# 데이터프로그래밍 기초 4일차

2026-1 DS Bootcamp

부산대학교
데이터사이언스전문대학원
석사과정 박민서

PUSAN NATIONAL UNIVERSITY
Graduate School of Data Science

# CONTENTS

1 Function (Level Up)

2 File Read & Write

3 Exception

4 Practice

5 Homework

## ▪ 재귀 함수 (Recursion)

- 함수 내부에서 원래의 함수를 다시 호출하는 것

```python
# 재귀 없이 팩토리얼 함수 구현
def factorial(n):
    res = 1
    if n == 0:
        return res

    for i in range(1, n + 1):
        res *= i

    return res

for num in range(11):
    print(f"{num}! = {factorial(num)}")
```

✓ 0.0s

```
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 재귀 함수 (Recursion)

- 함수 내부에서 원래의 함수를 다시 호출하는 것

```python
# 재귀로 팩토리얼 함수 구현
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

for num in range(11):
    print(f"{num}! = {factorial(num)}")
```

✓ 0.0s

```
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 재귀 함수 (Recursion)

- 함수 내부에서 원래의 함수를 다시 호출하는 것

```python
# 재귀로 피보나치 함수 구현
# 피보나치 수열
# 1, 2 번째 수 = 1
# 3번째 수 = 1번째 수+ 2번째 수
# ...
# n번째 수 = n - 1번째 수 + n - 2번째 수
import time

def fibonacci(n):
    if n == 1 or n == 2:
        return 1
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)

for i in range(1, 51, 10):
    start = time.time()
    result = fibonacci(i)
    end = time.time()
    print(f"피보나치 수열 {i}번째 수 = {result}, 실행 시간 = {end - start:.6f}(s)")
```

✓ 23.9s

```
피보나치 수열 1번째 수 = 1, 실행 시간 = 0.000000(s)
피보나치 수열 11번째 수 = 89, 실행 시간 = 0.000000(s)
피보나치 수열 21번째 수 = 10946, 실행 시간 = 0.001001(s)
피보나치 수열 31번째 수 = 1346269, 실행 시간 = 0.176147(s)
피보나치 수열 41번째 수 = 165580141, 실행 시간 = 23.810397(s)
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 메모화 (Memoization)

- 재귀 함수의 기하급수적인 호출 횟수를 줄이기 위한 방법
- 결과값을 미리 저장(메모)하고 이를 활용하여 다음 함수 호출의 결과에 이용

```python
# 재귀함수의 문제를 해결
fibo_dict = { # 결과를 저장할 자료형
    1: 1,
    2: 1
}

def fibonacci(n):
    if n in fibo_dict:
        return fibo_dict[n]  # 저장된 결과 이용 (시간 절약)
    else:
        res = fibonacci(n - 1) + fibonacci(n - 2)
        fibo_dict[n] = res   # fibo_dict에 메모
        return res

for i in range(1, 51, 10):
    start = time.time()
    result = fibonacci(i)
    end = time.time()
    print(f"피보나치 수열 {i}번째 수 = {result}, 실행 시간 = {end - start:.6f}(s)")
```

✓ 0.0s

```
피보나치 수열 1번째 수 = 1, 실행 시간 = 0.000000(s)
피보나치 수열 11번째 수 = 89, 실행 시간 = 0.000000(s)
피보나치 수열 21번째 수 = 10946, 실행 시간 = 0.000000(s)
피보나치 수열 31번째 수 = 1346269, 실행 시간 = 0.000000(s)
피보나치 수열 41번째 수 = 165580141, 실행 시간 = 0.000000(s)
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 메모화 (Memoization)

- 재귀 함수의 기하급수적인 호출 횟수를 줄이기 위한 방법
- 결과값을 미리 저장(메모)하고 이를 활용하여 다음 함수 호출의 결과에 이용

```python
# 재귀함수의 문제를 해결
fibo_dict = { # 결과를 저장할 자료형
    1: 1,
    2: 1
}

def fibonacci(n):
    if n in fibo_dict:
        return fibo_dict[n]  # 저장된 결과 이용 (시간 절약)
    else:
        res = fibonacci(n - 1) + fibonacci(n - 2)
        fibo_dict[n] = res   # fibo_dict에 메모
        return res

for i in range(1, 51, 10):
    start = time.time()
    result = fibonacci(i)
    end = time.time()
    print(f"피보나치 수열 {i}번째 수 = {result}, 실행 시간 = {end - start:.6f}(s)")
```

✓ 0.0s

```
피보나치 수열 1번째 수 = 1, 실행 시간 = 0.000000(s)
피보나치 수열 11번째 수 = 89, 실행 시간 = 0.000000(s)
피보나치 수열 21번째 수 = 10946, 실행 시간 = 0.000000(s)
피보나치 수열 31번째 수 = 1346269, 실행 시간 = 0.000000(s)
피보나치 수열 41번째 수 = 165580141, 실행 시간 = 0.000000(s)
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

- **여러 값 or 튜플로 반환**

```python
# 하나의 값 반환하는 함수
def my_func1(n):
    return n + 10

# 여러개의 값 반환하는 함수 (튜플로 반환)
def my_func2(n):
    return n + 10, n + 20

num1 = 100
res1 = my_func1(num1)
print(res1)

res2 = my_func2(num1)
print(res2)
```

✓ 0.0s

```
110
(110, 120)
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 지역변수 (Local Variable) & 전역변수 (Global Variable)

- 지역변수: 함수 내부에서만 사용하는 변수 ☞ 함수 밖에서 사용 불가
- 전역변수: 프로그램 전체에서 사용할 수 있는 변수 ☞ 함수 밖에서 정의한 후 사용 가능

```
def my_func():
    y = 10      # 함수 내부에서 y 변수 정의 (지역변수)
    print(y)    # 10 출력

my_func()
print(y)        # y가 정의가 되어있지 않음
```

⊗ 0.0s

```
10


---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[3], line 6
      3     print(y)    # 10 출력
      5 my_func()
----> 6 print(y)        # y가 정의가 되어있지 않음

NameError: name 'y' is not defined
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 지역변수 (Local Variable) & 전역변수 (Global Variable)

- 지역변수: 함수 내부에서만 사용하는 변수 ☞ 함수 밖에서 사용 불가
- 전역변수: 프로그램 전체에서 사용할 수 있는 변수 ☞ 함수 밖에서 정의한 후 사용 가능

```
y = 20      # 함수 밖에서 정의 (전역변수)
def my_func():
    y = 10      # 함수 내부에서 y 변수 정의 (지역변수)
    print(y)    # 10 출력

my_func()
print(y)
# 함수 밖에서 정의된 전역변수 y에 저장된 값 사용
# 지역변수 y는 사용되지 않음
```

✓ 0.0s

```
10
20
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 지역변수 (Local Variable) & 전역변수 (Global Variable)

- global: 함수 내에서 전역변수를 사용할 때 사용 ☞ 많이 사용할 수록 코드의 혼란 발생

```
y = 20      # 함수 밖에서 정의 (전역변수)
def my_func():
    global y
    y = 5       # 전역변수 y 값 20 -> 5으로 수정
    print(y)    # 5 출력

my_func()
print(y)
# 수정된 값 5가 다시 출력
```

✓ 0.0s

```
5
5
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 람다 (Lambda) 함수

- 간단한 함수를 보다 효율적으로 선언하는 방법
- “lambda 매개변수: 표현식”의 구조

```
# 간단한 함수 정의
def my_func(x):
    return x + 20

num1 = 10
res = my_func(num1)
print(res)

