

8장. 예외처리 & 파일 입,출력



목 차

1

예외 처리

2

파일 입출력

3

파일 입출력 응용 프로그램

에러(Error)와 예외(Exception)

에러(Error)

- **구문(syntax) 오류** : 문법에 맞지 않거나 오타가 났을 경우 발생하는 오류
IDE에서 실행 전에 알 수 있음

예외(Exception)

- **실행(runtime) 오류** : 문법적인 오류는 없지만 실행(run) 될 때 에러가 발생하는 것을 말한다.

예) 파일을 읽어 사용하려는데 파일이 없는 경우,
리스트 값을 출력하려는데 리스트 요소가 없는 경우 등..

에러가 발생되면 프로그램의 동작이 중지 또는 종료된다



예외(Exception)

← → ↺ docs.python.org/ko/3/tutorial/errors.html

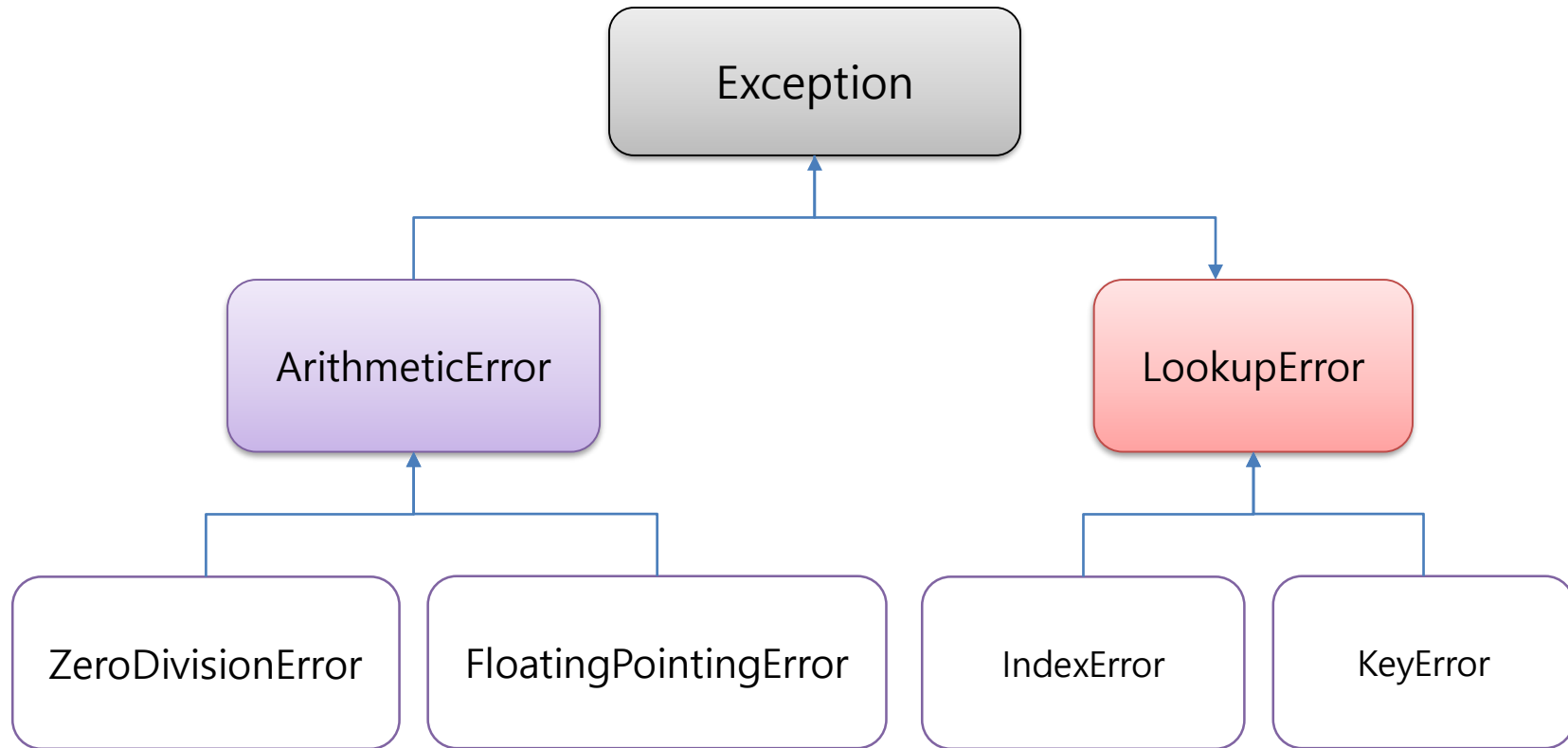
Python » Korean 3.10.0 3.10.0 Documentation » 파이썬 자습서 » 8. 에러와 예외

