

# 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



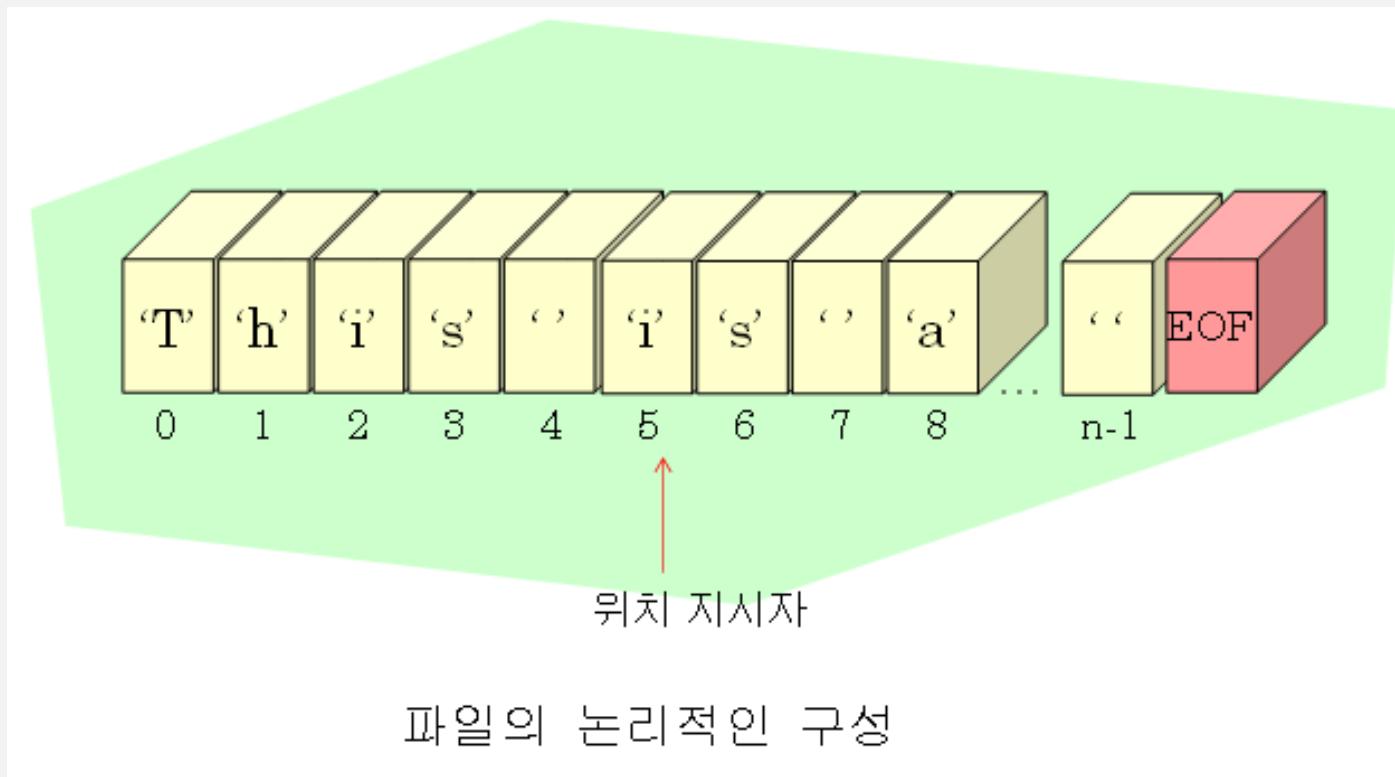
<https://beencomplex.wordpress.com/2015/09/17/i-had-it-with-these-motherfilipin-files-formats-on-this-motherflipin-internet/>

# What is a file ? - file 의 필요성



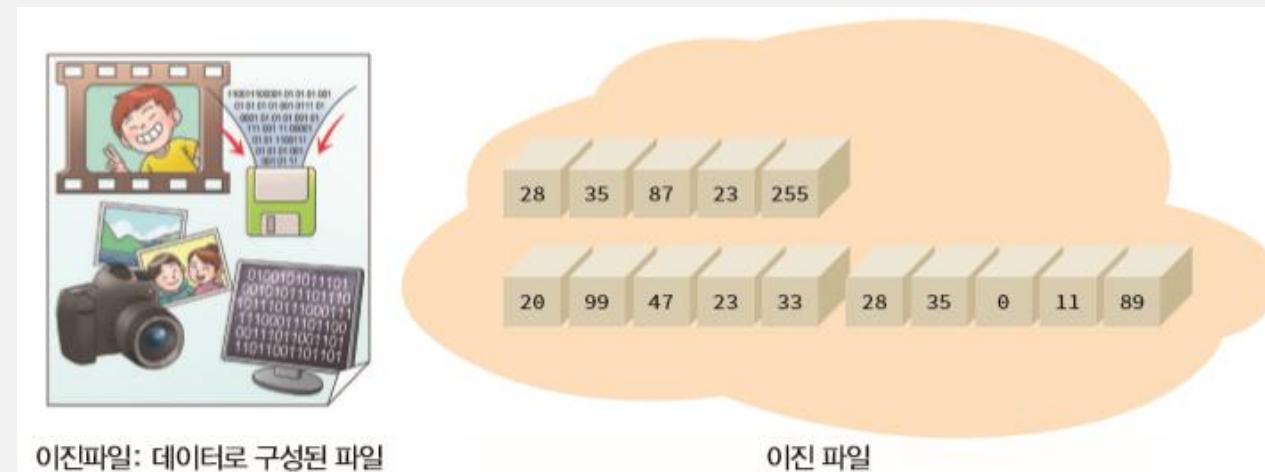
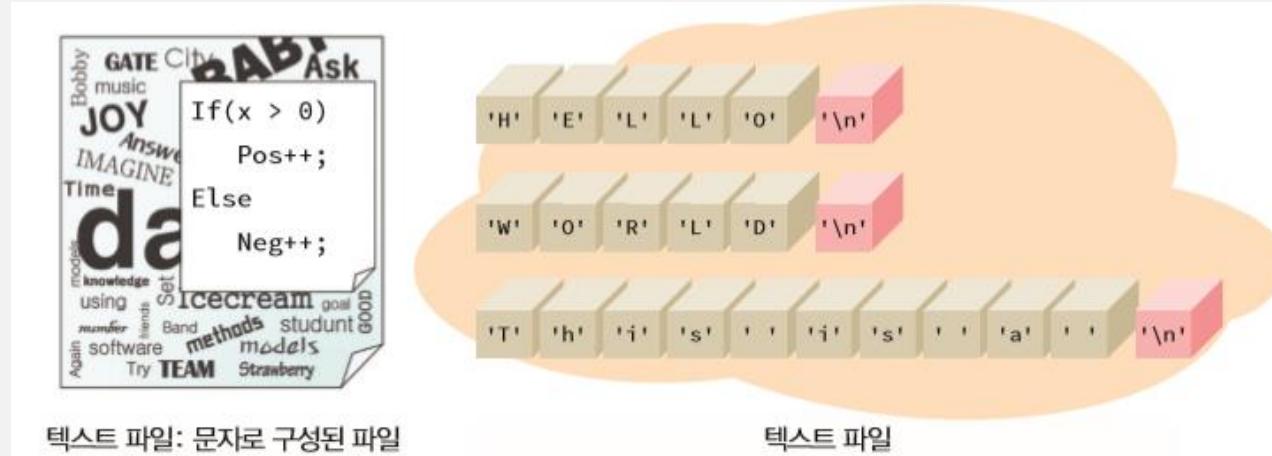
# What is a file ? - 논리적인 file 구조