# 람다 함수를 이용한 my_func 함수 기능 구현
my_func2 = lambda x:x+20
res2 = my_func2(10)
print(res2)
```

✓ 0.0s

```
30
30
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 람다 (Lambda) 함수

- 간단한 함수를 보다 효율적으로 선언하는 방법
- “lambda 매개변수: 표현식”의 구조

```
# lambda 함수 활용 (리스트 정렬)
nums = [10, -3, 7, -1]
nums.sort() # 기본 정렬
print(f"기본 정렬 결과: {nums}")

nums.sort(key=lambda x: abs(x)) # 절댓값 기준으로 정렬
print(f"abs 기준 정렬 결과: {nums}")

nums.sort(key=lambda x: -abs(x)) # 절댓값 기준으로 내림차순 정렬
# nums.sort(key=lambda x: abs(x), reverse=True)
print(f"abs 기준 내림차순 정렬 결과: {nums}")
```

✓ 0.0s

```
기본 정렬 결과: [-3, -1, 7, 10]
abs 기준 정렬 결과: [-1, -3, 7, 10]
abs 기준 정렬 결과: [10, 7, -3, -1]
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 람다 (Lambda) 함수

- 간단한 함수를 보다 효율적으로 선언하는 방법
- “lambda 매개변수: 표현식”의 구조

```
# 1차 정렬 기준: x // 10 값 기준 오름차순 -> 십의 자리 기준 오름차순
# 2차 정렬 기준: -x값 기준 오름차순 -> 내림차순 정렬
nums = [12, 5, 23, 7, 18, 3, 30]
nums.sort(reverse=True)
nums.sort(key=lambda x:x // 10)
print(f"정렬 결과: {nums}")

nums = [12, 5, 23, 7, 18, 3, 30]
nums.sort(key=lambda x: (x // 10, -x))  # 여러 기준도 lambda함수 활용해 한 번에 입력
print(f"정렬 결과: {nums}")
```

✓ 0.0s

```
정렬 결과: [7, 5, 3, 18, 12, 23, 30]
정렬 결과: [7, 5, 3, 18, 12, 23, 30]
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 람다 (Lambda) 함수

- 간단한 함수를 보다 효율적으로 선언하는 방법
- “lambda 매개변수: 표현식”의 구조

```python
# lambda 함수 활용 (딕셔너리 정렬)
scores = {
    "민수": 90,
    "영희": 85,
    "철수": 95
}

result = sorted(scores.items(), key=lambda x: x[1])
print(f"key=lambda x: x[1] 정렬 결과: {result}")

result = sorted(scores.items(), key=lambda x: x[1], reverse=True)
print(f"key=lambda x: x[1], reverse=True: {result}")
```

✓ 0.0s

```
key=lambda x: x[1] 정렬 결과: [('영희', 85), ('민수', 90), ('철수', 95)]
key=lambda x: x[1], reverse=True: [('철수', 95), ('민수', 90), ('영희', 85)]
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 람다 (Lambda) 함수

- 간단한 함수를 보다 효율적으로 선언하는 방법
- “lambda 매개변수: 표현식”의 구조

```
scores = {
    "민수": 90,
    "영희": 85,
    "철수": 90,
    "지수": 85,
    "현우": 90
}
# dict형.items() -> (key, value)로 이루어진 리스트
# x[0] = key, x[1] = value

result = sorted(scores.items(), key=lambda x: (x[1], x[0]))
print(f"key=lambda x: (x[1], x[0]) 정렬")
print(f"결과: {result}")
```

✓ 0.0s

```
key=lambda x: (x[1], x[0]) 정렬
결과: [('영희', 85), ('지수', 85), ('민수', 90), ('철수', 90), ('현우', 90)]
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 매핑 (Map) 함수

- list의 요소를 함수에 넣고 반환된 값으로 새로운 리스트를 구성
- map(함수, 리스트)

```python
# 제곱을 반환해주는 함수
def square(x):
    return x * x

my_list = [1, 2, 3, 4, 5]
res1 = map(square, my_list)
# my_list의 요소를 square 함수를 거쳐 나온 반환값으로 변환
print(res1) # <map ~~~> = 제너레이터 반환 (generator)
res2 = list(map(square, my_list))
print(res2) # list로 반환
```

✓ 0.0s

```
<map object at 0x000001A2A1487B80>
[1, 4, 9, 16, 25]
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 매핑 (Map) 함수

- list의 요소를 함수에 넣고 반환된 값으로 새로운 리스트를 구성
- map(함수, 리스트)

```python
# 제곱을 반환해주는 함수
def square(x):
    return x * x

my_list = [1, 2, 3, 4, 5]
res1 = map(square, my_list)
# my_list의 요소를 square 함수를 거쳐 나온 반환값으로 변환
print(res1) # <map ~~~> = 제너레이터 반환 (generator)
res2 = list(map(square, my_list))
print(res2) # list로 반환
```

✓ 0.0s

```
<map object at 0x000001A2A1487B80>
[1, 4, 9, 16, 25]
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 매핑 (Map) 함수

- list의 요소를 함수에 넣고 반환된 값으로 새로운 리스트를 구성
- map(함수, 리스트)

```
my_list = list(map(int, input().split()))
# 1 2 3 4 5 입력 -> [1, 2, 3, 4, 5] 반환
print(my_list)

my_list = list(map(float, input().split()))
# 1 2 3 4 5 입력 -> [1.0, 2.0, 3.0, 4.0, 5.0] 반환
print(my_list)

my_list = list(map(str, input().split()))
# 1 2 3 4 5 입력 -> ["1", "2", "3", "4", "5"] 반환
print(my_list)
```

