



파이썬기초

8주차 - 3교시
파일 입출력 함수





학습내용

- 파일 입출력 함수



학습목표

- 파일 처리를 위한 파일 입출력 함수에 대해 이해하고 이를 활용하여 코딩할 수 있다.



생각해 봅시다

파일처리를 위해 필요한
기능들에 대해 생각해봅시다.



01



파일 입출력 함수



1 | 파일 입출력 함수

01 파일 저장 / 화면 출력

- 텍스트파일 인코딩 / 디코딩
→ **문자코드**(아스키, 유니코드(UTF-8(한글))
- 이진파일 인코딩 / 디코딩
→ **미디어별**(이미지 파일, 동영상 파일) 저장형식

1 | 파일 입출력 함수

02 open() 함수

- 파일을 열 때, **파일유형과 사용 방법을 정의함**



텍스트 파일 모드

'r' , 'w' , 'a'



이진 파일 모드

'rb' , 'wb'

1 | 파일 입출력 함수

02 open() 함수

- 파일은 파일모드에서 **정의한 방법으로만 사용 가능함**
- 읽기모드('r')

➤ 파일들에 대해 **입력함수**들만 사용 가능함

- 쓰기모드('w' 또는 'a')

➤ 파일에 대해 **출력함수**들만 사용 가능함

1 | 파일 입출력 함수

03 open() 함수에서 사용하는 입출력 함수

표준함수	사용모드	대상 파일	기능
read()	읽기	텍스트/이진 파일	파일 전체를 읽음
read(n)	읽기	텍스트/이진 파일	n개 바이트를 읽음
readline()	읽기	텍스트 파일	한 라인을 읽음
readlines()	읽기	텍스트 파일	파일 전체 내용을 라인 단위 리스트로 반환함
write()	쓰기	텍스트/이진 파일	문자열을 있는 그대로 출력함
writelines()	쓰기	텍스트 파일	리스트의 모든 원소를 출력함

2 | 텍스트 파일 입력

01 텍스트 파일 입력 예제 1

```
In [36]: file5 = open("data.txt", "r", encoding="utf-8")
          txt = file5.read()
          print("read() 반환값 ", "="*30)
          print(txt)
          file5.close()
```

2 | 텍스트 파일 입력

01 텍스트 파일 입력 예제 1

결과

```
read() 반환값 =====  
안녕하세요.  
처음 해보는 파이썬 프로그래밍  
파일 다루는 방법을 공부하고 있습니다.  
화이팅!
```

2 | 텍스트 파일 입력

02 텍스트 파일 입력 예제 2

```
In [37]: file5 = open("data.txt", "r", encoding="utf-8")
          txt = file5.read(3)
          print("read(3) 반환값 ", "="*30)
          print(txt)
          file5.close()
```

결과

```
read(3) 반환값 =====
안녕하
```

2 | 텍스트 파일 입력

03 텍스트 파일 입력 예제 3

```
In [38]: file5 = open("data.txt", "r", encoding="utf-8")
          print("readline() 반환값 ", "="*30)
          txt = file5.readline()
          print(txt)
          txt = file5.readline()
          print(txt)
          txt = file5.readline()
          print(txt)
          file5.close()
```

2 | 텍스트 파일 입력

03 텍스트 파일 입력 예제 3

결과

```
readline() 반환값 =====  
안녕하세요.  
처음 해보는 파이썬 프로그래밍  
파일 다루는 방법을 공부하고 있습니다.
```

2 | 텍스트 파일 입력

04 텍스트 파일 입력 예제 4

```
In [40]: file5 = open("data.txt", "r", encoding="utf-8")
          txt = file5.readlines()
          print("readline() 반환값 ", "="*30)
          print(txt)
          file5.close()
```

결과

```
readline() 반환값 =====
['안녕하세요.Wn', '처음 해보는 파이썬 프로그래밍Wn', '파일 다루는
방법을 공부하고 있습니다.Wn', '화이팅!']
```

2 | 텍스트 파일 입력

05 텍스트 파일 입력 예제 5

```
In [41]: print("data.txt 파일 내용 읽기/출력")  
         for element in txt :  
             print(element, end = "" )
```

결과

data.txt 파일 내용 읽기/출력
안녕하세요.
처음 해보는 파이썬 프로그래밍
파일 다루는 방법을 공부하고 있습니다.
화이팅!