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



# What is a file ? - file의 종류

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

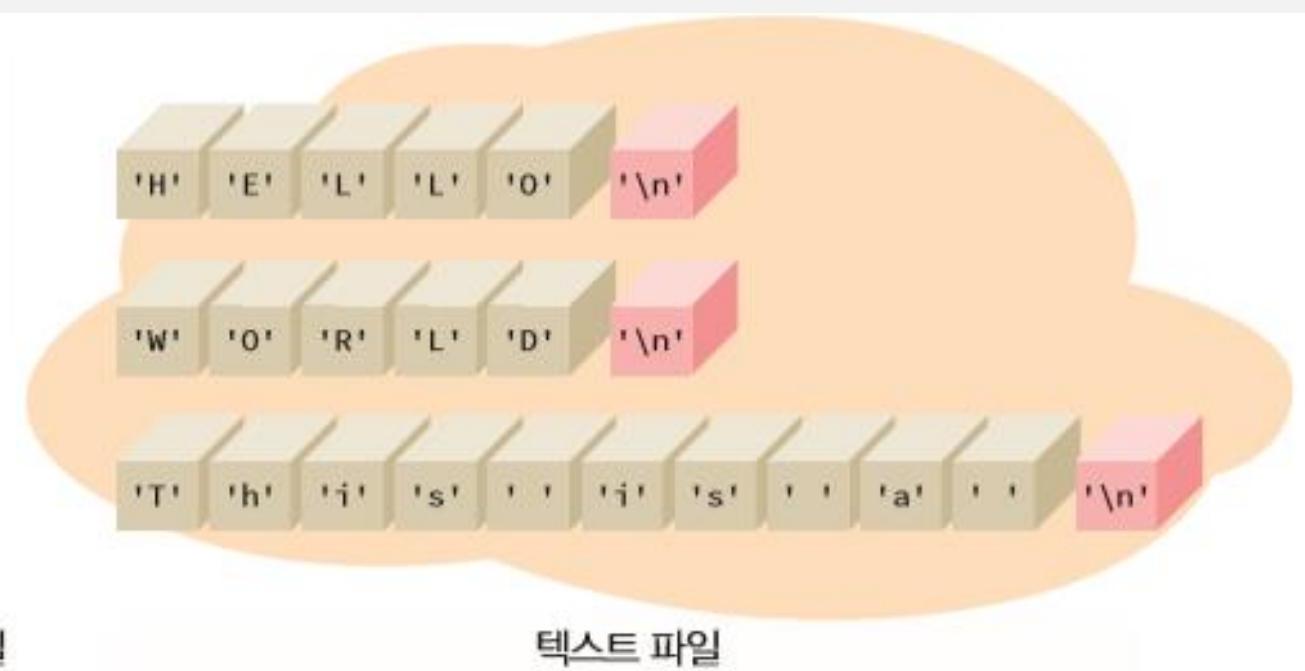


# What is a file ? - file의 종류

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



텍스트 파일: 문자로 구성된 파일

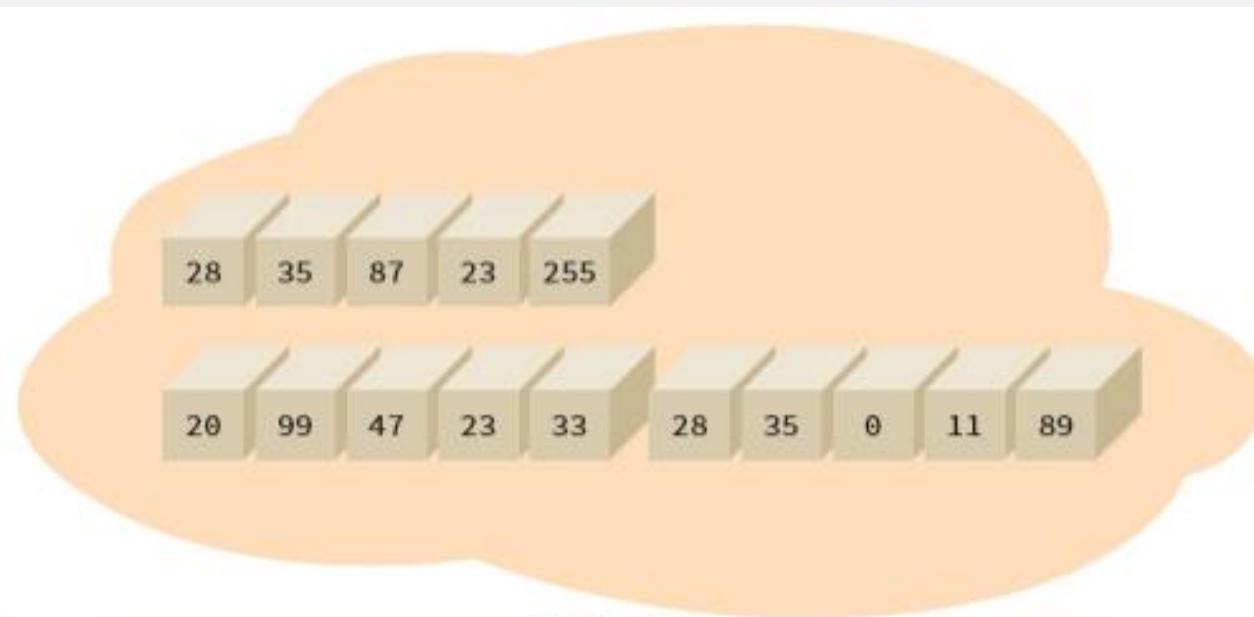


# What is a file ? - file의 종류

- 이진 file(binary file)
    - 사람이 읽을 수 없으나 computer는 읽을 수 있는 file
    - 문자 data가 아닌 이진 data 형태로 직접 저장
    - 특정 program에 의해서만 판독이 가능
- 예) sound file, image file, execution file 등



이진파일: 데이터로 구성된 파일



이진 파일

# What is a file ? - file의 특징

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

# What is a file ? - file의 특징

- File의 특징 : file 이름
  - 대부분의 OS에서는 file명의 일부가 해당 file에 어떤 유형의 data가 저장되어 있는지를 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**
    - 예 2) Python file : file.**py**  
**학부.hwp**  
**list.doc**  
**test.cpp**  
**sender.java**
  - 확장자는 OS에 따라 다를 수 있음

# What is a file ? - file의 특징

- File의 특징 : file 위치 (1/2)
    - 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
- 예) image\_file = “c:/images/jpg/flower.jpg”

# What is a file ? - file의 특징

- File의 특징 : file 위치 (2/2)
  - 절대경로
    - 저장매체의 root 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 : pwd / dir

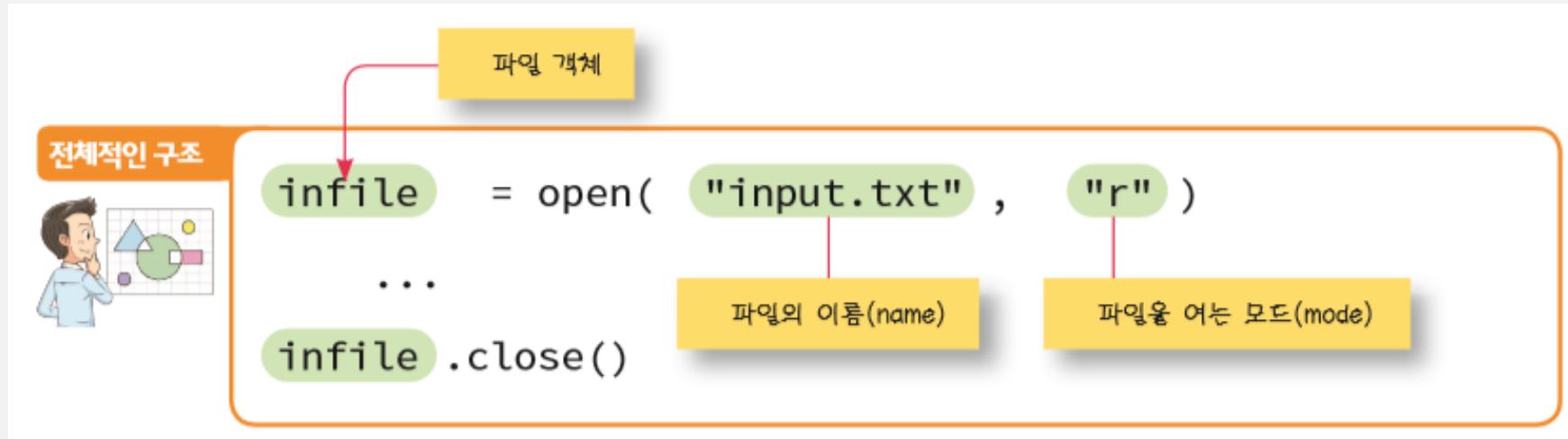
# What is a file ? - file의 특징

- File의 특징 : Meta 정보

- 메타정보 예
  - EXIF: exchangeable image file format
- 디지털 카메라의 image file 안에 저장되어 있는 file 형식
  - data와 함께 카메라 제조사(Marker)
  - 카메라 모델(Model), 에디터(Software)
  - 사진을 보정한 날짜(Date time), EXIF 버전(EXIF Version)
  - 촬영한 날짜(Shoot Date time)
  - GPS
  - Web에 올려진 사진의 실제 크기(Image Size)
  - 노출시간(Exposure Time)
  - 촬영 프로그램(Exposure Program)
  - 렌즈초점 길이(Focal Length)
  - 조리개수치(F-Number)
  - 셔터 스피드(Shutter Speed)
  - 플래시 사용여부 등 세부적인 부가 정보

<https://www.sony.co.kr/electronics/support/articles/S500078506>

# Text file 읽고 쓰기



# Opening a file

- open() 함수 호출
  - open()을 이용하여 물리 file을 열어 **file object**을 생성
    - 이후 생성된 file object를 통하여 program 내에서 file에 대한 read/write operation을 수행
    - File object를 생성하고 나면 program내에서 **물리적 file명 대신에 file object를 이용**하여 처리
  - file object = open(file명, 열기 모드)
    - f = open("input.txt", "r")
    - f = open("output.txt", "w")
    - f = open("append.txt", "a")
- file 닫기
  - file-object.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.



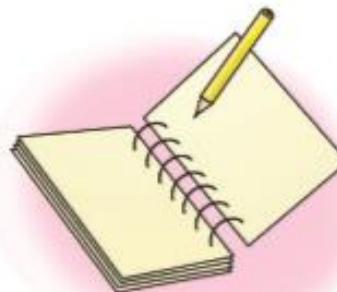
"r"

파일을 처음부터 읽는다.



"w"

파일의 처음부터 쓴다.  
만약 파일이 존재하면  
기존의 내용이 지워진다.



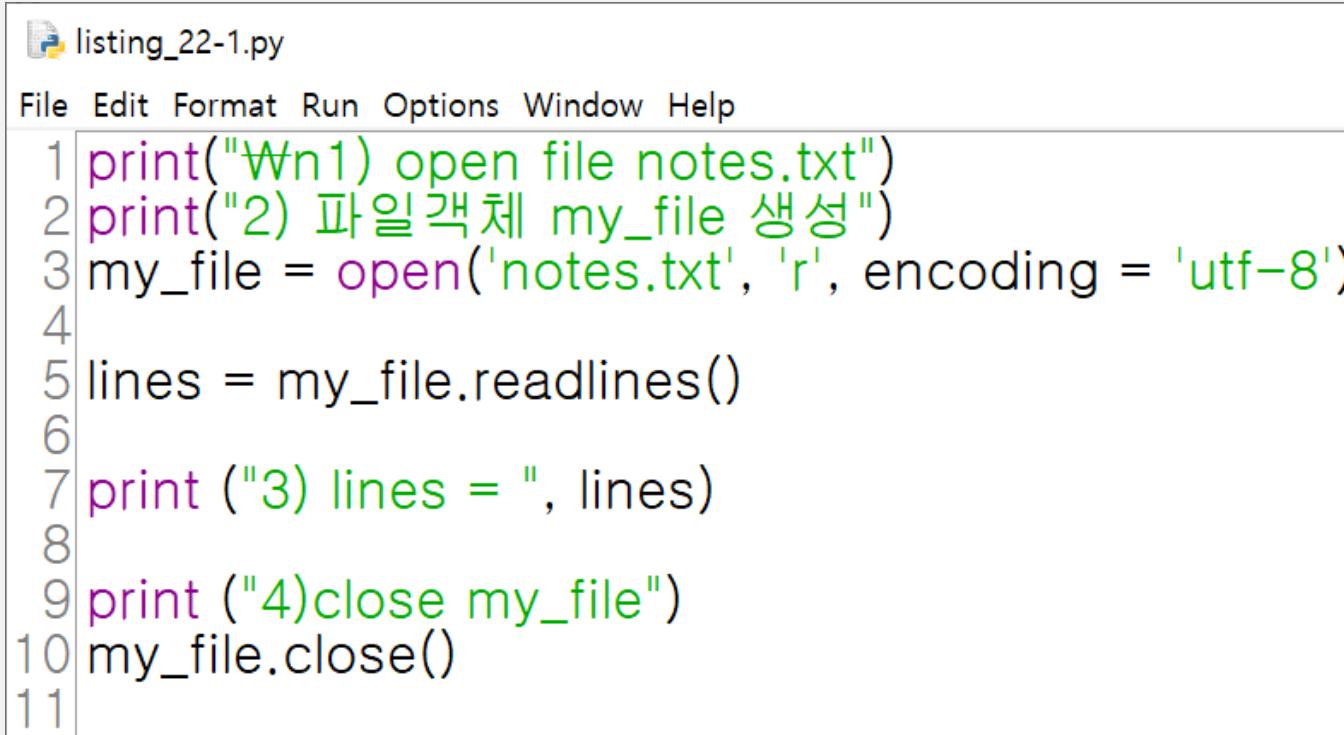
"a"

파일의 끝에 쓴다.  
파일이 없으면 생성된다.

