

File Input and Output

By 윤명근 / 박수현

수업목표

- What is a file ?
- Opening a file
- Reading a file
- Text files and binary files
- Writing to a file
- Pickle

What is a file ?

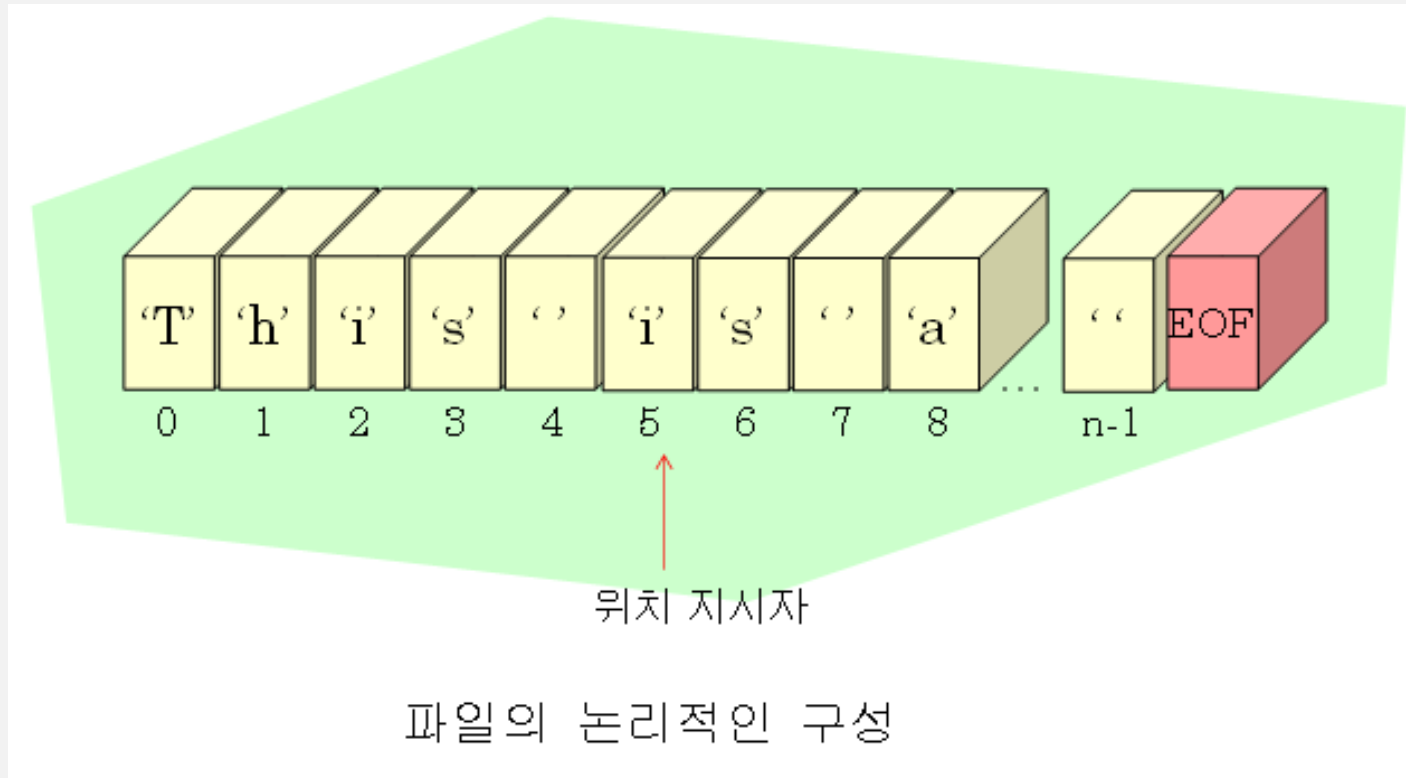
- File
 - Byte의 모음으로 hard disc, SSD(Solid State Disk), Cloud, 기타 저장장치(memory storage) 등에 저장됨
 - Text, 그림, 음악, program, 연락처, execution file 등 다양한 정보 저장

What is a file - file 의 필요성



What is a file - 논리적인 file 구조

- File 안에는 byte 들이 순차적으로 저장되어 있고 맨 끝에는 **End-of-file(EOF) marker**가 있음
- 모든 file은 입출력 동작이 발생하는 위치를 나타내는 **위치 표시자(position indicator)**를 가지고 있음

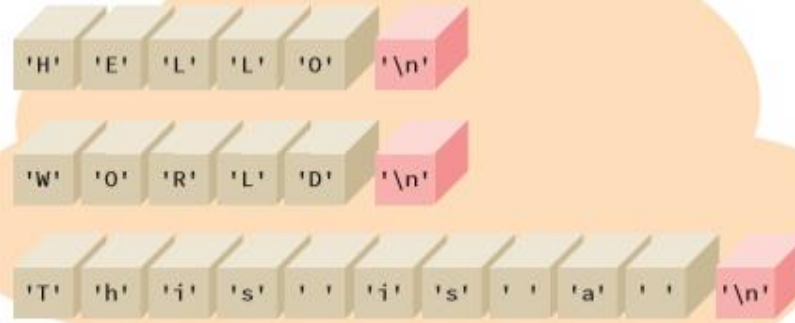


What is a file - file의 종류

- 텍스트 파일(text file)
- 이진 파일(binary file)



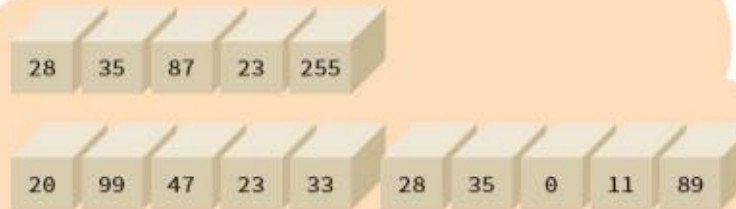
텍스트 파일: 문자로 구성된 파일



텍스트 파일



이진파일: 데이터로 구성된 파일



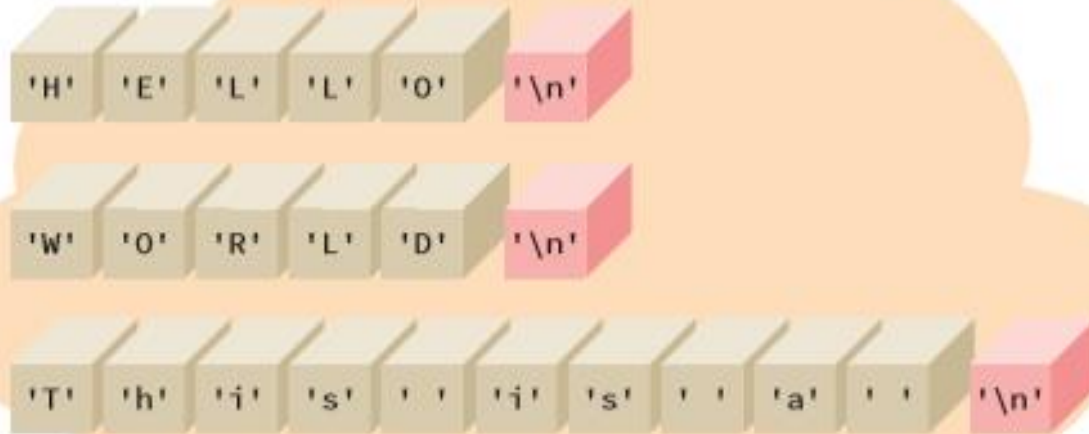
이진 파일

What is a file - file의 종류

- 텍스트 파일(text file)
 - 사람이 읽을 수 있는 ASCII code로 구성
 - 연속적인 line으로 구성, 각 line의 끝을 알리는 줄 바꿈 문자 “\n”로 종료
- 예) 메모장으로 작성한 파일, python source file 등



텍스트 파일: 문자로 구성된 파일



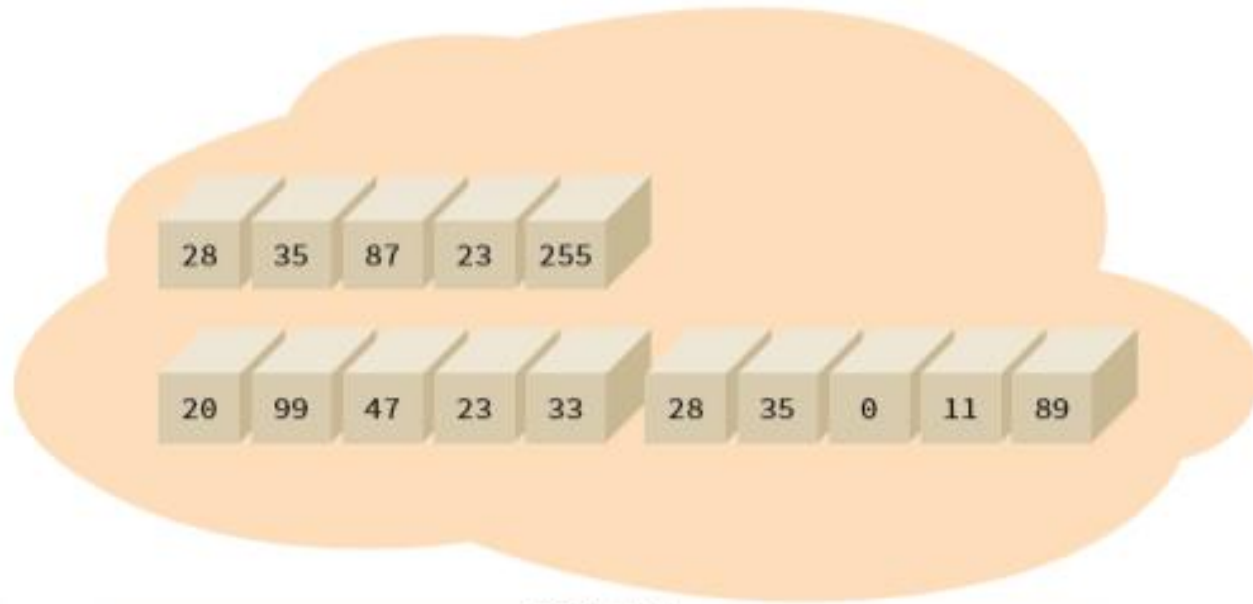
텍스트 파일

What is a file - file의 종류

- 이진 파일(binary file)
 - 사람이 읽을 수 없으나 컴퓨터는 읽을 수 있는 file
 - 문자 데이터가 아닌 이진 데이터가 직접 저장
 - 특정 프로그램에 의해서만 판독이 가능예) sound file, image file, execution file 등



이진파일: 데이터로 구성된 파일



이진 파일

What is a file - file의 특징

- File의 특징
 - File 이름
 - File 위치
 - File에 들어있는 data의 종류를 가리키는 유형, 사진, 음악, text
 - 크기
 - File의 크기 (byte 수)

What is a file - file의 특징

- File의 특징 : file 이름
 - 대부분의 OS에서는 file명의 일부가 해당 file에 어떤 유형의 data가 저장되어 있는 지를 나타냄
 - “.” notation 사용
 - “.” 다음에 오는 부분이 : 확장자 (extension)
 - 예 1) text file : my_letters.txt
mp3 file : my_song.mp3
실행(execution) file : chrome.exe
flower.png
 - 예2) Python file : file.py
학부.hwp
list.doc
test.cpp
hakbu.java
 - 확장자는 OS에 따라 다를 수 있음

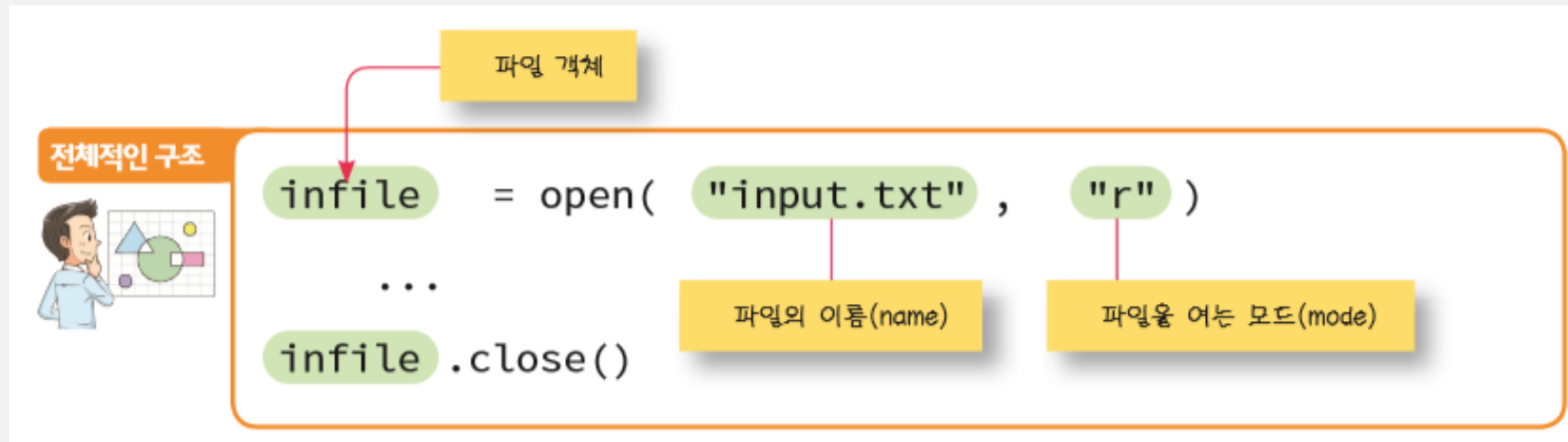
What is a file - file의 특징

- File의 특징 : file 위치
 - directory / folder
 - File을 group화하는 수단
 - folder 구조 / directory 구조
 - folder나 directory가 배열되고 연결되는 방식
 - 경로(path)
 - File이 어느 folder 구조에 있는지 설명하는 것
예) C:\Program Files (x86)\Internet Explorer\iexplorer.exe
 - File의 전체 경로명을 이용해 file을 가져오는 방법
image_file = "c:/images/jpg/flower.jpg"

What is a file - file의 특징

- File의 특징 : file 위치
 - 절대경로
 - 저장매체의 맨 위쪽 directory까지 전부 포함
 - C:\Program Files (x86)\Internet Explorer\iexplorer.exe
 - 상대경로
 - Directory 상의 현재 위치로부터 상대적인 위치 표현
 - ‘.’은 현재 directory
 - ‘..’은 부모 directory (한 단계 위의 directory)
 - 현재위치를 알려주는 명령어
 - 현재 작업 directory (cwd : current working directory)
 - » LINUX : pwd
 - » Windows : dir

Text 파일 읽고 쓰기



Opening a file

- open() 함수 호출
 - File을 열면 file객체가 생성
 - File객체는 프로그램 내에서 file에 접근하는 데 사용
 - File객체를 생성하고 나면 프로그램 내에서 **file명 대신에 file객체를 이용** 하여 처리
 - file객체 = open(file명, 열기 모드)
 - f = open("input.txt", "r")
 - f = open("output.txt", "w")
 - f = open("append.txt", "a")
- file 닫기
 - file객체.close()
 - f.close()

Opening a file - file mode

파일 모드	모드 이름	설명
"r"	읽기 모드(read mode)	파일의 처음부터 읽는다.
"w"	쓰기 모드(write mode)	파일의 처음부터 쓴다. 파일이 없으면 생성된다. 만약 파일이 존재하면 기존의 내용은 지워진다.
"a"	추가 모드(append mode)	파일의 끝에 쓴다. 파일이 없으면 생성된다.
"r+"	읽기와 쓰기 모드	파일에 읽고 쓸 수 있는 모드이다. 모드를 변경하려면 seek()가 호출되어야 한다.