2 | 텍스트 파일 입력

06 for문 이용한 라인 단위 입력 처리

- readline()을 사용하지 않고 **라인 단위**로 입력파일을 처리할 수 있음

```
In [40]: # for문 이용한 라인 단위 입력 처리
file6 = open("data.txt", "r", encoding="utf-8")
for line in file6 :
    print(line, end="")
file6.close()
```


2 | 텍스트 파일 입력

06 for문 이용한 라인 단위 입력 처리

- readln()을 사용하지 않고 **라인 단위**로 입력파일을 처리할 수 있음

결과

안녕하세요.

처음 해보는 파이썬 프로그래밍

파일 다루는 방법을 공부하고 있습니다.

화이팅!

3 | 텍스트 파일 출력

01 텍스트 파일 출력

- 'w' 또는 'a' 모드 설정



write()

문자열 파일 출력(쓰기)



writeline()

라인단위 리스트 파일 출력(쓰기)

3 | 텍스트 파일 출력

02 텍스트 파일 출력 예제

```
In [47]: # 텍스트 파일 출력
f1 = open("data.txt", "r", encoding="utf-8")
f2 = open("result.txt", "w", encoding="utf-8")

f2.write("[write() 함수 출력]\n")
f2.write("제목 : 파일 출력(쓰기) 테스트 ")
f2.write("한줄 입력....")
```

3 | 텍스트 파일 출력

02 텍스트 파일 출력 예제

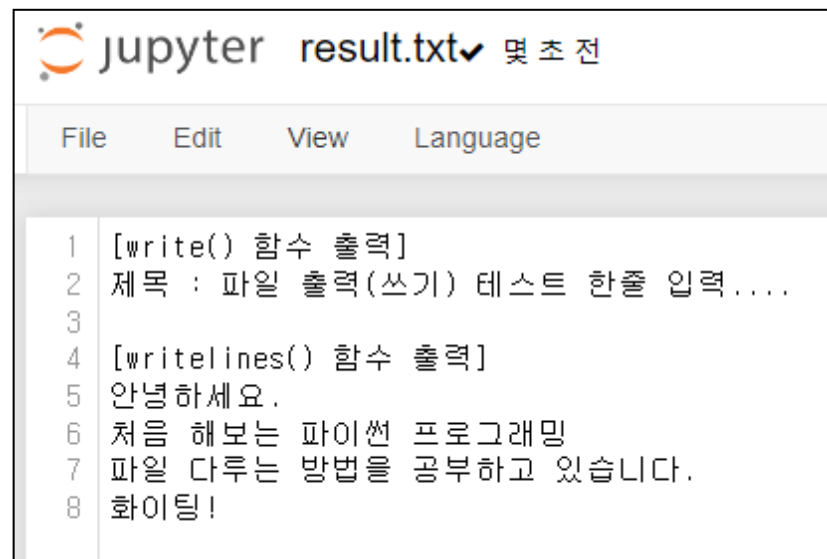
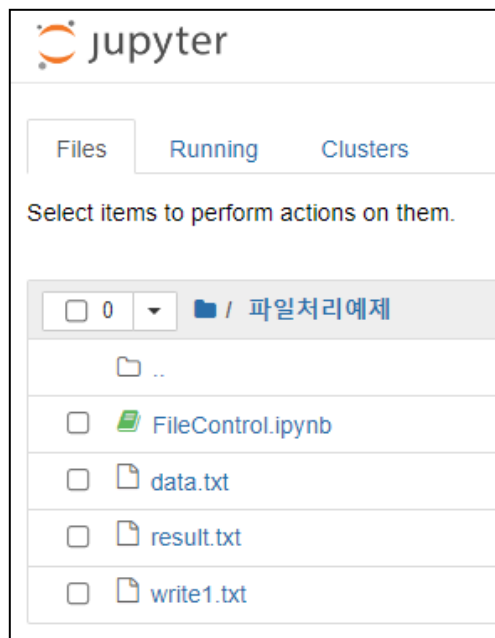
```
In [47]: f2.write("WnWn[writelines() 함수 출력]Wn")  
list_line = f1.readlines()  
f2.writelines(list_line)  
  
f1.close()  
f2.close()  
  
print("쓰기 완료")
```