8.2. 예외 ¶

문장이나 표현식이 문법적으로 올바르다 할지라도, 실행하려고 하면 에러를 일으킬 수 있습니다. 실행 중에 감지되는 에러들을 *예외*라고 부르고 무조건 치명적이지는 않습니다: 파이썬 프로그램에서 이것들을 어떻게 다루는지 곧 배우게 됩니다. 하지만 대부분의 예외는 프로그램이 처리하지 않아서, 여기에서 볼 수 있듯이 에러 메시지를 만듭니다:

```
>>> 10 * (1/0)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: division by zero
>>> 4 + spam*3
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'spam' is not defined
>>> '2' + 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
```

Exception 계층도



예외 처리(Exception)

❖ **예외 처리 방법** : try ~ except 구문

try:

예외가 발생할 가능성이 있는 코드

except 예외 클래스:

예외가 발생했을 경우 실행 코드

예외 처리(Exception)

- 숫자를 입력할 곳에 문자를 입력하여 예외 발생

숫자를 입력하세요 : ㅍ

Traceback (most recent call last):

File "[C:/pyworks/cho8/try_except/value_error.py](#)", line 3,

x = int(input("숫자를 입력하세요 : "))

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

```
while True:
```

```
    try:
```

```
        x = input("숫자를 입력하세요 : ")
```

```
        num = int(x)
```

```
        print(num)
```

```
        break
```

```
    except ValueError:
```

```
        print("유효한 숫자가 아닙니다. 다시 입력해 주세요.")
```

예외 처리(Exception)

숫자 추측 게임

- 예외 처리

```
import random

com = random.randint(1, 30)
while True:
    try:
        guess = int(input("맞혀보세요(1~30): "))

        if guess > 30 or guess < 1:
            print("입력 범위가 아니에요. 다시 입력해 주세요: ")
        elif guess == com:
            print("정답!!")
            break
        elif guess > com:
            print("너무 커요")
        elif guess < com:
            print("너무 작아요")

    except ValueError:
        print("앗! 유효한 숫자가 아닙니다. 다시 입력해 주세요")
```


다중 예외 처리(Exception)

다중 try ~ except 구문

try:

실행 코드

except 오류 Type1 as e

문제 발생시 실행코드

except 오류 Type2 as e

문제 발생시 실행코드

오류 메시지 표시

try:

```
data = [59, 0, 4, 116, 303]
```

```
x = input("정수 입력(0~4까지 입력해주세요)>")
```

```
num = int(x)
```

```
print(data[num])
```

```
except ValueError as e:
```

```
print("값에 문제가 있습니다.")
```

```
except IndexError as e:
```

```
print("0~4까지 입력해주세요")
```

예외 처리(Exception)

❖ try ~ except ~ finally 구문

finally 구문을 반드시 실행한다.

```
def divide(x, y):  
    try:  
        result = x / y  
        print(result)  
    except ZeroDivisionError:  
        print("0으로 나눌수 없습니다.")  
    finally:  
        print("여기는 반드시 수행되는 구간입니다.")
```

```
#divide(2, 1)  
divide(2, 0)
```

raise ~ Exception

▶ 예외 처리를 미루기 – **raise** 사용

사용하는 곳에서 발생 시킴(강제로 프로그램 종료)

```
class Animal:
    def cry(self):
        raise NotImplementedError

class Dog(Animal):
    #pass
    def cry(self):
        print("왈~ 왈~")

class Cat(Animal):
    def cry(self):
        print("야~옹 야~옹")

dog = Dog()
dog.cry()

cat = Cat()
cat.cry()
```