✓ 13.6s

```
1 2 3 4 5
Press 'Enter' to confirm your input or 'Escape' to cancel
[1, 2, 3, 4, 5]
[1.0, 2.0, 3.0, 4.0, 5.0]
['1', '2', '3', '4', '5']
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# File Read & Write

## ▪ 파일 처리 (.txt 파일 처리)

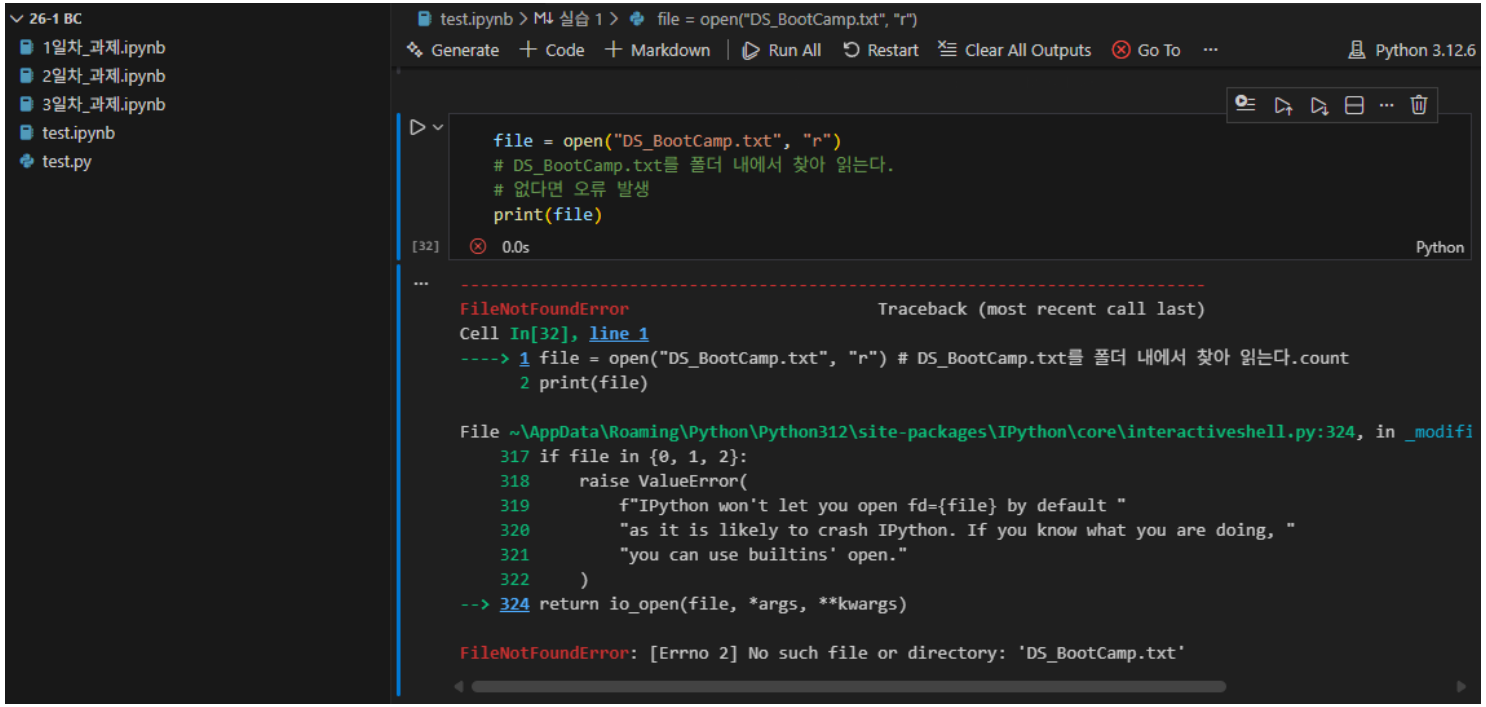
- open( ): 파일을 열 때 사용
- 변수 = open('파일 경로', '읽기 모드')
- 파일 경로 ("C:₩Users₩minsu₩OneDrive₩바탕 화면₩26-1 BC"에서 작업 중)
  - 절대 경로: "C:₩Users₩minsu₩OneDrive₩바탕 화면₩26-1 BC₩[디렉토리 명]₩...₩[파일명]"
  - 상대 경로: "[디렉토리 명]₩...₩[파일명]" ☞ 작업 폴더 기준으로 다음 경로

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# File Read & Write

## ▪ 파일 열기 & 닫기

- open( ): 파일을 열 때 사용　　close( ): 파일을 닫을 때 사용
- 변수 = open('파일 경로', '읽기 모드')

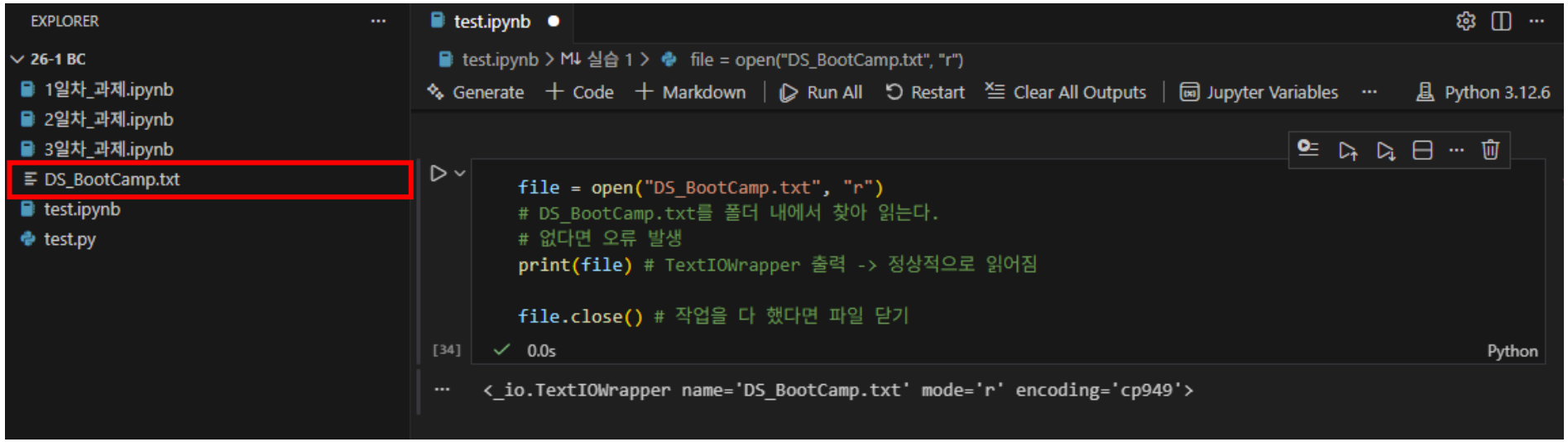



윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# File Read & Write

## ▪ 파일 열기 & 닫기

- open( ): 파일을 열 때 사용　　close( ): 파일을 닫을 때 사용
- 변수 = open('파일 경로', '읽기 모드')




윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# File Read & Write

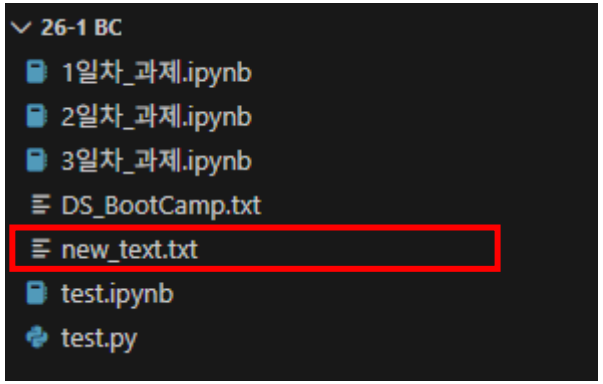


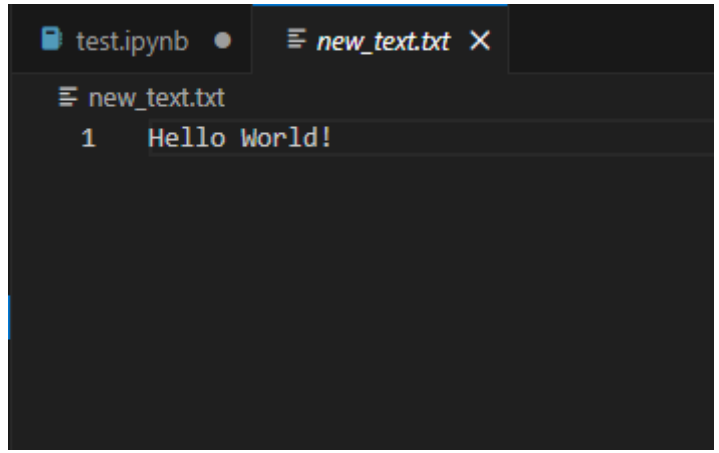
## ▪ 파일 열기 & 닫기

- open( ): 파일을 열 때 사용　　close( ): 파일을 닫을 때 사용
- 변수 = open('파일 경로', '읽기 모드')

```
file = open("new_text.txt", "w")
# 현재 작업 폴더에 new_text.txt 파일을 생성하여 쓰는 작업
file.write("Hello World!")

file.close() # 작업을 다 했다면 파일 닫기
```

✓ 0.0s




윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

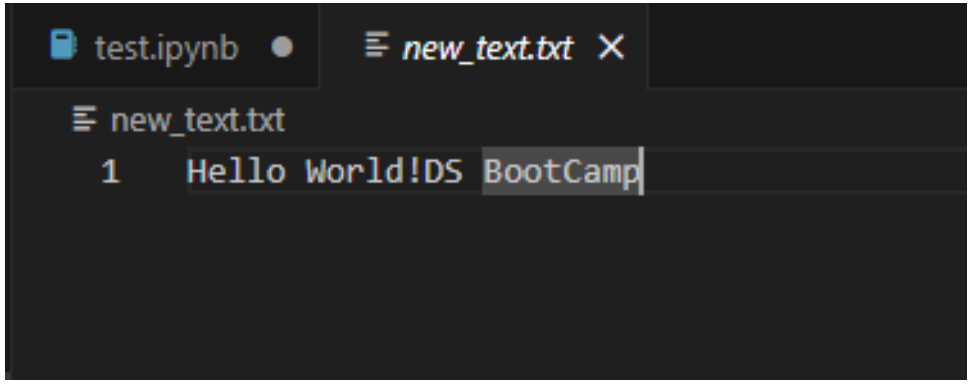
# File Read & Write

## ▪ 파일 열기 & 닫기

- open( ): 파일을 열 때 사용　　close( ): 파일을 닫을 때 사용
- 변수 = open('파일 경로', '읽기 모드')