"r"

파일을 처음부터 읽는다.



"w"

파일의 처음부터 쓴다.
만약 파일이 존재하면
기존의 내용이 지워진다.



"a"

파일의 끝에 쓴다.
파일이 없으면 생성된다.

Opening a file

- 열기모드
 - r: 읽기 모드 (기본)
 - w: 쓰기 모드
 - a: 추가 모드
 - b: 이진 모드 (binary)
 - 이진 file: 동영상, 음악, 그림 등 텍스트가 아닌 file
 - t: 텍스트 모드 (기본)
 - +: 갱신 모드

Opening a file

listing_22-1.py

```
File Edit Format Run Options Window Help
1 print("\n1) open file notes.txt")
2 print("\n2) 파일객체 my_file 생성")
3 my_file = open('notes.txt', 'r')
4
5 lines = my_file.readlines()
6
7 print ("\n3)lines = ", lines)
8
9 print ("\n4)close my_file")
10 my_file.close()
11
```

- 존재하지 않는 file (notes.txt)을 읽으려 하면 에러 발생

```
1) open file notes.txt
2) 파일객체 my_file 생성
Traceback (most recent call last):
  File "C:\프과소사\프과소사-강의예제\listing_22-1.py", line 3, in <module>
    my_file = open('notes.txt', 'r')
FileNotFoundError: [Errno 2] No such file or directory: 'notes.txt'
>>>
```

```

1 class File_1:
2     def __init__(self):
3         print("i-1) Constructor에서 notes.txt file을 open")
4         try:
5             self.my_file = open('notes.txt', 'r')
6         except:
7             print(">> Warningin 생성자) notes.txt file이 존재하지 않음 !")
8
9     def read_file(self):
10        # line 단위로 file에서 읽어옴
11        try:
12            print("r-1) read_file")
13            lines = self.my_file.readlines()
14            print("r-2) lines = ", lines)
15
16        except Exception as e:
17            print(">> Warning in read_file) ", e)
18
19    def __del__(self):
20        try:
21            print ("d-1) Destructor에서 my_file을 close")
22            self.my_file.close()
23
24        except Exception as e:
25            print(">> Warning in 소멸자) ", e)
26
27 class Main():
28     def main(self):
29         ex_file = File_1()
30         ex_file.read_file()
31
32 #main
33 exe_main = Main()
34 exe_main.main()
35

```

- 존재하지 않는 file (notes.txt) 에러 발생시 exception handler를 이용한 경우

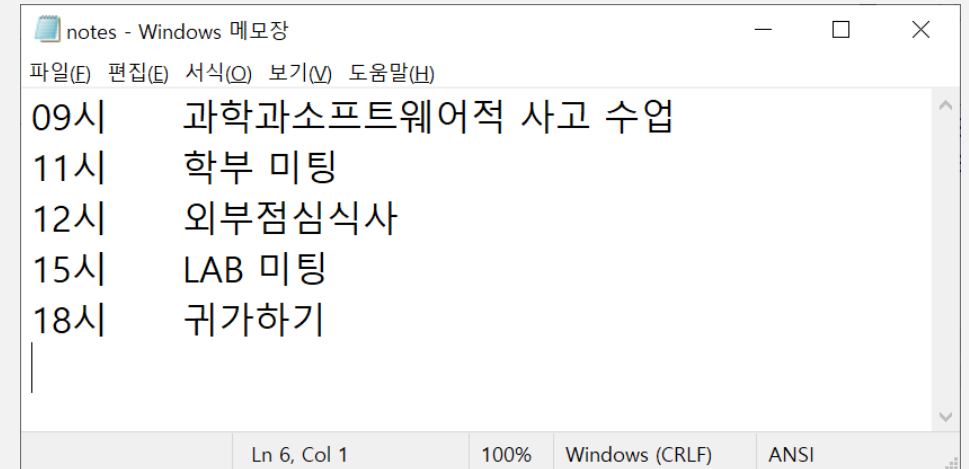
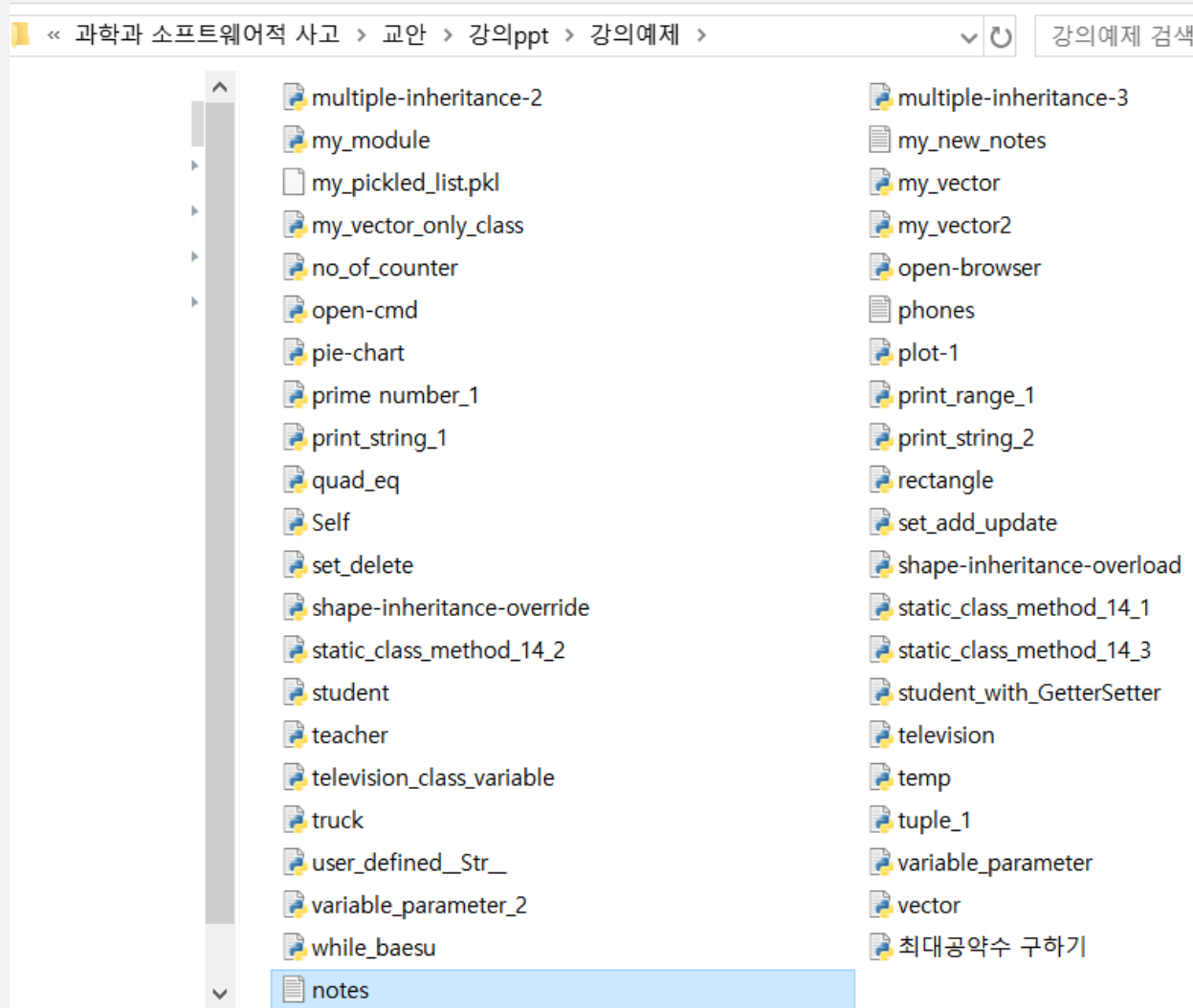
i-1) Constructor에서 notes.txt file을 open
>> Warningin 생성자) notes.txt file이 존재하지 않음 !

r-1) read_file
>> Warning in read_file) 'File_1' object has no attribute 'my_file'

d-1) Destructor에서 my_file을 close
>> Warning in 소멸자) 'File_1' object has no attribute 'my_file'
>>>

Opening a file

- notes.txt file을 동일 directory에 생성



Opening a file

listing_22-1.py

File Edit Format Run Options Window Help

```
1 print("\n1) open file notes.txt")
2 print("\n2) 파일객체 my_file 생성")
3 my_file = open('notes.txt', 'r')
4
5 lines = my_file.readlines()
6
7 print ("\n3)lines = ", lines)
8
9 print ("\n4)close my_file")
10 my_file.close()
11
```

```
1) open file notes.txt
2) 파일객체 my_file 생성
3)lines = ['09시    과학과소프트웨어적 사고 수업\n', '11시    학부 미팅\n',
'12시    외부점심식사\n', '15시    LAB 미팅\n', '18시    귀가하기\n']
4)close my_file
>>>
```

```

1 class File_1:
2     def __init__(self):
3         print("Wni-1) Constructor에서 notes.txt file을 open")
4         try:
5             self.my_file = open('notes.txt', 'r')
6         except:
7             print(">> Warningin 생성자) notes.txt file이 존재하지 않음 ! ")
8
9     def read_file(self):
10        # line 단위로 file에서 읽어옴
11        try:
12            print("Wnr-1) read_file")
13            lines = self.my_file.readlines()
14            print("r-2) lines = ", lines)
15
16        except Exception as e:
17            print(">> Warning in read_file) ", e)
18
19    def __del__(self):
20        try:
21            print ("Wnd-1) Destructor에서 my_file을 close")
22            self.my_file.close()
23
24        except Exception as e:
25            print(">> Warning in 소멸자) ", e)
26
27 class Main():
28     def main(self):
29         ex_file = File_1()
30         ex_file.read_file()
31
32 #main
33 exe_main = Main()
34 exe_main.main()
35

```

i-1) Constructor에서 notes.txt file을 open

r-1) read_file

r-2) lines = ['09시 과학과소프트웨어적 사고 수업Wn', '11시 학부 미팅Wn',
, '12시 외부점심식사Wn', '15시 LAB 미팅Wn', '18시 귀가하기Wn']

d-1) Destructor에서 my_file을 close

^^^

```

1 class File_1:
2     def __init__(self):
3         print("Wni-1) Constructor에서 notes.txt file을 open")
4         try:
5             self.my_file = open('notes.txt', 'r')
6         except:
7             print(">> Warningin 생성자) notes.txt file이 존재하지 않음 ! ")
8
9     def read_file(self):
10        # line 단위로 file에서 읽어옴
11        try:
12            print("Wnr-1) read_file")
13            first_line = self.my_file.readline()
14            second_line = self.my_file.readline()
15            third_line = self.my_file.readline()
16            fourth_line = self.my_file.readline()
17            fifth_line = self.my_file.readline()
18
19            print ("r-2) first line = ", first_line)
20            print ("r-3) second line = ", second_line)
21            print ("r-4) third line = ", third_line)
22            print ("r-5) fourth line = ", fourth_line)
23            print ("r-6) fifth line = ", fifth_line)
24
25        except Exception as e:
26            print(">> Warning in read_file) ", e)
27
28    def __del__(self):
29        try:
30            print ("Wnd-1) Destructor에서 my_file을 close")
31            self.my_file.close()
32
33        except Exception as e:
34            print(">> Warning in 소멸자) ", e)
35
36 class Main():
37     def main(self):
38         ex_file = File_1()
39         ex_file.read_file()
40
41 #main
42 exe_main = Main()
43 exe_main.main()
44 print("Wnmain) End.")

```

• notes.txt 내용을 한 줄씩 출력 - 1

i-1) Constructor에서 notes.txt file을 open

r-1) read_file

r-2) first line = 09시 과학과소프트웨어적 사고 수업

r-3) second line = 11시 학부 미팅

r-4) third line = 12시 외부점심식사

r-5) fourth line = 15시 LAB 미팅

r-6) fifth line = 18시 귀가하기

d-1) Destructor에서 my_file을 close
main) End.

```

1 class File_1:
2     def __init__(self):
3         print("Wni-1) Constructor에서 notes.txt file을 open")
4         try:
5             self.my_file = open('notes.txt', 'r')
6         except:
7             print(">> Warningin 생성자) notes.txt file이 존재하지 않음 ! ")
8
9     def read_file(self):
10        # line 단위로 file에서 읽어옴
11        try:
12            print("r-1) 한 줄씩 출력")
13            lines = self.my_file.readlines()
14            print("r-2) lines = ", lines)
15
16            i = 1
17            for each_line in lines:
18                #each_line.rstrip() # 각 줄에서 줄바꿈 문자를 제거
19                print ( ">> {} 번째 줄 내용 {}".format(i, each_line.rstrip()))
20                i = i + 1
21
22        except Exception as e:
23            print(">> Warning in read_file) ", e)
24
25    def __del__(self):
26        try:
27            print ("Wnd-1) Destructor에서 my_file을 close")
28            self.my_file.close()
29
30        except Exception as e:
31            print(">> Warning in 소멸자) ", e)
32
33    class Main():
34        def main(self):
35            ex_file = File_1()
36            ex_file.read_file()
37
38    #main
39    exe_main = Main()
40    exe_main.main()
41    print("Wnmain) End.")
42

```

- notes.txt 내용을
for문을 이용하여
한 줄씩 출력 - 2

i-1) Constructor에서 notes.txt file을 open

r-1) 한 줄씩 출력

r-2) lines = ['09시 과학과소프트웨어적 사고 수업Wn', '11시 학부 미팅Wn', '12시
외부점심식사Wn', '15시 LAB 미팅Wn', '18시 귀가하기Wn']

>> 1 번째 줄 내용 09시 과학과소프트웨어적 사고 수업

>> 2 번째 줄 내용 11시 학부 미팅

>> 3 번째 줄 내용 12시 외부점심식사

>> 4 번째 줄 내용 15시 LAB 미팅

>> 5 번째 줄 내용 18시 귀가하기

d-1) Destructor에서 my_file을 close

main) End.