# Opening a file

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

# Opening a file



```
listing_22-1.py
File Edit Format Run Options Window Help
1 print("1) open file notes.txt")
2 print("2) 파일 객체 my_file 생성")
3 my_file = open('notes.txt', 'r', encoding = 'utf-8')
4
5 lines = my_file.readlines()
6
7 print ("3) lines = ", lines)
8
9 print ("4) close my_file")
10 my_file.close()
11
```

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

```
1) open file notes.txt
2) 파일 객체 my_file 생성
Traceback (most recent call last):
  File "C:\소사\강의 예제\listing_22-1.py", line 5, in <module>
    my_file = open('notes.txt', 'r', encoding = 'cp949')
FileNotFoundError: [Errno 2] No such file or directory: 'notes.txt'
```

```

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

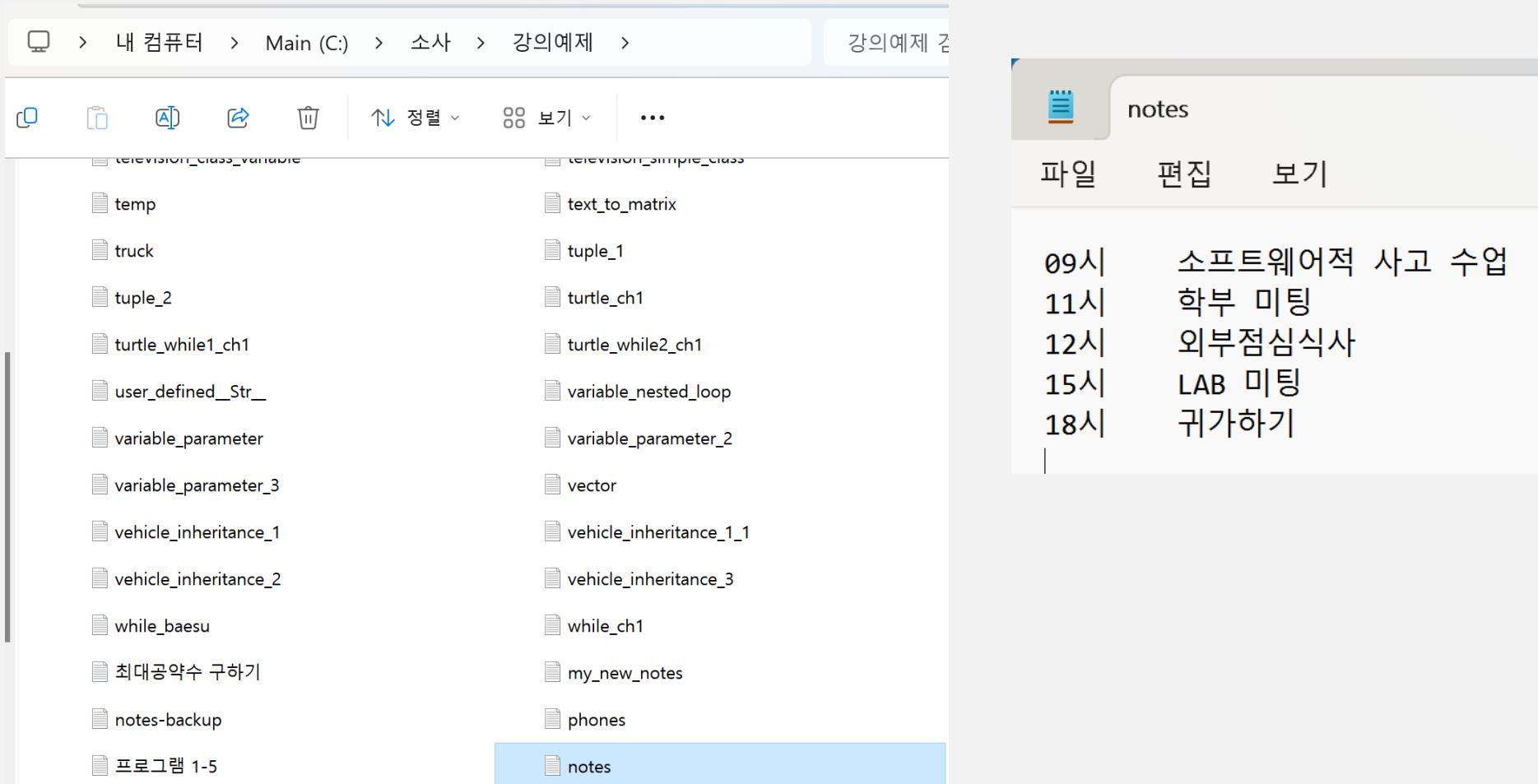
```

- 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("1) open file notes.txt")
2 print("2) 파일객체 my_file 생성")
3
4 #my_file = open('notes.txt', 'r', encoding = 'utf-8')
5 my_file = open('notes.txt', 'r', encoding = 'cp949')
6
7 lines = my_file.readlines()
8
9 print ("3) lines = ", lines)
10
11 print ("4)close my_file")
12 my_file.close()
13
```

```
1) open file notes.txt
2) 파일객체 my_file 생성
3) lines = ['09시₩t소프트웨어적 사고 수업₩n', '11시₩t학부 미팅₩n', '12시₩t외부점심식사₩n',
   '15시₩tLAB 미팅₩n', '18시₩t귀가하기₩n']
4)close my_file
```

- **Encoding**

- 'euc-kr', 'utf-8'
- cp949 (Code Page 949) : MS Windows

File Edit Format Run Options Window Help

```
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             self.my_file = open('notes.txt', 'r', encoding = 'euc-kr')
7         except:
8             print(">> Warning in 생성자) notes.txt file이 존재하지 않음 ! ")
9
10    def read_file(self):
11        # line 단위로 file에서 읽어옴
12        try:
13            print("Wnr-1) read_file")
14            lines = self.my_file.readlines()
15            print("r-2) lines = ", lines)
16
17        except Exception as e:
18            print(">> Warning in read_file) ", e)
19
20    def __del__(self):
21        try:
22            print ("Wnd-1) Destructor에서 my_file을 close")
23            self.my_file.close()
24
25        except Exception as e:
26            print(">> Warning in 소멸자) ", e)
27
28 class Main():
29     def main(self):
30         ex_file = File_1()      i_1) Constructor에서 notes.txt file을 open
31         ex_file.read_file()    r-1) read_file
32         r-2) lines = ['09시 Wt소프트웨어적 사고 수업\n', '11시 Wt학부 미팅\n', '12시 Wt외부점심식사\n', '15시 WtLAB 미팅\n', '18시 Wt평가하기\n']
33 #main
34 exe_main = Main()
35 exe_main.main()
36
```

```

File Edit Format Run Options Window Help
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', encoding = 'euc-kr')
6
7         except:
8             print(">> Warningin 생성자) notes.txt file이 존재하지 않음 ! ")
9
10    def read_file(self):
11        # line 단위로 file에서 읽어옴
12        try:
13            print("Wnr-1) read_file")
14            first_line = self.my_file.readline()
15            second_line = self.my_file.readline()
16            third_line = self.my_file.readline()
17            fourth_line = self.my_file.readline()
18            fifth_line = self.my_file.readline()
19
20            print ("r-2) first line = ", first_line)
21            print ("r-3) second line = ", second_line)
22            print ("r-4) third line = ", third_line)
23            print ("r-5) fourth line = ", fourth_line)
24            print ("r-6) fifth line = ", fifth_line)
25
26        except Exception as e:
27            print(">> Warning in read_file) ", e)
28
29    def __del__(self):
30        try:
31            print ("Wnd-1) Destructor에서 my_file을 close")
32            self.my_file.close()
33
34        except Exception as e:
35            print(">> Warning in 소멸자) ", e)
36
37 class Main():
38     def main(self):
39         ex_file = File_1()
40         ex_file.read_file()
41
42 #main
43 exe_main = Main()
44 exe_main.main()
45 print("Wnmain) End.")
46

```

- notes.txt 내용을 한 줄씩 출력 - 1

i\_1) Constructor에서 notes.txt file을 open  
 r-1) read\_file  
 r-2) first line = 09시 소프트웨어적 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', encoding = 'euc-kr')
6
7         except:
8             print(">> Warningin 생성자) notes.txt file이 존재하지 않음 ! ")
9
10    def read_file(self):
11        # line 단위로 file에서 읽어옴
12        try:
13            print("r-1) 한 줄씩 출력")
14            lines = self.my_file.readlines()
15            print("r-2) lines = ", lines)
16
17            i = 1
18            for each_line in lines:
19                #each_line.rstrip() # 각 줄에서 줄바꿈 문자를 제거
20                print ( ">> {} 번째 줄 내용 {}".format(i, each_line.rstrip()))
21                i = i + 1
22
23        except Exception as e:
24            print(">> Warning in read_file) ", e)
25
26    def __del__(self):
27        try:
28            print ("Wnd-1) Destructor에서 my_file을 close")
29            self.my_file.close()
30
31        except Exception as e:
32            print(">> Warning in 소멸자) ", e)
33
34 class Main():
35     def main(self):
36         ex_file = File_1()
37         ex_file.read_file()
38
39 #main
40 exe_main = Main()
41 exe_main.main()
42 print("Wnmain) End.")
43

