

Lab6. FILE IO

CSED101 LAB

파일에 데이터 쓰기

파일을 연다.



파일에 데이터를
읽거나 쓴다.



파일을 닫는다.

파일객체

파일이름

파일 열기 모드

```
f = open( "input.txt", "r" )
```

...

```
f.close()
```

파일 열기 모드	모드 이름	설명
"r"	읽기 모드 (read)	파일을 읽기만 할 때 사용. 파일의 처음부터 읽는다.
"w"	쓰기 모드 (write)	파일에 쓸 때 사용. 파일의 처음부터 쓴다. 파일이 없으면 생성된다. 파일이 존재하면 기존의 내용은 지워진다.
"a"	추가 모드 (append)	파일의 마지막에 새로운 내용을 추가 시킬 때 사용 파일이 없으면 생성된다.

파일 쓰기

■ `f = open("new.txt", "w")`

```
f = open("new.txt", "w")

for i in range(1, 6):
    data = "%d line\n" % i
    f.write(data)

f.close()
```

new.txt

```
1 line
2 line
3 line
4 line
5 line
```

파일 읽기(1/2)

(1) *readline()*

파일에서 한 줄 읽어옴

```
f = open("new.txt", "r")
line = f.readline()
print(line, end='')
line = f.readline()
print(line, end='')
f.close()
```

```
1 line
2 line
>>>
```

(2) *readlines()*

모든 라인을 한꺼번에 읽어서 각각의 줄을 요소로 갖는 리스트를 반환

```
f = open("new.txt", "r")
lines = f.readlines()
print(lines)
f.close()
```

```
['1 line\n', '2 line\n', '3 line\n', '4 line\n', '5 line\n']
>>>
```

파일 읽기(1/2)

(3) *f.read()*:

파일을 전부 읽은 문자열을 반환

```
f = open("new.txt", "r")
```

```
data = f.read()  
print(data, end='')
```

```
f.close()
```

```
1 line  
2 line  
3 line  
4 line  
5 line  
>>>
```

(4) *for* 문 사용

```
f = open("new.txt", "r")
```

```
for line in f:  
    print(line, end='')
```

```
f.close()
```

```
1 line  
2 line  
3 line  
4 line  
5 line  
>>>
```

파일에 내용 추가하기

```
f = open("new.txt", "a")

for i in range(6, 11):
    data = "%d line\n" % i
    f.write(data)

f.close()
```

new.txt

1 line
2 line
3 line
4 line
5 line
6 line
7 line
8 line
9 line
10 line

추가된 내용

파일을 열고 자동으로 닫기(with~as)

- open() ~ close() 사용

```
f = open( "output.txt", "w" )  
f.write("Python is fun!")  
f.close()
```

- with open() as 사용

```
with open( "output.txt", "w" ) as f :  
    f.write("Python is fun!")
```

with 블록을 벗어나는 순간 파일을 자동으로 닫아준다.

파일 포인터

- 파일 포인터
: 파일의 현재 위치를 가리키는 것
- *f.tell()*: 현재 파일 포인터의 위치 반환
- *f.seek()*: 지정하는 곳으로 포인터의 위치를 변경

test.txt

```
first line
second line
third line
```

```
>>> f = open("test.txt", "r")
>>> f.tell()
0
>>> f.readline()
'first line\n'
>>> f.tell()
11
>>> 
```

※ 운영체제 별로 개행문자가 달라서 결과가 다르게 나올 수 있음

파일 - 실습

- 파일명을 입력 받아, 해당 파일을 한 줄씩 읽어 파일의 내용을 모두 대문자로 출력하는 프로그램을 작성하시오. 단, 파일이 없는 경우 "파일이 존재하지 않는다" 정도의 메시지를 출력할 것!

(힌트)

- 문자열을 대문자로 변환

```
>>> "hello".upper()
```

```
'HELLO'
```

- 문자열을 소문자로 변환

```
>>> "World".lower()
```

```
'world'
```

- `os.path.exists(파일명)`: 파일이나 디렉토리가 존재하는지 검사

```
>>> import os
```

```
>>> os.path.exists("test.txt")
```

```
True
```

```
Enter a file name: test.txt
FIRST LINE
SECOND LINE
THIRD LINE
>>>
```

Problem

- 아래의 score.txt를 읽어서 학생들의 성적을 처리하여 그 결과를 report.txt로 출력하는 프로그램을 작성하시오.

(학점기준)

- 평균이 90이상이면 A
- 80이상~90미만이면 B
- 70이상~80미만이면 C
- 60이상~70미만이면 D
- 60미만이면 F

(주의사항)

학생수를 예시의 학생수처럼 고정해서 구현하지 말 것!

- 제출파일명: 학번_Lab6.py

<score.txt> 예시

- 각 줄(line) 단위로 한 학생의 학번, 중간고사 점수, 기말고사 점수가 순서대로 기록되어 있다.

202301	89	78
202302	76	84
202303	67	42
202304	66	88
202305	98	99

<report.txt> 예시

- 학번, 평균(학점) 순으로 아래와 같이 파일에 기록해 주세요.

202301	83.5(B)
202302	80.0(B)
202303	54.5(F)
202304	77.0(C)
202305	98.5(A)