

파일 다루기

이번 장에서는 파이썬에서 파일 생성, 읽기, 쓰기 등 파일 다루기에 대하여 공부합니다.



1. 파일
2. 파일 다루는 절차
3. 파일 열기
4. 파일 다루기 예
5. 다른 함수들
6. 파일 관리 함수





1. 파일

□ 파일(file)

- 정보를 저장하는 기본 단위

□ 파일의 종류

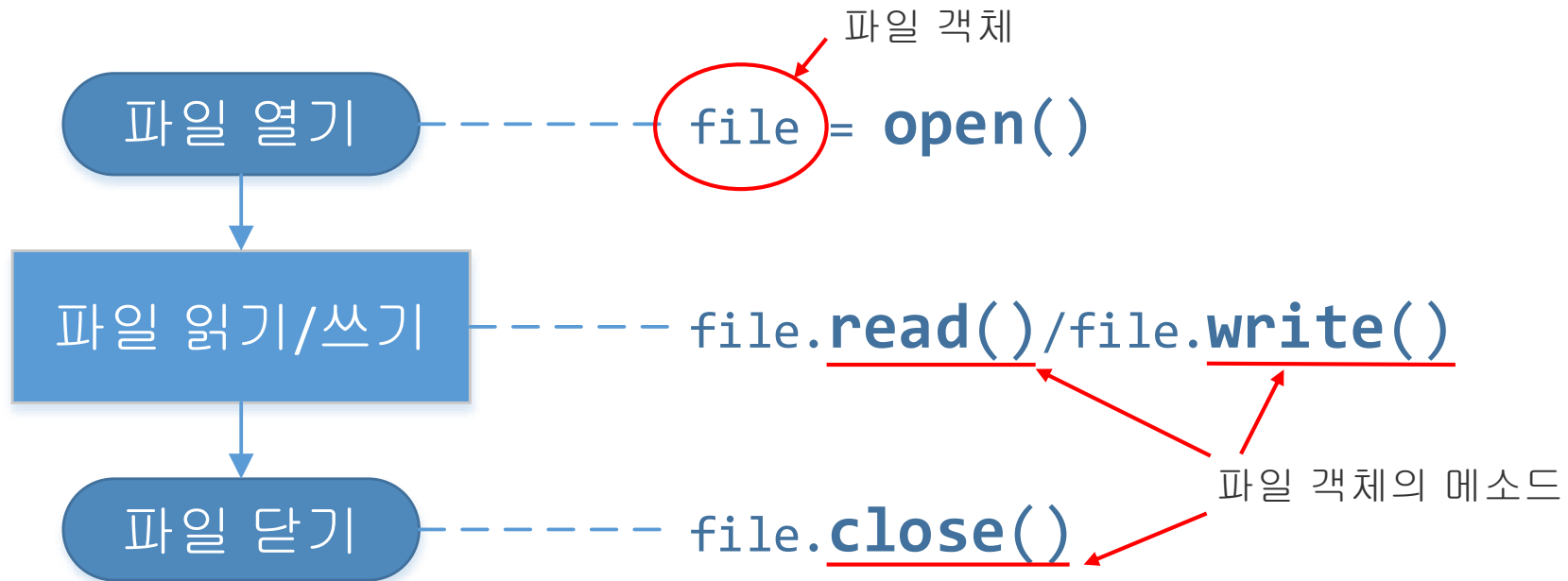
바이너리 파일	텍스트 파일
<ul style="list-style-type: none">• 컴퓨터만 이해할 수 있는 형태인 이진(법) 형식으로 저장된 파일• 일반적으로 메모장으로 열면 내용이 깨져 보임(메모장에서 해석 불가)• 엑셀 파일, 워드 파일 등	<ul style="list-style-type: none">• 사람도 이해할 수 있는 형태인 문자열 형식으로 저장된 파일• 메모장으로 열면 내용 확인이 가능• 메모장에 저장된 파일, HTML 파일, 파이썬 코드 파일 등



2. 파일 다루는 절차

□ 파일 다루는 절차

- open -> read / write -> close





3. 파일 열기

□ 파일 열기

```
f = open( "파일명" , "파일열기모드" ,  
          encoding= 'utf-8(한글로 인코딩)' )
```

▪ 파일 열기 모드

모드	의미
'r'	읽기용으로 열기 (기본값)
'w'	쓰기용으로 열기. 이미 같은 경로에 파일이 존재하면 파일내용을 비움.
'x'	배타적 생성모드로 열기. 파일이 존재하면 IOError 예외 일으킴.
'a'	쓰기용으로 열기. 단, 'w'와는 달리 이미 같은 파일이 존재하는 경우 기존 내용에 덧붙이기를 함.
'b'	바이너리 모드
't'	텍스트 모드(기본값)
'+'	읽기/쓰기용으로 파일 읽기



3. 파일 열기

□ 파일 열기

- with ~ as 를 사용하면 명시적으로 close() 함수를 호출하지 않아도 블록이 끝날 때 파일이 닫힘

with open(“파일명” , “파일열기모드”) **as** 파일객체 :
읽기 / 쓰기 명령

블록이 끝날 때,
파일이 닫힘



4. 파일 다루기 예

□ 예

```
f = open( "hello.txt" , "w" )  
f.write( 'Hello World\n' )  
f.write( 'I love you\n' )  
f.close()
```

□ 예

```
f = open( "hello.txt" , "r" )  
contents = f.read()      # 파일의 끝까지 읽음  
print(contents)  
f.close()
```