```

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

i\_1) Constructor에서 notes.txt file을 open  
r-1) 한 줄씩 출력  
r-2) lines = ['09시Wt소프트웨어적 사고 수업\n', '11시Wt학부 미팅\n', '12시Wt외부점심식사\n', '15시WtLAB 미팅\n', '18시Wt귀가하기\n']  
>> 1 번째 줄 내용 09시 소프트웨어적 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.

```

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', encoding = 'euc-kr')
6         except:
7             print(">> Warning in 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-2) my_file을 r mode로 open")
23                self.my_file = open('notes.txt', 'r', encoding = 'cp949')
24            elif mode == "a":
25                print ("Wn0-3) my_file을 a 모드로 open")
26                self.my_file = open('notes.txt', 'a', encoding = 'cp949')
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    def append(self):
47        print("Wna-1) notes.txt file을 append 모드로 open")
48        self.open_mode("a")
49
50        print("a-2) 새로운 일정 추가 : 19시 가족과 함께 저녁식사")
51        self.my_file.write('19시 가족과 함께 저녁식사')
52
53        print("a-3) close my_file")
54        self.close_file()
55
56        print("a-4) read mode로 my_file open")
57        self.open_mode("r")
58        self.read_file()
59
60    def __del__(self):
61        try:
62            print ("Wnd-1) Destructor에서 my_file을 close")
63            self.my_file.close()
64
65        except Exception as e:
66            print(">> Warning in 소멸자) ", e)
67
68 class Main():
69     def main(self):
70         ex_file = File_1()
71         ex_file.read_file()
72         ex_file.append()
73
74     #main
75     print("1) main")
76     exe_main = Main()
77     exe_main.main()
78     print("Wn2) end")
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
625
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503

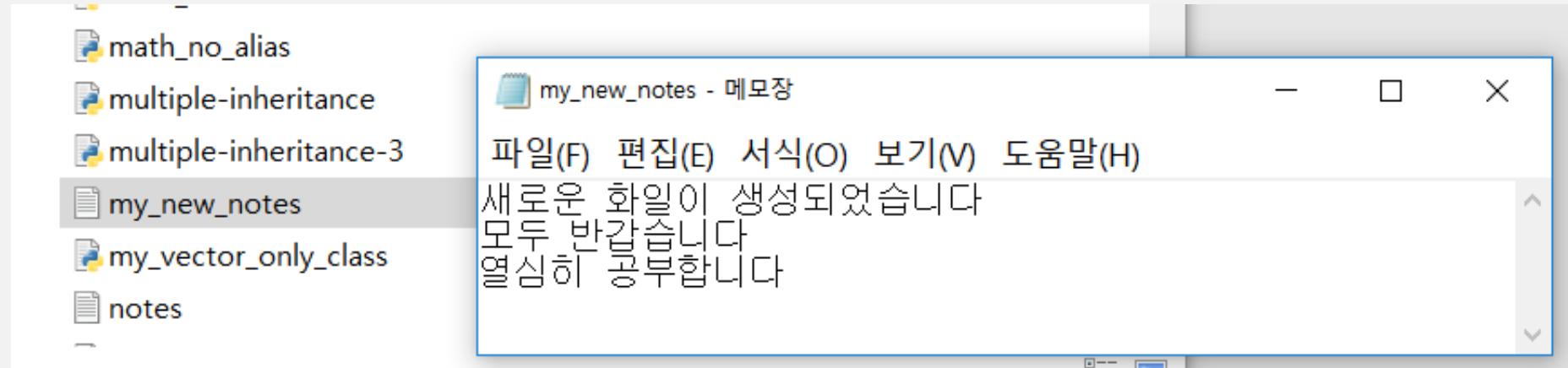
```

 listing\_22-4.py

```
File Edit Format Run Options Window Help
1 print("Wn1) w mode로 my_new_notes.txt open 후 파일객체 new_file 생성")
2 new_file = open("my_new_notes.txt", 'w', encoding = 'cp949')
3
4 print("2) write() method로 새로운 내용을 쓰기")
5 new_file.write("새로운 파일이 생성되었습니다Wn")
6 new_file.write("모두 반갑습니다Wn")
7 new_file.write("열심히 공부합니다")
8
9 print ("3) close new_file")
10 new_file.close()
11
12 print ("4) read mode로 파일객체 my_file 생성")
13 my_file = open('my_new_notes.txt', 'r', encoding = 'cp949')
14
15 lines = my_file.readlines()
16 print ("5) lines = ", lines)
17
18 print ("6) 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 파일이 생성되었습니다Wn', '모두 반갑습니다Wn', '열심히 공부합니다']
- 6) close my\_file

- 새 file my\_new\_notes.txt에 대하여 w mode로 내용 쓰기



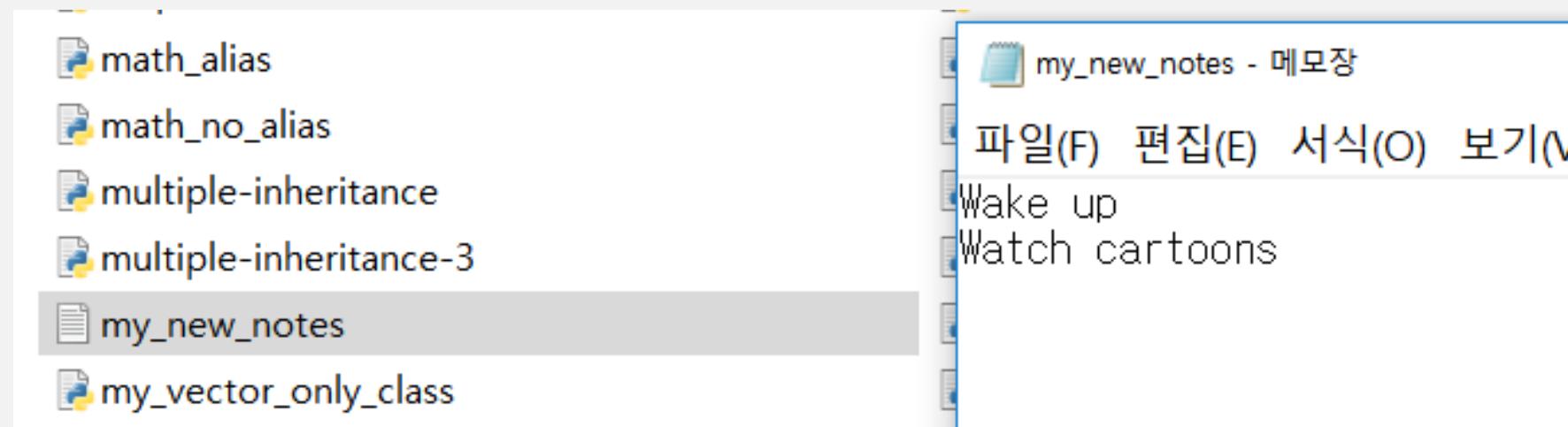
- 새 file `my_new_notes.txt`에 대하여 `w mode`로 내용 쓰기

```
1 print("Wn1) w mode로 기존 my_new_notes.txt open 후 파일객체 the_file 생성")
2 the_file = open('my_new_notes.txt', 'w', encoding = 'cp949')
3
4 print("2) write() method로 새로운 내용을 쓰기")
5 the_file.write("Wake upWn")
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', encoding = 'cp949')
13
14 lines = my_file.readlines()
15 print ("5) lines = ", lines)
16 print ("6) 기존 파일 my_new_notes.txt의 내용이 새로이 변경되었음")
17
18 print ("7) close my_file")
19 my_file.close()
```

- 기존 file에 쓰기  
모드 w 사용하기

- 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 upWn', 'Watch cartoons']
- 6) 기존 파일 my\_new\_notes.txt의 내용이 새로이 변경되었음
- 7) close my\_file

- 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("Wn{i_1} file_name : {}".format(file_name))
8             # print("i_2) Constructor에서 file을 r 모드로 open")
9         try:
10             self.f_name = file_name
11             self.my_file = open(self.f_name, 'r', encoding = 'cp949')
12         except:
13             print("Wn>> Warningin 생성자 {} 이 존재하지 않음 ! ".W
14                 format(file_name))
15
16     def open_mode(self, mode):
17
18         try:
19             if D:
20                 print("Wno-1) open_mode() / file_name : {}, mode : {} ".W
21                     format(self.f_name, mode))
22
23             if mode == "r":
24                 if D:
25                     print ("Wno-2) {}을 r mode로 open".format(self.f_name))
26                     self.my_file = open(self.f_name, 'r', encoding = 'cp949')
27
28             elif mode == "a":
29                 if D:
30                     print ("Wno-3) {}을 a 모드로 open".format(self.f_name))
31                     self.my_file = open(self.f_name, 'a', encoding = 'cp949')
32
33             elif mode == "w":
34                 if D:
35                     print ("Wno-4) {}을 w 모드로 open".format(self.f_name))
36                     self.my_file = open(self.f_name, 'w', encoding = 'cp949')
37
38         except Exception as e:
39             print("Wn>> Warning in open_mode() : ", e)
40

```

```

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 = ['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
c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp949'>를 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.txt를 a 모드로 open
a-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>를 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'>를 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\n', '모두 반갑습니다\n', '열심히 공부합니다\n', '19시 가족과 함께 저녁식사']
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다
>> 4 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.txt를 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'>를 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\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'>를 close
2) Main End.