반드시 구현하도록 만드는 예외 처리

try ~ except ~ else 구문

```
# try ~ except ~ else 구문  
# 에러가 없으면 try ~ else 실행  
# 에러가 있으면 try ~ except 실행
```

```
try:  
    print("1번")  
    with open('2021kbo.txt') as f:  
        lines = f.readlines()
```

```
except:  
    print("2번")
```

```
else:  
    print("3번")  
    for i in lines:  
        print(i, end='')
```

1번
3번
NC
키움
기아
삼성

```
try:  
    print("1번")  
    with open('2022kbo.txt') as f:  
        lines = f.readlines()
```

에러가 있으면

1번
2번

사용자 정의 예외 처리

➤ 예외 만들기

프로그램 수행 도중 특별한 경우 예외를 만들어서 사용한다.
이때 내장 클래스인 Exception 클래스를 상속하여 만든다.

```
class MyError(Exception):  
    pass  
  
def say_nick(nick):  
    if nick == '바보':  
        raise MyError()  
    print(nick)  
  
say_nick('영웅')  
say_nick('바보')
```



```
class MyError(Exception):  
    def __str__(self):  
        return "허용되지 않는 별명입니다."  
  
def say_nick(nick):  
    if nick == '바보':  
        raise MyError()  
    print(nick)  
  
try:  
    say_nick('영웅')  
    say_nick('바보')  
except MyError as e:  
    print(e)
```

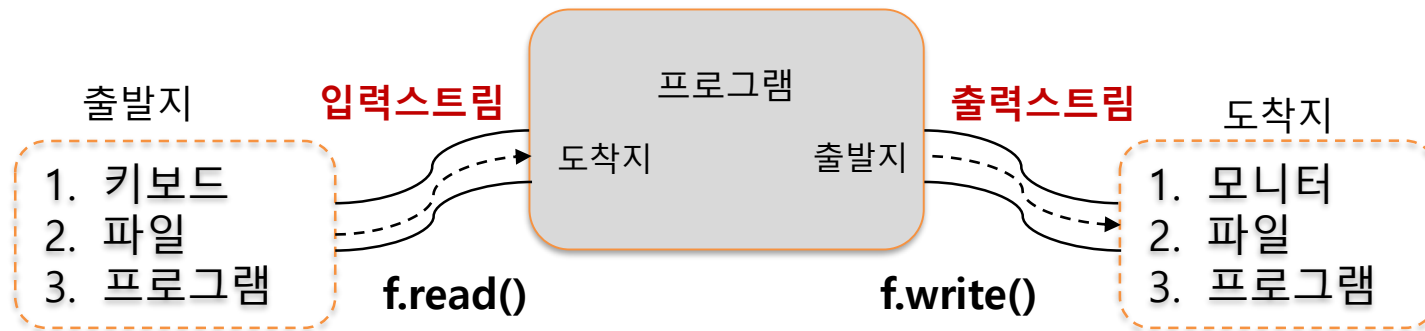
메시지가 출력되려면
__str__ 메서드를 구현
해야 함

입, 출력 스트림

● 스트림(stream)

자료흐름이 물의 흐름과 같다는 뜻이다. 입출력 장치는 매우 다양하기 때문에 프로그램 호환성이 떨어짐

- 입력 스트림 – 동영상을 재생하기 위해 동영상 파일에서 자료를 읽을때 사용함
- 출력 스트림 – 사용자가 쓴 글을 파일에 저장할 때는 출력 스트림 사용함

