



문화 A0019

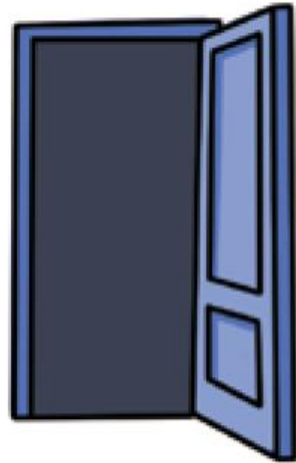
파이썬프로그래밍

김 태 완

kimtwan21@dongduk.ac.kr

파일 읽기

- 파일 읽기 3 단계
 - 파일 열기
 - 파일 읽기
 - 파일 닫기



1단계
파일 열기



2단계
파일 읽기



3단계
파일 닫기

파일 읽기

- 파일 읽기 3 단계

- 파일 열기

- 파일을 열기 위해서는 open() 함수에서 파일명을 지정하고, 읽기(Read)를 의미하는 "r"로 설정함
 - 모드(mode) : open() 함수의 마지막 매개변수
 - 파일을 열 때 어떤 용도로 열지 결정함
 - 파일 읽기용은 "r"을 씀

변수명 = open("파일경로/파일이름", "r")

| 종류 | 설명 |
|----|--|
| 생략 | r과 동일하다. |
| r | 읽기 모드, 기본값이다. |
| w | 쓰기 모드, 기존에 파일이 있으면 덮어쓴다. |
| r+ | 읽기/쓰기 겸용 모드이다. |
| a | 쓰기 모드, 기존에 파일이 있으면 이어서 쓴다. append의 약어이다. |
| t | 텍스트 모드, 텍스트 파일을 처리한다. 기본값이다. |
| b | 이진 모드, 이진 파일을 처리한다. |

- 파일 읽기

- 파일 닫기

파일 읽기

- 파일 읽기 3 단계
 - 파일 열기
 - **파일 읽기**
 - 파일에서 데이터를 읽어올 수 있는 상태
 - **파일 닫기**
 - 파일과 관련된 모든 작업이 끝나면 파일을 정상적으로 닫아줘야 함
 - 파일을 닫기 위해 사용하는 변수는 `open()` 함수로 열었던 변수명임

```
변수명.close();
```

파일 읽기

- 파일 읽기 3 단계
 - 파일 열기
 - 파일 읽기
 - 파일 닫기

```
f = open("20230511.txt", 'r')  
  
line = f.readline()  
print(line)  
  
f.close()
```

파일 읽기

- 파일을 열 때 오류 처리
 - 열고자 하는 파일이 없을 때 아래와 같이 오류 메시지 출력

```
Traceback (most recent call last):
  File "c:\Users\enoug\Desktop\python\tmp.py", line 85, in <module>
    f = open("tony.txt", 'r')
FileNotFoundError: [Errno 2] No such file or directory: 'tony.txt'
```

- 예외 처리

```
import os
filename = "tony.txt"
if os.path.exists(filename):
    f = open(filename, 'r')
    lines = f.readlines()
    ...#생략
    f.close()
else:
    print("%s 파일이 없습니다." %filename)
```

