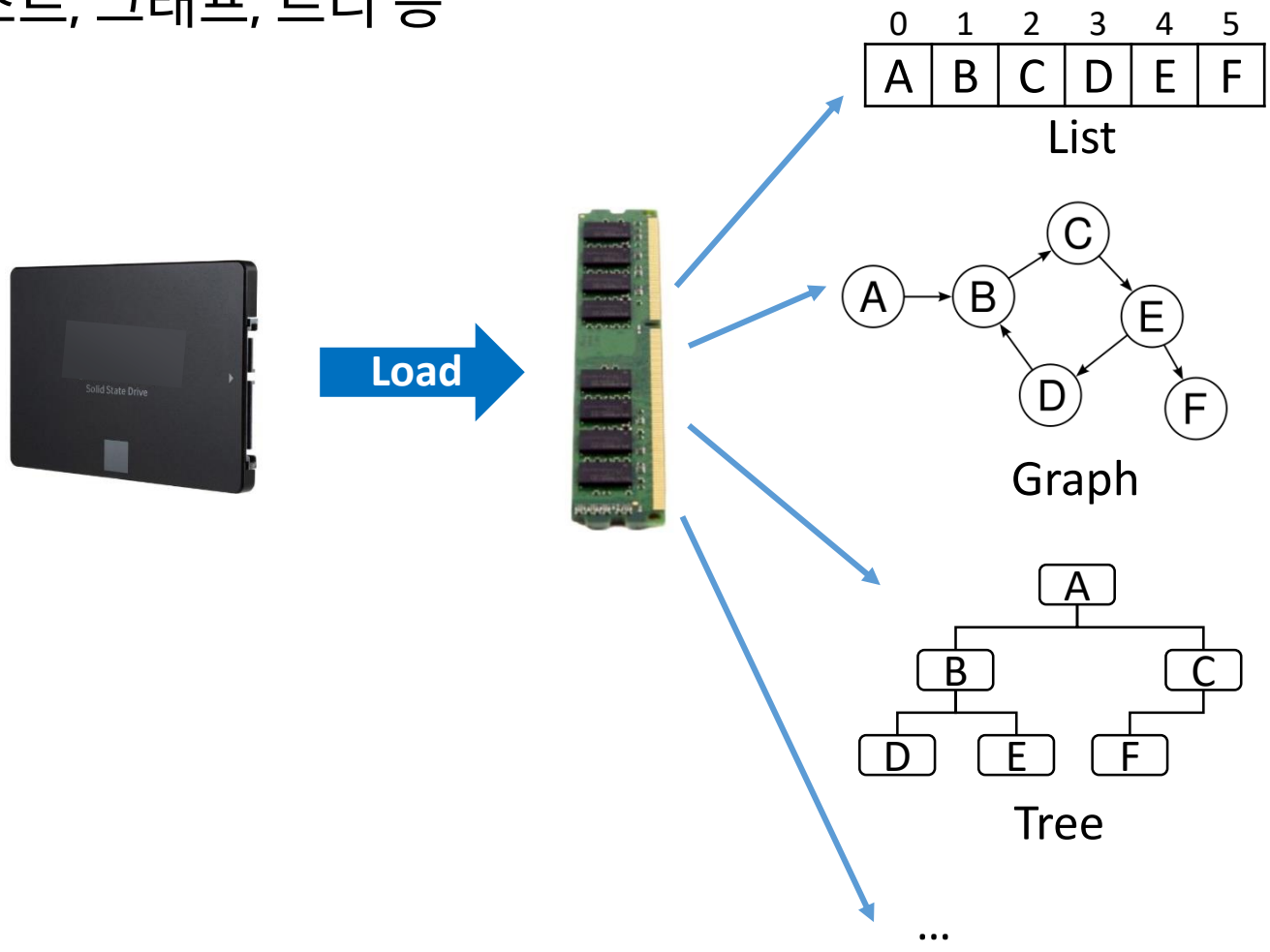
- 예3) 텍스트 파일 쓰기

```
filename = 'c:/work/newfile.txt'
outputFile = open(filename, 'w')
content = 'This is myfile\nCreated by Python'
outputFile.write(content)
outputFile.close()
```