```
file = open("new_text.txt", "a")
# 현재 작업 폴더에 new_text.txt 파일을 열어 덧붙이는 작업
# 파일이 없다면 생성해서 작업
file.write("DS BootCamp")

file.close() # 작업을 다 했다면 파일 닫기
```

✓ 0.0s




윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# File Read & Write

## ▪ 파일 열기 & 닫기

- with: 파일의 open과 close 함수 사용의 실수를 방지 ☞ 작업 중 에러가 나도 자동으로 close
- **with open('파일 경로', '읽기 모드') as 변수명:**
  **작업 코드**

```
file = open("new_text.txt", "a")
# 현재 작업 폴더에 new_text.txt 파일을 열어 덧붙이는 작업
# 파일이 없다면 생성해서 작업
file.write("DS BootCamp")

file.close() # 작업을 다 했다면 파일 닫기
```

✓ 0.0s

=

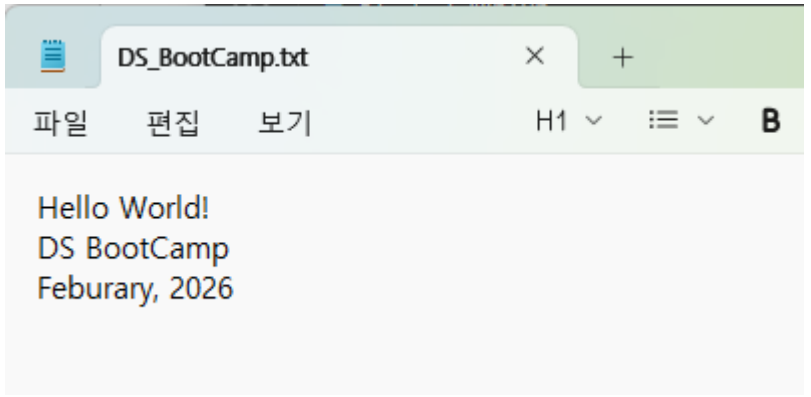
```
with open("new_text.txt", "a") as file:
    file.write("DS BootCamp")
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# File Read & Write

## ▪ 파일 읽기

- read( ): 파일 전체를 하나의 문자열로 읽음




```
with open("DS_BootCamp.txt", "r") as f:
    content = f.read()

print(content)
```

✓ 0.0s

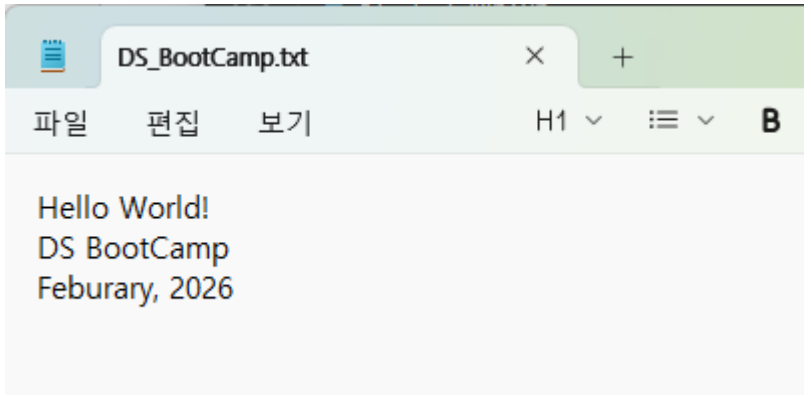
```
Hello World!
DS BootCamp
Feburary, 2026
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# File Read & Write

## ▪ 파일 읽기

- readline( ): 파일 내에서 한 줄만 읽음 (“₩n”까지)
- 일반적으로 반복문과 함께 사용




```
with open("DS_BootCamp.txt", "r") as f:
    content = f.readline()

print(content)
```

✓ 0.0s

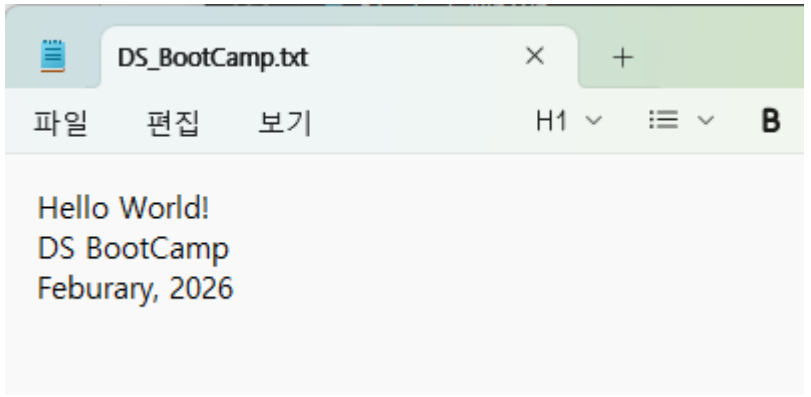
```
Hello World!
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# File Read & Write

## ▪ 파일 읽기

- readlines( ): 파일 내의 모든 line을 읽고 리스트로 반환
- 각 요소는 한 줄에 있는 문자열




```
with open("DS_BootCamp.txt", "r") as f:
    content = f.readlines()

print(content)
```

✓ 0.0s

```
['Hello World!\n', 'DS BootCamp\n', 'Feburary, 2026']
```

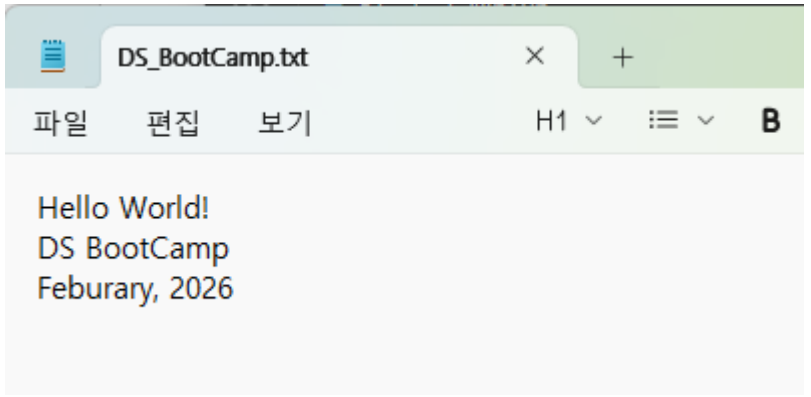
윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# File Read & Write



## ▪ 파일 읽기

- readlines( ): 파일 내의 모든 line을 읽고 리스트로 반환
- 각 요소는 한 줄에 있는 문자열




```
with open("DS_BootCamp.txt", "r") as f:
    lines = f.readlines()
    for line in lines:
        print(line.strip())
```