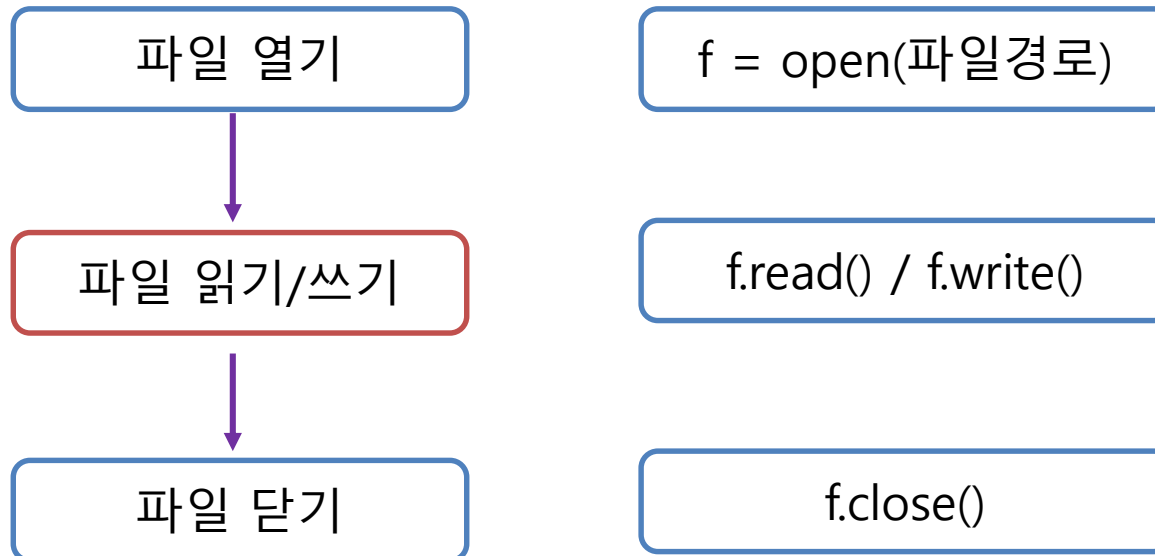


파일 입출력

● 파일 입출력의 필요성

프로그램 실행 중에 메모리에 저장된 데이터는 프로그램이 종료되면 사라진다.
데이터를 프로그램이 종료된 후에도 계속해서 사용하려면 파일에 저장하고
필요할 때 파일을 읽어서 데이터를 사용할 수 있다.

● 파일 입출력 프로세스



파일 쓰기

● 파일 쓰기

write() 함수 – 파일에 내용을 쓰는 함수로 문자열만 쓰기 가능

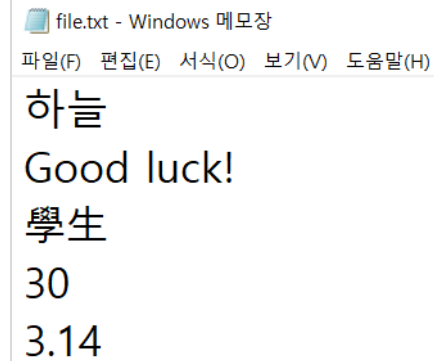
`open("파일 위치", 'w')`

모드	설명
w	쓰기
r	읽기
a	추가 쓰기

```
f = open("C:/pyfile/file.txt", 'w')

f.write("하늘\n")
f.write('Good luck!\n')
f.write('學生\n')
f.write('30\n')
f.write('3.14\n')

f.close()
```



file.txt - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

하늘
Good luck!
學生
30
3.14

파일 쓰기(추가모드)

● 파일 쓰기(추가 모드)

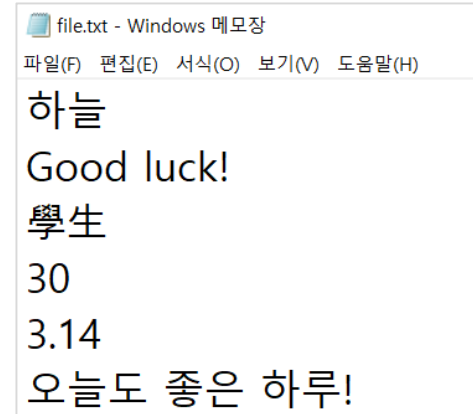
'w' 모드는 초기화 되어 저장되므로 추가할때는 'a'모드 사용

`open("파일 위치", 'a')`

```
f = open("C:/pyfile/file.txt", 'w')

f.write("하늘\n")
f.write('Good luck!\n')
f.write('學生\n')
f.write('30\n')
f.write('3.14\n')
f.write("오늘도 좋은 하루!")

f.close()
```



file.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

하늘
Good luck!
學生
30
3.14
오늘도 좋은 하루!

