



# Day41; 20221103

날짜	
유형	
태그	

GitHub - u8yes/Python

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

<https://github.com/u8yes/Python>

u8yes/Python



1 Contributor 0 Issues 2 Stars 0 Forks

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b75e3d71-c1f1-41b6-885f-052072a1d97c/09.\\_%EC%98%88%EC%99%B8%EC%B2%98%EB%A6%AC.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b75e3d71-c1f1-41b6-885f-052072a1d97c/09._%EC%98%88%EC%99%B8%EC%B2%98%EB%A6%AC.pdf)

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/dbfe67f3-2936-4152-85cc-6ab5c23bb218/10.\\_%ED%8C%8C%EC%9D%BC%EC%9E%85%EC%B6%9C%EB%A0%A5.pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/dbfe67f3-2936-4152-85cc-6ab5c23bb218/10._%ED%8C%8C%EC%9D%BC%EC%9E%85%EC%B6%9C%EB%A0%A5.pdf)

---

■ 관리자: Anaconda Prompt (Anaconda3)

```
(base) C:\WINDOWS\system32>conda activate
(base) C:\WINDOWS\system32>conda activate tf_cpu
(tf_cpu) C:\WINDOWS\system32>cd D:\heaven_dev\workspaces\Python\src(python)
(tf_cpu) C:\WINDOWS\system32>cd d:
D:\heaven_dev\workspaces\Python\src(python)
(tf_cpu) C:\WINDOWS\system32>d:
(tf_cpu) D:\heaven_dev\workspaces\Python\src(python)>_
```

```
In [3]: try:
        print( 1 / 0 )
        except:
            print("예외가 발생했습니다.")

        print("프로그램 종료")
```

```
예외가 발생했습니다.
프로그램 종료
```

```
In [25]: my_list = [1,2,3]

# my_list[4] # IndexError: list index out of range

try:
    print("첨자(인덱스)를 입력하세요")
    index = int(input())
    print(my_list[index] / 0)
except IndexError as e:
    print(f"잘못된 첨자입니다({e})")
except ValueError as e:
    print(f"잘못된 값입니다({e})")
except ZeroDivisionError as e:
    print(f"0으로 나눌 수 없습니다({e})")

print("예외가 났던 안 났던 프로그램 종료합니다.")
```

첨자(인덱스)를 입력하세요  
5  
잘못된 첨자입니다(list index out of range)  
예외가 났던 안 났던 프로그램 종료합니다.

```
In [40]: my_list = [1,2,3]

# my_list[4] # IndexError: list index out of range

try:
    print("첨자(인덱스)를 입력하세요")
    index = int(input())
    print(f"my_list[{index}]:{my_list[index]}")
except Exception as e:
    print(f"예외가 발생했습니다({e})")
else:
    print("리스트의 요소 출력 성공(성공시에만 또는 텍스트, finally와 다른 점, 한번이라도 예외가 생기면 else를 수행하지 않음)")

print("예외가 났던 안 났던 프로그램 종료합니다.")
```

첨자(인덱스)를 입력하세요  
1  
my\_list[1]:2  
리스트의 요소 출력 성공(성공시에만 또는 텍스트, finally와 다른 점, 한번이라도 예외가 생기면 else를 수행하지 않음)  
예외가 났던 안 났던 프로그램 종료합니다.

```
In [42]: my_list = [1,2,3]

try:
    print("인덱스를 입력하세요: ")
    index = int(input())
    print(f"my_list[{index}]:{my_list[index]}")
except Exception as e:
    print(f"예외가 발생했습니다({e})")
finally:
    print("무슨 일이 있어도 마무리합니다.")
```

인덱스를 입력하세요:

4

예외가 발생했습니다(list index out of range)

무슨 일이 있어도 마무리합니다.

## - 강제 예외 발생\_20221103

```
In [49]: text = input()

if text.isdigit() == False:
    raise Exception("입력 받은 문자열이 숫자로 구성되어 있지 않습니다.")

print("프로그램 종료")
```

1234이상

```
Exception                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_13720\2438931714.py in <module>
      2
      3 if text.isdigit() == False:
----> 4     raise Exception("입력 받은 문자열이 숫자로 구성되어 있지 않습니다.")
      5
      6 print("프로그램 종료")
```

**Exception:** 입력 받은 문자열이 숫자로 구성되어 있지 않습니다.

```
In [54]: try:
        text = input()

        if text.isdigit() == False:
            raise Exception("입력 받은 문자열이 숫자가 아닙니다.")

    except Exception as e:
        print(f"예외가 발생했습니다 : {e}")

    print("프로그램 종료")
```

```

예외가 발생했습니다 : 입력 받은 문자열이 숫자가 아닙니다.
프로그램 종료
```

---

```
In [73]: def some_function():
        print("1-10 사이의 수를 입력하세요 : ")
        num = int(input())

        if num < 1 or num > 10:
            raise Exception(f"{num} -> 유효하지 않은 숫자입니다.")
        else:
            print(f"입력한 수는 {num}입니다.")
```

```
In [78]: try:
        some_function()
    except Exception as e: # e는 임의로 정한 매개변수
        print(f"예외가 발생했습니다 : {e}")
```

```
1-10 사이의 수를 입력하세요 :
20
예외가 발생했습니다 : 20 -> 유효하지 않은 숫자입니다.
```