\\

```

listing_class_22-3.py
File Edit Format Run Options Window Help
1 class File_1:
2     def __init__(self):
3         print("Wni-1) Constructor에서 notes.txt file을 r 모드로 open")
4         try:
5             self.my_file = open('notes.txt', 'r')
6         except:
7             print(">> Warningin 생성자) notes.txt file이 존재하지 않음 ! ")
8
9     def close_file(self):
10        try:
11            print ("Wna-1) my_file을 close")
12            self.my_file.close()
13
14        except Exception as e:
15            print(">> Warning in close_file() : ", e)
16
17    def open_mode(self, mode):
18        try:
19            print("o-1) mode : ", mode)
20
21            if mode == "r":
22                print ("Wno-1) my_file을 r mode로 open")
23                self.my_file = open('notes.txt', 'r')
24            elif mode == "a":
25                print ("Wn0-2) my_file을 a mode로 open")
26                self.my_file = open('notes.txt', 'a')
27
28        except Exception as e:
29            print(">> Warning in close_file() : ", e)
30
31    def read_file(self):
32        # line 단위로 file에서 읽어옴
33        try:
34            lines = self.my_file.readlines()
35            print("r-1) lines = ", lines)
36
37            i = 1
38            for each_line in lines:
39                #each_line.rstrip() # 각 줄에서 줄바꿈 문자를 제거
40                print ( ">> {} 번째 줄 내용 : {}".format(i, each_line.rstrip()))
41                i = i + 1
42
43        except Exception as e:
44            print(">> Warning in read_file) ", e)
45

```

```

46
47 def append(self):
48     print("Wna-1) notes.txt file을 append 모드로 open")
49     self.open_mode("a")
50
51     print("a-2) 새로운 일정 추가 : 19시 가족과 함께 저녁식사")
52     self.my_file.write('Wn19시    가족과 함께 저녁식사')
53
54     #print("a-3) close my_file")
55     #self.close_file()
56
57     print("a-4) read mode로 my_file open")
58     self.open_mode("r")
59     self.read_file()
60
61 def __del__(self):
62     try:
63         print ("Wnd-1) Destructor에서 my_file을 close")
64         self.my_file.close()
65
66     except Exception as e:
67         print(">> Warning in 소멸자) ", e)
68
69 class Main():
70     def main(self):
71         ex_file = File_1()
72         ex_file.read_file()
73         ex_file.append()
74
75 #main
76 exe_main = Main()
77 exe_main.main()
78 print("Wnmain) End.")
79

```

- notes.txt에 새로운 내용을 append

```

i-1) Constructor에서 notes.txt file을 r 모드로 open
r-1) lines = ['09시    과학과소프트웨어적 사고 수업Wn', '11시    학부 미팅W
n', '12시    외부점심식사Wn', '15시    LAB 미팅Wn', '18시    귀가하기']
>> 1 번째 줄 내용 : 09시    과학과소프트웨어적 사고 수업
>> 2 번째 줄 내용 : 11시    학부 미팅
>> 3 번째 줄 내용 : 12시    외부점심식사
>> 4 번째 줄 내용 : 15시    LAB 미팅
>> 5 번째 줄 내용 : 18시    귀가하기

```

```

a-1) notes.txt file을 append 모드로 open
o-1) mode : a

```

```

0-2) my_file을 a mode로 open
a-2) 새로운 일정 추가 : 19시 가족과 함께 저녁식사
a-4) read mode로 my_file open
o-1) mode : r

```

```

o-1) my_file을 r mode로 open
r-1) lines = ['09시    과학과소프트웨어적 사고 수업Wn', '11시    학부 미팅W
n', '12시    외부점심식사Wn', '15시    LAB 미팅Wn', '18시    귀가하기Wn', '
19시    가족과 함께 저녁식사']
>> 1 번째 줄 내용 : 09시    과학과소프트웨어적 사고 수업
>> 2 번째 줄 내용 : 11시    학부 미팅
>> 3 번째 줄 내용 : 12시    외부점심식사
>> 4 번째 줄 내용 : 15시    LAB 미팅
>> 5 번째 줄 내용 : 18시    귀가하기
>> 6 번째 줄 내용 : 19시    가족과 함께 저녁식사

```

```

d-1) Destructor에서 my_file을 close

```

```

main) End.

```

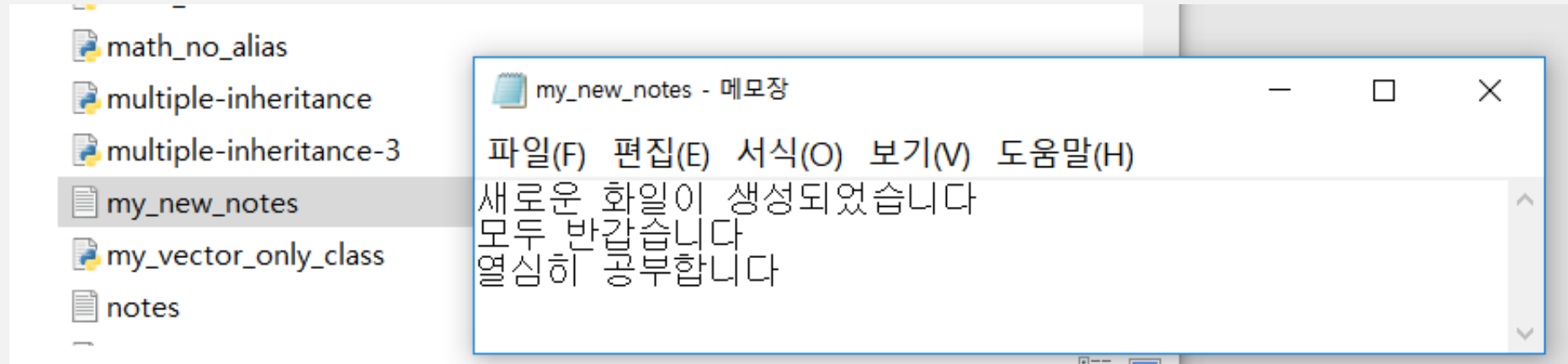


```
1 print("\n1) w mode로 my_new_notes.txt open 후 파일객체 new_file 생성")
2 new_file = open("my_new_notes.txt", 'w')
3
4 print("\n2) write() method로 새로운 내용을 쓰기")
5 new_file.write("새로운 화일이 생성되었습니다\n")
6 new_file.write("모두 반갑습니다\n")
7 new_file.write("열심히 공부합니다")
8
9 print ("\n3) close new_file")
10 new_file.close()
11
12 print ("\n4) read mode로 파일객체 my_file 생성")
13 my_file = open('my_new_notes.txt', 'r')
14
15 lines = my_file.readlines()
16 print ("\n5) lines = ", lines)
17
18 print ("\n6) close my_file")
19 my_file.close()
```

```
1) w mode로 my_new_notes.txt open 후 파일객체 new_file 생성
2) write() method로 새로운 내용을 쓰기
3) close new_file
4) read mode로 파일객체 my_file 생성
5) lines = ['새로운 화일이 생성되었습니다\n', '모두 반갑습니다\n', '열심히 공부합니다']
6) close my_file
```

- 새 file my_new_notes.txt에 대하여 w mode로 내용 쓰기

- 새 file my_new_notes.txt에 대하여 w mode로 내용 쓰기



```

1 print("\n1) w mode로 기존 my_new_notes.txt open 후 파일객체 the_file 생성")
2 the_file = open('my_new_notes.txt', 'w')
3
4 print("2) write() method로 새로운 내용을 쓰기")
5 the_file.write("Wake up\n")
6 the_file.write("Watch cartoons")
7
8 print ("3) close the_file")
9 the_file.close()
10
11 print ("4) read mode로 파일객체 my_file 생성")
12 my_file = open('my_new_notes.txt', 'r')
13
14 lines = my_file.readlines()
15 print ("5) lines = ", lines)
16 print ("6) 기존 화일 my_new_notes.txt의 내용이 새로이 변경되었음")
17
18 print ("7) close my_file")
19 my_file.close()
20

```

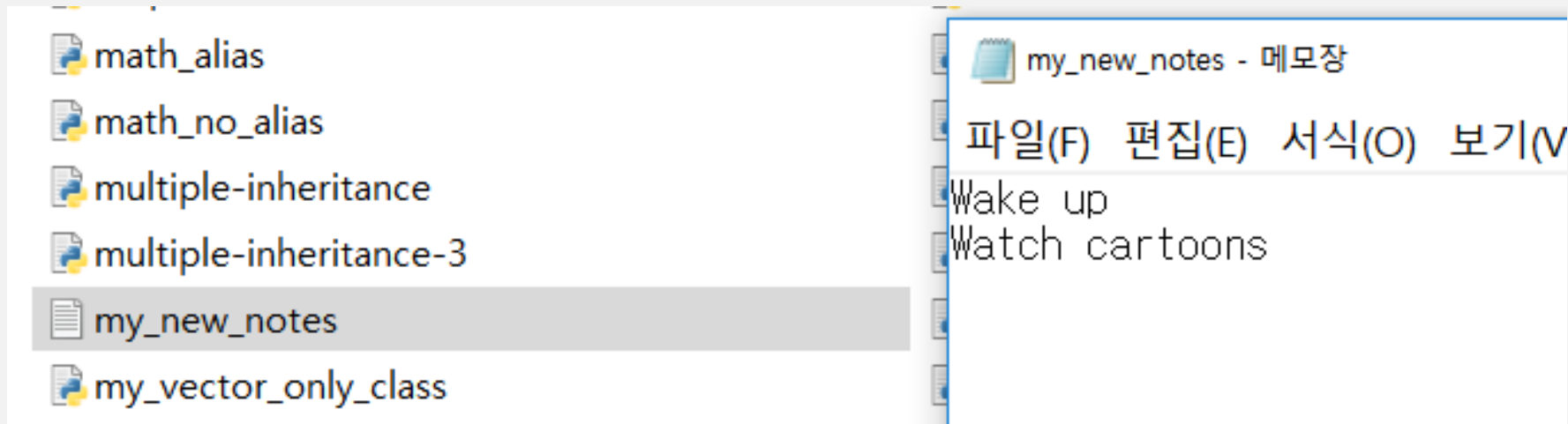
```

1) w mode로 기존 my_new_notes.txt open 후 파일객체 the_file 생성
2) write() method로 새로운 내용을 쓰기
3) close the_file
4) read mode로 파일객체 my_file 생성
5) lines = ['Wake up\n', 'Watch cartoons']
6) 기존 화일 my_new_notes.txt의 내용이 새로이 변경되었음
7) close my_file
>>>

```

- 기존 file 에
쓰기 모드 w
사용하기

- 만일 동일한 이름의 file이 존재하면 (예 : **my_new_notes.txt**)
“w” mode로 “쓰기”시 **기존에 있던 내용은 없어지고 새로운
내용으로 rewrite**



```

1 D = True
2 #D = False
3
4 class File_1:
5     def __init__(self, file_name):
6         if D:
7             print("\n1) file_name : {}".format(file_name))
8         # print("1_2) Constructor에서 file을 r 모드로 open")
9         try:
10             self.f_name = file_name
11         # self.my_file = open(self.f_name, 'r')
12         except:
13             print("\n>> Warningin 생성자 {} 이 존재하지 않음 ! ".format(file_name))
14
15     def open_mode(self, mode):
16
17         try:
18             if D:
19                 print("\no-1) open_mode() / file_name : {}, mode : {} ".format(self.f_name, mode))
20
21             if mode == "r":
22                 if D:
23                     print ("\no-2) {}을 r mode로 open".format(self.f_name))
24                     self.my_file = open(self.f_name, 'r')
25
26             elif mode == "a":
27                 if D:
28                     print ("\no-3) {}을 a mode로 open".format(self.f_name))
29                     self.my_file = open(self.f_name, 'a')
30
31             elif mode == "w":
32                 if D:
33                     print ("\no-4) {}을 w mode로 open".format(self.f_name))
34                     self.my_file = open(self.f_name, 'w')
35
36         except Exception as e:
37             print("\n>> Warning in open_mode() : ", e)
38

```

```

1) main
i_1) file_name : my_new_notes.txt
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake upWn', 'Watch cartoons']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons

c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp949'>을 close

a-1) append()
a-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : a
o-3) my_new_notes.txt을 a mode로 open

a-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>을 append 모드로 open
a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사

c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>을 close
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake upWn', 'Watch cartoonsWn', '19시 가족과 함께 저녁식사']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons
>> 3 번째 줄 내용 : 19시 가족과 함께 저녁식사
Ww-1) write_file()
w-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake upWn', 'Watch cartoonsWn', '19시 가족과 함께 저녁식사']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons
>> 3 번째 줄 내용 : 19시 가족과 함께 저녁식사
Ww-1) write_file()
w-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : w
o-4) my_new_notes.txt을 w mode로 open

w-3) w mode로 새로운 내용쓰기

c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='w' encoding='cp949'>을 close
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['새로운 화일이 생성되었습니다Wn', '모두 반갑습니다Wn', '열심히 공부합니다']
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다

d-1) Destructor에서 <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp949'>을 close

2) Main End.