```

새 file  
my\_new\_notes.txt  
에 대하여 w mode  
로 내용 쓰기 (1/4)

```

41 def close_file(self):
42     try:
43         if D:
44             print("Wnc_1) close_file()")
45             print ("c-2) {}을 close".format(self.my_file))
46
47             self.my_file.close()
48
49 except Exception as e:
50     print("Wn>> Warning in close_file() : ", e)
51
52 def read_file(self):
53     # line 단위로 file에서 읽어옴
54     try:
55         if D:
56             print("r-1) read_file()")
57
58             self.open_mode("r")
59
60             lines = self.my_file.readlines()
61             if D:
62                 print("Wnr-2) lines = ", lines)
63
64             i = 1
65             for each_line in lines:
66                 #each_line.rstrip() # 각 줄에서 줄바꿈 문자를 제거
67                 print (">> {} 번째 줄 내용 : {}".W
68                     format(i, each_line.rstrip()))
69             i = i + 1
70
71 except Exception as e:
72     print("Wn>> Warning in read_file) ", e)
73

```

```

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 = ['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
c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp949'>를 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.txt를 a 모드로 open
a-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>를 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'>를 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 생성되었습니다Wn', '모두 반갑습니다Wn', '열심히 공부합니다Wn', '19시 가족과 함께 저녁식사']
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다
>> 4 번째 줄 내용 : 19시 가족과 함께 저녁식사
w-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.txt를 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'>를 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 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'>를 close
2) Main End.

```

새 file  
my\_new\_notes.txt  
에 대하여 w mode  
로 내용 쓰기 (2/4)

```

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

```

```

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 = ['새로운 파일이 생성되었습니다Wn', '모두 반갑습니다Wn', '열심
히 공부합니다!']
>> 1 번째 줄 내용 : 새로운 파일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다
c_1) close_file()
c_2) <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp
949'>를 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.txt를 a 모드로 open
a-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp
949'>를 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'>를 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 파일이 생성되었습니다Wn', '모두 반갑습니다Wn', '열심
히 공부합니다Wn', '19시    가족과 함께 저녁식사']
>> 1 번째 줄 내용 : 새로운 파일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다
>> 4 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.txt를 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'>를 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 파일이 생성되었습니다Wn', '모두 반갑습니다Wn', '열심
히 공부합니다!']
>> 1 번째 줄 내용 : 새로운 파일이 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'>를 close
2) Main End.

```

새 file  
my\_new\_note  
s.txt에 대하여  
w mode로 내  
용 쓰기 (3/4)

```

109 def __del__(self):
110     try:
111         if D:
112             print ("Wn-1) Destructor에서 {}을 close".W
113                 format(self.my_file))
114             self.my_file.close()
115
116     except Exception as e:
117         print("Wn>> Warning in 소멸자) ", e)
118
119 class Main():
120     def main(self):
121
122 #       f_name = input("W>> File 이름을 입력하세요 : ")
123 f_name = "my_new_notes.txt"
124
125 ex_file = File_1(f_name)
126 ex_file.read_file()
127 ex_file.close_file()
128
129 ex_file.append()
130 ex_file.close_file()
131
132 ex_file.read_file()
133
134 ex_file.write_file()
135 ex_file.close_file()
136
137 ex_file.read_file()
138
139 #main
140 if D:
141     print("Wn1) main")
142
143 exe_main = Main()
144 exe_main.main()
145
146 if D:
147     print("Wn2) Main End.")
148

```

```

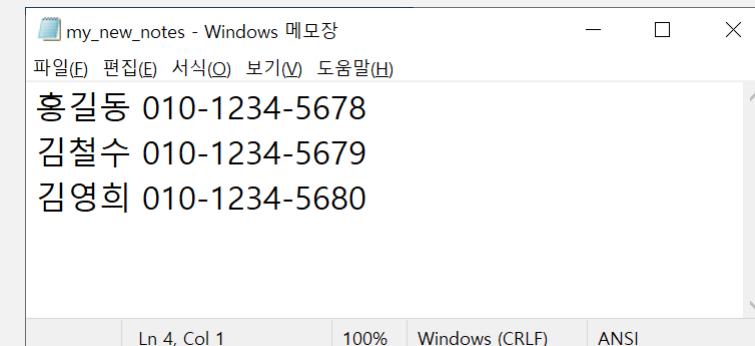
1) main
i_1) file_name : my_new_notes.txt
r-1) read_file()
o-1) open_mode() / file_name : my_new_notes.txt, mode : r
o-2) my_new_notes.txt를 r mode로 open
r-2) lines = ['새로운 화일이 생성되었습니다Wn', '모두 반갑습니다Wn', '열심히 공부합니다']
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다
c_1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='r' encoding='cp949'>를 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.txt를 a 모드로 open
a-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>를 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'>를 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 생성되었습니다Wn', '모두 반갑습니다Wn', '열심히 공부합니다Wn', '19시 가족과 함께 저녁식사']
>> 1 번째 줄 내용 : 새로운 화일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다
>> 4 번째 줄 내용 : 19시 가족과 함께 저녁식사
w-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.txt를 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'>를 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 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'>를 close
2) Main End.

```

- 만일 file이 존재하는지를 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", encoding = 'cp949')
4
5 if os.path.isfile("my_new_notes.txt"):
6     outfile = open("my_new_notes.txt", "w", encoding = 'cp949')
7     print("Warning) 동일한 이름의 파일이 이미 존재합니다. ")
8 else :
9     print(">> my_new_notes.txt를 w 모드로 open합니다.")
10    outfile = open("my_new_notes.txt", "w", encoding = 'cp949')
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()
```

>> 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("Wn{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', encoding = 'cp949')
16        except:
17            print("Wn>> Warningin 생성자 {} 이 존재하지 않음 ! ".W
18                  format(file_name))
19
20    def open_mode(self, mode):
21        try:
22            if D:
23                print("Wno{i-1} open_mode() / file_name : {}, mode : {} ".W
24                  format(self.f_name, mode))
25
26            if mode == "r":
27                if D:
28                    print ("o-2) {}을 r mode로 open".format(self.f_name))
29                    self.my_file = open(self.f_name, 'r', encoding = 'cp949')
30
31            elif mode == "a":
32                if D:
33                    print ("o-3) {}을 a 모드로 open".format(self.f_name))
34                    self.my_file = open(self.f_name, 'a', encoding = 'cp949')
35
36            elif mode == "w":
37                if D:
38                    print ("o-4) {}을 w 모드로 open".format(self.f_name))
39                    self.my_file = open(self.f_name, 'w', encoding = 'cp949')
40
41        except Exception as e:
42            print("Wn>> Warning in open_mode() : ", e)
43

```

# User-defined exception handler를 이용한 새 file my\_new\_notes.txt에 대하여 w mode로 내용 쓰기

1) main

i-1) file\_name : my\_new\_notes.txt

r-1) read\_file()  
>> 1 번째 줄 내용 : 흥길동 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.txt를 a 모드로 open

a-4) <`_io.TextIOWrapper` name='my\_new\_notes.txt' mode='a' encoding='cp949'>을 append 모드로 open

c-1) close\_file()  
c-2) <`_io.TextIOWrapper` name='my\_new\_notes.txt' mode='a' encoding='cp949'>을 close

r-1) read\_file()  
>> 1 번째 줄 내용 : 흥길동 010-1234-5678  
>> 2 번째 줄 내용 : 김철수 010-1234-5679  
>> 3 번째 줄 내용 : 김영희 010-1234-5680  
>> 4 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

MyFileNotFoundException) >>> Warning) 동일한 이름의 파일이 이미 존재합니다.  
>>> 현존하는 파일에 새로운 내용으로 overwrite하시겠습니까 ? Y or N : Y

o-1) open\_mode() / file\_name : my\_new\_notes.txt, mode : w  
o-4) my\_new\_notes.txt를 w 모드로 open

wc) 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()  
>> 1 번째 줄 내용 : 새로운 파일이 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.

```

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

```

1) main

i-1) file\_name : my\_new\_notes.txt

r-1) read\_file()  
 >> 1 번째 줄 내용 : 흥길동 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.txt를 a 모드로 open

a-4) <`_io.TextIOWrapper` name='my\_new\_notes.txt' mode='a' encoding='cp949'>을 append 모드로 open

c-1) close\_file()  
 c-2) <`_io.TextIOWrapper` name='my\_new\_notes.txt' mode='a' encoding='cp949'>을 close

r-1) read\_file()  
 >> 1 번째 줄 내용 : 흥길동 010-1234-5678  
 >> 2 번째 줄 내용 : 김철수 010-1234-5679  
 >> 3 번째 줄 내용 : 김영희 010-1234-5680  
 >> 4 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

MyFileNotFoundException) >>> Warning) 동일한 이름의 파일이 이미 존재합니다.  
 >>> 현존하는 파일에 새로운 내용으로 overwrite하시겠습니까 ? Y or N : Y

o-1) open\_mode() / file\_name : my\_new\_notes.txt, mode : w  
 o-4) my\_new\_notes.txt를 w 모드로 open

wc) 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()  
 >> 1 번째 줄 내용 : 새로운 파일이 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.

```

98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137

```

```

1) main
i-1) file_name : my_new_notes.txt
r-1) read_file()
>> 1 번째 줄 내용 : 흥길동 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.txt를 a 모드로 open
a-4) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>을 append 모드로 open
c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'>을 close
r-1) read_file()
>> 1 번째 줄 내용 : 흥길동 010-1234-5678
>> 2 번째 줄 내용 : 김철수 010-1234-5679
>> 3 번째 줄 내용 : 김영희 010-1234-5680
>> 4 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
MyFileNotFoundException) >>> Warning) 동일한 이름의 파일이 이미 존재합니다.
>>> 현존하는 파일에 새로운 내용으로 overwrite하시겠습니까 ? Y or N : Y
o-1) open_mode() / file_name : my_new_notes.txt, mode : w
o-4) my_new_notes.txt를 w 모드로 open
wc) 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()
>> 1 번째 줄 내용 : 새로운 파일이 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.

