

```
1 Lab. 일반 text file 다루기
2
3 1. open() / read()
4   f = open('stockcode.txt', 'r')
5   data = f.read() # 모든 내용을 한꺼번에 읽기
6   print(data)
7   f.close()
8
9
10 2. open() / readline()
11   f = open('stockcode.txt', 'r')
12   line_num = 1
13   line = f.readline() # text file을 줄 단위로 읽기
14   while line:
15       print('%d %s' %(line_num, line), end='')
16       line = f.readline()
17       line_num += 1
18   f.close()
19
20
21 3. open() / readlines()
22   f = open('stockcode.txt', 'r')
23   lines = f.readlines() # text file을 한 줄씩 읽어서 각 줄을 요소로 하는 list 반환
24   # read()와 마찬가지로 모든 내용을 한꺼번에 읽기 때문에 파일 크기가 매우 크면 메모리 문제 발생 가능.
25   # print(lines)
26   for line_num, line in enumerate(lines):
27       print('%d %s' %(line_num+1, line), end='')
28   f.close()
29
30
31 4. with ~ as : 파일 열고 자동으로 닫기
32   with open('stockcode.txt', 'r') as f:
33       for line_num, line in enumerate(f.readlines()):
34           print('%d %s' %(line_num+1, line), end='')
35
36
37 5. write()
38   text = input('파일에 저장할 내용을 입력하세요: ')
39   f = open('mydata.txt', 'w')
40   f.write(text) # text를 file로 저장
41   f.close()
42
43
44 6. writelines()
45   count = 1
46   data = []
47   print('파일저장을 끝내려면 내용을 입력하지 말고 [Enter]를 누르세요')
48   while True:
49       text = input('%d 파일에 저장할 내용을 입력하세요: ' %count)
50       if text == '':
51           break
```

```
52     data.append(text+'\n')
53     count += 1
54
55     f = open('mydata.txt', 'w')
56     f.writelines(data)    #file에 한 줄씩 쓰기
57     f.close()
58
59
60 7. binary file open() / read() / write()
61     bufsize = 1024
62     f = open('img_sample.jpg', 'rb')
63     h = open('img_sample_copy.jpg', 'wb')
64
65     data = f.read(bufsize)
66     while data:
67         h.write(data)
68         data = f.read(bufsize)
69
70     print("File Copy Successfully.")
71     f.close()
72     h.close()
73
74
75 8. seek() /
76     spos = 105    # 파일을 읽는 위치 지정
77     size = 500    # 읽을 크기를 지정
78
79     f = open('stockcode.txt', 'r')
80     h = open('stockcode_part.txt', 'w')
81
82     f.seek(spos)    #file의 특정 위치로 이동
83     data = f.read(size)
84     h.write(data)
85     print("File contents copied successfully.")
86     h.close()
87     f.close()
88
89
90 9. file 크기 구하기
91     from os.path import getsize
92
93     file1 = 'stockcode.txt'
94     file2 = 'img_sample.jpg'
95     file_size1 = getsize(file1)    #file 크기 구하기
96     file_size2 = getsize(file2)
97
98     print('File Name: %s \tFile Size: %d' %(file1, file_size1))
99     print('File Name: %s \tFile Size: %d' %(file2, file_size2))
100
101
102 10. directory list
```

```
103 import os, glob
104
105 folder = 'C:/PythonHome'
106 file_list = os.listdir(folder)
107 print(file_list)
108
109 files = '*.txt'
110 file_list = glob.glob(files) #인자로 입력된 조건이나 경로에 해당하는 file들을 list로 반환
111 # glob()는 OS의 wildcard character(*) 사용 가능
112 print(file_list)
113
114
115 11. shutil.rmtree()
116 import shutil
117 import os
118
119 target_folder = 'C:/PythonHome/tmp'
120 print('[%s] 하위 모든 디렉터리 및 파일들을 삭제합니다.' %target_folder)
121 for file in os.listdir(target_folder):
122     print(file)
123 k = input('[%s]를 삭제하겠습니까? (y/n) ' %target_folder)
124 if k == 'y':
125     try:
126         shutil.rmtree(target_folder)
127         #인자로 입력된 경로에 해당하는 directory와 하위 directory 및 모든 file들을 일괄적으로 삭제
128         print('[%s]의 모든 하위 디렉터리와 파일들을 삭제했습니다.' %target_folder)
129     except Exception as e:
130         print(e)
131
132
133 12. os.path.exists()
134 import os
135 from os.path import exists
136
137 dir_name = input('새로 생성할 디렉터리 이름을 입력하세요: ')
138 if not exists(dir_name):
139     os.mkdir(dir_name)
140     print('[%s] 디렉터리를 생성했습니다.' %dir_name)
141 else:
142     print('[%s]은(는) 이미 존재합니다.' %dir_name)
143
144
145 13. os.path.isfile() / os.path.isdir()
146 import os
147 from os.path import exists, isdir, isfile
148
149 files = os.listdir()
150 for file in files:
151     if isdir(file):
152         print('DIR: %s' %file)
153
```

```
154     for file in files:
155         if isfile(file):
156             print('FILE: %s' %file)
```