✓ 0.0s

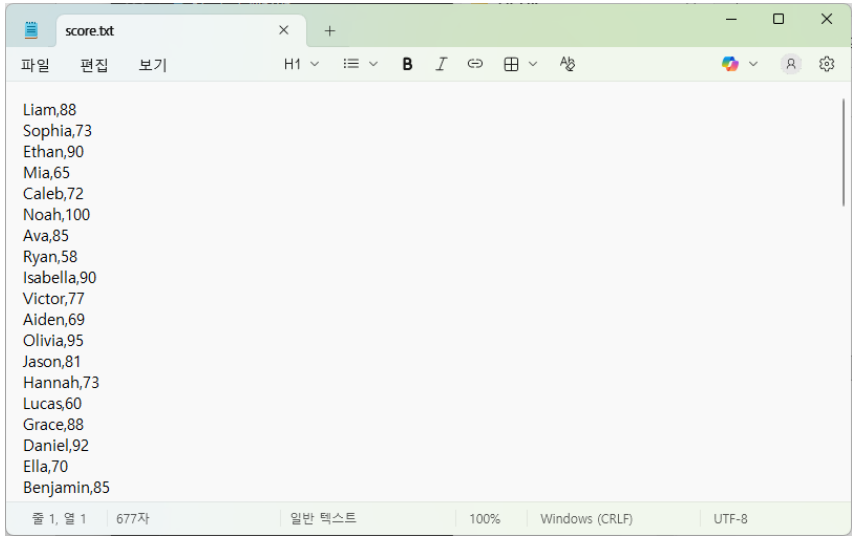
```
Hello World!
DS BootCamp
Feburary, 2026
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# Q&A

## ▪ 실습 1

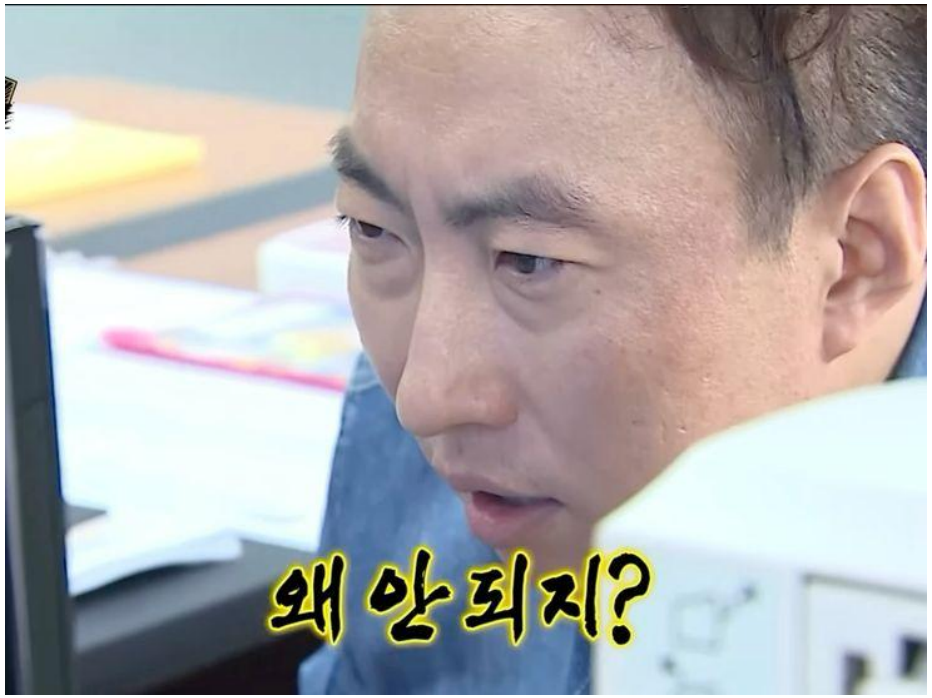
- ‘score.txt’ 파일을 열어 다음 작업을 수행하세요.
  - 학생 수를 출력하세요.
  - 전체 학생들의 평균 점수를 구하세요.
    (가능하다면 소수점 둘째자리까지)
  - 70점 이상의 점수를 받은 학생 수를 출력하세요.
  - 최고 점수를 받은 학생들 이름을 출력하세요. (여러 학생 존재 가능)

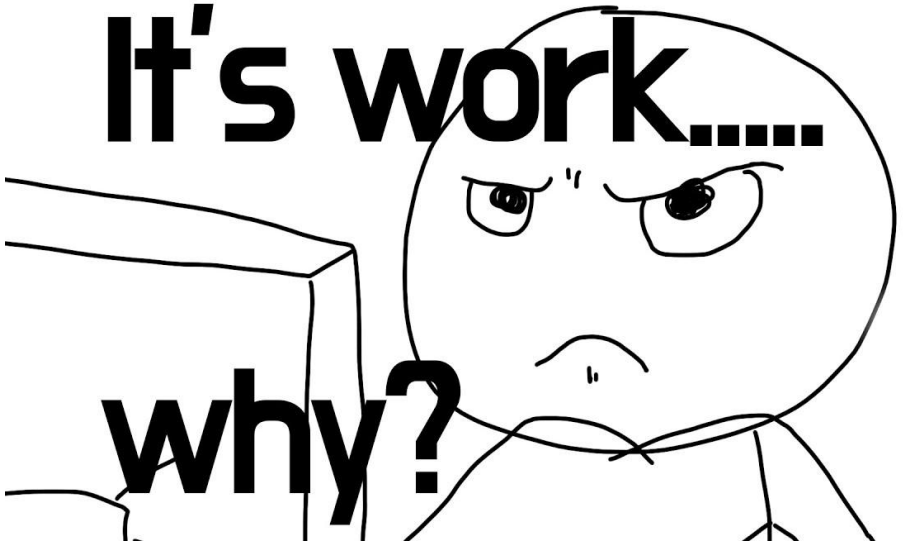



```
학생 수: 68명
평균 점수: 77.96점
최고 점수 / 최저 점수: 100 / 47
70점 이상 학생 수: 50
최고점을 받은 학생 목록: ['Noah', 'Logan']
```

# Exception

- 에러 (Error) 해석




윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

- 에러 (Error) 해석

```
num = int("BootCamp")
print(f"number : {num}")
```

0.0s

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[54], line 1
----> 1 num = int("BootCamp")
      2 print(f"number : {num}")

ValueError: invalid literal for int() with base 10: 'BootCamp'
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# Exception

- 에러 (Error) 해석

```
a = 1

if a < 10
    print("a < 10")
```

0.0s

```
Cell In[55], line 3
  if a < 10
          ^
SyntaxError: expected ':'
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# Exception

- 에러 (Error) 해석

```
file = open("DS_BootCamp.txt", "r")
# DS_BootCamp.txt를 폴더 내에서 찾아 읽는다.
# 없다면 오류 발생
print(file)
```