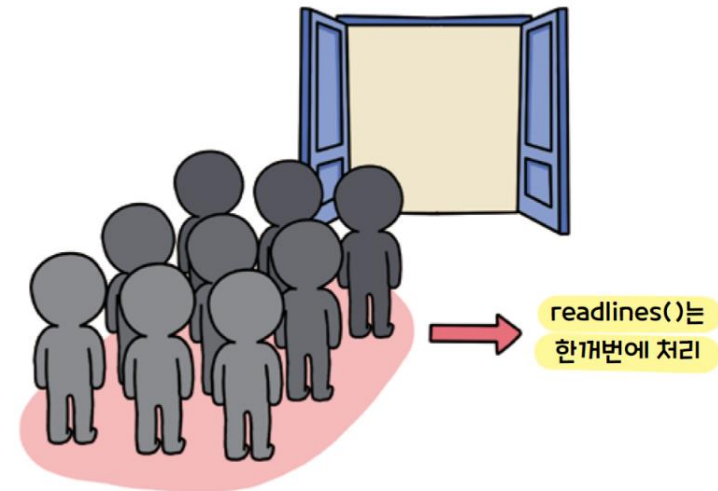
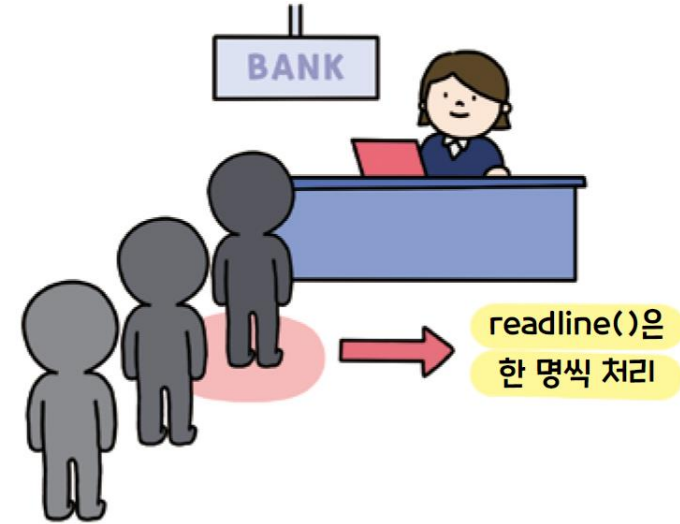
파일 읽기

- 파일 또는 폴더가 이미 존재하는 지 확인
 - `os.path.exists("1.txt")`
 - `os.path.isfile("1.txt")`
 - `os.path.isdir("c:/python")`

```
import os
filename = "tony.txt"
if os.path.isfile(filename):
    f = open(filename, 'r')
    lines = f.readlines()
    ...#생략
    f.close()
else:
    print("%s 파일이 없습니다." %filename)
```

파일 읽기

- 파일에 담아둔 데이터를 읽기 위한 함수
 - `readline()`
 - 파일의 내용을 한 행씩 읽음
 - 파일에 100개의 행이 있다면 100번을 반복
 - `readlines()`
 - 파일의 내용을 한꺼번에 읽어 리스트에 저장



파일 읽기

- 파일에 담아둔 데이터를 읽기 위한 함수
 - readline()

```
f = open("/mnt/disk/aaa.txt", 'r')

while True:
    line = f.readline()
    if not line: # if line == "":
        break
    print(line)

f.close()
```

파일 읽기

- 파일에 담아둔 데이터를 읽기 위한 함수
 - readlines()

```
f = open("/mnt/disk/aaa.txt", 'r')

lines = f.readlines()

for line in lines:
    print(line)

f.close()
```

파일 읽기

- 파일에 담아둔 데이터를 읽기 위한 함수
 - readlines()
 - 리스트 : ["1 번째 줄입니다.\n", "2 번째 줄입니다.\n", ..., "10 번째 줄입니다.\n"]
 - 파일을 읽을 때 줄 끝의 줄 바꿈(\n) 문자를 제거하고 사용해야 할 경우가 많다.
 - 다음처럼 strip 함수를 사용하면 줄 바꿈 문자를 제거할 수 있다.

```
f = open("/mnt/disk/aaa.txt", 'r')
lines = f.readlines()

for line in lines:
    print(line.strip())

f.close()
```

파일 읽기

- 파일에 담아둔 데이터를 읽기 위한 함수
 - `read ()` : 전체가 하나의 문자열

```
f = open("/home/taewan/kick.txt", 'r')  
  
data = f.read()  
  
print(data)  
  
f.close()
```

파일 쓰기

- 원하는 결과를 파일에 씀
- 파일 쓰기 3 단계

- 파일 열기 `변수명 = open("파일경로/파일이름", "w")`

- 파일을 쓰기 위해서도 `open()` 함수에서 파일명을 지정하고, 파일 쓰기 모드를 지정
- 이때 파일 경로에 같은 이름의 파일이 있다면 기존 파일을 덮어씀
 - 같은 이름의 파일이 없다면 파일을 새로 생성함
- 쓰기용은 `open()` 함수의 매개변수로 "w"를 사용함