파일 쓰기

- 문자형이 아닌 숫자형을 파일에 쓰기

정수나 실수형인 경우 - 문자열 포매팅 방법 사용

```
f = open('c:/pyfile/number.txt', 'w')
#f.write(14)
f.write("%d\n" % 14)
f.write("%.1f\n" % 7.3)
sum_v = "%d" % eval("100+200") # eval(문자열식) -> 숫자 계산
f.write(sum_v)
f.close()
```

```
Traceback (most recent call last):
  File "C:/pyworks/cho8/fwrite2.py", line 3, in <module>
    f.write(300)
TypeError: write() argument must be str, not int
```

타입 에러

파일 읽기

- 파일 읽기

`read()` 함수 – 파일의 내용 전체를 읽어서 문자열로 돌려준다.

`open("파일 위치", 'r')`

```
f = open("C:/pyfile/file.txt", 'r')  
  
data = f.read()  
print(data)  
  
f.close()
```

```
하늘  
Good luck!  
學生  
30  
3.14  
오늘도 좋은 하루!
```

파일 쓰기 & 읽기

- 문자형을 반복해서 쓰기 & 읽기

```
#파일 쓰기
f = open("C:/pyfile/repeat.txt", 'w')

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

#파일 읽기
f = open("C:/pyfile/repeat.txt", 'r')

text = f.read()
print(text)
f.close()
```

1번째 줄입니다
2번째 줄입니다
3번째 줄입니다
4번째 줄입니다
5번째 줄입니다
6번째 줄입니다
7번째 줄입니다
8번째 줄입니다
9번째 줄입니다
10번째 줄입니다

파일 쓰기 - 리스트형

- 리스트형 자료를 파일에 쓰기

```
f = open('C:/pyfile/2021kbo.txt', 'w')
team = ['NC', '키움', '기아', '삼성', '롯데', 'LG', 'KT', 'SSG']
"""
for i in team:
    f.write(i + ', ')
"""
for i in range(len(team)):
    f.write(team[i] + " ")
f.close()

f = open('C:/pyfile/2021kbo.txt', 'r')
data = f.read()
print(data)
f.close()
```

2021kbo.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

NC 키움 기아 삼성 롯데 LG KT SSG

파일 입출력 연습 문제

계절(season)을 파일에 저장하고, 콘솔에 읽어오는 프로그램을 작성하세요.

- 입,출력 파일 이름: season.txt

season = ['봄', '여름', '가을', '겨울']

👉 실행 결과

봄
여름
가을
겨울

```
season = ['봄', '여름', '가을', '겨울']
f = open("c:/pyfile/season.txt", 'w')
for i in season:
    f.write(i + '\n')

# for i in range(len(season)):
#     f.write(season[i] + '\n')
f.close()

f = open("c:/pyfile/season.txt", 'r')
data = f.read()
print(data)
f.close()
```

파일 읽기

- 파일 읽기 – 줄 단위로 읽기

readline() 함수 – 한 줄 읽기

readlines() 함수 – 파일의 내용 전체를 리스트로 돌려 준다.

```
f = open('c:/pyfile/season.txt', 'r')
#reason = f.readline() # 한줄(첫줄) 읽기
#print(reason)
seasons = f.readlines() # 전체 라인 읽기
print(seasons) # 리스트로 반환
print(seasons[0])
print(seasons[-1])

for season in seasons:
    #print(season)
    print(season.strip()) #한줄 공백 제거

f.close()
```

파일 읽기 – 랜덤 출력

리스트형 자료 랜덤하게 출력하기 – 예외 처리하기

```
import random

try:
    f = open('C:/pyfile/2021kb0.txt', 'r')
    data = f.read().split()
    #print(data)
    team = random.choice(data)
    print(team)
    f.close()
except FileNotFoundError:
    print("파일을 찾을 수 없습니다.")
```