```

새 file
my_new_notes.txt
에 대하여 w mode
로 내용 쓰기

```

39 def close_file(self):
40     try:
41         if D:
42             print("Wnc_1) close_file()")
43             print ("c-2) {}을 close".format(self.my_file))
44
45             self.my_file.close()
46
47     except Exception as e:
48         print("Wn>> Warning in close_file() : ", e)
49
50 def read_file(self):
51     # line 단위로 file에서 읽어옴
52     try:
53         if D:
54             print("r-1) read_file()")
55
56             self.open_mode("r")
57
58             lines = self.my_file.readlines()
59             if D:
60                 print("Wnr-2) lines = ", lines)
61
62             i = 1
63             for each_line in lines:
64                 #each_line.rstrip() # 각 줄에서 줄바꿈 문자를 제거
65                 print ( ">> {} 번째 줄 내용 : {}".format(i, each_line.rstrip()))
66                 i = i + 1
67
68     except Exception as e:
69         print("Wn>> Warning in read_file) ", e)
70

```

```

1) main
i_1) file_name : my_new_notes.txt
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r

o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake upWn', 'Watch cartoons']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons

c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp5
49'>을 close

a-1) append()
a-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : a

o-3) my_new_notes.txt을 a 모드로 open

a-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp
949'>을 append 모드로 open
a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사

c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp
949'>을 close
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r

o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake upWn', 'Watch cartoonsWn', '19시 가족과 함께 저녁
식사']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons
>> 3 번째 줄 내용 : 19시 가족과 함께 저녁식사
Ww-1) write_file()
w-2) file name : my_new_notes.txt

```

```

o-1) open_mode() / file_name : my_new_notes.txt, mode : r

o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake upWn', 'Watch cartoonsWn', '19시 가족과 함께 저녁
식사']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons
>> 3 번째 줄 내용 : 19시 가족과 함께 저녁식사
Ww-1) write_file()
w-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : w

o-4) my_new_notes.txt을 w 모드로 open

w-3) w 모드로 새로운 내용쓰기

c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='w' encoding='cp
949'>을 close
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r

o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['새로운 화일이 생성되었습니다Wn', '모두 반갑습니다Wn', '열심
히 공부합니다']
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다

d-1) Destructor에서 <_io.TextIOWrapper name='my_new_notes.txt' mode='r'
encoding='cp949'>을 close

2) Main End.

```

새 file
my_new_notes.txt
에 대하여 w mode
로 내용 쓰기

```

71 def append(self):
72     try:
73
74         if D:
75             print("\Wna-1) append()")
76             print("a-2) file name : {}".format(self.f_name))
77
78         self.open_mode("a")
79         if D:
80             print("\Wna-2) {}을 append 모드로 open".format(self.my_file))
81             print("a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사")
82
83         #         input_line = input(">> 새로운 일정을 추가하세요 : ")
84         #         self.my_file.write(input_line)
85         self.my_file.write('\Wn19시    가족과 함께 저녁식사')
86
87     except Exception as e:
88         print("\Wn>> Warning in append() ", e)
89
90
91 def write_file(self):
92     if D:
93         print("\Ww-1) write_file()")
94         print("w-2) file name : {}".format(self.f_name))
95
96         self.open_mode("w")
97
98         if D:
99             print("\Wnw-3) w 모드로 새로운 내용쓰기")
100
101         self.my_file.write("새로운 화일이 생성되었습니다\Wn")
102         self.my_file.write("모두 반갑습니다\Wn")
103         self.my_file.write("열심히 공부합니다")
104
105 def __del__(self):
106     try:
107         if D:
108             print("\Wnd-1) Destructor에서 {}을 close".format(self.my_file))
109             self.my_file.close()
110
111     except Exception as e:
112         print("\Wn>> Warning in 소멸자) ", e)
113

```

```

1) main
i-1) file_name : my_new_notes.txt
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r

o-2) my_new_notes.txt를 r mode로 open

r-2) lines = ['Wake up\Wn', 'Watch cartoons']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp949'>을 close

a-1) append()
a-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : a

o-3) my_new_notes.txt을 a 모드로 open

a-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>을 append 모드로 open
a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>을 close
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r

o-2) my_new_notes.txt를 r mode로 open

r-2) lines = ['Wake up\Wn', 'Watch cartoons\Wn', '19시    가족과 함께 저녁
식사']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons
>> 3 번째 줄 내용 : 19시    가족과 함께 저녁식사
Ww-1) write_file()
w-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : r

o-2) my_new_notes.txt를 r mode로 open

r-2) lines = ['Wake up\Wn', 'Watch cartoons\Wn', '19시    가족과 함께 저녁
식사']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons
>> 3 번째 줄 내용 : 19시    가족과 함께 저녁식사
Ww-1) write_file()
w-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : w

o-4) my_new_notes.txt을 w 모드로 open

w-3) w 모드로 새로운 내용쓰기

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='w' encoding='cp949'>을 close
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r

o-2) my_new_notes.txt를 r mode로 open

r-2) lines = ['새로운 화일이 생성되었습니다\Wn', '모두 반갑습니다\Wn', '열심
히 공부합니다']
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다

d-1) Destructor에서 <_io.TextIOWrapper name='my_new_notes.txt' mode='r'
encoding='cp949'>을 close

2) Main End.

```

새 file
my_new_notes.txt
에 대하여 w mode
로 내용 쓰기


```

114 class Main():
115     def main(self):
116
117 #         f_name = input(">> File 이름을 입력하세요 : ")
118         f_name = "my_new_notes.txt"
119
120         ex_file = File_1(f_name)
121         ex_file.read_file()
122         ex_file.close_file()
123
124         ex_file.append()
125         ex_file.close_file()
126
127         ex_file.read_file()
128
129         ex_file.write_file()
130         ex_file.close_file()
131
132         ex_file.read_file()
133
134 #main
135 if D:
136     print("\n1) main")
137
138 exe_main = Main()
139 exe_main.main()
140
141 if D:
142     print("\n2) Main End.")
143

```

```

1) main
i-1) file_name : my_new_notes.txt
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake upWn', 'Watch cartoons']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp949'>을 close

a-1) append()
a-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : a

o-3) my_new_notes.txt을 a 모드로 open

a-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>을 append 모드로 open
a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>을 close
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake upWn', 'Watch cartoonsWn', '19시 가족과 함께 저녁식사']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons
>> 3 번째 줄 내용 : 19시 가족과 함께 저녁식사
Ww-1) write_file()
w-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake upWn', 'Watch cartoonsWn', '19시 가족과 함께 저녁식사']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons
>> 3 번째 줄 내용 : 19시 가족과 함께 저녁식사
Ww-1) write_file()
w-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : w
o-4) my_new_notes.txt을 w 모드로 open

w-3) w 모드로 새로운 내용쓰기

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='w' encoding='cp949'>을 close
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['새로운 화일이 생성되었습니다Wn', '모두 반갑습니다Wn', '열심히 공부합니다']
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다

d-1) Destructor에서 <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp949'>을 close

2) Main End.

```

새 file
my_new_notes.txt
에 대하여 w mode
로 내용 쓰기


```

1) main
i_1) file_name : my_new_notes.txt
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake up\n', 'Watch cartoons']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons

c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp949'>을 close

a-1) append()
a-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : a
o-3) my_new_notes.txt을 a mode로 open

a-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>을 append mode로 open
a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사

c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>을 close
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake up\n', 'Watch cartoons\n', '19시 가족과 함께 저녁식사']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons
>> 3 번째 줄 내용 : 19시 가족과 함께 저녁식사
Ww-1) write_file()
w-2) file name : my_new_notes.txt

```

```

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['Wake up\n', 'Watch cartoons\n', '19시 가족과 함께 저녁식사']
>> 1 번째 줄 내용 : Wake up
>> 2 번째 줄 내용 : Watch cartoons
>> 3 번째 줄 내용 : 19시 가족과 함께 저녁식사
Ww-1) write_file()
w-2) file name : my_new_notes.txt

o-1) open_mode() / file_name : my_new_notes.txt, mode : w
o-4) my_new_notes.txt을 w mode로 open

w-3) w mode로 새로운 내용쓰기

c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='w' encoding='cp949'>을 close
r-1) read_file()

o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt을 r mode로 open

r-2) lines = ['새로운 화일이 생성되었습니다\n', '모두 반갑습니다\n', '열심히 공부합니다']
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다

d-1) Destructor에서 <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp949'>을 close

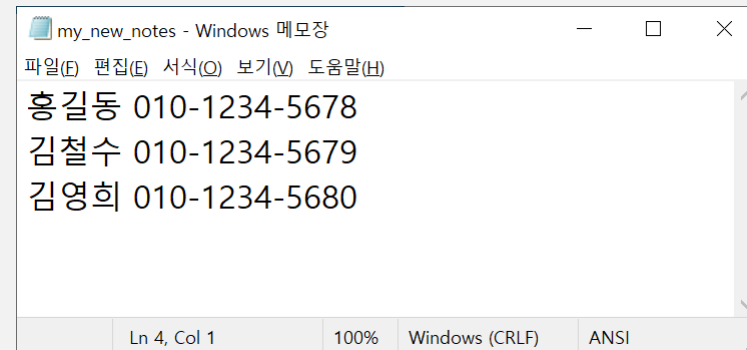
2) Main End.

```

- 만일 file이 존재하는 지를 검사하려면 다음과 같이 수행

```
filewrite1.py
File Edit Format Run Options Window Help
1 import os.path
2
3 #outfile = open("my_new_notes.txt", "w")
4
5 if os.path.isfile("my_new_notes.txt"):
6     outfile = open("my_new_notes.txt", "w")
7     print("\nWarning) 동일한 이름의 파일이 이미 존재합니다. ")
8 else :
9     print(">> my_new_notes.txt를 w 모드로 open합니다.")
10    outfile = open("my_new_notes.txt", "w")
11    outfile.write("홍길동 010-1234-5678\n")
12    outfile.write("김철수 010-1234-5679\n")
13    outfile.write("김영희 010-1234-5680\n")
14
15 outfile.close()
16
```

>> my_new_notes.txt를 w 모드로 open합니다.



Warning) 동일한 이름의 파일이 이미 존재합니다.

```

1 D = True
2 #D = False
3
4 import os.path
5
6 class File_1:
7
8     def __init__(self, file_name):
9
10         if D:
11             print("\n i-1) file_name : {}".format(file_name))
12             # print("i-2) Constructor에서 file을 r 모드로 open")
13         try:
14             self.f_name = file_name
15             # self.my_file = open(self.f_name, 'r')
16         except:
17             print("\n >> Warning in 생성자) {} 이 존재하지 않음 ! ".format(file_name))
18
19     def open_mode(self, mode):
20         try:
21             if D:
22                 print("\n o-1) open_mode() / file_name : {}, mode : {}".format(self.f_name, mode))
23
24             if mode == "r":
25                 if D:
26                     print ("o-2) {}을 r mode로 open".format(self.f_name))
27                     self.my_file = open(self.f_name, 'r')
28
29             elif mode == "a":
30                 if D:
31                     print ("o-3) {}을 a 모드로 open".format(self.f_name))
32                     self.my_file = open(self.f_name, 'a')
33
34             elif mode == "w":
35                 if D:
36                     print ("o-4) {}을 w 모드로 open".format(self.f_name))
37                     self.my_file = open(self.f_name, 'w')
38
39         except Exception as e:
40             print("\n >> Warning in open_mode() : ", e)
41

```

User-defined
exception handler
를 이용한 새 file
my_new_notes.txt
에 대하여 w mode
로 내용 쓰기

```

1) main
i-1) file_name : my_new_notes.txt

r-1) read_file()
>> 1 번째 줄 내용 : 홍길동 010-1234-5678
>> 2 번째 줄 내용 : 김철수 010-1234-5679
>> 3 번째 줄 내용 : 김영희 010-1234-5680

a-1) append()
a-2) file name : my_new_notes.txt
a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사

o-1) open_mode() / file_name : my_new_notes.txt, mode : a
o-3) my_new_notes.txt을 a 모드로 open

a-4) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp
949'>을 append 모드로 open

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp
949'>을 close

r-1) read_file()
>> 1 번째 줄 내용 : 홍길동 010-1234-5678
>> 2 번째 줄 내용 : 김철수 010-1234-5679
>> 3 번째 줄 내용 : 김영희 010-1234-5680
>> 4 번째 줄 내용 :
>> 5 번째 줄 내용 : 19시 가족과 함께 저녁식사

w-1) write_file()
w-2) file name : my_new_notes.txt
w-3) User-defined Excpetion Handler 호출

MyFileException) >>> Warning) 동일한 이름의 파일이 이미 존재합니다.
>>> Warning) 동일한 이름의 파일이 이미 존재합니다.

>>> 현존하는 화일에 새로운 내용으로 overwrite하시겠습니까 ? Y or N : Y

o-1) open_mode() / file_name : my_new_notes.txt, mode : w
o-4) my_new_notes.txt을 w 모드로 open

wc) w 모드로 새로운 내용쓰기

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='w' encoding='cp
949'>을 close

r-1) read_file()
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다

d-1) Destructor에서 <_io.TextIOWrapper name='my_new_notes.txt' mode='w'
' encoding='cp949'>을 close

2) main End.

```

```

42 def close_file(self):
43     try:
44         if D:
45             print("\n<1> close_file()")
46             print ("<2> {}을 close".format(self.my_file))
47
48             self.my_file.close()
49
50     except Exception as e:
51         print("\n>> Warning in close_file() : ", e)
52
53 def read_file(self):
54     # line 단위로 file에서 읽어옴
55     try:
56         if D:
57             print("\n<1> read_file()")
58
59             # Code [1]
60             '''
61             self.open_mode("r")
62             lines = self.my_file.readlines()
63
64             if D:
65                 print("\n<2> lines = ", lines)
66
67             i = 1
68             for each_line in lines:
69                 #each_line.rstrip() # 각 줄에서 줄바꿈 문자를 제거
70                 print ( ">> {} 번째 줄 내용 : {}".format(i, each_line.rstrip()))
71                 i = i + 1
72             '''
73
74             # Code [2]
75             # 위의 Code [1] 부분을 아래와 같이 대체
76             # self.f_name을 open하여 사용. 이후 바로 자동으로 file close
77             with open(self.f_name) as f:
78                 i = 1
79                 for each_line in f:
80                     # each_line.rstrip() # 각 줄에서 줄바꿈 문자를 제거
81                     print ( ">> {} 번째 줄 내용 : {}".format(i, each_line.rstrip()))
82                     i = i + 1
83
84     except Exception as e:
85         print("\n>> Warning in read_file) ", e)

```

User-defined exception handler 를 이용한 새 file my_new_notes.txt 에 대하여 w mode 로 내용 쓰기

```

1) main
i-1) file_name : my_new_notes.txt

r-1) read_file()
>> 1 번째 줄 내용 : 홍길동 010-1234-5678
>> 2 번째 줄 내용 : 김철수 010-1234-5679
>> 3 번째 줄 내용 : 김영희 010-1234-5680

a-1) append()
a-2) file name : my_new_notes.txt
a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사

o-1) open_mode() / file_name : my_new_notes.txt, mode : a
o-3) my_new_notes.txt을 a 모드로 open

a-4) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp
949'>을 append 모드로 open

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp
949'>을 close

r-1) read_file()
>> 1 번째 줄 내용 : 홍길동 010-1234-5678
>> 2 번째 줄 내용 : 김철수 010-1234-5679
>> 3 번째 줄 내용 : 김영희 010-1234-5680
>> 4 번째 줄 내용 :
>> 5 번째 줄 내용 : 19시 가족과 함께 저녁식사

w-1) write_file()
w-2) file name : my_new_notes.txt
w-3) User-defined Exception Handler 호출

MyFileException) >>> Warning) 동일한 이름의 파일이 이미 존재합니다.
>>> Warning) 동일한 이름의 파일이 이미 존재합니다.

>>> 현존하는 화일에 새로운 내용으로 overwrite하시겠습니까? Y or N : Y

o-1) open_mode() / file_name : my_new_notes.txt, mode : w
o-4) my_new_notes.txt을 w 모드로 open

wc) w 모드로 새로운 내용쓰기

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='w' encoding='cp
949'>을 close

r-1) read_file()
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다

d-1) Destructor에서 <_io.TextIOWrapper name='my_new_notes.txt' mode='w'
' encoding='cp949'>을 close

2) main End.