결과

쓰기 완료

3 | 텍스트 파일 출력

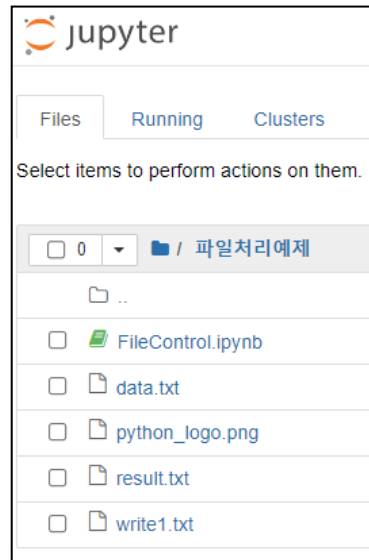
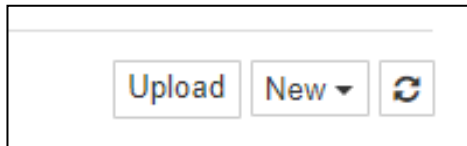
03 결과



4 | 이진 파일 입출력

01 원본 이미지 파일 준비

- upload 버튼을 눌러 파일 업로드



4 | 이진 파일 입출력

02 이진파일 입력과 출력

- 'rb' 또는 'wb'로 설정



'rb' 모드

read() 함수 사용



'wb' 모드

write() 함수 사용

4 | 이진 파일 입출력

03 사용 예제

```
In [53]: # 이진 파일 입출력
file_img = open("python_logo.png", "rb")
file_newImg = open("파이썬.png", "wb")

img = file_img.read()
# print(img)
```


4 | 이진 파일 입출력

03 사용 예제

```
In [53]: # 파일 쓰기, 파일 복사
file_newImg.write(img)

file_img.close()
file_newImg.close()

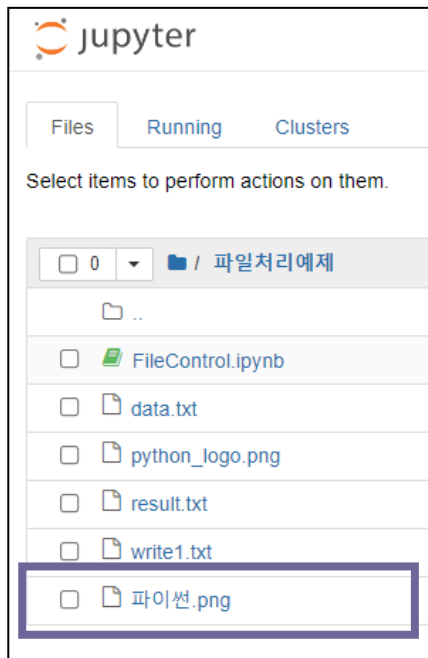
print("파일 복사 완료...")
```

결과

파일 복사 완료...