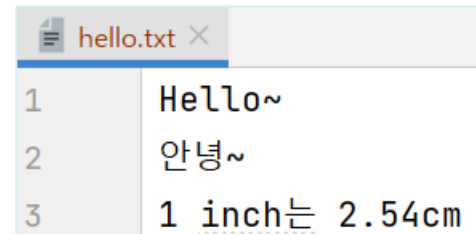

with ~ as 구문

- 자원누수 방지를 돕는 with ~ as 구문

with open(파일이름) **as** 파일 객체:
코드 블록

```
with open('hello.txt', 'w') as f:
    f.write('Hello~ \n')
    f.write('안녕~ \n')
    num = "1 inch는 %.2fcm" % 2.54
    f.write(num)

with open('hello.txt', 'r') as f:
    data = f.read()
    print(data)
```



hello.txt	
1	Hello~
2	안녕~
3	1 inch는 2.54cm

with ~ as 구문 예제

- 구구단 파일 만들기

```
2 x 1 = 2  
2 x 2 = 4  
2 x 3 = 6  
2 x 4 = 8  
2 x 5 = 10  
2 x 6 = 12  
2 x 7 = 14  
2 x 8 = 16  
2 x 9 = 18
```

```
3 x 1 = 3  
3 x 2 = 6  
3 x 3 = 9  
3 x 4 = 12  
3 x 5 = 15  
3 x 6 = 18  
3 x 7 = 21  
3 x 8 = 24  
3 x 9 = 27
```

```
with open('99.txt', 'w') as f:  
    for i in range(2, 10):  
        for j in range(1, 10):  
            gugudan = "%d x %d = %d" % (i, j, i*j)  
            f.write(gugudan)  
            f.write('\n')  
        f.write('\n')
```

영어 타자 연습 프로그램

● 영어 타자 연습 프로그램

게임 방법

- ① 파일 쓰기를 이용하여 word.txt 파일을 생성한다.
- ② 게임이 시작되면 영어 단어가 화면에 표시된다.
- ③ 사용자는 최대한 빠르고 정확하게 입력해야 한다.
- ④ 바르게 입력했으면 다음 문제로 넘어가고 "통과"를 출력한다.
- ⑤ 오타가 있으면 '오타! 다시 도전!'이 출력되고 같은 단어가 한 번 더 나온다.
- ⑥ 타자 게임 시간을 측정한다.

영어 타자 연습 프로그램

[타자 게임]준비되면 엔터!

-문제 1

grape

grape

통과!

-문제 2

potato

potata

오타! 다시 도전!

-문제 2

potato

potato

통과!

-문제 3

grape

-문제 9

tree

tree

통과!

-문제 10

garlic

garlic

통과!

타자 시간 : 34.46초

영어 타자 연습 프로그램

- word.txt 파일 만들고, 랜덤 추출하기

```
import random

with open("word.txt", 'w') as f:
    word = ['sky', 'happiness', 'human', 'earth', 'horse', 'cow',
            'penguin', 'description', 'egg', 'banana']

    for w in word:
        f.write(w + " ")

# 단어를 랜덤하게 뽑아내기
with open("word.txt", 'r') as f:
    #word = f.readlines()
    word = f.readline().split()
    w = random.choice(word)
    print(w)
```

영어 타자 연습 프로그램

```
import random
import time

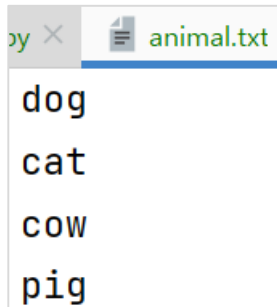
try:
    with open('word.txt', 'r') as f:
        word = f.readline().split()
    n = 1 #문제 번호

    print('[타자 게임] 준비되면 엔터')
    input()
    start = time.time()
```

```
while n < 11:
    print("-문제", n)
    q = random.choice(word) # 문제
    print(q)
    u = input() # 사용자 입력
    if q == u:
        print("통과!")
        n += 1
    else:
        print("오타! 다시 도전!")
end = time.time()
#print("게임 종료!")
et = end - start
print('타자 시간 : %.2f초' % et)
except FileNotFoundError as e:
    print("파일을 찾을 수 없습니다.")
```

readlines() 예제

- readlines() 을 이용한 이차원 리스트 만들기



animal.txt

```
dog
cat
cow
pig
```

```
with open('animal.txt', 'r') as f:
    #lines = f.readlines() #전체 읽기
    #print(lines) - 리스트로 반환
    # for line in lines:
    #     print(line)

    # 2차원 리스트에 저장
    animal = []
    for i in range(4):
        animal.append(f.readline().split())
    print(animal)

    # 리스트 목록 출력
    for i in animal:
        print(i)
```