---

```
In [73]: def some_function():
        print("1-10 사이의 수를 입력하세요 : ")
        num = int(input())

        if num < 1 or num > 10:
            raise Exception(f"{num} -> 유효하지 않은 숫자입니다.")
        else:
            print(f"입력한 수는 {num}입니다.")
```

```
In [78]: try:
        some_function()
    except Exception as e: # e는 임의로 정한 매개변수
        print(f"예외가 발생했습니다 : {e}")
```

1-10 사이의 수를 입력하세요 :  
20  
예외가 발생했습니다 : 20 -> 유효하지 않은 숫자입니다.

```
In [84]: def some_function_caller():
        try:
            some_function()
        except Exception as e:
            print(f"{1} 예외가 발생했습니다 : {e}")
            raise
```

```
In [85]: try:
        some_function_caller()
    except Exception as e:
        print(f"{2} 예외가 발생했습니다. {e}")
```

1-10 사이의 수를 입력하세요 :  
-1  
1 예외가 발생했습니다 : -1 -> 유효하지 않은 숫자입니다.  
2 예외가 발생했습니다. -1 -> 유효하지 않은 숫자입니다.

## - 사용자 정의 예외 형식\_20221103

```
In [93]: class MyException(Exception):  
         def __init__(self):  
             super().__init__("MyException 발생했습니다.")
```

```
In [94]: everything_is_fine = False  
  
if everything_is_fine == False:  
    raise MyException()
```

```
-----  
MyException                                Traceback (most recent call last)  
~AppData\Local\Temp\ipykernel_13720\384908216.py in <module>  
      2  
      3 if everything_is_fine == False:  
----> 4     raise MyException()  
  
MyException: MyException 발생했습니다.
```

정상

```
In [98]: class InvalidIntException(Exception):  
         def __init__(self, arg):  
             super().__init__(f"(정수가 아닙니다. : {arg})")  
  
         def convert_to_integer(text):  
             if text.isdigit():  
                 return int(text)  
             else:  
                 raise InvalidIntException(text) # InvalidIntException() 인스턴스 생성해줄
```

```
In [101]: try:  
           print("(사용자 정의 예외)숫자를 입력하세요.")  
           text = input()  
           number = convert_to_integer(text)  
       except InvalidIntException as e:  
           print(f"예외가 발생했습니다{e}")  
       else:  
           print(f"정수 형식으로 변환되었습니다. : {number} {type(number)}")  
  
print("프로그램 종료")
```

(사용자 정의 예외)숫자를 입력하세요.

45

정수 형식으로 변환되었습니다. : 45 <class 'int'>

프로그램 종료

## 예외상황

```
In [98]: class InvalidIntException(Exception):
        def __init__(self, arg):
            super().__init__(f"(정수가 아닙니다. : {arg})")

        def convert_to_integer(text):
            if text.isdigit():
                return int(text)
            else:
                raise InvalidIntException(text) # InvalidIntException()인스턴스 생성해줄
```

```
In [100]: try:
        print("(사용자 정의 예외)숫자를 입력하세요.")
        text = input()
        number = convert_to_integer(text)
    except InvalidIntException as e:
        print(f"예외가 발생했습니다{e}")
    else:
        print(f"정수 형식으로 변환되었습니다. : {number} {type(number)}")

    print("프로그램 종료")
```

(사용자 정의 예외)숫자를 입력하세요.  
사십오  
예외가 발생했습니다(정수가 아닙니다. : 사십오)  
프로그램 종료

## 파이썬-판다스

### 3-5.통계값 알아보기

- DataFrame.describe() : 적용 가능 컬럼에 대해 count, mean, std, min, max, Q1, Q2, Q3 등의 통계값 구하기
- Series.count() : 개수
- Series.sum() : 합계
- Series.mean() : 평균
- Series.std() : 표준편차
- Series.var() : 분산
- Series.median() : 중앙값
- Series.mode() : 최빈값
- Series.quantile() : 분위수

open('파일이름',mode)



단독으로만 쓰지 않고 같이 글자들을 붙어서 쓸 수도 있다.