```

```

87 def append(self):
88     try:
89         if D:
90             print("\na-1) append()")
91             print("a-2) file name : {}".format(self.f_name))
92             print("a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사")
93
94         input_line = input(">> 새로운 일정을 추가하세요 : ")
95         self.my_file.write(input_line)
96         self.open_mode("a")
97
98         if D:
99             print("\na-4) {}을 append 모드로 open".format(self.my_file))
100
101         self.my_file.write('\n19시 가족과 함께 저녁식사')
102         self.close_file()
103
104     except Exception as e:
105         print("\n>> Warning in append() ", e)
106
107 def write_file(self):
108     if D:
109         print("\nw-1) write_file()")
110         print("w-2) file name : {}".format(self.f_name))
111     try:
112         if os.path.isfile(self.f_name):
113             if D:
114                 print("w-3) User-defined Excpetion Handler 호출")
115
116                 raise MyFileException(">>> Warning) 동일한 이름의 파일이 이미 존재합니다. ")
117
118         else:
119             self.open_mode("w")
120             self.write_contents()
121
122     except MyFileException as e:
123         print(e)
124         answer = input("\n>>> 현존하는 화일에 새로운 내용으로 overwriteW
125 하시겠습니까 ? Y or N : ")
126
127         if answer == "Y":
128             self.open_mode("w")
129             self.write_contents()
130
131         else:
132             pass
133
134
135

```

User-defined exception handler 를 이용한 새 file my_new_notes.txt 에 대하여 w mode 로 내용 쓰기

```

1) main
i-1) file_name : my_new_notes.txt

r-1) read_file()
>> 1 번째 줄 내용 : 홍길동 010-1234-5678
>> 2 번째 줄 내용 : 김철수 010-1234-5679
>> 3 번째 줄 내용 : 김영희 010-1234-5680

a-1) append()
a-2) file name : my_new_notes.txt
a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사

o-1) open_mode() / file_name : my_new_notes.txt, mode : a
o-3) my_new_notes.txt을 a 모드로 open

a-4) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp
949'>을 append 모드로 open

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp
949'>을 close

r-1) read_file()
>> 1 번째 줄 내용 : 홍길동 010-1234-5678
>> 2 번째 줄 내용 : 김철수 010-1234-5679
>> 3 번째 줄 내용 : 김영희 010-1234-5680
>> 4 번째 줄 내용 :
>> 5 번째 줄 내용 : 19시 가족과 함께 저녁식사

w-1) write_file()
w-2) file name : my_new_notes.txt
w-3) User-defined Excpetion Handler 호출

MyFileException) >>> Warning) 동일한 이름의 파일이 이미 존재합니다.
>>> Warning) 동일한 이름의 파일이 이미 존재합니다.

>>> 현존하는 화일에 새로운 내용으로 overwrite하시겠습니까 ? Y or N : Y

o-1) open_mode() / file_name : my_new_notes.txt, mode : w
o-4) my_new_notes.txt을 w 모드로 open

wc) w 모드로 새로운 내용쓰기

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='w' encoding='cp
949'>을 close

r-1) read_file()
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다

d-1) Destructor에서 <_io.TextIOWrapper name='my_new_notes.txt' mode='w'
' encoding='cp949'>을 close

2) main End.

```



```

136 def write_contents(self):
137     if D:
138         print("\nwc) w 모드로 새로운 내용쓰기")
139
140     self.my_file.write("새로운 화일이 생성되었습니다\n")
141     self.my_file.write("모두 반갑습니다\n")
142     self.my_file.write("열심히 공부합니다")
143
144     self.close_file()
145
146 def __del__(self):
147     try:
148         if D:
149             print ("\nnd-1) Destructor에서 {}을 close".format(self.my_file))
150             self.my_file.close()
151
152     except Exception as e:
153         print("\n>> Warning in 소멸자) ", e)
154
155 class MyFileException(Exception):
156
157     def __init__(self, msg):
158
159         self.msg = msg
160         if D:
161             print("\nMyFileException) ", self.msg)
162
163
164 class Main():
165     def main(self):
166
167         # f_name = input(">> File 이름을 입력하세요 : ")
168         f_name = "my_new_notes.txt"
169
170         ex_file = File_1(f_name)
171         ex_file.read_file() # Code [2] 에서 이미 file close
172
173         ex_file.append() # append() 에서 이미 file close
174         ex_file.read_file()
175
176         ex_file.write_file() # write_file() 에서 이미 file close
177         ex_file.read_file()
178

```

User-defined exception handler 를 이용한 새 file my_new_notes.txt 에 대하여 w mode 로 내용 쓰기

```

1) main
i-1) file_name : my_new_notes.txt

r-1) read_file()
>> 1 번째 줄 내용 : 홍길동 010-1234-5678
>> 2 번째 줄 내용 : 김철수 010-1234-5679
>> 3 번째 줄 내용 : 김영희 010-1234-5680

a-1) append()
a-2) file name : my_new_notes.txt
a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사

o-1) open_mode() / file_name : my_new_notes.txt, mode : a
o-3) my_new_notes.txt을 a 모드로 open

a-4) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp
949'>을 append 모드로 open

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp
949'>을 close

r-1) read_file()
>> 1 번째 줄 내용 : 홍길동 010-1234-5678
>> 2 번째 줄 내용 : 김철수 010-1234-5679
>> 3 번째 줄 내용 : 김영희 010-1234-5680
>> 4 번째 줄 내용 :
>> 5 번째 줄 내용 : 19시 가족과 함께 저녁식사

w-1) write_file()
w-2) file name : my_new_notes.txt
w-3) User-defined Excpetion Handler 호출

MyFileException) >>> Warning) 동일한 이름의 파일이 이미 존재합니다.
>>> Warning) 동일한 이름의 파일이 이미 존재합니다.

>>> 현존하는 화일에 새로운 내용으로 overwrite하시겠습니까? Y or N : Y

o-1) open_mode() / file_name : my_new_notes.txt, mode : w
o-4) my_new_notes.txt을 w 모드로 open

wc) w 모드로 새로운 내용쓰기

c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='w' encoding='cp
949'>을 close

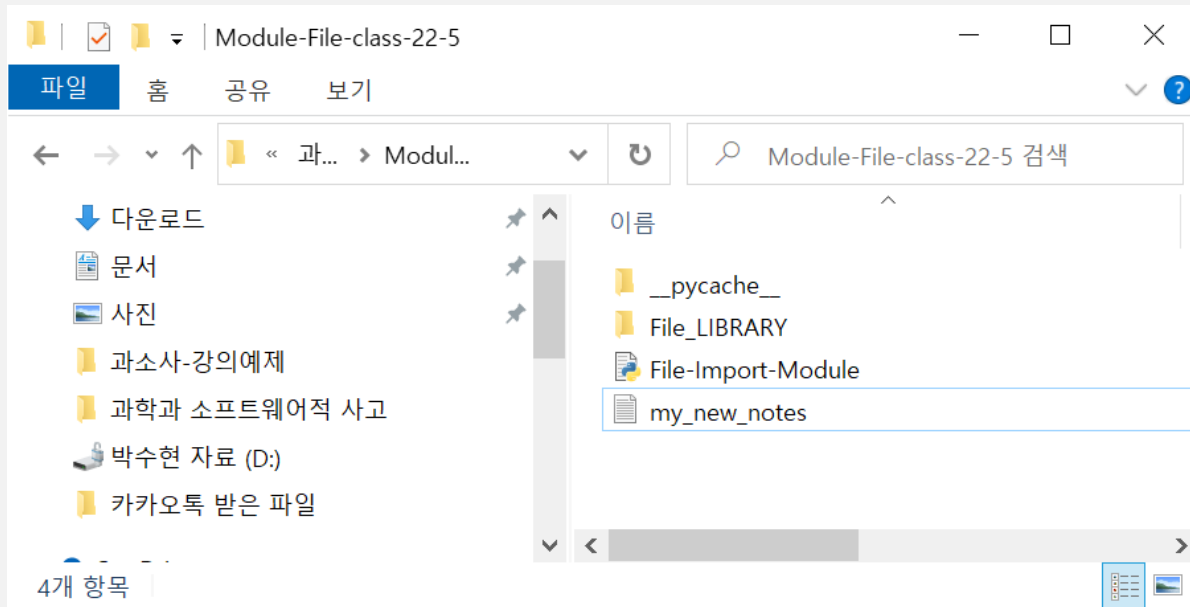
r-1) read_file()
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다

d-1) Destructor에서 <_io.TextIOWrapper name='my_new_notes.txt' mode='w'
encoding='cp949'>을 close

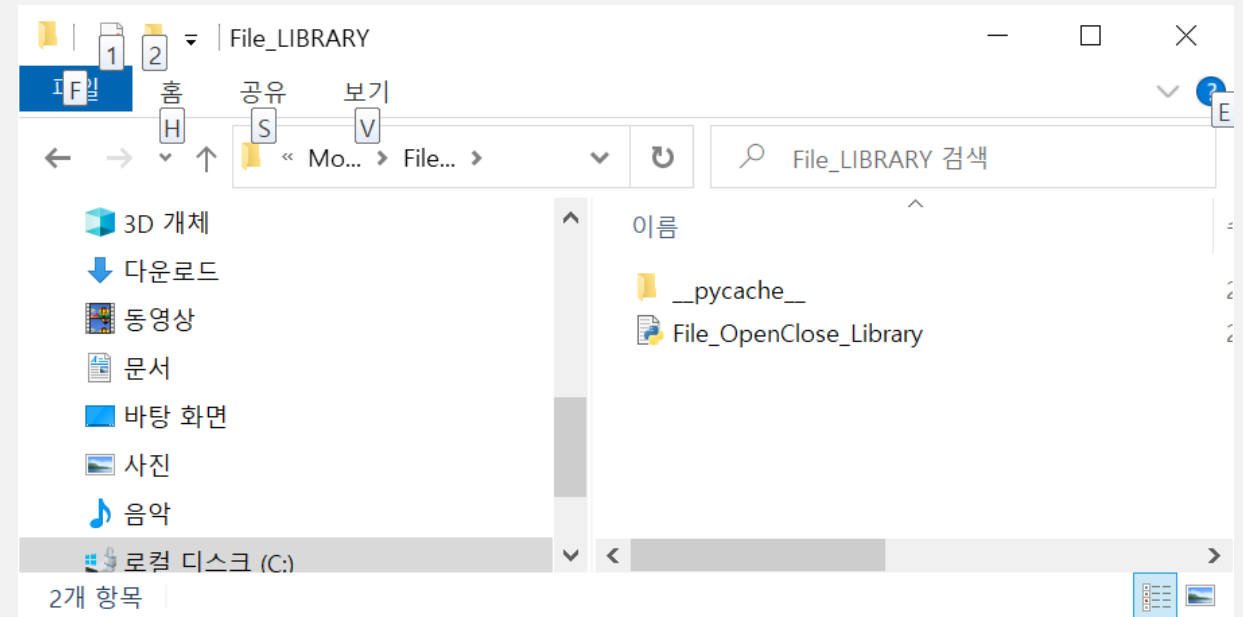
2) main End.

```

Module 사용



C:/과소사/과소사-강의예제/Module-File-class-22-5



C:/과소사/과소사-강의예제/Module-File-class-22-5/File_LIBRARY

```
1 D = True
2 #D = False
3
4 import sys
5 sys.path.append("C:/과소사/과소사-강의예제/Module-File-class-22-5/File_LIBRARY")
6
7 import File_OpenClose_Library as F_Lib
8
9
10 class Main():
11     def main(self):
12
13         # f_name = input(">> File 이름을 입력하세요 : ")
14         f_name = "my_new_notes.txt"
15
16         ex_file = F_Lib.File_1(f_name)
17         ex_file.read_file() # Code [2] 에서 이미 file close
18
19         ex_file.append() # append() 에서 이미 file close
20         ex_file.read_file()
21
22         ex_file.write_file() # write_file() 에서 이미 file close
23         ex_file.read_file()
24
25 #main
26
27 if __name__ == "__main__":
28
29     if D:
30         print("\n1) main")
31
32     exe_main = Main()
33     exe_main.main()
34
35     if D:
36         print("\n2) main End.")
37
38
```

Module 사용


```

1 D = True
2 #D = False
3
4 import os.path
5
6 class File_1:
7
8     def __init__(self, file_name):
9
10         if D:
11             print("Wni-1) file_name : {}".format(file_name))
12         # print("i-2) Constructor에서 file을 r 모드로 open")
13         try:
14             self.f_name = file_name
15             # self.my_file = open(self.f_name, 'r')
16         except:
17             print("Wn>> Warningin 생성자) {} 이 존재하지 않음 !".format(file_name))
18
19     def open_mode(self, mode):
20         try:
21             if D:
22                 print("Wno-1) open_mode() / file_name : {}, mode : {}".format(self.f_name, mode))
23
24             if mode == "r":
25                 if D:
26                     print ("o-2) {}을 r mode로 open".format(self.f_name))
27                     self.my_file = open(self.f_name, 'r')
28
29             elif mode == "a":
30                 if D:
31                     print ("o-3) {}을 a 모드로 open".format(self.f_name))
32                     self.my_file = open(self.f_name, 'a')
33
34             elif mode == "w":
35                 if D:
36                     print ("o-4) {}을 w 모드로 open".format(self.f_name))
37                     self.my_file = open(self.f_name, 'w')
38
39         except Exception as e:
40             print("Wn>> Warning in open_mode() : ", e)
41