파일 쓰기 – 입력 받기

◆ 입력 받아 파일 쓰기

```
sky  
바다  
오늘도 좋은 하루 되세요~  
10000  
12.345
```

```
f = open('input.txt', 'a')  
  
text = input("저장할 내용을 입력해 주세요 : ")  
f.write(text)  
f.write('\n')  
f.close()
```


성적 입력 처리

◆ 파일에 과목의 성적 저장하는 프로그램

```
추신수 80 90 85.0  
박인비 80 100 90.0  
손흥민 90 70 80.0
```

```
with open('score.txt', 'a') as f:  
    name = input("이름 입력 : ")  
    kor = int(input("국어 점수 : "))  
    math = int(input("수학 점수 : "))  
    avg = (kor + math) / 2  
  
    f.write(name + ' ')  
    f.write(str(kor) + ' ')  
    f.write(str(math) + ' ')  
    f.write(str(avg) + '\n')
```

성적 입력 처리 - 반복

◆ 반복해서 과목의 성적을 저장하는 프로그램

```
성적을 저장할까요?(y/n) y
이름 입력 : 박인비
국어 점수 : 95
수학 점수 : 90
성적을 저장할까요?(y/n) y
이름 입력 : 추신수
국어 점수 : 85
수학 점수 : 90
성적을 저장할까요?(y/n) y
이름 입력 : 손흥민
국어 점수 : 88
수학 점수 : 79
성적을 저장할까요?(y/n) n
```

```
추신수 80 90 85.0
박인비 80 100 90.0
손흥민 90 70 80.0
```

```
with open('scorelist.txt', 'a') as f:
    while True:
        try:
            key = input("성적을 저장할까요?(y/n) ")
            if key == 'n' or key == 'N':
                break
            elif key == 'y' or key == "Y":
                name = input("이름 입력 : ")
                kor = int(input("국어 점수 : "))
                math = int(input("수학 점수 : "))
                avg = (kor + math) / 2

                f.write(name + ' ')
                f.write(str(kor) + ' ')
                f.write(str(math) + ' ')
                f.write(str(avg) + '\n')
            else:
                print("잘못된 입력입니다. 다시 입력해 주세요 ")
        except ValueError:
            print("잘못된 입력입니다. 다시 입력해 주세요 ")
    print("입력을 종료합니다.")
```

성적표 만들기

◆ 성적 파일을 읽어서 성적표 만들기

***** 성적표 *****

=====

이름	국어	수학	총점	평균
----	----	----	----	----

=====

홍길동	80	90	170	85.0
-----	----	----	-----	------

이순신	90	95	185	92.5
-----	----	----	-----	------

황진이	85	90	175	87.5
-----	----	----	-----	------

성춘향	90	90	180	90.0
-----	----	----	-----	------

***** 과목별 평균 *****

국어 : 86.25, 수학 : 91.25

```
with open('scorelist.txt', 'r') as f:
    li = []
    for i in range(4):
        li.append(f.readline().split())
        #print(li)

    for i in range(4):
        li[i][1] = int(li[i][1])
        li[i][2] = int(li[i][2])
        li[i].append(li[i][1] + li[i][2])
        li[i].append(li[i][3] / 2)
        #print(li)
```

성적표 만들기

◆ 성적 파일을 읽어서 성적표 만들기

```
print("***** 성적표 *****")
print("=====")
print(" 이름   국어   수학   총점   평균")
print("=====")
for i in range(4):
    print(" {}  {}  {}  {}  {}".format(li[i][0],
                                       li[i][1], li[i][2], li[i][3], li[i][4]))
print()

print("***** 과목별 평균 *****")
sum_subj = [0, 0]
for i in range(4):
    sum_subj[0] += li[i][1]
    sum_subj[1] += li[i][2]
print(" 국어 : {0}, 수학 : {1}".format(sum_subj[0]/4, sum_subj[1]/4))
```

바이너리 파일 읽고 쓰기

◆ 바이너리 파일

바이너리 파일이란 화상, 음성 등의 대부분의 파일로 0과 1로 이루어진 파일이다.

