
Lecture 9. 파일 입출력과 예외 처리

BAE SEKANG(배세강)



Contents

1. File I/O

- 파일의 종류
- file mode
- open()
- 파일 쓰기
- 파일 읽기
- with
- encoding

2. try-except

- Syntax Error & Exception Error
- try & except
- else
- finally



File I/O

♣ 파일의 종류

- text file: 사람이 읽을 수 있는 문자로 구성된 파일(.txt, .py, .csv, ...)
- binary file: 이진 형식의 데이터로 저장된 파일(.jpg, .mp3, .exe, ...)



0	0011 0000	L	0100 1100	g	0110 0111	"	0010 0010
1	0011 0001	M	0100 1101	h	0110 1000	'	0010 0111
2	0011 0010	N	0100 1110	i	0110 1001	(	0010 1000
3	0011 0011	O	0100 1111	j	0110 1010	)	0010 1001
4	0011 0100	P	0101 0000	k	0110 1011	*	0010 1010
5	0011 0101	Q	0101 0001	l	0110 1100	+	0010 1011
6	0011 0110	R	0101 0010	m	0110 1101	,	0010 1100
7	0011 0111	S	0101 0011	n	0110 1110	-	0010 1101
8	0011 1000	T	0101 0100	o	0110 1111	.	0010 1110
9	0011 1001	U	0101 0101	p	0111 0000	/	0010 1111
		V	0101 0110	q	0111 0001	:	0011 1010
A	0100 0001	W	0101 0111	r	0111 0010	;	0011 1011
B	0100 0010	X	0101 1000	s	0111 0011	=	0011 1101
C	0100 0011	Y	0101 1001	t	0111 0100	?	0011 1111
D	0100 0100	Z	0101 1010	u	0111 0101		
E	0100 0101			v	0111 0110		
F	0100 0110	a	0110 0001	w	0111 0111		
G	0100 0111	b	0110 0010	x	0111 1000		
H	0100 1000	c	0110 0011	y	0111 1001		
I	0100 1001	d	0110 0100	z	0111 1010		
J	0100 1010	e	0110 0101				
K	0100 1011	f	0110 0110	!	0010 0001		



File I/O

♣ file mode

- 파일을 열 때 선택하는 방식

mode	의미
r	read: 파일을 읽는 모드
w	write: 파일을 새롭게 쓰는 모드
a	append: 파일을 이어서 쓰는 모드
x	create: 파일을 새로 생성하는 모드



File I/O

♣ open()

- text file: 사람이 읽을 수 있는 문자로 구성된 파일(.txt, .py, .csv, ...)
- binary file: 이진 형식의 데이터로 저장된 파일(.jpg, .mp3, .exe, ...)

```
# f에 파일 객체가 저장된다.
```

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

```
# 파일을 사용한 후 닫아야 한다.
```

```
f.close()
```

- 파일 객체: 파일의 데이터, 지금까지 읽은 위치, 파일 이름, 열린 모드 등의 복합적인 정보를 담고 있는 Object



File I/O

♣ 파일 쓰기

- `write()`: 파일을 새로 만든다.
- `append()`: 기존 파일의 맨 마지막 부분부터 새로운 내용을 추가한다.
- `writelines()`: list, tuple 등에 담긴 문자열을 한 번에 파일에 쓴다.

\n을 붙여야 행이 나뉜다!

```
lines = ["1번 행\n", "2번 행\n", "3번 행\n"]  
f = open("test.txt", "w", encoding="utf-8")  
f.writelines(lines)  
f.close()
```



File I/O

♣ 파일 읽기

- `read()`: 파일 내용 전체를 하나의 `str`로 가져온다.
- `readline()`: 파일을 한 줄 읽어온다.
- `readlines()`: 파일의 모든 줄을 읽은 후, 각 줄을 요소로 한 `list`를 반환한다.

```
f = open("input.txt", "r")  
file_contents = f.read()  
  
print(file_contents)  
  
f.close()
```



File I/O

♣ Example: 파일 읽기

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

while True:
    line = f.readline()

    # 파일을 다 읽으면 반복문을 빠져나옴
    if not line:
        break

    # strip()은 \n 제거 용도
    print(line.strip())
```



File I/O

[Practice]

practice1.txt 파일 내용 전체를 출력하는 프로그램을 작성한다.

[Practice]

practice1.txt 파일의 각 줄을 요소로 한 list를 출력한다.



File I/O

♣ with

- with 블록을 통해 수동으로 닫는 close()를 사용하지 않아도 된다.
- with open() as [파일 객체 변수]:

```
with open("test.txt", "r", encoding="utf-8") as f:  
    data = f.read()  
    print(data)  
# 이 지점에 도달하면 f는 자동으로 closed
```

Q. 굳이 close()를 해야 하는 이유가 뭘까?



File I/O

♣ encoding

- 인코딩: 문자를 이진 데이터로 변환하는 규칙
- ASCII, UTF-8, EUC-KR, CP949, ...

```
# 파일 저장할 때 UTF-8로 통일하는 것이 표준
with open("test.txt", "w", encoding="utf-8") as f:
    f.write("한글 데이터")
```

```
with open("test.txt", "r", encoding="utf-8") as f:
    print(f.read())
```



try-except

♣ Error

- 구문 오류(Syntax Error): 문법 규칙을 어겨서 발생한 오류
- 예외(Exception): 문법은 옳으나 코드 실행 중 예상치 못한 상황 발생

구분	Syntax Error	Exception
발생 시점	실행 전(컴파일 단계)	실행 중(런타임 단계)
원인	오타, 들여쓰기 실수 등	없는 파일 열기, 0으로 나누기 등
해결 방법	코드 수정	try-except



try-except

♣ try

- try 부분: 일단 실행할 코드
- except 부분: 예외가 발생할 시 실행할 코드

```
try:
    with open("test.txt", "r", encoding="utf-8") as f:
        data = f.read()
        num = int(data)
except FileNotFoundError:
    print("파일이 없습니다. ")
except ValueError:
    print(" 파일 내용이 숫자가 아닙니다.")
```



try-except

♣ else, finally

- else: 에러가 발생하지 않았을 때만 추가로 실행
- finally: 에러 유무와 관계없이 항상 마지막으로 실행

```
try:
    f = open("test.txt", "r")
except FileNotFoundError:
    print("파일이 없습니다.")
else:
    print(f.read())
    f.close()
finally:
    print("3. 모든 작업을 마쳤습니다.")
```



try-except

[Practice]

practice3.txt 파일을 읽어와 전체 내용을 출력하되, practice3.txt 파일이 존재하지 않으면 “존재하지 않습니다!”를 출력한 후 사용자에게 글을 입력 받아 practice3.txt를 새로 만드는 프로그램을 작성한다.



THANK YOU!

Any Question?