```

```

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

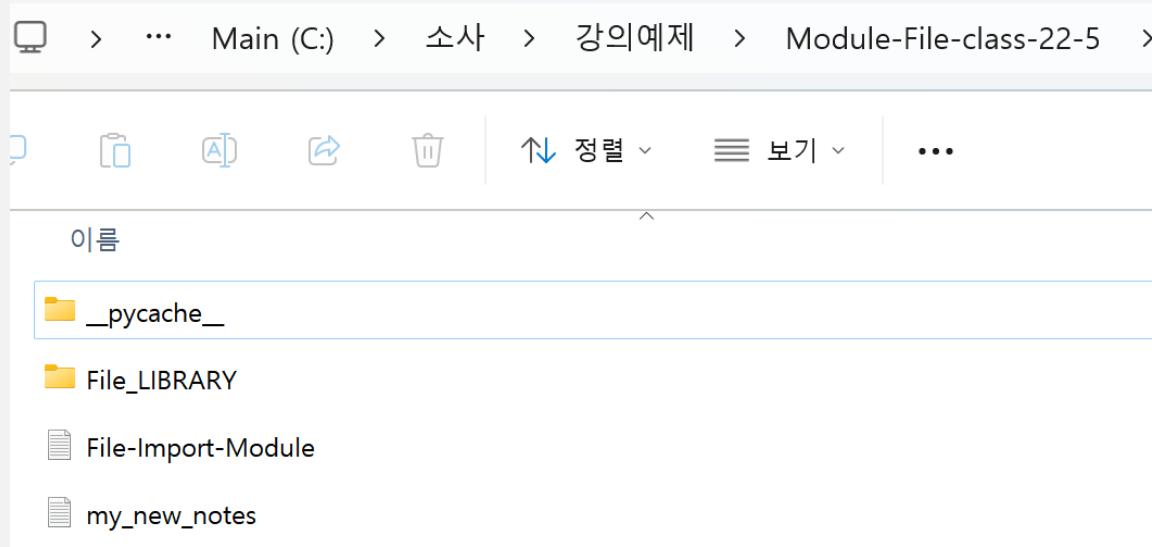
```

```

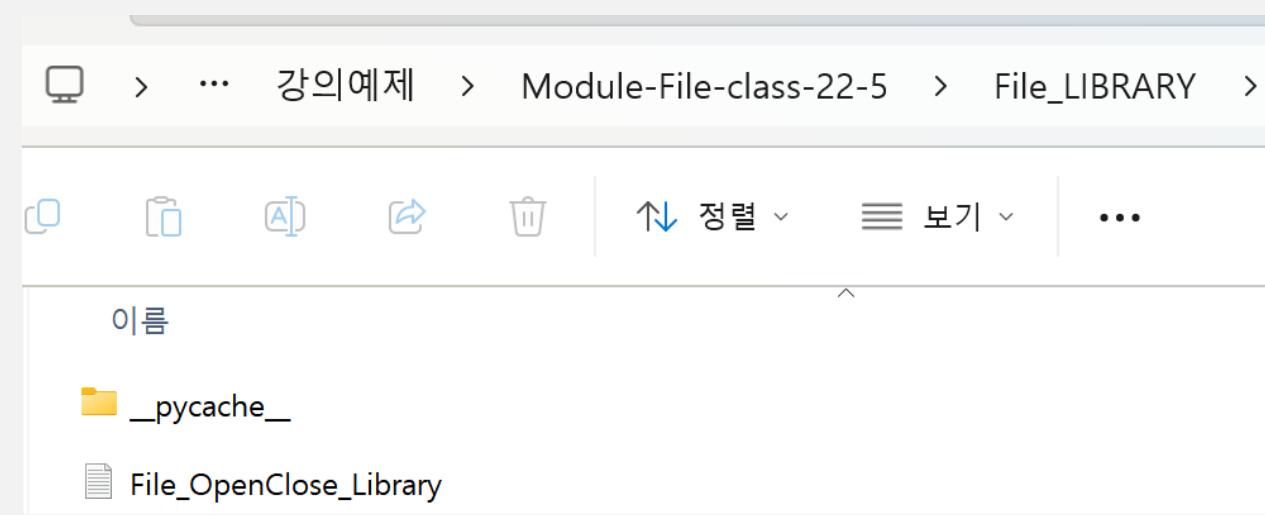
1) main
i-1) file_name : my_new_notes.txt
r-1) read_file()
>> 1 번째 줄 내용 : 홍길동 010-1234-5678
>> 2 번째 줄 내용 : 김철수 010-1234-5679
>> 3 번째 줄 내용 : 김영희 010-1234-5680
a-1) append()
a-2) file name : my_new_notes.txt
a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사
o-1) open_mode() / file_name : my_new_notes.txt, mode : a
o-3) my_new_notes.txt를 a 모드로 open
a-4) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'
'을 append 모드로 open
c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'
'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 : 19시 가족과 함께 저녁식사
w-1) write_file()
w-2) file name : my_new_notes.txt
w-3) User-defined Exception Handler 호출
MyFileNotFoundException) >>> Warning) 동일한 이름의 파일이 이미 존재합니다.
>>> 현존하는 파일에 새로운 내용으로 overwrite하시겠습니까 ? Y or N : Y
o-1) open_mode() / file_name : my_new_notes.txt, mode : w
o-4) my_new_notes.txt를 w 모드로 open
wc) 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()
>> 1 번째 줄 내용 : 새로운 파일이 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:/소사/강의예제/Module-File-class-22-5



C:/소사/강의예제/Module-File-class-22-5/File\_LIBRARY

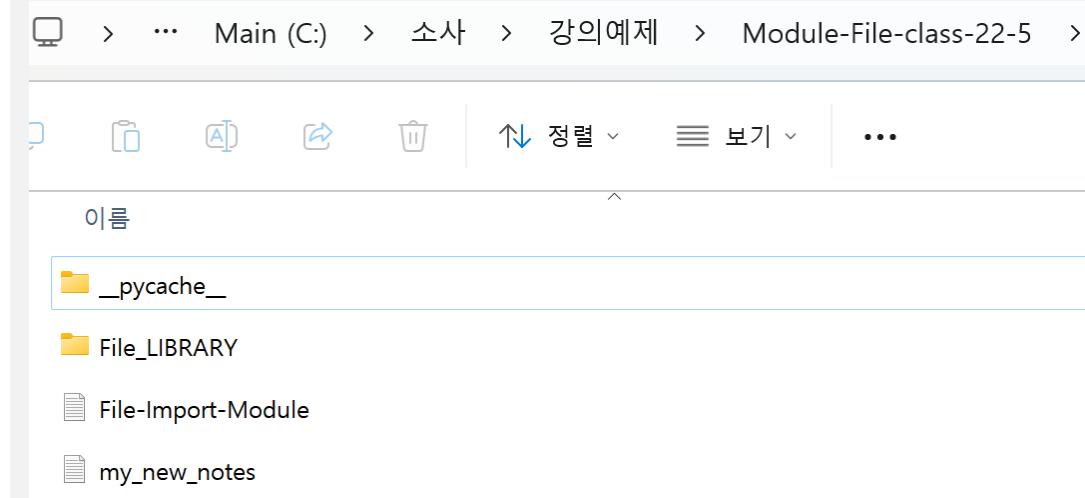
File Edit Format Run Options Window Help

```

1 D = True
2 #D = False
3
4 import sys
5 sys.path.append("C:/소사/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 if __name__ == "__main__":
26
27     if D:
28         print("Wn1) main")
29
30     exe_main = Main()
31     exe_main.main()
32
33     if D:
34         print("Wn2) main End.")
35
36

```

## Module 사용



C:/소사/강의예제/Module-File-class-22-5

```

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 ! ".W
18                  format(file_name))
19
20    def open_mode(self, mode):
21
22        try:
23            if D:
24                print("Wno-1) open_mode() / file_name : {}, mode : {} ".W
25                  format(self.f_name, mode))
26
27            if mode == "r":
28                if D:
29                    print ("o-2) {}을 r mode로 open".format(self.f_name))
30                    self.my_file = open(self.f_name, 'r', encoding = 'cp949')
31
32            elif mode == "a":
33                if D:
34                    print ("o-3) {}을 a 모드로 open".format(self.f_name))
35                    self.my_file = open(self.f_name, 'a', encoding = 'cp949')
36
37            elif mode == "w":
38                if D:
39                    print ("o-4) {}을 w 모드로 open".format(self.f_name))
40                    self.my_file = open(self.f_name, 'w', encoding = 'cp949')
41
42        except Exception as e:
43            print("Wn>> Warning in open_mode() : ", e)

```

```

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

```

# Module 사용

# Module 사용

```
89 def append(self):
90     try:
91         if D:
92             print("Wn a-1) append()")
93             print("a-2) file name : {}".format(self.f_name))
94             print("a-3) 새로운 일정 추가 : 19시 가족과 함께 저녁식사")
95
96     #     input_line = input("> 새로운 일정을 추가하세요 : ")
97     #     self.my_file.write(input_line)
98     #     self.open_mode("a")
99
100    if D:
101        print("Wn a-4) {}을 append 모드로 open".format(self.my_file))
102
103    self.my_file.write('Wn19시 가족과 함께 저녁식사')
104    self.close_file()
105
106 except Exception as e:
107     print("Wn>> Warning in append()", e)
108
109
110 def write_file(self):
111     if D:
112         print("Wn w-1) write_file()")
113         print("w-2) file name : {}".format(self.f_name))
114
115     try:
116         if os.path.isfile(self.f_name):
117             if D:
118                 print("w-3) User-defined Exception Handler 호출")
119
120             raise MyFileNotFoundException("Wn>> Warning) 동일한 이름의 파일이 이미 존재합니다.")
121
122         else:
123             self.open_mode("w")
124             self.write_contents()
125
126     except MyFileNotFoundException as e:
127         print(e)
128         answer = input("Wn>> 현존하는 파일에 새로운 내용으로 overwrite하시겠습니까 ? Y or N : ")
129
130
131         if answer == "Y":
132             self.open_mode("w")
133             self.write_contents()
134
135         else:
136             pass
137
```

```
1) main
i-1) file_name : my_new_notes.txt
r-1) read_file()
>> 1 번째 줄 내용 : 새로운 파일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다
>> 4 번째 줄 내용 : 19시 가족과 함께 저녁식사
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.txt를 a 모드로 open
a-4) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'
'>를 append 모드로 open
c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'
'>를 close
r-1) read_file()
>> 1 번째 줄 내용 : 새로운 파일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다
>> 4 번째 줄 내용 : 19시 가족과 함께 저녁식사
>> 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
MyFileNotFoundException) >>> Warning) 동일한 이름의 파일이 이미 존재합니다.
>>> Warning) 동일한 이름의 파일이 이미 존재합니다.
>>> 현존하는 파일에 새로운 내용으로 overwrite하시겠습니까 ? Y or N : Y
o-1) open_mode() / file_name : my_new_notes.txt, mode : w
o-4) my_new_notes.txt를 w 모드로 open
wc) w 모드로 새로운 내용쓰기
c-1) close_file()
c-2) <_io.TextIOWrapper name='my_new_notes.txt' mode='w' encoding='cp949'
'>를 close
r-1) read_file()
>> 1 번째 줄 내용 : 새로운 파일이 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'
'>를 close
2) main End.
```

# Module 사용