4 | 이진 파일 입출력

04 결과 확인



5 | CSV 파일 입출력

01 텍스트 파일에서 리스트 표현

- 한 학생의 성적 데이터 처리



라인 단위

한 학생의 자료, 자료값들을 의미에 따라
동일한 순서로 나열함



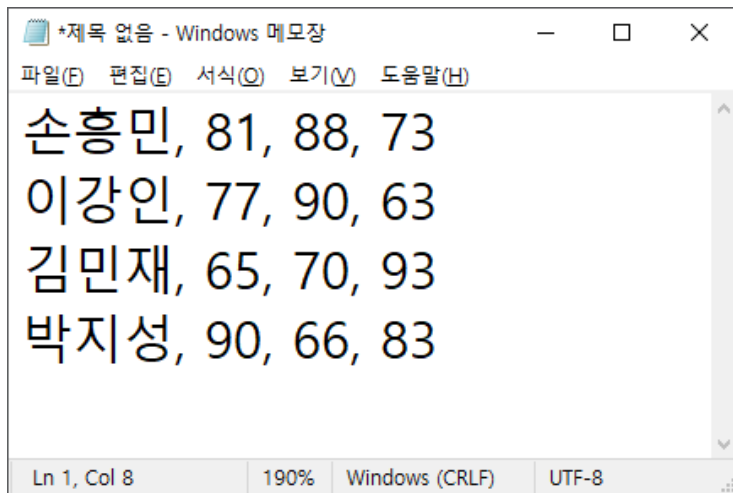
한 라인에서 데이터 구분

공백, 콤마, 기타 구분자

5 | CSV 파일 입출력

01 텍스트 파일에서 리스트 표현

- csv 파일 : **콤마**로 데이터 구분



5 | CSV 파일 입출력

02 CSV 파일(또는 공백구분 파일) 입력

- 라인단위로 읽고, 라인의 자료값들을 구분해야 함
 - 라인 단위로 문자열을 추출하고 **필요한 부분의 문자열들을 분리**해서 리스트로 추출함

부분 문자열 분리

`rstrip()`과 `split()` 함수 사용

5 | CSV 파일 입출력

03 rstrip()과 split() 함수 사용 예제

- rstrip() 사용 예제

```
In [56]: data_line = "손흥민, 81, 88, 73"  
         line = data_line.rstrip()  
         print("line.rstrip()의 결과:", line)
```

결과

line.rstrip()의 결과: 손흥민, 81, 88, 73

5 | CSV 파일 입출력

03 `rstrip()`과 `split()` 함수 사용 예제

- `split()` 함수 사용 예제

```
In [57]: data_line = "손흥민, 81, 88, 73"
```

```
line = data_line.split(',')  
print("line.split(',')의 결과:", line)
```

5 | CSV 파일 입출력

03 rstrip()과 split() 함수 사용 예제

- split() 함수 사용 예제

```
In [57]: scoreList = []  
         for i in range(2, len(line)) :  
             score = int(line[i])  
             scoreList.append(score)  
         print("line의 수치값 리스트:", scoreList)
```

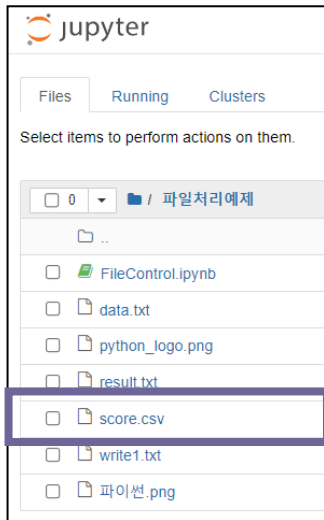
결과

```
line.split(',')의 결과: ['손흥민', '81', '88', '73']  
line의 수치값 리스트: [88, 73]
```


5 | CSV 파일 입출력

04 CSV 파일 입출력 예제

- score.csv 파일을 현재 작업 폴더에 업로드



5 | CSV 파일 입출력

04 CSV 파일 입출력 예제

In [67]: # 성적 처리 예제

```
original_file = open ("score.scv", "r", encoding="utf-8")
```

```
result_file = open ("result-score.txt", "w", encoding="utf-8")
```

```
result_file.writelines("[ 성적 처리 결과 ]\n")
```

5 | CSV 파일 입출력

04 CSV 파일 입출력 예제

```
In [67]: for line in original_file :  
          student = (line.rstrip()).split(",")  
          total = 0  
          for i in range(1, len(student)) :  
              total = total + int(student[i])
```

5 | CSV 파일 입출력

04 CSV 파일 입출력 예제

```
In [67]: avg = total / (len(student)-1)

studentResult ="이름:%s 총점: %3d 평균:%5.1fWn" % (student[0],
total, avg)

result_file.write(studentResult)
```

