

# 예외처리

1. 예외처리 문법
2. 의도적인 오류 발생
3. 사용자 정의 오류 클래스 만들기

## ■ 오류가 발생하는 여러가지 상황

### ● ZeroDivisionError

```
2 / 0
```

```
ZeroDivisionError: division by zero
```

### ● IndexError

```
list = []; list[0]
```

```
IndexError: list index out of range
```

### ● TypeError

```
4 + ""
```

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

### ● FileNotFoundError

```
file = open('python.txt', 'r')
```

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

## ■ 예외처리 문법

### ● 기본 구조

```
try:  
    명령문1 실행  
    명령문2 실행  
except [발생 오류 [as 오류 메시지 변수]]:  
    오류 발생 시 처리할 명령문 실행
```

- 오류가 발생하면 해당 라인에서 동작을 멈추고 except 블록으로 이동
- 오류가 발생하지 않으면 except 블록은 실행되지 않음

### ● 작성 형태 3가지

```
try:  
    ...  
except:  
    ...
```

```
try:  
    ...  
except 오류명:  
    ...
```

```
try:  
    ...  
except 오류명 as 변수:  
    ...
```

## ■ 예외처리 문법

### ● try - except 사용

- ZeroDivisionError

```
try:
    print(2 / 0)
except ZeroDivisionError as e:
    print('예외발생')
    print(e)
```

예외발생  
division by zero

- IndexError

```
try:
    list = []
    print(list[0])
except IndexError as e:
    print('예외발생')
    print(e)
```

예외발생  
list index out of range

- FileNotFoundError

```
try:
    file = open('python.txt', 'r')
except FileNotFoundError as e:
    print('예외발생')
    print(e)
```

예외발생  
[Errno 2] No such file or directory: 'python.txt'

## ■ 예외처리 문법

### ● try - except - else 사용

- 오류가 발생하지 않으면 else 블록 실행
- 반드시 except 절 다음에 위치

```
try:
    file = open('python.txt')
except FileNotFoundError as e:
    print('예외발생')
    print(e)
else:
    print(file.read())
```

예외발생

[Errno 2] No such file or directory: 'python.txt'

```
try:
    file = open('harry_potter.txt')
except FileNotFoundError as e:
    print('예외발생')
    print(e)
else:
    print(file.read())
```

1

Harry Potter and the Sorcerer's Stone

CHAPTER ONE

THE BOY WHO LIVED

Mr. and Mrs. Dursley, of number four, Privet Drive, were proud to say that they were perfectly normal, thank you very much. They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense.

## ■ 예외처리 문법

### ● try - except - finally 사용

- 오류 발생 유무와 관계없이 항상 실행
- 반드시 실행되어야 하는 경우 사용

```
try:
    file = open('song.txt', encoding='utf8')
    text = file.read()
    print(text)
    text += 1
except FileNotFoundError as e:
    print('오류발생', e)
except TypeError as e:
    print('오류발생', e)
finally:
    file.close()
    print('file close')
```

동해물과 백두산이 마르고 닳도록

하느님이 보우하사 우리나라 만세

오류발생 can only concatenate str (not "int") to str

file close

## ■ 연습문제

### ● 데이터베이스를 활용하는 코드에 예외처리 적용하기

1. SQL 표현이 잘못된 경우 예외처리
2. 어떤 경우로 종료되더라도 `close()`가 수행되도록 `finally` 처리

```
import sqlite3

conn = sqlite3.connect('geoje.db')

cursor = conn.cursor()

sql = '''
    SELECT id, name, desc, mapinfo, latitude, longitude
    FORM tour
''' # FROM tour

cursor.execute(sql)
result = cursor.fetchmany(10)
for row in result:
    print(row[1], row[4], row[5])

cursor.close()
conn.close()
```

거제대교	34.8839990	128.4742000
한려해상	34.8649712	128.7271122
양지암	34.8803759	128.7381372
외도보타니아	34.7691855	128.711822
오색바위	34.7381883	128.6583798
신선대	34.7379442	128.6625034
바람의언덕	34.7445821	128.6630829
칠천연륙교	34.9783618	128.6518552
거제해금강	34.7328499	128.6838576
거가대교	35.0149387	128.7555727

## ■ 의도적인 오류 발생

● raise

raise 예외

```
class Car:
    def move(self):
        raise NotImplementedError

class Taxi(Car):
    pass

taxi = Taxi()
taxi.move()
```

```
NotImplementedError
d:\study\python\openeg\09-예외처리.ipynb
      6      pass
      8 taxi = Taxi()
---->  9 taxi.move()

d:\study\python\openeg\09-예외처리.ipynb
      2 def move(self):
---->  3      raise NotImplementedError

NotImplementedError:
```

- (모듈 등) 기능을 활용하려는 개발자가 해야 할 일을 알려주기 위한 목적

```
class Car:
    def move(self):
        raise NotImplementedError

class Taxi(Car):
    def move(self): # 오버라이드
        print('fast move')

taxi = Taxi()
taxi.move()
```

fast move



## ■ 사용자 정의 오류 클래스 만들기

### ● 기본 구조

```
class CustomError(Exception):  
    pass
```

- 파이썬에 미리 만들어놓은 많은 오류 모듈이 있긴 하지만  
현재 개발하려는 서비스 상황에 알맞은 오류를 찾을수 없을 때  
직접 오류를 만들어서 쓰는 것도 가능

★ Exception이라는 클래스를 상속받는 것만으로 가능

### ● CustomError 적용

```
import random  
  
class CustomError(Exception):  
    def __init__(self, msg=''):  
        super().__init__(msg)  
  
num = random.randint(1, 10)  
  
if num > 5: raise CustomError('5보다 큰 숫자')  
else: print(num)
```

```
CustomError                                Traceback (most recent call last)  
d:\study\python\openeg\09-예외처리.ipynb 15 in <cell>  
     5 def __init__(self, msg): self.msg = msg  
     7 num = random.randint(1, 10)  
----> 9 if num > 5: raise CustomError('5보다 큰 숫자')  
    10 else: print(num)  
  
CustomError: 5보다 큰 숫자
```

## ■ 연습문제

### ● 사용자 인증 기능 작성하기

- 조건 1) NotExistIdError, WrongPasswordError 클래스 작성
- 조건 2) 오류 메시지를 전달할 수 있도록 생성자 작성
- 조건 3) id/pw가 blue/1234가 아닌 경우

'ID를 확인해주세요', 'PW를 확인해주세요' 메시지 출력

```
class NotExistIdError( ① ):
    # 생성자 작성 ( ② )
class WrongPasswordError( ③ ):
    # 생성자 작성 ( ④ )
```

```
login('white', 1234)
```

NotExistIdError: ID를 확인해주세요.

```
login('blue', 4321)
```

WrongPasswordError: PW를 확인해주세요.

```
def login(id, pw):
    # id 비교 조건문 작성 ( ⑤ )
    # 조건에 따라 예외 발생 ( ⑥ )
    # pw 비교 조건문 작성 ( ⑦ )
    # 조건에 따라 예외 발생 ( ⑧ )
    print('로그인 되었습니다.')
login('blue', 1234)
```