

08장 예외 처리

8.1 오류는 어떨 때 발생하는가?

```
[ch08/ex01_error.py]

#-*- coding:utf-8 -*-

# 오류는 어떨 때 발생하는가?

# 1. 디렉토리 안에 없는 파일을 열려고 시도했을 때 발생하는 오류
# f = open("없는파일.txt" 'r')
'''
Traceback (most recent call last):
  File "D:/dev/workspace/workspace-pycharm/foundation/ch08_exceptionHandler/ex01_error.py", line 6, in
<module>
    f = open("없는파일.txt" 'r')
FileNotFoundError: [Errno 2] No such file or directory: '없는파일.txt'
'''

# 2. 0으로 다른 숫자를 나누는 경우에 발생하는 오류
# 4 / 0
'''
Traceback (most recent call last):
  File "D:/dev/workspace/workspace-pycharm/foundation/ch08_exceptionHandler/ex01_error.py", line 14, in
<module>
    4 / 0
ZeroDivisionError: division by zero
'''

# 3. a[4]는 a 리스트에서 얻을 수 없는 값이어서 발생하는 오류
a = [1,2,3]
print(a[4])
'''
Traceback (most recent call last):
  File "D:/dev/workspace/workspace-pycharm/foundation/ch08_exceptionHandler/ex01_error.py", line 24, in
<module>
    print(a[4])
IndexError: list index out of range
'''
```

8.2 오류 예외 처리 기법

- try except: try 블록 수행 중 오류가 발생하면 except 블록이 수행된다. 하지만 오류가 발생하지 않으면 except 블록은 수행되지 않는다.
- try else: else절은 예외가 발생하지 않은 경우에 실행되면 반드시 except절 바로 다음에 위치해야 한다.
- try finally: finally절은 try문 수행 도중 예외 발생 여부에 상관없이 항상 수행된다.

```
[ch08/ex02_try.py]

#-*- coding:utf-8 -*-

# 오류 예외 처리 기법
# try, except 문

try:
```

```

4 / 0
except ZeroDivisionError as e: # 예외처리(Exception Handler)
    # print(e)
    pass

# try ... finally
f = open("foo.txt", 'w')
try:
    pass
finally: # 항상 실행해야하는 실행문을 기입한다.
    f.close()

# 여러 개의 오류 처리하기
try:
    a = [1, 2]
    print(a[3])
    4 / 0
except ZeroDivisionError:
    print("0으로 나눌 수 없습니다.")
except IndexError:
    print("인덱싱할 수 없습니다.")

try:
    a = [1,2]
    print(a[3])
    4 / 0
except (ZeroDivisionError, IndexError) as e:
    print(e)

```

8.3 오류 회피하기

- 프로그래밍을 하다 보면 특정 오류가 발생할 경우 그냥 통과시켜야 할 때가 있다.

[ch08/ex03_pass.py]

```

#-*- coding:utf-8 -*-

try:
    f = open("나없는파일", 'r')
except FileNotFoundError: # 파일이 없더라도 오류를 발생시키지 않고 통과한다.
    pass

```

8.4 오류 일부러 발생시키기

- 파이썬은 raise라는 명령어를 이용해 오류를 강제로 발생시킬 수 있다.

[ch08/ex04_raise.py]

```

#-*- coding:utf-8 -*-

# 오류 일부러 발생시키기
class Bird:
    def fly(self):
        raise NotImplementedError

class Eagle(Bird):

```

```

pass

# class Eagle(Bird):
#     def fly(self):
#         print("very fast")

eagle = Eagle()
eagle.fly()

'''
Traceback (most recent call last):
  File "...", line 16, in <module>
    eagle.fly()
  File "...", line 6, in fly
    raise NotImplementedError
NotImplementedError
'''

```

8.5 예외 만들기

- 프로그램 수행 도중 특수한 경우에만 예외 처리를 하기 위해서 종종 사용자 정의 예외를 만들어서 사용한다.
- 예외는 파이썬 내장 클래스인 Exception 클래스를 상속하여 만들 수 있다.

[ch08/ex05_myError.py]

```

#-*- coding:utf-8 -*-

# __str__메서드는 print(e) 처럼 오류 메시지를 print문으로 출력할 경우에 호출되는 메서드이다.
class MyError(Exception):
    def __str__(self):
        return "허용되지 않는 별명입니다."

def say_nick(nick):
    if nick == '바보':
        raise MyError()
    print(nick)

try:
    say_nick("천사")
    say_nick("바보")
except MyError as e:
    print("허용되지 않는 별명입니다.")
    print(e)

'''
천사
허용되지 않는 별명입니다.
허용되지 않는 별명입니다.
'''

```