5 | CSV 파일 입출력

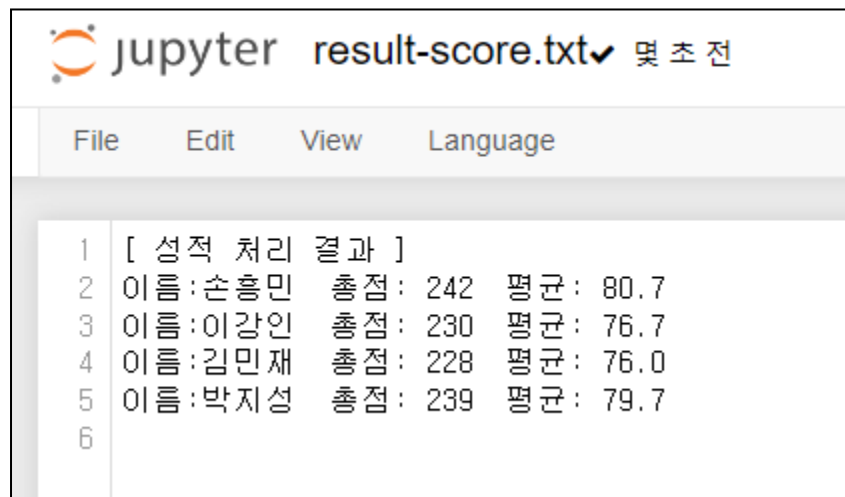
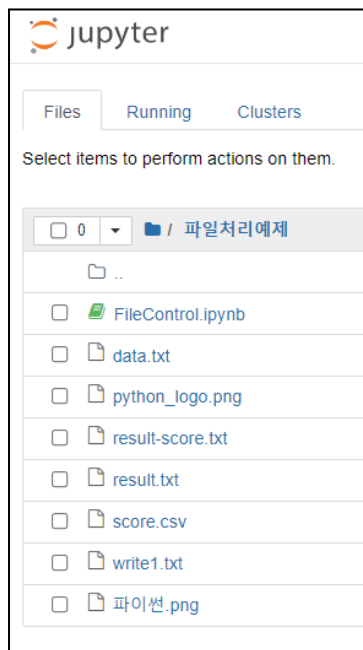
04 CSV 파일 입출력 예제

```
In [67]: original_file.close()  
         result_file.close()  
  
         print("성적처리 완료 ...")
```

결과 성적처리 완료 ...

5 | CSV 파일 입출력

05 결과





Q1

open()함수로 파일을 열 때 사용 가능한 파일 모드가
아닌 것은?

- ① rw
- ② w
- ③ a
- ④ rb



Q1

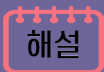
open()함수로 파일을 열 때 사용 가능한 파일 모드가 아닌 것은?

- 1 rw
- 2 w
- 3 a
- 4 rb



정답

1번



해설

파일 open 시 사용 가능한 모드로는 'r', 'w', 'a', 'rb', 'wb' 가 있습니다.



Q2

파일 읽기 함수에서 모든 라인을 리스트로
반환해 주는 함수는?

- ① read()
- ② readlines()
- ③ readline()
- ④ read(3)



Q2

파일 읽기 함수에서 모든 라인을 리스트로 반환해 주는 함수는?

- ① read()
- ② readlines()
- ③ readline()
- ④ read(3)



정답

2번



해설

readlines() 함수는 파일 전체 내용을 라인 단위의 문자열을 리스트로 반환해 줍니다.



Q3

파일 출력 함수에서 문자열 하나를 그대로 출력(쓰기)하는 함수는?

- ① readline()
- ② read()
- ③ write()
- ④ writeline()



Q3

파일 출력 함수에서 문자열 하나를 그대로 출력(쓰기)하는 함수는?

- ① readline()
- ② read()
- ③ write()
- ④ writeline()



정답

3번



해설

문자열 하나를 그대로 출력하는 함수는 write() 함수입니다.



입력과 출력

- 프로그램을 기준으로 파일을 읽어오는 것을 파일 입력이라고 하고 프로그램에서 바깥 쪽으로 내보내는 것을 파일 출력이라 함

텍스트파일

- 문자열로 저장된 파일

바이너리 파일

- 이미지와 동영상 같은 비트 단위로 저장되는 형태의 파일



파일 처리단계

- 파일 실행(더블 클릭) → 파일 사용(읽기/쓰기 등) → 파일 닫기
- `open()` 함수 → 파일 사용(읽기/쓰기 등) → `close()` 함수



`open()` 함수

- 변수 = `open("파일명", "파일모드")`
- 텍스트 파일 모드 : 'r' , 'w' , 'a'
- 이진 파일 모드 : 'rb' , 'wb'



- 📌 with 블록을 이용한 파일 처리
 - with 블록을 이용하면 블록이 끝날 때 자동으로 close() 실행

📌 입출력 함수

- read()
- readline()
- readlines()
- write()
- writeline()



파이썬기초

NEXT

파일 입출력의 활용