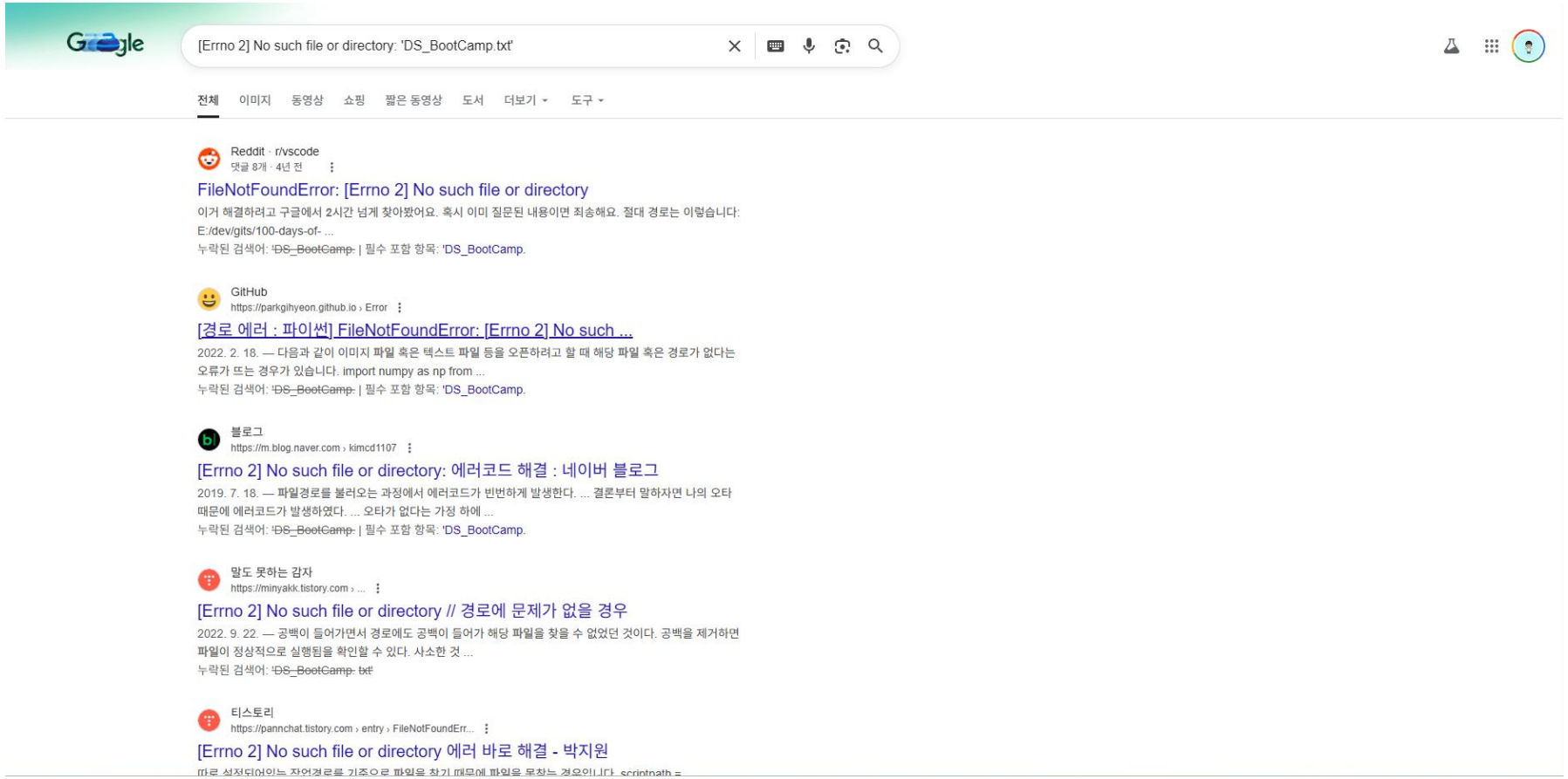
0.0s Python

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[32], line 1
----> 1 file = open("DS_BootCamp.txt", "r") # DS_BootCamp.txt를 폴더 내에서 찾아 읽는다.count
      2 print(file)

File ~\AppData\Roaming\Python\Python312\site-packages\IPython\core\interactiveshell.py:324, in _modifi
    317 if file in {0, 1, 2}:
    318     raise ValueError(
    319         f"IPython won't let you open fd={file} by default "
    320         "as it is likely to crash IPython. If you know what you are doing, "
    321         "you can use builtins' open."
    322     )
--> 324 return io_open(file, *args, **kwargs)

FileNotFoundError: [Errno 2] No such file or directory: 'DS_BootCamp.txt'
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# Exception

- 에러 (Error) 해석




윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# Exception

## ▪ 예외 처리 (Exception Handling)

- 예외: 프로그램 실행 중 발생하는 오류
- 예외 처리: 예외를 해결하는 모든 것

```
my_str = "안녕하세요"

print(my_list[1])
```

0.3s

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1], line 3
      1 my_str = "안녕하세요"
----> 3 print(my_list[1])

NameError: name 'my_list' is not defined
```

해결

```
my_str = "안녕하세요"

my_list = [1, 2, 3, 4, 5]
print(my_list[1])
```

0.0s

```
2
```

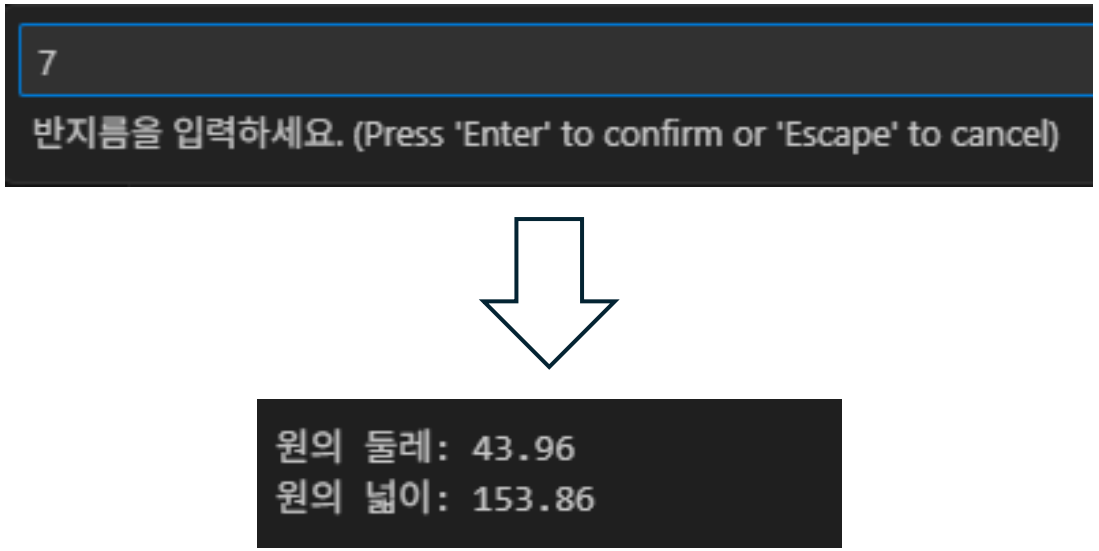
윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

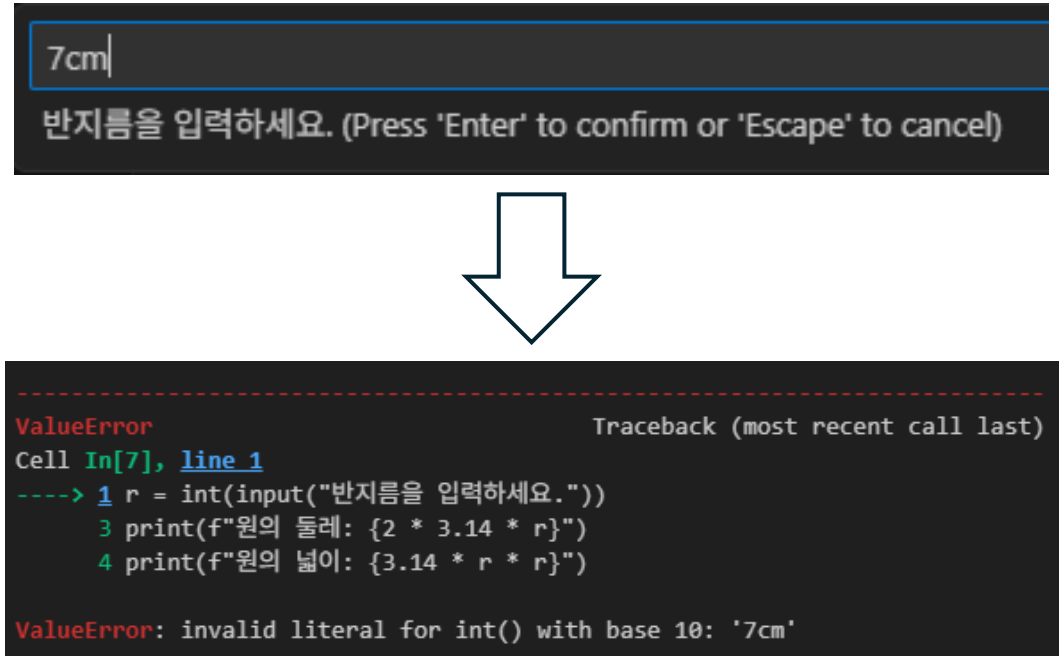
# Exception

## ▪ 예외 처리 (Exception Handling)

- 조건문 사용한 예외 처리