- 파일 쓰기
- 파일 닫기

`변수명.close();`

| 종류 | 설명 |
|----|--|
| 생략 | r과 동일하다. |
| r | 읽기 모드, 기본값이다. |
| w | 쓰기 모드, 기존에 파일이 있으면 덮어쓴다. |
| r+ | 읽기/쓰기 겸용 모드이다. |
| a | 쓰기 모드, 기존에 파일이 있으면 이어서 쓴다. append의 약어이다. |

파일 쓰기

- 원하는 문자열을 파일에 쓰는 함수
 - write()
 - 한 줄씩 쓰기
 - writelines()
 - 모든 줄 쓰기

한 줄씩 쓰기: write()

```
with open('myfile.txt', 'w') as f:  
    for line in lines:  
        f.write(line)
```

모든 줄 쓰기: writelines()

vs.

```
with open('myfile.txt', 'w') as f:  
    f.writelines(lines)
```

파일 쓰기

- 원하는 문자열을 파일에 쓰는 함수
 - write()

```
f = open("result.txt", 'w')

for i in range(1, 11):
    data = "%d번째 줄입니다.\n" % i
    f.write(data)

f.close()
```

파일 쓰기

- 원하는 문자열을 파일에 쓰는 함수
 - `writelines()`
 - 리스트의 요소를 하나씩 기록해주는 역할만을 하기 때문에 자동 줄 바꿈 기능이 없기 때문에 수동으로 줄 바꿈 ("`\n`")을 해줘야 함

```
f = open("result.txt", 'w')

res = ['안녕하세요\n', '저는 누구입니다.\n',
       '만나서 반갑습니다.\n']

f.writelines(res)

f.close()
```


파일 쓰기

- 파일을 열 때, 모드 값 중 ("w"가 아닌 "a"를 사용하는 경우)
 - 기존 파일을 열어 그 뒤에 문자열 쓰기

```
f = open("result.txt", 'a')

for i in range(1, 11):
    data = "%d번째 줄입니다.\n" % i
    f.write(data)

f.close()
```

파일 닫기를 신경 쓰지 않고 싶을 때.

- with ~ as 구문 사용
 - 기존에는 파일을 열면 항상 close 해주는 것이 좋다.
 - 하지만 with ~ as 구문을 사용하면 들여쓰기가 벗어나는 순간 열린 파일 객체 f가 자동으로 close

```
f = open("foo.txt", 'w')  
f.write("Life is too short, you need python")  
f.close()
```



```
with open("foo.txt", "w") as f:  
    f.write("Life is too short, you need python")
```

Examples

- **예제 1 :** 폴더에 'star.txt' 파일을 생성한 후 다음과 같이 써보세요.

```
*****
```

```
*****
```

```
***
```

```
*
```

Examples

- **예제 2** : 학번을 입력 받아 “id.txt”에 적어보자.

20231234

Examples

- **예제 3** : “id.txt”을 열어 학번 다음 줄에 이름까지 적어보자.

```
20231234
```

```
김태완
```

Examples

- **예제 4 :** 50명의 학생의 시험 점수 (100점 만점)가 기록되어 있는 score.txt 파일을 읽어 n명의 평균 점수를 구해보자.

```
f = open("score.txt", 'r')  
sum = 0  
average = 0.0  
  
## coding here ##  
  
print(average)
```

- **실행 화면**

```
70.14
```

Examples

- **예제 5** : 50명의 학생의 시험 점수 (100점 만점)가 기록되어 있는 score2.txt 파일을 읽어 n명의 평균 점수를 구해보자.

```
f = open("score.txt", 'r')
sum = 0
average = 0.0

## coding here ##

print(average)
```

- **실행 화면**

```
42.28
```

감사합니다

kimtwan21@dongduk.ac.kr

김 태 완