```
138 def write_contents(self):
139     if D:
140         print("Wnwc) w 모드로 새로운 내용쓰기")
141
142         self.my_file.write("새로운 파일이 생성되었습니다Wn")
143         self.my_file.write("모두 반갑습니다Wn")
144         self.my_file.write("열심히 공부합니다")
145
146         self.close_file()
147
148     def __del__(self):
149         try:
150             if D:
151                 print ("Wnd-1) Destructor에서 {}을 close".format(self.my_file))
152             self.my_file.close()
153
154         except Exception as e:
155             print("Wn>> Warning in 소멸자) ", e)
156
157 class MyFileNotFoundException(Exception):
158
159     def __init__(self, msg):
160
161         self.msg = msg
162
163         if D:
164             print("WnMyFileNotFoundException) ", self.msg)
```

```
1) main
i-1) file_name : my_new_notes.txt
r-1) read_file()
>> 1 번째 줄 내용 : 새로운 파일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다
>> 4 번째 줄 내용 : 19시 가족과 함께 저녁식사

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.txt를 a 모드로 open

a-4) <_io.TextIOWrapper name='my_new_notes.txt' mode='a' encoding='cp949'
'>를 append 모드로 open

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

r-1) read_file()
>> 1 번째 줄 내용 : 새로운 파일이 생성되었습니다
>> 2 번째 줄 내용 : 모두 반갑습니다
>> 3 번째 줄 내용 : 열심히 공부합니다
>> 4 번째 줄 내용 : 19시 가족과 함께 저녁식사
>> 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

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

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

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

wc) w 모드로 새로운 내용쓰기

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

r-1) read_file()
>> 1 번째 줄 내용 : 새로운 파일이 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' e
ncoding='cp949'>를 close

2) main End.
```

# 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)

| A         | B  | C | D       |
|-----------|----|---|---------|
| 1/2/2029  | 5  | 8 | red     |
| 1/3/2029  | 5  | 2 | green   |
| 1/3/2029  | 9  | 1 | blue    |
| 1/3/2030  | 10 | 2 | yellow  |
| 1/12/2031 | 11 | 4 | pink    |
| 1/12/2031 | 4  | 1 | skyblue |

# CSV(Comma Separated Value) 처리



File Edit Format Run Options Window Help

```
1 # 파일을 연다.
2 f = open("data.csv", "r", encoding = 'cp949')
3
4 # 파일 안의 각 줄을 처리한다.
5 for line in f.readlines():
6     line = line.strip()          # 공백 문자를 없앰
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       |
|-----------|----|---|---------|
| 1/2/2029  | 5  | 8 | red     |
| 1/3/2029  | 5  | 2 | green   |
| 1/3/2029  | 9  | 1 | blue    |
| 1/3/2030  | 10 | 2 | yellow  |
| 1/12/2031 | 11 | 4 | pink    |
| 1/12/2031 | 4  | 1 | skyblue |

```
1/2/2029,5,8,red
1/2/2029
5
8
red
1/3/2029,5,2,green
1/3/2029
5
2
green
1/3/2029,9,1,blue
1/3/2029
9
1
blue
1/3/2030,10,2,yellow
1/3/2030
10
2
yellow
1/12/2031,11,4,pink
1/12/2031
11
4
pink
1/12/2031,4,1,skyblue
1/12/2031
4
1
skyblue
```

# Reading a file

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

# Text files and binary files

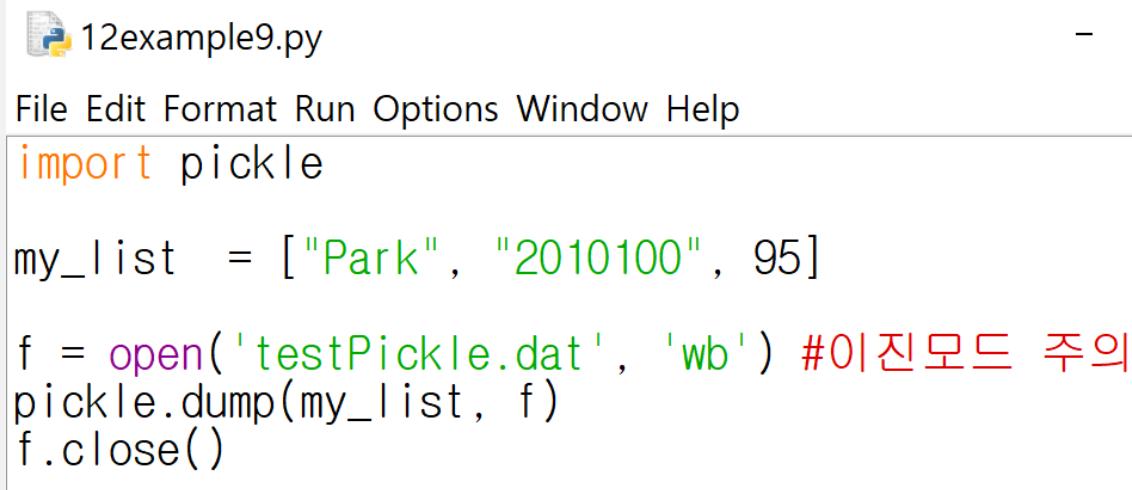
- Text file
  - ASCII code 또는 UNICODE 만으로 구성된 file
  - open() 함수 텍스트 모드로 작업
    - 예: “rt”, “r” → 기본 모드
  - 글자, 숫자, 개행문자 등으로 구성
  - Windows notepad.exe로 내용을 읽을 수 있음
  - LINUX cat 명령어로 내용을 읽을 수 있음
- Binary file
  - 그림, 동영상, 음악 등 텍스트 이외의 정보 저장
  - open() 함수 text binary mode로 작업
    - 예: “rb”
- hwp file은 어떤 file인가?

# Pickle

- 피클(pickle)
  - 다양한 data 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 data 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

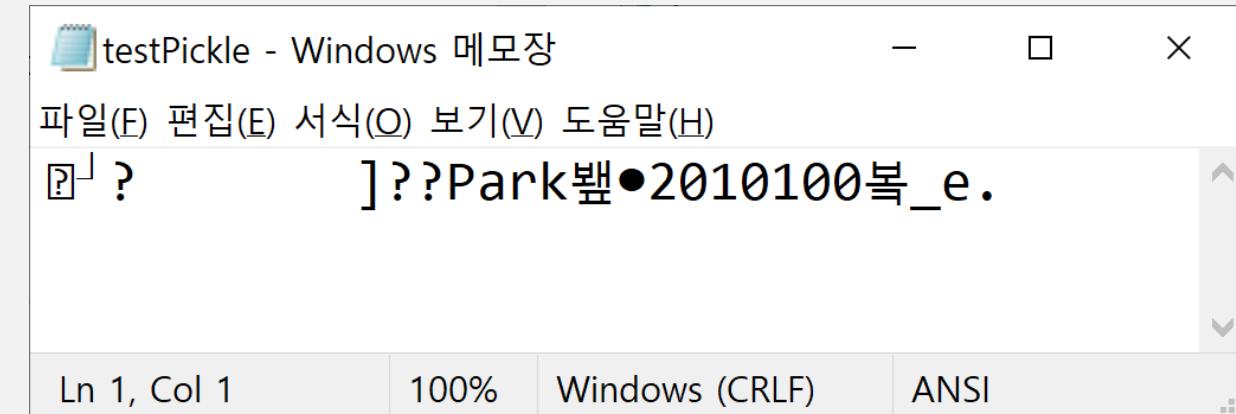
- 피클링 (pickling)
  - dump() 함수 사용
  - file을 이진모드로 열어야 함



```
12example9.py
File Edit Format Run Options Window Help
import pickle

my_list = ["Park", "2010100", 95]

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



# Pickle

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

12example9\_class.py

```
File Edit Format Run Options Window Help
import pickle

class Pickle1:
    def __init__(self):
        self.my_list = ["Park", "2010100", 95]

    def pickle_dump(self):
        f = open('testPickle.dat', 'wb') #이진모드 주의
        pickle.dump(self.my_list, f)
        f.close()

class Main:
    def __init__(self):
        pass

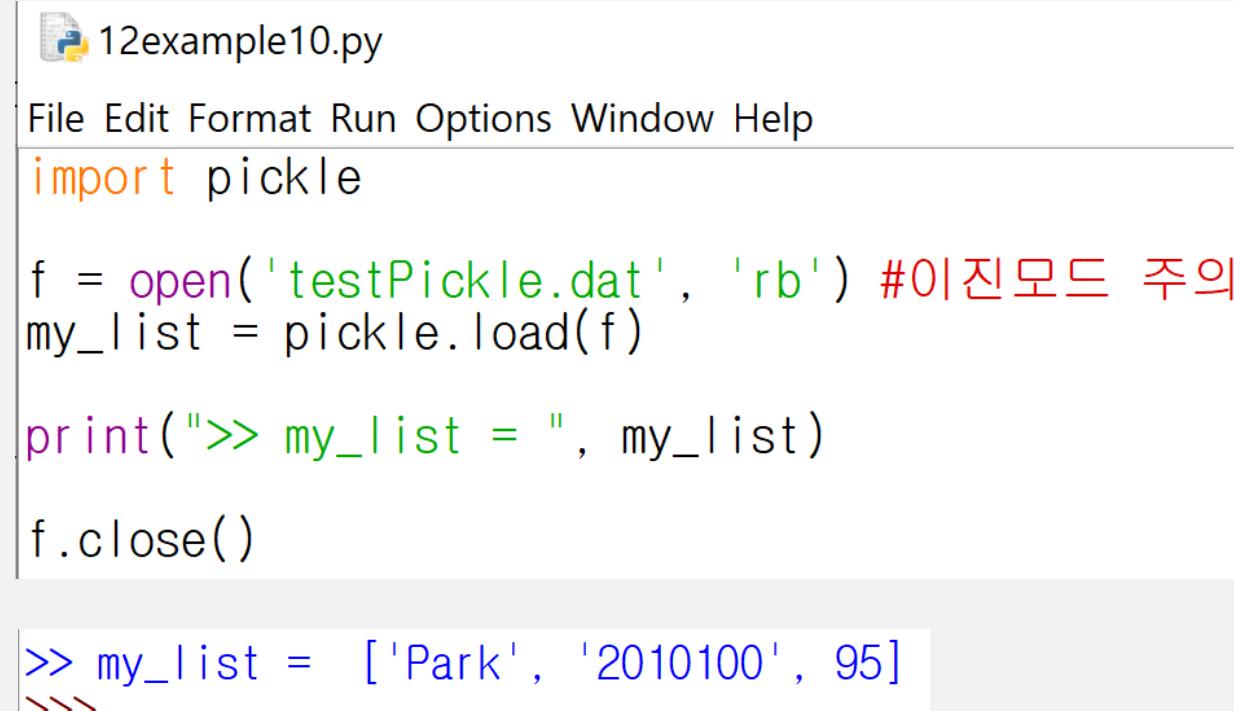
    def main(self):
        pick = Pickle1()
        pick.pickle_dump()