```
r = int(input("반지름을 입력하세요."))

print(f"원의 둘레: {2 * 3.14 * r}")
print(f"원의 넓이: {3.14 * r * r}")
```




윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# Exception

## ▪ 예외 처리 (Exception Handling)

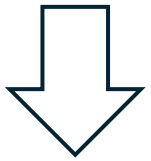
- 조건문 사용한 예외 처리
- 모든 예외가 발생할 경우 고려

```
r = input("반지름을 입력하세요. (정수)")

if r.isdigit():  # isdigit(): 입력한 문자열이 숫자인지 확인
    r = int(r)
    print(f"원의 둘레: {2 * 3.14 * r}")
    print(f"원의 넓이: {3.14 * r * r}")
else:
    print("정수가 입력되지 않았습니다.")
```

✓ 1.7s

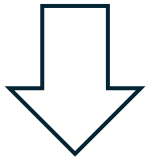
```
정수가 입력되지 않았습니다.
```

```
7
반지름을 입력하세요. (정수) (Press 'Enter' to confirm or 'Escape' to cancel)
```



```
원의 둘레: 43.96
원의 넓이: 153.86
```

```
7cm
반지름을 입력하세요. (정수) (Press 'Enter' to confirm or 'Escape' to cancel)
```



```
정수가 입력되지 않았습니다.
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# Exception

## ▪ 예외 처리 (Exception Handling)

- try ~ except 구문: 조건문을 활용한 예외처리의 단점 보완 ☞ “모든 예외는 예측할 수 없다.”
- try:

  예외 발생 가능성 있는 코드

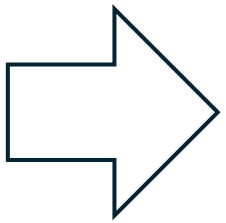
  except:

  예외가 발생했을 때 실행할 코드

```
r = input("반지름을 입력하세요. (정수)")

if r.isdigit():  # isdigit(): 입력한 문자열이 숫자인지 확인
    r = int(r)
    print(f"원의 둘레: {2 * 3.14 * r}")
    print(f"원의 넓이: {3.14 * r * r}")
else:
    print("정수가 입력되지 않았습니다.")
```

✓ 1.7s

```
정수가 입력되지 않았습니다.
```



```
try:
    r = int(input("반지름을 입력하세요."))

    print(f"원의 둘레: {2 * 3.14 * r}")
    print(f"원의 넓이: {3.14 * r * r}")
except:
    print("프로그램이 잘못되었습니다.")
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 예외 처리 (Exception Handling)

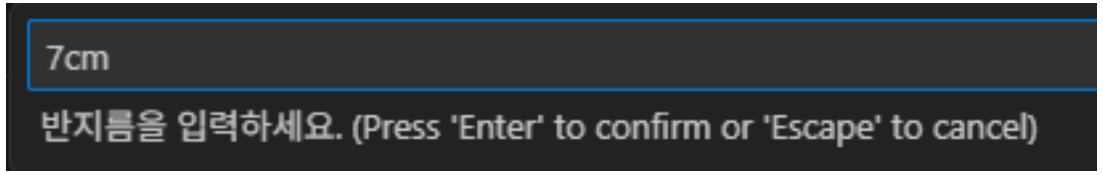
- try ~ except 구문: 조건문을 활용한 예외처리의 단점 보완 ☞ “모든 예외는 예측할 수 없다.”
- **try:**

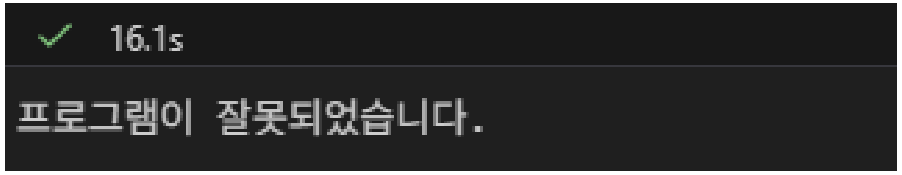
  **예외 발생 가능성 있는 코드**

  **except:**

  **예외가 발생했을 때 실행할 코드**

```
try:
    r = int(input("반지름을 입력하세요."))

    print(f"원의 둘레: {2 * 3.14 * r}")
    print(f"원의 넓이: {3.14 * r * r}")
except:
    print("프로그램이 잘못되었습니다.")
```



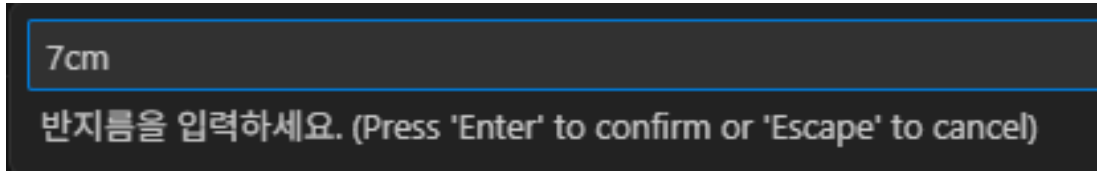



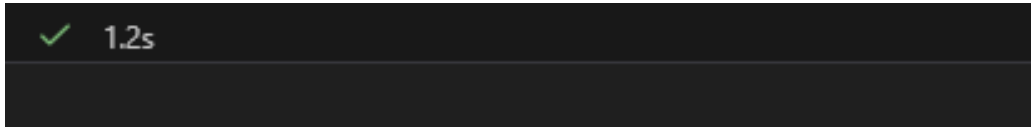

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

## ▪ 예외 처리 (Exception Handling)

- pass: 아무것도 하지 않고 지나갈 때 사용

```
try:
    r = int(input("반지름을 입력하세요."))