Dictionary

- 파일과 같은 대용량 데이터를 메모리에 적재하기 위해서는 적합한 데이터 구성(data organization)이 필요
  - 리스트, 그래프, 트리 등



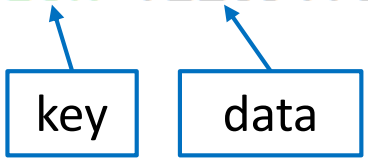
Dictionary

- Dictionary는 키(key) 값으로 데이터(data)에 바로 접근할 수 있는 데이터 구성
- 사전에서 단어의 철자(key)로 단어의 뜻(data)을 바로 찾을 수 있는 것과 같은 유사한 원리로 데이터를 구성함
- 대용량 데이터에서 키 값으로 원하는 데이터를 빠르게 접근 할 때 효율적

Dictionary

- Python Dictionary: 키(key)와 데이터(data)의 쌍으로 데이터를 저장하고 접근
- Python Dictionary 형식: **{ key : data }**
  - key와 data는 문자열, 정수, 실수 등 모든 유형이 가능
  - 예) 각 나라의 환율을 표현하는 dictionary의 예
    - dictionary의 선언

```
exchangeRate = {'USD':1208.00, 'JPY':1033.54,
                'EUR':1269.79, 'CNY':173.18}
```



    - key 값을 이용한 data의 접근

```
>>> exchangeRate['EUR']
1269.79
```

Dictionary Operations

```
d = {'USD':1208.00, 'JPY':1033.54, 'EUR':1269.79, 'CNY':173.18}
```

Operation	Description	Example
`d[key] = data`	key 값에 data를 갱신하거나 새로운 key와 data를 추가	`d['USD'] = 1198.87`
`del d[key]`	key와 data쌍을 삭제	`del(d['USD'])`
`d.keys()`	dictionary에 있는 모든 key를 객체 형태로 반환	`for key in d.keys():` `  print(key)`

Dictionary vs. List

- list
  - index로 직접 value를 접근
  - index 값이 어떤 의미인지 알기 어려움(단순 정수)
  - index에 따라 value가 순차적으로 저장됨 (정렬 가능)

- dictionary
  - key에 연관된 value를 접근
  - key값의 의미를 알기 쉬움
  - key와 value가 저장되는 위치를 알 수 없음(정렬 불가능)

```
exchangeRate = [1208.00, 1033.54,
                1269.79, 173.18]
```

index	value
0	1208.00
1	1033.54
2	1269.79
3	173.18

exchangeRate[1]

```
exchangeRate = {'USD':1208.00, 'JPN':1033.54,
                'EUR':1269.79, 'CNY':173.18}
```

key	value
'JPY'	1033.54
'EUR'	1269.79
'CNY'	173.18
'USD'	1208.00

exchangeRate['EUR']

Dictionary vs. List

- List를 이용한 환율 변환 프로그램

```
exchangeRate = [1208.00, 1033.54, 1269.79, 173.18]
print('This program convert KRW to other currency')
amount = int(input('Enter amount of KRW: '))
symbol = input('Enter USD, JPY, EUR or CNY: ')
if symbol == 'USD':
    print(symbol, 'rate:', exchangeRate[0])
    result = amount / exchangeRate[0]
elif symbol == 'JPY':
    print(symbol, 'rate:', exchangeRate[1])
    result = amount / exchangeRate[1]
elif symbol == 'EUR':
    print(symbol, 'rate:', exchangeRate[2])
    result = amount / exchangeRate[2]
elif symbol == 'CNY':
    print(symbol, 'rate:', exchangeRate[3])
    result = amount / exchangeRate[3]

print('Result:', format(result,'.2f'), symbol)
```