## open( , mode, ...) 함수

문자	의 미
'r'	읽기용으로 열기(기본값).
'w'	쓰기용으로 열기, 이미 같은 경로에 파일이 존재하면 파일 내용을 비움.
'x'	배타적 생성모드로 열기. 파일이 존재하면 IOError 예외 일으킴.
'a'	쓰기용으로 열기. 단, 'w'와는 달리 이미 같은 경로에 파일이 존재하는 경우 기존 내용에 덧붙임.
'b'	바이너리 모드
't'	텍스트 모드(기본값)
'+'	읽기/쓰기용으로 파일 읽기

바이너리 - 정수, 실수, 문자, 문자열, boolean, list, tuple, dictionary 같은 것들(Python 기준)

```
In [8]: # file에 Write 하기_20221103

file = open("test.txt", 'wt') # file을 open하겠다. # write로 설정했기 때문에 출력stream을 만들어줌.
file.write("hello")
file.close() # 내 자신의 주소값을 null로 만들어준다. # 파이썬이 메모리를 소멸시켜줌.
```

```
In [10]: # file에서 데이터 읽어오기
file = open("test.txt", 'rt') # rt는 디폴트이다. read, text모드
str = file.read()
print(str)
file.close()

hello
```


TCL	COMMIT ROLLBACK SAVEPOINT	<ul style="list-style-type: none"> <li>DML문에 의한 데이터 변경과 같은 작업은 트랜잭션으로 볼 수 있는데, 이러한 작업을 완료 또는 취소하기 위한 명령이다.</li> </ul>
DCL	GRANT REVOKE	<ul style="list-style-type: none"> <li>데이터베이스 및 해당 구조에 대한 액세스 권한을 부여하거나 제거한다.</li> </ul>

## grant

미국식[grænt]  영국식[gra:nt] 


(동사)

1 (특히 공식적법적으로) 승인[허락]하다

The bank finally granted a £500 loan to me. 

은행에서 마침내 나에게 500파운드의 융자를 승인했다.


2 (내키지 않지만) 인정하다

She's a smart woman, I grant you, but she's no genius. 

그녀가 똑똑한 여자인 건 해, 그건 인정하지. 하지만 천재는 아냐.

(명사)

1 (정부나 단체에서 주는) 보조금

student grants 

학비 보조금

영어사전 다른 뜻 2

## revoke

미국식[riˈvəʊk]  영국식[riˈvəʊk] 

(동사)

폐지[철회/취소]하다

영어사전 다른 뜻 1

```
In [6]: # file = open("text.txt", 'wt')

with open("test.txt", "w+t") as file:
    # with open()집중관리하기 시작할, #사용하고 있는지 안 하고 있는지 감시할, 안 하면 알아서 close()해줄,
    file.write("john3:16")
```

```
In [7]: with open("test.txt", 'a+t') as file: # w는 삭제하고 새로 write해줄, a는 기존 내용에 덧붙여짐,
        file.write(', World')
```

```
In [9]: with open("test.txt", 'r+t') as file:
        str = file.read()
        print(str) # john3:16, World # close()가 없어도 전혀 문제가 되지 않는다.
```

john3:16, World

### ### 텍스트 파일 읽기 / 쓰기

##### 1. 문자열을 담은 리스트를 파일에 쓰는 `writelines()` 메서드

```
In [10]: 1 lines = ["we'll find a way we always have - Interstellar\n", # \n은 줄바꿈이다,
                2 "I'll find you and I'll kill you - Taken\n",
                3 "I'll be back - Terminator 2\n"]
```

```
In [12]: with open('movie.quotes.txt', 'w') as file:
        for line in lines:
            file.write(line)
```

```
In [13]: with open('movie.quotes2.txt', 'w') as file:
        file.writelines(lines) # 반복문을 이용하지 않고 한번에 write 되게 해줄.
```

## 텍스트 파일 읽기 / 쓰기

### 2. 줄 단위로 텍스트를 읽는 `readline()` & `readlines()` 메서드

```
In [15]: with open('movie_quotes.txt', 'r') as file:
          line = file.readline()

          while line != '':
              print(line, end= ' ')
              line = file.readline() # 한 자 한 자씩 읽어옴.
```

we'll find a way we always have - Interstellar  
I'll find you and I'll kill you - Taken  
I'll be back - Terminator 2

```
In [16]: with open('movie_quotes.txt', 'r') as file:
          lines = file.readlines() # 라인별로 읽어옴.
          line = ''

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

we'll find a way we always have - Interstellar  
I'll find you and I'll kill you - Taken  
I'll be back - Terminator 2

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8b11052c-303b-42ad-9839-5d9f3efdd09d/test.txt>

john3:16, World

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b55f811b-4849-41a4-8aef-4c6ca1b1a3f7/movie\\_quotes.txt](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b55f811b-4849-41a4-8aef-4c6ca1b1a3f7/movie_quotes.txt)

we'll find a way we always have - Interstellar  
I'll find you and I'll kill you - Taken  
I'll be back - Terminator 2

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b36bb2ae-3312-4de4-9a1e-a28e4a365edc/movie\\_quotes2.txt](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/b36bb2ae-3312-4de4-9a1e-a28e4a365edc/movie_quotes2.txt)

we'll find a way we always have - Interstellar  
I'll find you and I'll kill you - Taken  
I'll be back - Terminator 2

## 텍스트 파일 읽기

줄 단위로 텍스트를 읽는 `readline()` & `readlines()`  
메서드

ex1)

```
with open('movie_quotes.txt', 'r') as file:
    line = file.readline()
    while line != "":
        print(line, end="")
        line = file.readline()
```

ex2)

```
with open('movie_quotes.txt', 'r') as file:
    lines = file.readlines()
    line = ""
    for line in lines:
        print(line, end="")
```