```

Module 사용

```

42 def close_file(self):
43     try:
44         if D:
45             print("Wnc-1) close_file()")
46             print("c-2) {}을 close".format(self.my_file))
47
48             self.my_file.close()
49
50     except Exception as e:
51         print("Wn>> Warning in close_file() : ", e)
52
53 def read_file(self):
54     # line 단위로 file에서 읽어옴
55     try:
56         if D:
57             print("Wnr-1) read_file()")
58
59             # Code [1]
60             '''
61             self.open_mode("r")
62             lines = self.my_file.readlines()
63
64             if D:
65                 print("Wnr-2) lines = ", lines)
66
67             i = 1
68             for each_line in lines:
69                 #each_line.rstrip() # 각 줄에서 줄바꿈 문자를 제거
70                 print ( ">> {} 번째 줄 내용 : {}".format(i, each_line.rstrip()))
71                 i = i + 1
72             '''
73
74             # Code [2]
75             # 위의 Code [1] 부분을 아래와 같이 대체
76             # self.f_name을 open하여 사용. 이후 바로 자동으로 file close
77             with open(self.f_name) as f:
78                 i = 1
79                 for each_line in f:
80                     # each_line.rstrip() # 각 줄에서 줄바꿈 문자를 제거
81                     print ( ">> {} 번째 줄 내용 : {}".format(i, each_line.rstrip()))
82                     i = i + 1
83
84     except Exception as e:
85         print("Wn>> Warning in read_file) ", e)
86

```

```

87 def append(self):
88     try:
89         if D:
90             print("Wna-1) append()")
91             print("a-2) file name : {}".format(self.f_name))
92             print("a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사")
93
94             # input_line = input(">> 새로운 일정을 추가하세요 : ")
95             # self.my_file.write(input_line)
96             self.open_mode("a")
97
98             if D:
99                 print("Wna-4) {}을 append 모드로 open".format(self.my_file))
100
101             self.my_file.write('Wn19시 가족과 함께 저녁식사')
102             self.close_file()
103
104     except Exception as e:
105         print("Wn>> Warning in append() ", e)
106
107
108 def write_file(self):
109     if D:
110         print("Wnw-1) write_file()")
111         print("w-2) file name : {}".format(self.f_name))
112     try:
113         if os.path.isfile(self.f_name):
114             if D:
115                 print("w-3) User-defined Excpetion Handler 호출")
116
117                 raise MyFileException(">>> Warning) 동일한 이름의 파일이 이미 존재합니다. ")
118
119             else:
120                 self.open_mode("w")
121                 self.write_contents()
122
123     except MyFileException as e:
124         print(e)
125         answer = input("Wn>>> 현존하는 화일에 새로운 내용으로 overwriteW
126 하시겠습니까 ? Y or N : ")
127
128
129         if answer == "Y":
130             self.open_mode("w")
131             self.write_contents()
132
133         else:
134             pass
135

```

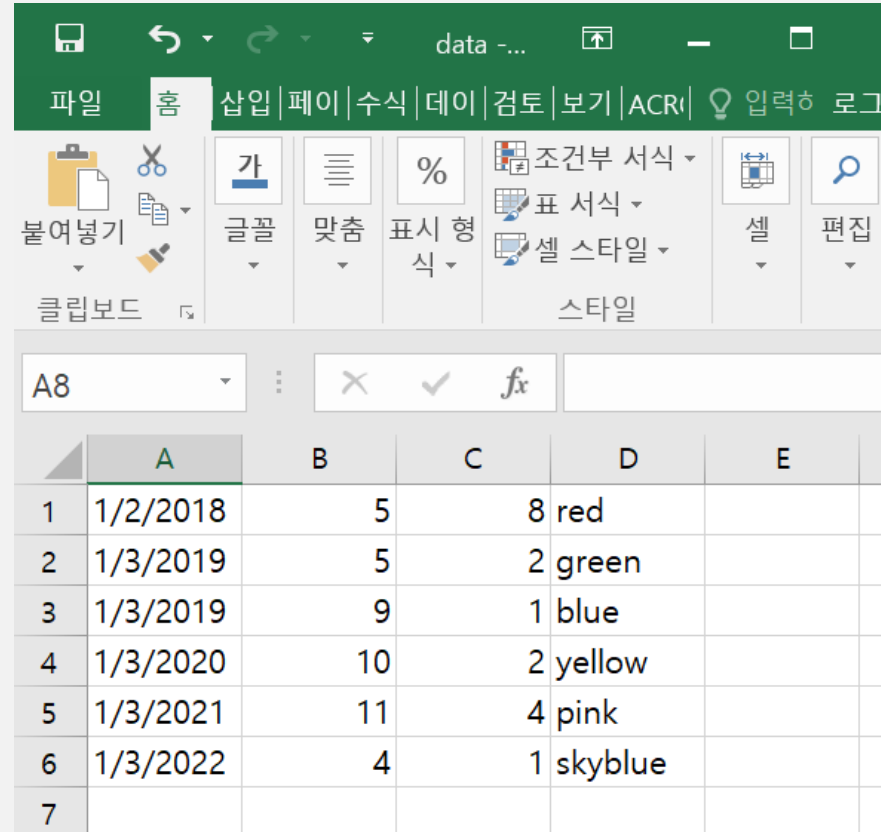
Module 사용

File Edit Format Run Options Window Help

```
136 def write_contents(self):
137     if D:
138         print("\nwc) w 모드로 새로운 내용쓰기")
139
140     self.my_file.write("새로운 화일이 생성되었습니다\n")
141     self.my_file.write("모두 반갑습니다\n")
142     self.my_file.write("열심히 공부합니다")
143
144     self.close_file()
145
146 def __del__(self):
147     try:
148         if D:
149             print ("\nd-1) Destructor에서 {}을 close".format(self.my_file))
150             self.my_file.close()
151
152     except Exception as e:
153         print("\n>> Warning in 소멸자 ", e)
154
155 class MyFileException(Exception):
156
157     def __init__(self, msg):
158
159         self.msg = msg
160         if D:
161             print("\nMyFileException) ", self.msg)
162
```

CSV(Comma Separated Value) 처리

- CSV
 - Spread sheet (excel 등) 나 database 에서 가장 널리 사용되는 입출력 형식
 - Python에서는 csv라는 모듈을 제공 (아래 그림 : data.csv)



The screenshot shows the Microsoft Excel interface with a file named 'data.csv' open. The ribbon is set to '홈' (Home). The data is organized in a table with columns A through E. The data rows are numbered 1 through 6, with row 7 being empty. The data in the table is as follows:

	A	B	C	D	E
1	1/2/2018	5	8	red	
2	1/3/2019	5	2	green	
3	1/3/2019	9	1	blue	
4	1/3/2020	10	2	yellow	
5	1/3/2021	11	4	pink	
6	1/3/2022	4	1	skyblue	
7					

CSV(Comma Separated Value) 처리

css.py

File Edit Format Run Options Window Help

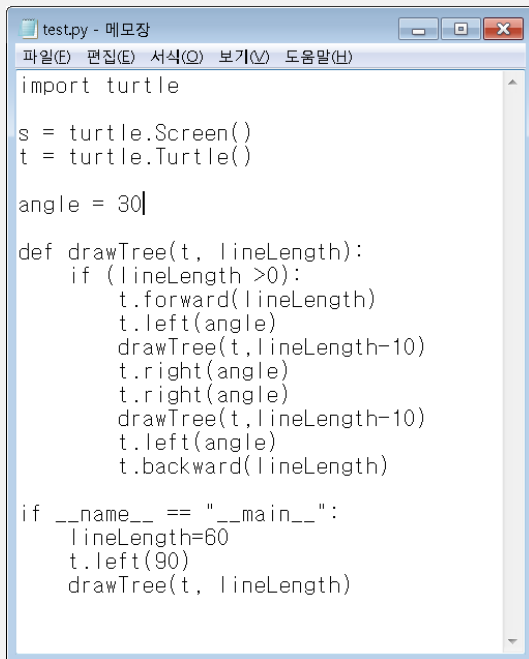
```
1 # 파일을 연다.  
2 f = open("data.csv", "r")  
3  
4 # 파일 안의 각 줄을 처리한다.  
5 for line in f.readlines():  
6     line = line.strip()      # 공백 문자를 없앴  
7     print(line)             # line을 출력  
8     parts = line.split(",") # 줄을 쉼표로 분리  
9  
10    for part in parts: # 각 줄의 필드를 출력한다.  
11        print(" ", part)  
12
```

	A	B	C	D	E
1	1/2/2018	5	8	red	
2	1/3/2019	5	2	green	
3	1/3/2019	9	1	blue	
4	1/3/2020	10	2	yellow	
5	1/3/2021	11	4	pink	
6	1/3/2022	4	1	skyblue	
7					

```
1/2/2018,5,8,red  
1/2/2018  
5  
8  
red  
1/3/2019,5,2,green  
1/3/2019  
5  
2  
green  
1/3/2019,9,1,blue  
1/3/2019  
9  
1  
blue  
1/3/2020,10,2,yellow  
1/3/2020  
10  
2  
yellow  
1/3/2021,11,4,pink  
1/3/2021  
11  
4  
pink  
1/3/2022,4,1,skyblue  
1/3/2022  
4  
1  
skyblue
```

Opening a file

- 존재하지 않는 file을 쓰는 것은 허용됨
 - 동일한 이름의 file이 존재했다면, 기존 file의 내용이 모두 삭제되므로 주의 필요



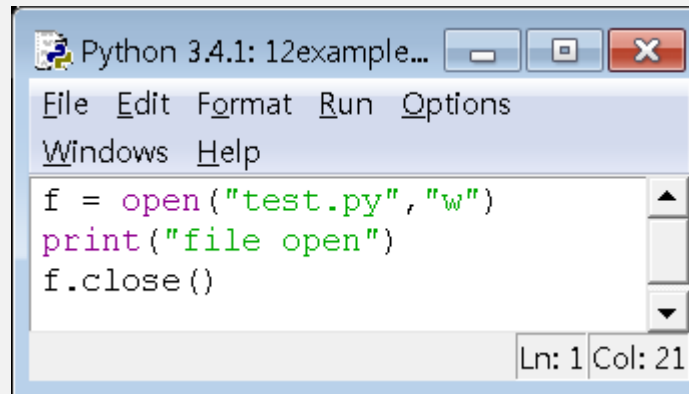
```
import turtle

s = turtle.Screen()
t = turtle.Turtle()

angle = 30

def drawTree(t, lineLength):
    if (lineLength > 0):
        t.forward(lineLength)
        t.left(angle)
        drawTree(t, lineLength-10)
        t.right(angle)
        t.right(angle)
        drawTree(t, lineLength-10)
        t.left(angle)
        t.backward(lineLength)

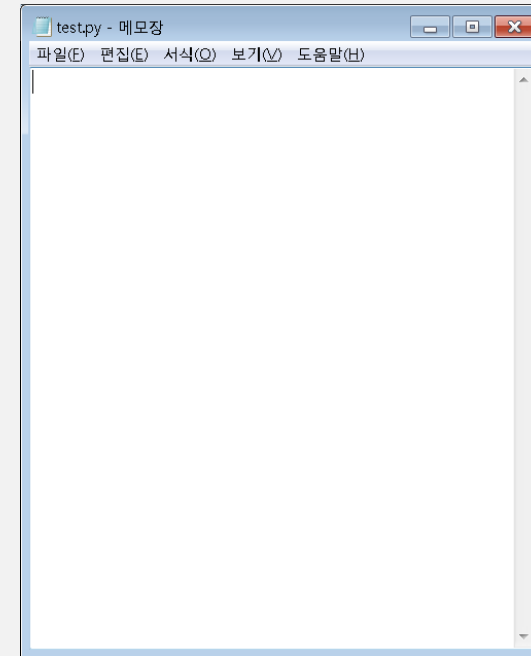
if __name__ == "__main__":
    lineLength=60
    t.left(90)
    drawTree(t, lineLength)
```



```
File Edit Format Run Options
Windows Help

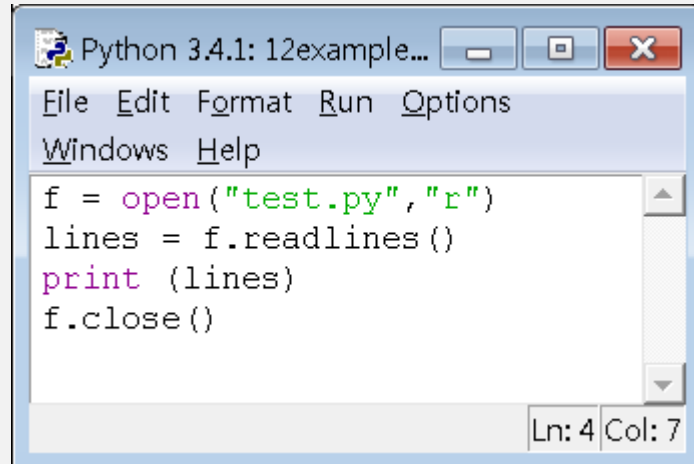
f = open("test.py", "w")
print("file open")
f.close()

Ln: 1 Col: 21
```