Dictionary vs. List

- Dictionary를 이용한 환율 변환 프로그램

```
exchangeRate = {'USD':1208.00, 'JPY':1033.54,
                'EUR':1269.79, 'CNY':173.18}
print('This program convert KRW to other currency')
amount = int(input('Enter amount of KRW: '))
symbol = input('Enter USD, JPY, EUR or CNY: ')
result = amount / exchangeRate[symbol]

print(symbol, 'rate:', exchangeRate[symbol])
print('Result:', format(result,'.2f'), symbol)
```

```
This program convert KRW to other currency
Enter amount of KRW: 10000
Enter USD, JPY, EUR or CNY: USD
USD rate: 1208.0
Result: 8.28 USD
>>>
```

Computational Problem

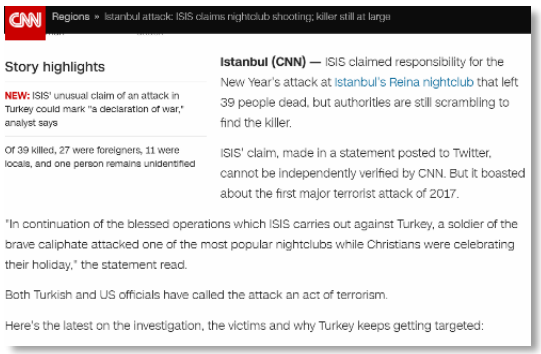
The Problem

World News Analysis

2017년 1월 2일의 BBC, NBC, CNN의 헤드라인 뉴스 기사의 텍스트를 분석하여 기사에서 사용된 단어의 횟수를 파악하고, 사용자로 부터 정수값을 입력 받아 그 이상으로 사용된 단어를 출력

- 세 개의 뉴스 기사의 텍스트는 하나의 파일(news.txt)로 통합하여 제공 됨
- stopwords.txt로 제공 된 단어들은 횟수를 파악 하지 않음

  - Stop words : a, an, the, am, he, she, who, where 등의 단어



World News Analysis

Problem Analysis

프로그램 실행 시, 사용자로 부터 정수값을 입력 받음

news.txt 파일 내용에서 사용된 각 각의 단어 빈도를 파악하되, stopwords.txt 파일 내용에 있는 단어의 빈도는 파악하지 않음

단어의 빈도를 모두 파악하였으면, 사용자가 입력한 정수값 보다 큰 빈도를 가지는 단어를 모두 출력

World News Analysis

Data Representation

Variable Name	Description
newsText	news.txt 파일의 텍스트
stopwordsText	stopwords.txt 파일의 텍스트
newsWordList	news.txt의 모든 단어 리스트
stopWordList	stopwords.txt의 모든 단어 리스트
FreqWordsDict	newsWordList의 각 단어의 사용 빈도 (dictionary)
threshold	임계값(사용자 입력 값)