`open("파일 위치", 'wb')`

모드	설명
wb	쓰기
rb	읽기

```
with open('data.bin', 'wb') as f:  
    text = "비가 내린다."  
    f.write(text.encode())
```

코드화하다.

```
with open('data.bin', 'rb') as f:  
    data = f.read()  
    text = data.decode()  
    print(text)
```

복호화하다.

glob 모듈(Module)

◎ glob 모듈

- 파일 입출력 관련 프로그램을 만들때 특정 디렉터리에 있는 파일 이름을 모두 알아야 할때 사용.
- 파일 이름을 리스트로 반환해 준다.

```
import glob

data = glob.glob("c:/pyworks/ch06/*.py")
print(data)
```

```
['C:/pyworks/ch06\\argv_test.py', 'C:/pyworks/ch06\\calendar_ex.py', 'C:/pyworks/ch06\\dice.py', 'C:/pyworks/ch06\\dice2.py',
```

연습 문제

c 드라이브에서 모든 폴더를 읽어서 pyfile을 찾고, pyfile 폴더에서 모든 txt 파일을 읽어와서 결과를 출력해 보시오.

```
import os
import glob

# pyfile 찾기
os.chdir('c:/')
dir = os.popen('dir')
print(dir.read())

# 모든 txt파일 읽기
file = glob.glob('c:/pyfile/*.txt')
print(file)

# 특정 파일의 내용 읽기
with open('c:/pyfile/kbo2021.txt', 'r') as f:
    data = f.read()
    print(data)
```

pickle 모듈

◎ pickle 모듈

- 객체의 형태를 그대로 유지하면서 파일에 저장하고 불러올 수 있는 모듈이다.
- 이때 객체란, 리스트나 딕셔너리등의 자료구조도 포함한다.

```
import pickle

with open('data.txt', 'wb') as f:
    li = ['강아지', '고양이', '닭']
    dic = {1:'강아지', 2:'고양이', 3:'닭'}
    pickle.dump(li, f)
    pickle.dump(dic, f)

with open('data.txt', 'rb') as f:
    li = pickle.load(f)
    dic = pickle.load(f)
    print(li)
    print(dic)
```

모드	설명
pickle.dump	쓰기
pickle.load	읽기

로그인 구현 예제

❖ 로그인 구현하기

```
PS C:\pyworks\file_io> python login.py
Enter userID: sky
Enter password:
로그인에 성공하였습니다.
PS C:\pyworks\file_io> python login.py
Enter userID: river
Enter password:
아이디나 비밀번호가 다릅니다.
```

- 아이디와 비밀번호를 입력 받는다.
(비밀번호를 표시하지 않도록 getpass 모듈을 import함)
- member.txt 읽어온다.
- 아이디와 비밀번호를 2차원 리스트로 만든다.
- 입력한 아이디와 비밀번호가 member.txt와 일치하면
'로그인에 성공했습니다.' 아니면 '아이디나 비밀번호가 다릅니다.' 표시

로그인 구현 예제

```
from getpass import getpass
```

```
def login():
```

```
    # 아이디는 8자 이하, 비밀번호는 10자만 입력
```

```
    userid = input("Enter userID: ")
```

```
    userpw = getpass("Enter password: ") # 비밀번호 감추기
```

```
    member = []
```

```
    with open('member.txt') as f:
```

```
        lines = f.readlines()
```

```
    for i in lines: # 요소를 분리
```

```
        member.append([i[0:10], i[10:]])
```

```
    #print(member)
```

sky	s12345678!
mountain	m12345678%
river	r12345678@

로그인 구현 예제

```
access = False # 로그인 성공 여부
for user, passwd in member:
    if user.strip() == userid and passwd.strip() == userpw:
        print("로그인에 성공하였습니다.")
        access = True
        break

if access == False:
    print("아이디나 비밀번호가 다릅니다.")
login()
```