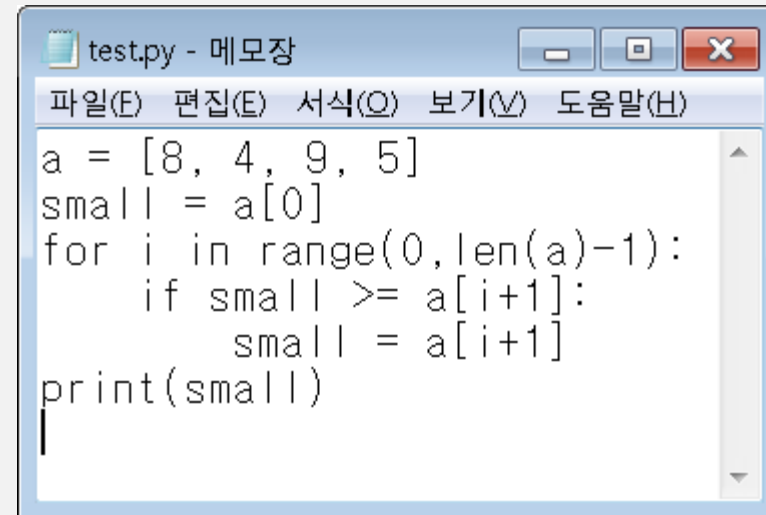


Reading a file

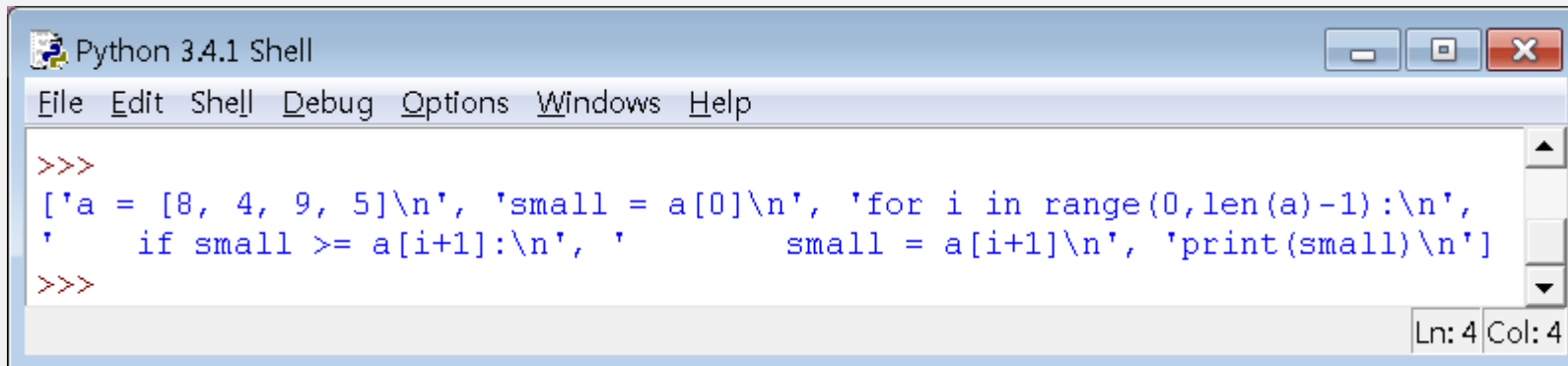
- `file객체.readlines()`
 - file 전체를 읽어서 줄 단위로 구성된 list 반환



```
Python 3.4.1: 12example...
File Edit Format Run Options
Windows Help
f = open("test.py", "r")
lines = f.readlines()
print (lines)
f.close()
Ln: 4 Col: 7
```



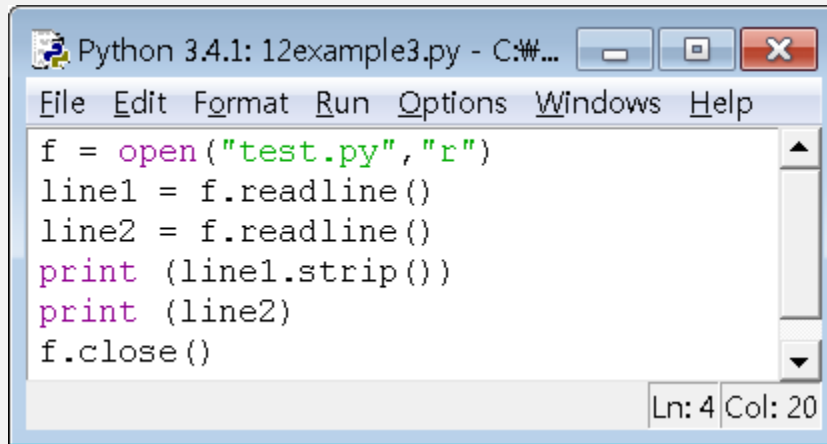
```
test.py - 메모장
파일(E) 편집(E) 서식(O) 보기(V) 도움말(H)
a = [8, 4, 9, 5]
small = a[0]
for i in range(0, len(a)-1):
    if small >= a[i+1]:
        small = a[i+1]
print(small)
```



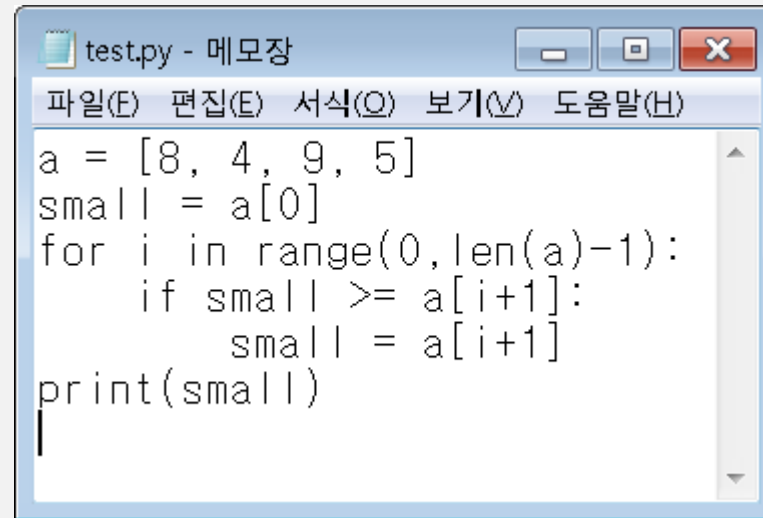
```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>>
['a = [8, 4, 9, 5]\n', 'small = a[0]\n', 'for i in range(0, len(a)-1):\n',
  '    if small >= a[i+1]:\n', '        small = a[i+1]\n', 'print(small)\n']
>>>
Ln: 4 Col: 4
```

Reading a file

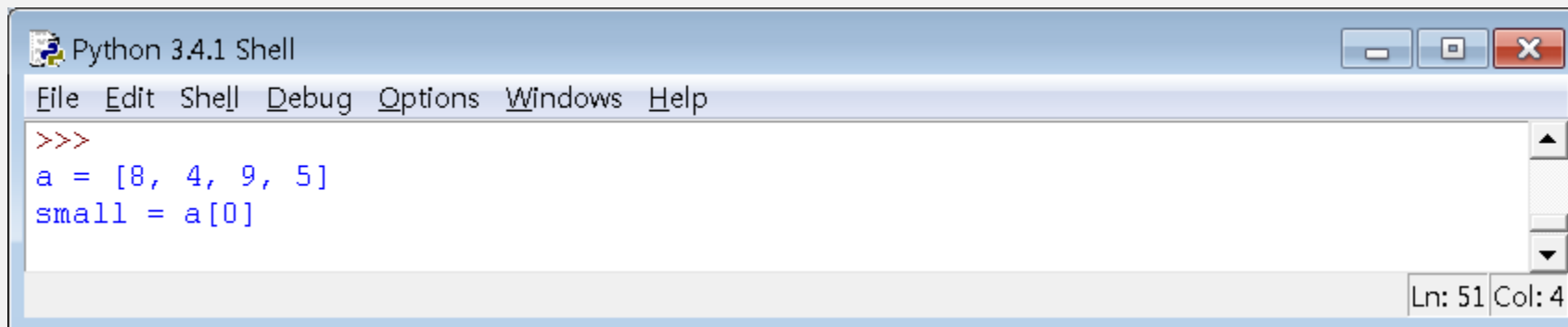
- file객체.readline()
 - file의 한 줄을 읽어서 문자열로 반환
 - 리스트.strip()은 문자열 처음과 끝의 공백문자 제거



```
Python 3.4.1: 12example3.py - C:\#\nFile  Edit  Format  Run  Options  Windows  Help\nf = open("test.py", "r")\nline1 = f.readline()\nline2 = f.readline()\nprint (line1.strip())\nprint (line2)\nf.close()\nLn: 4 Col: 20
```



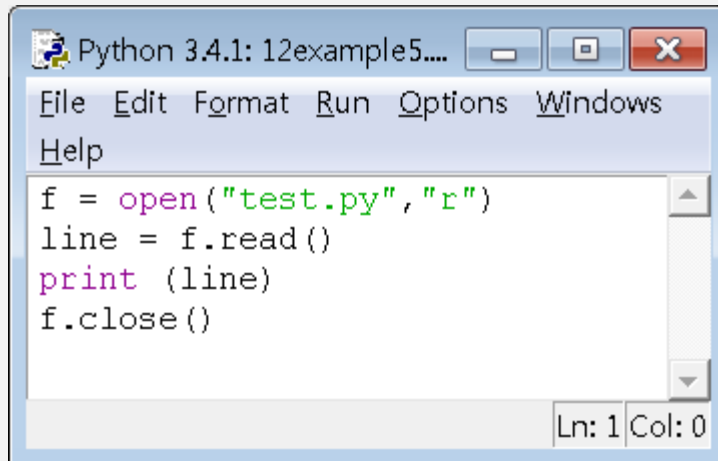
```
test.py - 메모장\n파일(E)  편집(E)  서식(O)  보기(V)  도움말(H)\na = [8, 4, 9, 5]\nsmall = a[0]\nfor i in range(0, len(a)-1):\n    if small >= a[i+1]:\n        small = a[i+1]\nprint(small)
```



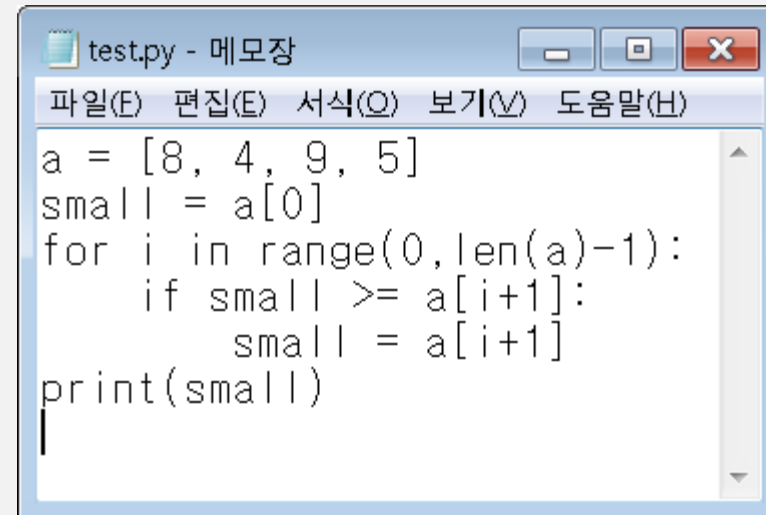
```
Python 3.4.1 Shell\nFile  Edit  Shell  Debug  Options  Windows  Help\n>>>\na = [8, 4, 9, 5]\nsmall = a[0]\nLn: 51 Col: 4
```


Reading a file

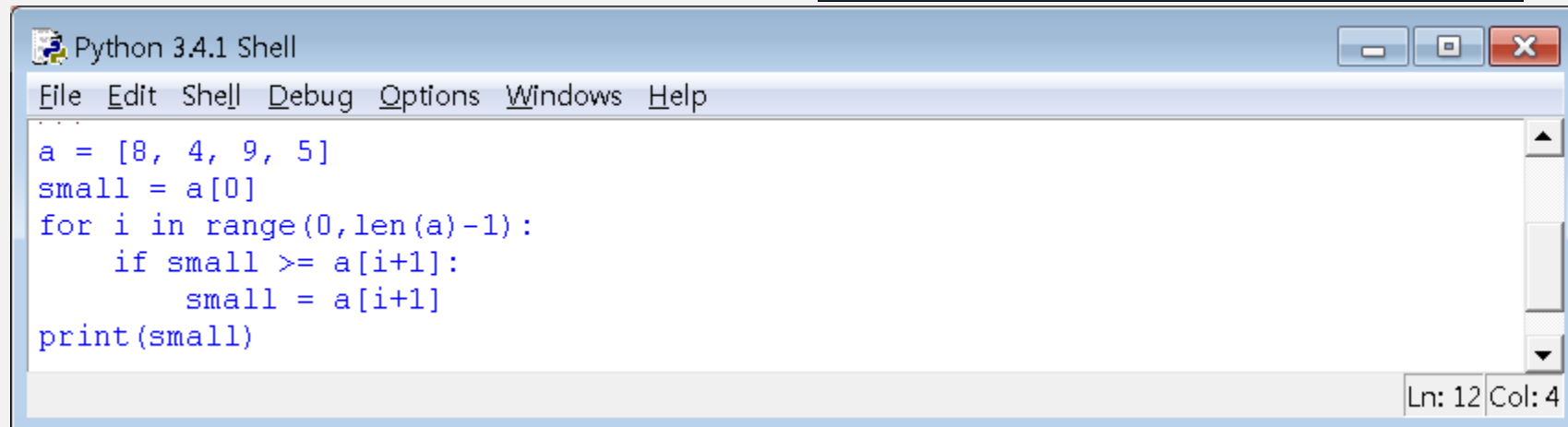
- file객체.read()
 - file 전체를 읽어서 문자열로 반환



```
Python 3.4.1: 12example5...
File Edit Format Run Options Windows
Help
f = open("test.py", "r")
line = f.read()
print (line)
f.close()
Ln: 1 Col: 0
```



```
test.py - 메모장
파일(E) 편집(E) 서식(O) 보기(V) 도움말(H)
a = [8, 4, 9, 5]
small = a[0]
for i in range(0, len(a)-1):
    if small >= a[i+1]:
        small = a[i+1]
print(small)
```



```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
a = [8, 4, 9, 5]
small = a[0]
for i in range(0, len(a)-1):
    if small >= a[i+1]:
        small = a[i+1]
print(small)
Ln: 12 Col: 4
```

Reading a file

- file을 읽다가 맨 처음으로 이동하고 싶은 경우
 - seek(0) 함수 사용
 - f.seek(0)

Text files and binary files


- 텍스트 file
 - 아스키 코드 또는 유니코드 만으로 구성된 file
 - open() 함수 텍스트 모드로 작업
 - 예: “rt”, “r” → 기본 모드
 - 글자, 숫자, 개행문자 등으로 구성
 - 윈도우 notepad.exe로 내용을 읽을 수 있음
 - 리눅스 cat 명령어로 내용을 읽을 수 있음
- 이진 file
 - 그림, 동영상, 음악 등 텍스트 이외의 정보 저장
 - open() 함수 텍스트 바이너리 모드로 작업
 - 예: “rb”
- hwp file은 어떤 file인가?