    print(f"원의 둘레: {2 * 3.14 * r}")
    print(f"원의 넓이: {3.14 * r * r}")
except:
    pass
```



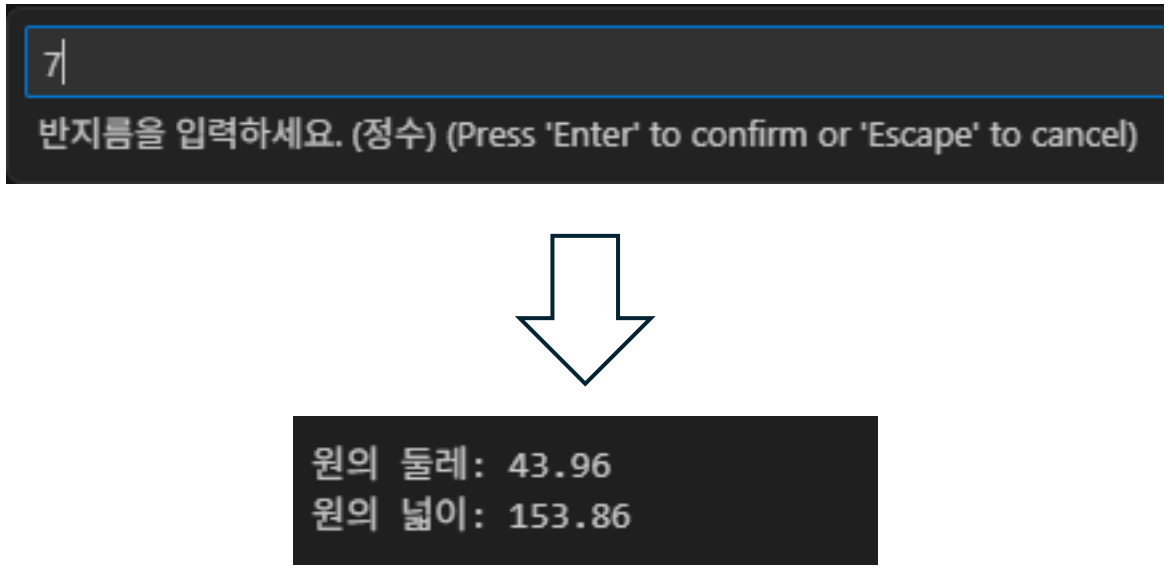

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

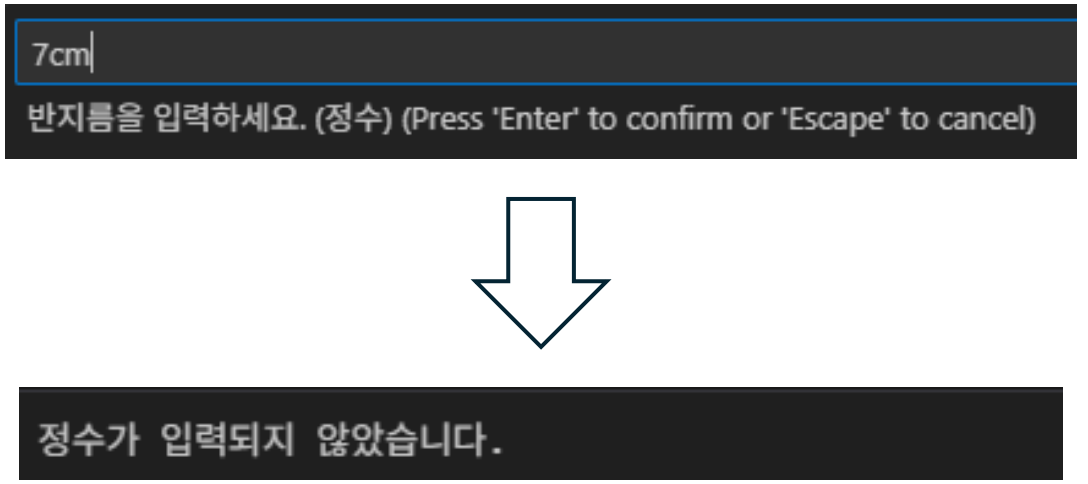
# Exception

## ▪ 예외 처리 (Exception Handling)

- try except + else
- try문에 있는 코드가 정상 작동할 때, 그 다음 실행할 코드를 else문에 작성

```
try:
    r = int(input("반지름을 입력하세요. (정수)"))
except:
    print("정수가 입력되지 않았습니다.")
else:
    print(f"원의 둘레: {2 * 3.14 * r}")
    print(f"원의 넓이: {3.14 * r * r}")
```



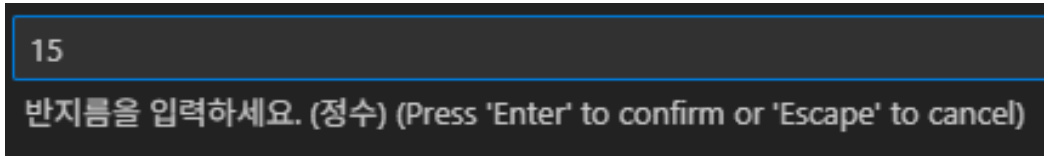




윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

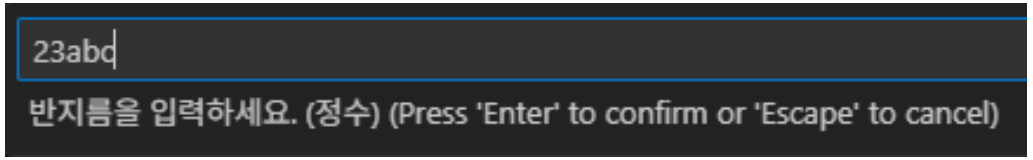
## ▪ 예외 처리 (Exception Handling)

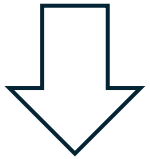
- try except + else + finally
- finally: 예외가 발생하든 그렇지 않든 무조건 실행

```
try:
    r = int(input("반지름을 입력하세요. (정수)"))
    print(f"원의 둘레: {2 * 3.14 * r}")
    print(f"원의 넓이: {3.14 * r * r}")
except:
    print("정수가 입력되지 않았습니다.")
else:
    print("예외가 발생하지 않았습니다.")
finally:
    print("프로그램이 끝났습니다.")
```




```
원의 둘레: 94.2
원의 넓이: 706.5
예외가 발생하지 않았습니다.
프로그램이 끝났습니다.
```



```
정수가 입력되지 않았습니다.
프로그램이 끝났습니다.
```

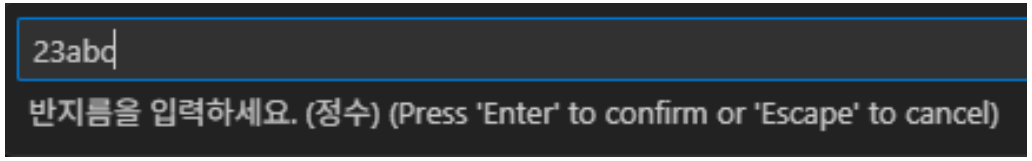
윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

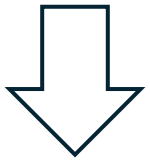
# Exception

## ▪ 예외 처리 (Exception Handling)

- try except ~ as
- Exception: 모든 예외의 종류를 포함한 키워드

```
try:
    r = int(input("반지름을 입력하세요. (정수)"))
    print(f"원의 둘레: {2 * 3.14 * r}")
    print(f"원의 넓이: {3.14 * r * r}")
except:
    print("정수가 입력되지 않았습니다.")
else:
    print("예외가 발생하지 않았습니다.")
finally:
    print("프로그램이 끝났습니다.")
```



```
에러 type: <class 'ValueError'>
exception: invalid literal for int() with base 10: '23abc'
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

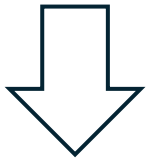
# Exception

## ▪ 예외 처리 (Exception Handling)

- raise: 예외를 강제로 발생시키는 키워드

```
r = int(input("반지름을 입력하세요. (정수)"))

if r <= 0:
    raise NotImplementedError

print(f"원의 둘레: {2 * 3.14 * r}")
print(f"원의 넓이: {3.14 * r * r}")
```

```
0
반지름을 입력하세요. (정수) (Press 'Enter' to confirm or 'Escape' to cancel)
```



```
---------------------------------------------------------------------------
NotImplementedError                       Traceback (most recent call last)
Cell In[10], line 4
      1 r = int(input("반지름을 입력하세요. (정수)"))
      3 if r <= 0:
----> 4     raise NotImplementedError
      6 print(f"원의 둘레: {2 * 3.14 * r}")
      7 print(f"원의 넓이: {3.14 * r * r}")

NotImplementedError:
```

윤인성, 「혼자 공부하는 파이썬 개정판」, 한빛미디어, 2022, 292-396쪽
https://www.nise.go.kr/sedu/pe/pe3/page3_07.html

# Homework

## ▪ 4일차 과제

- GitHub 사이트에서 “4일차_과제.ipynb” 다운로드
- 코드 작성 후, “**본인이름**_4일차_과제.ipynb”로 저장
- 저장한 과제 파일 전송 (이메일 주소: minsuh99@pusan.ac.kr)

  기한: ~ 2/6 PM 13:59:59

# Q&A