

파일

파일에 문자열 쓰기

형식

```
file = open('hello.txt', 'w') # hello.txt 파일을 쓰기 모드(w)로 열기. 파일 객체 반환
file.write('Hello, world!')   # 파일에 문자열 저장
file.close()                  # 파일 객체 닫기
```

- open() 으로 파일을 열어서 파일 객체를 얻는다.
- 파일모드 'w'는 쓰다(write)의 의미를 가진다.
- write() 를 통해 문자열을 입력한다.

파일에서 문자열 읽기

형식

```
file = open('hello.txt', 'r') # hello.txt 파일을 읽기 모드(r)로 열기. 파일 객체 반환
s = file.read()               # 파일에서 문자열 읽기
print(s)                      # Hello, world!
file.close()                  # 파일 객체 닫기
```

- 파일을 읽을 때도 open 함수로 파일을 열어서 파일 객체를 얻는다.
- read() 로 파일의 내용을 읽는다.
- 단, 이때는 파일 모드를 읽기 모드 'r'(read)로 지정한다.

기존파일에 문자열 추가하기

형식

```
file = open('hello.txt', 'a')
s = file.write('add new line..!!')
file.close()
```

- 'a'는 추가모드로 파일의 마지막에 새로운 내용을 추가할때 사용한다.

자동으로 파일 객체 닫기

형식

```
with open(파일이름, 파일모드) as 파일객체:  
    코드
```

- 파일을 `open()` 할때마다 매번 `close()` 로 닫는것이 번거로울때 `with as`를 사용한다.
- `close()` 없이도 자동으로 파일객체를 닫아준다.

writelines() 및 readline() 메소드

형식1

```
lists = ['리스트1\n', '리스트2\n', '리스트3\n']  
  
with open('hello.txt', 'w') as myfile: # hello.txt 파일을 쓰기 모드(w)로 열기  
    myfile.writelines(lists)
```

- `writelines()` 는 리스트에 들어있는 문자열을 파일에 쓴다.
- 리스트의 각 문자열 끝에 개행 문자 `\n`을 붙여줘야 줄바꿈이 된다.

형식2

```
with open('hello.txt', 'r') as file: # hello.txt 파일을 읽기 모드(r)로 열기  
    line = None # 변수 line을 None으로 초기화  
    while line != '':  
        line = file.readline()  
        print(line.strip('\n'))
```

- `readline()` 으로 파일을 읽을 때는 `while` 반복문을 활용한다.
- 더 이상 읽을 줄이 없을 때는 빈 문자열을 반환하는데, 이를 조건으로 반복문을 탈출한다.
- 이와같은 방법으로 길이에 상관없이 파일을 읽어올 수 있다.

피클링(pickling) 및 언피클링(unpickling)

파이썬 객체를 파일에 저장하는 과정을 피클링(pickling)이라고 하고, 파일에서 객체를 읽어오는 과정을 언피클링(unpickling)이라고 한다.

해당 작업은 pickle 모듈의 dump() 및 load()를 사용한다.

형식1

```
import pickle

변수1 = '문자열'
변수2 = 17
with open('파일명.p', 'wb') as myfile: #파일을 바이너리 쓰기 모드(wb)로 열기
    pickle.dump(변수1, myfile)
    pickle.dump(변수2, myfile)
```

형식2

```
import pickle

with open('파일명.p', 'rb') as myfile: #파일을 바이너리 읽기 모드(rb)로 열기
    변수1 = pickle.load(myfile)
    변수2 = pickle.load(myfile)
```

파일 모드 종류

모드	설명
r	읽기 모드 : 파일을 읽기만 할 때 사용(파일이 없으면 에러)
r+	읽기 or 쓰기 모드 : 파일이 없으면 에러
w	쓰기 모드 : 파일에 내용을 입력할 때 사용(파일이 없으면 생성)
w+	읽기 or 쓰기 모드 : 파일이 없으면 생성
a	추가 모드 : 파일의 마지막에 새로운 내용을 추가할 때 사용(파일이 없으면 생성)
a+	읽기 or 추가 모드 : 파일이 없으면 생성

- +는 각자 조금 다른 작업을 한다.
 - w+ : 기존 파일에 있던 데이터를 완전 지우고 새로 내용을 입력합니다.

- r+ : 기존 파일에 있던 데이터를 그대로 두고 그 위에 내용을 덮어씹습니다.
- a+ : 기존 파일에 있던 데이터를 그대로 두고 마지막에 내용을 입력합니다.

예제] 15file.py

```

9 print("="*30)
0 print("새파일01.txt")
1 print("="*30)
2 f_open = open("새파일01.txt", mode='wt', encoding='utf-8')
3 for i in range(1, 21):
4     data = "%d번째 줄입니다.\n" % i
5     f_open.write(data)
6 f_open.close()
7

```

```

1 f_read = open("새파일01.txt", mode='rt', encoding='utf-8')
2 while True:
3     line = f_read.readline()
4     if not line: break
5     print(line)
6 f_read.close()
7

```

```

1 f_add = open('새파일01.txt', mode='at', encoding='utf-8')
2 f_add.write("추가하는 내용입니다.")
3 f_add.writelines(["줄바꿈은 되나요?\n", "안되면 개행문자를 넣어주세요."])
4 f_add.write("마지막 라인입니다.")
5 f_add.close()
6

```

```

3 print("="*30)
4 print("새파일02.txt")
5 print("="*30)
6
7 with open("새파일02.txt", mode='wt', encoding='utf-8') as myfile:
8     for i in range(1, 16):
9         data = "%d라인 입력합니다.\n" % i
10        myfile.write(data)
11
12 with open("새파일02.txt", mode='rt', encoding='utf-8') as myfile:
13     line = None
14     while line != '':
15         line = myfile.readline()
16         print(line.strip('\n'))
17

```

여기까지 작성하세요.

```
67 import pickle
68
69 name = '개발자'
70 age = 99
71 address = '서울시 중구 세종대로'
72 times = {'JAVA': 20, 'HTML': 2, 'Oracle': 10, 'Python': 3}
73
74 with open('developer.p', 'wb') as file:
75     pickle.dump(name, file)
76     pickle.dump(age, file)
77     pickle.dump(address, file)
78     pickle.dump(times, file)
79
80 with open('developer.p', 'rb') as file:
81     name = pickle.load(file)
82     age = pickle.load(file)
83     address = pickle.load(file)
84     times = pickle.load(file)
85     print("이름", name)
86     print("나이", age)
87     print("주소", address)
88     print("배당시간", times)
89
```