#main
if __name__ == "__main__":
    try:
        pic_main = Main()
        pic_main.main()

    except Exception as e:
        print("\n>> Warning : ", e)
```

# Pickle

- 언피클링(unpickling)
  - load() 함수 사용
  - file을 이진모드로 열어야 함



```
12example10.py
File Edit Format Run Options Window Help
import pickle

f = open('testPickle.dat', 'rb') #이진모드 주의
my_list = pickle.load(f)

print(">> my_list = ", my_list)

f.close()

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

# Pickle

- 피클링(pickling): 객체를 file로 저장
  - `dump()` 함수 사용 (`write()` 대신)
- 언피클링(unpickling): file로부터 읽어드린 내용을 객체로 변환
  - `load()` 함수 사용 (`read()` 대신)

12example10\_class.py

```
File Edit Format Run Options Window Help
import pickle

class Pickle1:

    def __init__(self):
        self.my_list = ["Park", "2010100", 95]

    def pickle_dump(self):
        f = open('testPickle.dat', 'wb') #이진모드 주의
        pickle.dump(self.my_list, f)
        f.close()

    def unpickle_load(self):
        f = open('testPickle.dat', 'rb') #이진모드 주의
        loaded_list = pickle.load(f)
        print(">> my_list = ", loaded_list)
        f.close()

class Main:
    def __init__(self):
        pass

    def main(self):
        pick = Pickle1()
        pick.pickle_dump()
        pick.unpickle_load()

#main
if __name__ == "__main__":
    try:
        pic_main = Main()
        pic_main.main()
    except Exception as e:
        print("\n>> Warning : ", e)

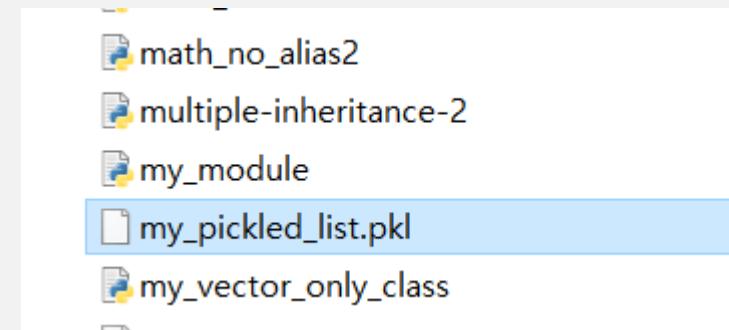
    >> my_list = ['Park', '2010100', 95]
    >>>
```

```

1 import pickle
2
3 print("Wn1) pickling 시작")
4
5 my_list = ['Fred', 73, 'Hello there', 81.9876e-13]
6 print("2) my_list = ", my_list)
7
8 print("3) wb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성")
9 pickle_file = open('my_pickled_list.pkl', 'wb')
10
11 print("4) dump")
12 pickle.dump(my_list, pickle_file)
13 print("5) pickle_file = ", pickle_file)
14
15 print("6) close pickl_file")
16 pickle_file.close()
17
18 print("7) unpickling 시작")
19 print("8) rb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성")
20 pickle_file = open('my_pickled_list.pkl', 'rb')
21
22 print("9) load")
23 unpickled_list = pickle.load(pickle_file)
24
25 print("10) close pickl_file")
26 pickle_file.close()
27
28 print ("11) 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]



```

1 import pickle
2
3 D = True
#D = False
4
5 class Pickle1:
6
7     def __init__(self):
8         self.my_list = ['Fred', 73, 'Hello there', 81.9876e-13]
9         if D:
10             print("i) self.my_list = ", self.my_list)
11
12     def pickle_dump(self):
13         if D:
14             print("p1) wb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성")
15             pickle_file = open('my_pickled_list.pkl', 'wb') #이진모드 주의
16
17         if D:
18             print("p2) dump")
19             pickle.dump(self.my_list, pickle_file)
20
21         if D:
22             print("p3) close pickl_file")
23             pickle_file.close()
24
25     def unpickle_load(self):
26         if D:
27             print("u1) unpickling 시작")
28             print("u2) rb 모드로 my_pickled_list.pkl 오픈 후 pickle_file 객체생성")
29             pickle_file = open('my_pickled_list.pkl', 'rb')
30
31         if D:
32             print("u3) load")
33             unpickled_list = pickle.load(pickle_file)
34
35         if D:
36             print("u4) close pickl_file")
37             pickle_file.close()
38
39
40         print (">> unpickled_list = ", unpickled_list)
41

```

```

x42 class Main:
43     def __init__(self):
44         pass
45
46     def main(self):
47         if D:
48             print("Wn1) 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("Wn>> Warning : ", e)
62

```

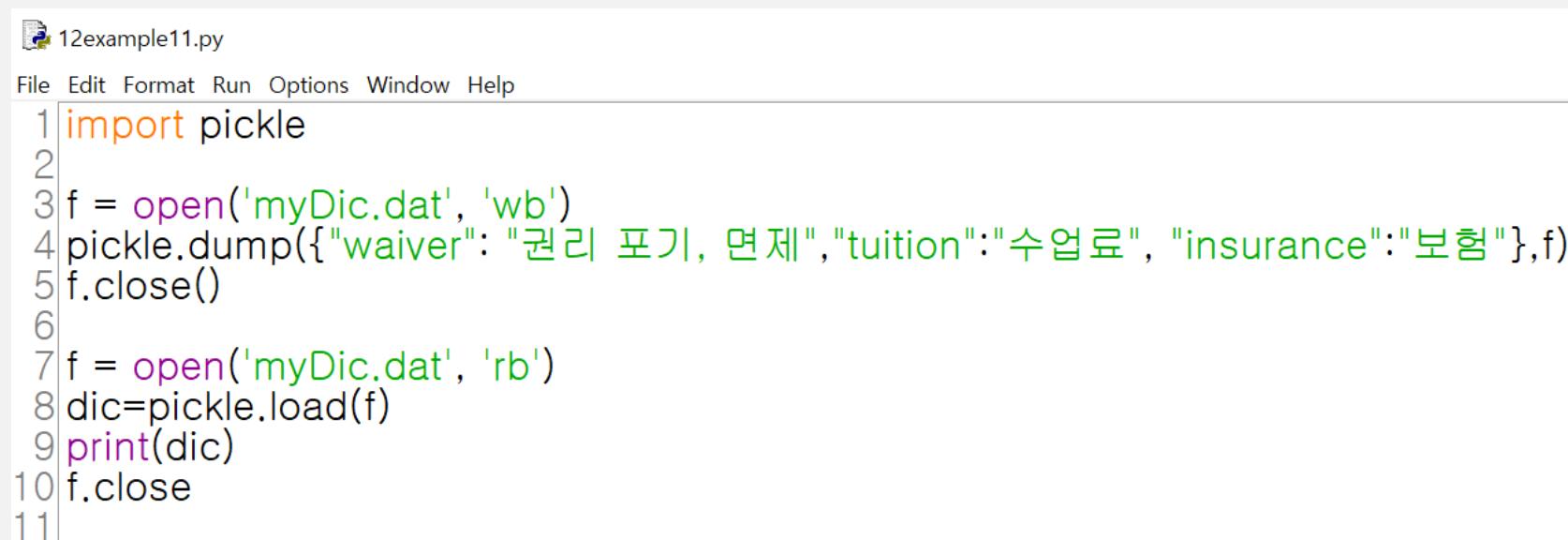
1) pickling 시작

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]

# 실습

- Pickling을 이용해서 Dictionary Data type을 file로 저장하고 읽는 program을 작성 하시오.
  - 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 program을 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)
      - data의 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    |
| 휴斯顿    | 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.
    - file 객체.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 halfhid 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 1 100% Windows (CRLF) UTF-8

# 숙제

- 영어 사전

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

| vocabulary - Windows 메모장 |             |               |    |            |                 |             |
|--------------------------|-------------|---------------|----|------------|-----------------|-------------|
|                          |             |               |    |            |                 |             |
| 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

- 숙제 1 :
  - 수업자료에서 제시한 Pickling을 이용한 Dictionary Data type을 file로 저장하고 읽는 program을 class로 재작성 하시오.
    - dictionary data
- {"waiver": "권리 포기, 면제", "tuition": "수업료", "insurance": "보험"}

# 숙제

- File Input and Output
  - 숙제 2  
레 미제라블 텍스트 file("Les\_Miserables-Victor\_Hugo.txt")을 읽어 들여 등장하는 단어들의 횟수를 모두 구하고 zipf's law와의 관련성을 분석하시오.  
file 이름 : LesMiserables-이름-학번.py
  - 숙제 3  
"vocabulary.txt"는 인터넷에서 다운받은 영단어 텍스트 file인데 내용이 뒤섞여 있다. file 내용을 이용해서 "단어:의미"로 구성된 사전을 구현하시오.  
file 이름 : vocabulary-이름-학번.py
- Les\_Miserables-Victor\_Hugo.txt ,vocabulary.txt 파일은 수업자료를 업로드하는 공지에 올려놓았습니다.
- 마감일 : 자율적으로 해보세요. 제출하지 않아도 됩니다.