Pickle

- 피클(pickle)
 - 다양한 데이터 타입이 혼재되어 있는 list, tuple, object 등을 file로 저장하고 싶을 때 사용
 - list 안의 항목들이 문자열이 아닐 때도 있음
 - ['Fred', 73, 'Hello there', 81.9876e-13] list를 file에 저장하면 모든 정보가 문자열로만 저장됨
 - 피클을 사용하면 저장된 file을 원래 데이터 타입으로 자동으로 변환이 가능함
 - 피클링(pickling): 객체를 file로 저장
 - dump() 함수 사용 (write() 대신)
 - 언피클링(unpickling): file로부터 읽어드린 내용을 객체로 변환
 - load() 함수 사용 (read() 대신)

Pickle

- 피클링 (Pickling)
 - dump 함수 사용
 - file을 이진모드로 열어야 함


 12example9.py

File Edit Format Run Options Window Help

```
import pickle
```

```
my_list = ["Yoon", "2010100", 95]
```

```
f = open('testPickle.dat', 'wb') #이진모드 주의
pickle.dump(my_list, f)
f.close()
```

 testPickle - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

```
| □ ]q (X  Yoonq rX• 2010100q K_e.
```

Pickle

- 피클링 (Pickling)
 - dump 함수 사용
 - file을 이진모드로 열어야 함

```
12example9_class.py
File Edit Format Run Options Window Help
1 import pickle
2
3 class Pickle1:
4
5     def __init__(self):
6         self.my_list = ["Yoon", "2010100", 95]
7
8     def pickle_dump(self):
9         f = open('testPickle.dat', 'wb') #이진모드 주의
10        pickle.dump(self.my_list, f)
11        f.close()
12
13 class Main:
14     def __init__(self):
15         pass
16
17     def main(self):
18         pick = Pickle1()
19         pick.pickle_dump()
20
21 #main
22 if __name__ == "__main__":
23     try:
24         pic_main = Main()
25         pic_main.main()
26
27     except Exception as e:
28         print("\n>> Warning : ", e)
29
```

Pickle

- 언피클링
 - load 함수 사용
 - file을 이진모드로 열어야 함

12example10.py

File Edit Format Run Options Window Help

```
1 import pickle
2
3 f = open('testPickle.dat', 'rb') #이진모드 주의
4 my_list = pickle.load(f)
5
6 print(">> my_list = ", my_list)
7
8 f.close()
9
```

```
>> my_list = ['Yoon', '2010100', 95]
```

Pickle

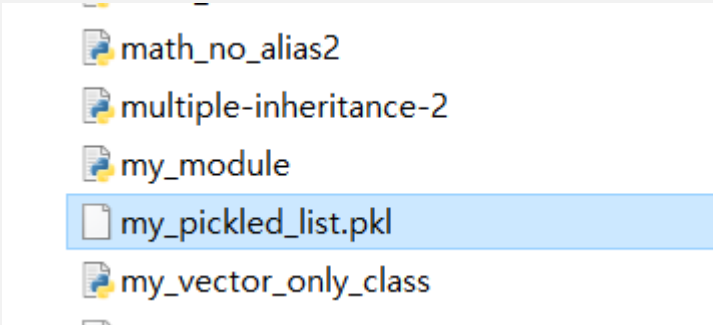
- 언피클링
 - load 함수 사용
 - file을 이진모드로 열어야 함

```
12example10_class.py
File Edit Format Run Options Window Help
1 import pickle
2
3 class Pickle1:
4
5     def __init__(self):
6         self.my_list = ["Yoon", "2010100", 95]
7
8     def pickle_dump(self):
9         f = open('testPickle.dat', 'wb') #이진모드 주의
10        pickle.dump(self.my_list, f)
11        f.close()
12
13    def unpickle_load(self):
14        f = open('testPickle.dat', 'rb') #이진모드 주의
15        loaded_list = pickle.load(f)
16        print(">> my_lsit = ", loaded_list)
17        f.close()
18
19 class Main:
20     def __init__(self):
21         pass
22
23     def main(self):
24         pick = Pickle1()
25         pick.pickle_dump()
26         pick.unpickle_load()
27
28 #main
29 if __name__ == "__main__":
30     try:
31         pic_main = Main()
32         pic_main.main()
33
34     except Exception as e:
35         print("\n>> Warning : ", e)
36
>> my_lsit = ['Yoon', '2010100', 95]
```



```
1 import pickle
2
3 print("\n1) pickling 시작")
4
5 my_list = ['Fred', 73, 'Hello there', 81.9876e-13]
6 print("\n2) my_list = ", my_list)
7
8 print("\n3) wb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성")
9 pickle_file = open('my_pickled_list.pkl', 'wb')
10
11 print("\n4) dump")
12 pickle.dump(my_list, pickle_file)
13 print("\n5) pickle_file = ", pickle_file)
14
15 print("\n6) close pickl_file")
16 pickle_file.close()
17
18 print("\n7) unpickling 시작")
19 print("\n8) rb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성")
20 pickle_file = open('my_pickled_list.pkl', 'rb')
21
22 print("\n9) load")
23 unpickled_list = pickle.load(pickle_file)
24
25 print("\n10) close pickl_file")
26 pickle_file.close()
27
28 print("\n11) recovered_list = ", unpickled_list)
29
```

1) pickling 시작
2) my_list = ['Fred', 73, 'Hello there', 8.19876e-12]
3) wb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성
4) dump
5) pickle_file = <_io.BufferedWriter name='my_pickled_list.pkl'>
6) close pickl_file
7) unpickling 시작
8) rb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성
9) load
10) close pickl_file
11) recovered_list = ['Fred', 73, 'Hello there', 8.19876e-12]



math_no_alias2
multiple-inheritance-2
my_module
my_pickled_list.pkl
my_vector_only_class

listing_22-6-class-pickle.py

File Edit Format Run Options Window Help

1 import pickle

2

3 D = True

4 #D = False

5

6 class Pickle1:

7

8 def __init__(self):

9 self.my_list = ['Fred', 73, 'Hello there', 81.9876e-13]

10 if D:

11 print("i) self.my_list = ", self.my_list)

12

13 def pickle_dump(self):

14 if D:

15 print("p1) wb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성")

16 pickle_file = open('my_pickled_list.pkl', 'wb') #이진모드 주의

17

18 if D:

19 print("p2) dump")

20 pickle.dump(self.my_list, pickle_file)

21

22 if D:

23 print("p3) close pickl_file")

24 pickle_file.close()

25

26 def unpickle_load(self):

27 if D:

28 print("u1) unpickling 시작")

29 print("u2) rb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성")

30 pickle_file = open('my_pickled_list.pkl', 'rb')

31

32 if D:

33 print("u3) load")

34 unpickled_list = pickle.load(pickle_file)

35

36 if D:

37 print("u4) close pickl_file")

38 pickle_file.close()

39

40 print(">> unpickled_list = ", unpickled_list)

41

class Main:

43 def __init__(self):

44 pass

45

46 def main(self):

47 if D:

48 print("\n1) pickling 시작")

49

50 pick = Pickle1()

51 pick.pickle_dump()

52 pick.unpickle_load()

53

54 #main

55 if __name__ == "__main__":

56 try:

57 pic_main = Main()

58 pic_main.main()

59

60 except Exception as e:

61 print("\n>> Warning : ", e)

62

math_no_alias2

multiple-inheritance-2

my_module

my_pickled_list.pkl

my_vector_only_class

1) pickling 시작

i) self.my_list = ['Fred', 73, 'Hello there', 8.19876e-12]

p1) wb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성

p2) dump

p3) close pickl_file

u1) unpickling 시작

u2) rb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성

u3) load

u4) close pickl_file

>> unpickled_list = ['Fred', 73, 'Hello there', 8.19876e-12]

...

소프트웨어학부

58

실습

- Pickling을 이용해서 Dictionary Data type을 file로 저장하고 읽는 프로그램을 작성하시오.
 - dictionary data
{ "waiver": "권리 포기, 면제", "tuition": "수업료", "insurance": "보험" }

```
12example11.py
File Edit Format Run Options Window Help
1 import pickle
2
3 f = open('myDic.dat', 'wb')
4 pickle.dump({"waiver": "권리 포기, 면제", "tuition": "수업료", "insurance": "보험"}, f)
5 f.close()
6
7 f = open('myDic.dat', 'rb')
8 dic=pickle.load(f)
9 print(dic)
10 f.close
11
```

실습

- 이전 Pickling을 이용해서 Dictionary Data type을 file로 저장하고 읽는 프로그램을 class로 재작성 하시오.

숙제

- 단어 숫자 분포

- 책을 구성하는 단어를 등장 빈도수로 나열하면 지프의 법칙(Zipf's law)을 발견할 수 있다고 한다.
 - Zipf's law, Zipf's distribution (지프의 법칙, 위키백과)
 - 데이터의 빈도는 순위에 반비례한다. 실험 결과, 영어에서 가장 많이 등장하는 단어 “the”는 전체 문서에서 7%의 빈도(전체 단어 약 백만 개 중 69,671회 등장)로 등장하며, 2위인 “of”는 약 3.5% 등장, 3위인 “and”는 약 2.4% 등장했다.
 - 어떤 나라에서 n번째로 인구가 많은 도시는 가장 인구가 많은 도시의 인구수의 $1/n$ 이 된다.
 - 포털 사이트의 가장 인기 있는 페이지도 지프 분포를 따른다.
 - 조화수열은 가장 간단한 형태의 지프 분포이다.
 - » $1, 1/2, 1/3, \dots, 1/n$

도시 이름	인구	인구/뉴욕 인구	지프의 법칙
뉴욕	831만 명	1/1	1/1
로스앤젤레스	383만 명	1/2.2	1/2
시카고	283만 명	1/2.9	1/3
휴스턴	220만 명	1/3.8	1/4
피닉스	155만 명	1/5.4	1/5

도시 이름	인구	인구/서울 인구	지프의 법칙
서울	1020만 명	1/1	1/1
부산	356만 명	1/2.9	1/2
인천	269만 명	1/3.8	1/3
대구	249만 명	1/4.1	1/4
대전	148만 명	1/6.9	1/5

숙제

- 단어 숫자 분포
 - 레 미제라블 텍스트 file("Les_Miserables-Victor_Hugo.txt")을 읽어 들여 등장하는 단어들의 횟수를 모두 구하고 zipf's law와의 관련성을 분석하시오.
 - 다음 내용을 먼저 학습하시오.
 - file객체.read().split()
 - f.read().split()
 - 사전 구성

```
wordCount={}
if w not in wordCount:
    wordCount[w]=1
else:
    wordCount[w] += 1
```
 - 스트링.replace(".",",")
 - y=x.replace(".",",").replace("?",",")
 - 스트링.upper()

숙제

Les_Miserables-Victor_Hugo - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Chapter one

Jean Valjean

One evening in October 1815, an hour before sunset, a man with a long beard and dusty, torn clothes walked into the town of Digne. He was in his late forties, of medium height, broad-shouldered and strong. A leather cap half-hid his face, which was sunburnt and shining with sweat. His rough yellow shirt was unbuttoned, revealing a hairy chest. On his back was a heavy soldier's bag, and in his hand was a large wooden stick.

The townspeople, who had never seen him before, watched with interest as he stopped for water at a fountain. Children followed him to the marketplace, where he stopped for more water at another fountain. He then crossed the square towards an inn, and entered by the kitchen door.

The innkeeper, who was also the cook, was busy with his pots and pans, preparing a meal for a group of travellers who were laughing and joking in the next room.

'What can I do for you, Monsieur?' he asked without looking up.

'A meal and a bed,' said the stranger.

'Of course.' The innkeeper turned to look at him. Then, seeing the visitors rough appearance, he added, 'If you can pay for it.'

'I have money.' The stranger produced an old leather purse from his jacket.

'Then you're welcome,' the innkeeper said.

The stranger smiled with relief and sat down by the fire. He did not see a young boy run out with a note that the innkeeper had quickly written. He did not see the boy return a short time later and whisper something to the innkeeper.

'When will the meal be ready?' the stranger asked.

'I'm sorry, Monsieur,' the innkeeper said. 'You can't stay here. I've got no free rooms.'

'Then put me in a stable. All I need is a quiet corner somewhere. After dinner

Ln 1, Col 1100%Windows (CRLF)UTF-8

숙제

- 영어 사전

- “vocabulary.txt”는 인터넷에서 다운받은 영단어 텍스트 file인데 내용이 뒤섞여있다. file 내용을 이 용해서 “단어:의미”로 구성된 사전을 구현하시오. 사용자로 부터 단어를 입력 받으면 단어의 의미가 화면에 출력되어야 한다.



1	environment	환경	16	present	"현재, 출석한, 제출하다"
2	company	회사	17	draw	그리다. 뽑다
3	government	"정부, 정치"	18	lose	"잃어버리다, 지다"
4	face	얼굴	19	provide	제공(공급)하다.
5	race	"경주, 인종"	20	allow	허락하다.
6	plant	"식물, 식물을 심다"	21	consider	"생각하다, 여기다"
7	fire	"화재, 발사하다"	22	realize	알다
8	matter	"문제, 중요하다"	23	mean	"의미하다, 비천한"
9	state	"상태, 국가, 말하다"	24	patient	인내심 있는. 환자
10	cause	"원인, 야기하다"	25	ancient	고대의
11	reason	"이유, 이성, "	26	individual	"개인의, 개인"
12	concern	"걱정, 관련되다"	27	positive	"긍정적인, 적극적인"
13	bear	"곰, 참다, 낳다"	28	genetic	유전의
14	raise	"들어올리다, 모금하다"	29	likely	할 것 같은
15	subject	주제. 과목	30	still	"여전히, 움직이지 않는"
1	planet	"행성, 지구"	16	relax	쉬다. 진정하다.
2	article	기사. 물품	17	suppose	"추측, 가정(하다)"
3	education	교육	18	require	요구(하다).
4	purpose	"목적, 의도"	19	affect	영향을 미치다
5	arm	"무기, 무장하다"	20	add	더하다. 덧붙여 말하다.
6	object	물체. 대상	21	involve	포함하다. 참여시키다
7	amount	양	22	prevent	막다. 못하게 하다.
8	rest	"나머지, 휴식(하다)"	23	fine	"좋은, 벌금"
9	process	"과정. 처리하다"	24	material	"물질, 재료, 물질의 "

숙제

- File Input and Output

- 숙제 1

레 미제라블 텍스트 file("Les_Miserables-Victor_Hugo.txt")을 읽어 들여 등장하는 단어들의 횟수를 모두 구하고 zipf's law와의 관련성을 분석하시오.

file 이름 : 이름-학번- LesMiserables.py

- 숙제 2

"vocabulary.txt"는 인터넷에서 다운받은 영단어 텍스트 file인데 내용이 뒤섞여있다. file 내용을 이용해서 "단어:의미"로 구성된 사전을 구현하시오.

file 이름 : 이름-학번- vocabulary.py

- Les_Miserables-Victor_Hugo.txt ,vocabulary.txt 화일은 수업자료를 업로드하는 공지에 올려놓았습니다.

- 마감일 : 2020. 12. 8(화) 13:00