4. 파일 다루기 예

□ 예

```
with open( "hello.txt" , "r" ) as f :  
    contents = f.read()  
    print(contents)
```

블록이 끝날 때,
f.close()를 호출하지 않아도
파일이 닫힘



4. 파일 다루기 예

□ 추가 쓰기(append) 모드 a

- 기존 내용의 뒤에 내용을 추가함

```
f = open( "hello.txt" , "a" )  
f.write( 'WnYou love meWn' )  
f.close()
```





4. 파일 다루기 예

□ 예

```
try :  
    f = open("live.txt", "rt")  
    text = f.read()  
    print(text)  
except FileNotFoundError :  
    print("파일이 없습니다.")  
finally :  
    f.close()
```



5. 다른 함수들

❑ readline()

- 한 줄씩 읽기
- 파일 마지막에서는 빈 문자열을 반환함

```
f= open("hello.txt", "r")
text = ""
line = 1
while True :
    row = f.readline()      # 파일을 한줄씩 읽음
    if not row:             # 파일의 마지막일때 row 값은 None
        break
    text += str(line) + " : " + row
    line += 1
f.close()
print(text)
```



5. 다른 함수들

□ writeline()

- 한 줄씩 쓰기

```
lines = ["we'll find a way we always have - Interstellar\n",  
        "I'll find you and I'll kill you - Taken\n",  
        "I'll be back - Terminator 2\n"]
```

```
with open('movie_quotes.txt', 'w') as file:  
    file.writelines(lines)
```

movie_quotes.txt 파일을 읽어보는 프로그램을 작성해보시오.



5. 다른 함수들

❑ movie_quotes.txt 파일을 읽어보는 프로그램

```
with open('movie_quotes.txt', 'r') as file:
    line = file.readline()

    while line != '':
        print(line, end='')
        line = file.readline()
```



6. 파일 관리 함수들

□ 파일 관리 함수는 파일 자체를 다루는 함수로서, `shutil` 모듈과 `os` 모듈에 정의되어 있음

□ 예

함수	설명
<code>shutil.copy(a, b)</code>	파일을 복사한다.
<code>shutil.copytree(a, b)</code>	디렉토리를 복사한다. 서브 디렉토리까지 전부 복사한다.
<code>shutil.move(a, b)</code>	파일을 이동한다.
<code>shutil.rmtree(path)</code>	디렉토리를 삭제한다.
<code>os.rename(a, b)</code>	파일 이름을 변경한다.
<code>os.remove(f)</code>	파일을 삭제한다.
<code>os.chmod(f, m)</code>	파일의 퍼미션을 변경한다.
<code>shutil.chown(f, u, g)</code>	파일의 소유권을 변경한다.
<code>os.link(a, b)</code>	하드 링크를 생성한다.
<code>os.symlink(a, b)</code>	심볼릭 링크를 생성한다.



6. 파일 관리 함수들

□ 예

- 파일을 복사

```
import shutil  
  
shutil.copy("hello.txt", "hello2.txt")
```

□ 예

- 디렉토리 만들기

```
import os  
  
os.mkdir("log")
```



6. 파일 관리 함수들

□ 예

- 디렉토리에 있는 파일 목록 구하기

```
import os  
  
folder = '.'  
file_list = os.listdir(folder)  
print(file_list)
```




```
import os
import random, datetime

if not os.path.isdir("log") :
    os.mkdir("log")

if not os.path.exists("log/count_log.txt") :
    f = open("log/count_log.txt", "w")
    f.write("기록 시작\n")
    f.close()

with open("log/count_log.txt", "a") as f :
    for i in range(1, 11) :
        stamp = str(datetime.datetime.now())
        value = random.random() * 1000000
        log_line = stamp + "\t" + str(value) + "값이 생성됨" + "\n"
        f.write(log_line)
```



```
with open("log/count_log.txt", "r") as f :  
    text = f.read()  
    word_list = text.split(" ")    # 빈 칸 기준으로 단어를 분리  
    line_list = text.split("\n")   # 한 줄씩 분리  
  
print("총 글자의 수: ", len(text))  
print("총 단어의 수: ", len(word_list))  
print("총 줄의 수: ", len(line_list))
```



```
file_name = input("파일 이름을 입력하시오: ")  
word_name = input("찾을 단어를 입력하시오: ")
```

```
with open(file_name, "r") as f :  
    text = f.read()  
    text = text.lower()  
    pos = text.find(word_name)  
    count = 0  
    while pos != -1 :  
        count += 1  
        pos = text.find(word_name, pos+1)  
  
print("개수는", count)
```



```
import zipfile

def compressZip(zipname, filename) :
    print("[%s] -> [%s] 압축...." %(filename, zipname))
    with zipfile.ZipFile(zipname, 'w') as ziph :
        ziph.write(filename)

    print("압축이 끝났습니다")

filename = input("파일이름을 입력하시오: ")
zipname = filename + '.zip'
compressZip(zipname, filename)
```



- 파일에서 특정 단어를 다른 단어로 교체하는 프로그램을 작성하시오.
예를 들어, "abc"를 "111"로 대체시킴.
파일 이름과 단어 2개는 키보드로 입력받는다.

- Zip 파일의 압축을 푸는 프로그램을 작성하시오.
(hint) p.20 프로그램의 6번째줄에서 `ziph.write()` 대신에 `ziph.extractall()` 사용

- 파일에서 알파벳의 개수를 구하는 프로그램을 작성하시오.
파일 이름은 키보드로 입력받는다.