World News Analysis

Algorithmic Thinking

① 사용자로부터 threshold 값을 입력

② news.txt로 부터 모든 단어를 추출하고 리스트 newsWordList로 구성

③ stopword.txt로 부터 모든 단어를 추출하고 리스트 stopWordList로 구성

④ stopWordList에 없는 newsWordList의 단어 빈도를 파악하여 freqWordDict에저장

⑤ 빈도가 threshold 이상의 단어를 모두 출력

World News Analysis

Algorithmic Thinking

활용 가능한 주요 Python 기능들

① 텍스트로부터 영단어를 추출

Regular Expression

② 단어들의 사용 빈도를 파악

dictionary = { *word*$_1$: *frequency*$_1$, *word*$_2$: *frequency*$_2$ … }

③ Stop word는 빈도 제외

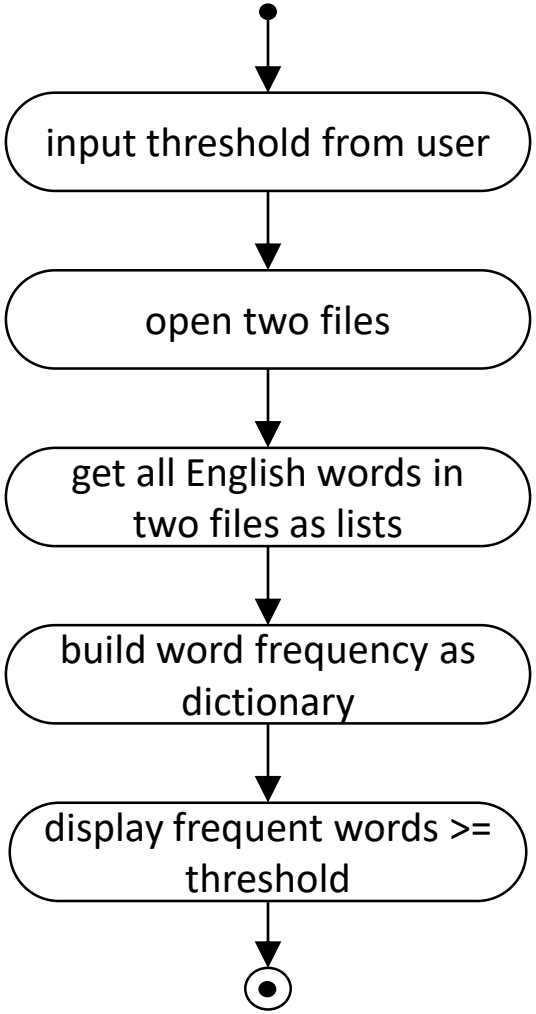
Membership operator

④ 특정 빈도 이상의 단어를 출력

```
if dictionary[ word ] >= threshold:
    print(word)
```

World News Analysis

Algorithmic Thinking & Decomposition



World News Analysis

Program Design

Program Introduction

Displaying welcome messages

Input

Input threshold from user
Read text from assigned file news.txt and stopwords.txt

Processing

Extract English words not in stopwords.txt, and count the frequency

Display results

Display the word's frequency

World News Analysis

Program Implementation

```
import re

def openFile(filename):
    file = open(filename)
    text = file.read()
    file.close()
    return text

def getAllEnglishWords(text):
    pattern = r'\b[A-Za-z]*\b'
    words = re.findall(pattern, text)
    return words
```

World News Analysis

Program Implementation

```
def buildWordFrequency(newsWordList, stopWordList):
    freqWordDict = {}
    for word in newsWordList:
        word = word.lower()
        if word not in stopWordList:
            if word not in freqWordDict:
                freqWordDict[word] = 1
            else:
                freqWordDict[word] = freqWordDict[word] + 1
    return freqWordDict

def displayFrequentWords(freqWordDict, threshold):
    for key in freqWordDict.keys():
        if freqWordDict[key] >= threshold:
            print(format(key,'10'), format(freqWordDict[key],'5'))
```

World News Analysis

Program Implementation

```
print('Welcome to World News Analyzer')
print('This program displays most frequent words in news.txt')

newsText = openFile('c:/work/news.txt')
stopwordText = openFile('c:/work/stopwords.txt')

newsWordList = getAllEnglishWords(newsText)
stopWordList = getAllEnglishWords(stopwordText)

freqWordDict = buildWordFrequency(newsWordList, stopWordList)

threshold = int(input('Enter the threshold value: '))
displayFrequentWords(freqWordDict, threshold)
```

World News Analysis

Program Execution

```
Welcome to World News Analyzer
This program displays most frequent words in news.txt
Enter the threshold value: 15
turkey          50
attack          57
istanbul        68
nightclub       44
gunman          15
shot            15
people          29
year            17
killed          19
turkish         21
```

✓ 2017년 1월 2일, 신문사들의 공통된 세계 뉴스 기사의 주제는 '터키 이스탄불의 한 나이트클럽에서 총격이 발생해 사상자가 발생한 것'임을 단어 빈도 분석을 통하여 추측할 수 있음