

## 33강 - 예외처리

### #01. 에러의 종류

종류	설명
Syntax Error	문법 에러. 파이썬의 작성규칙을 지키지 않아 발생하는 에러. 프로그램이 실행될 수 없는 상태
Runtime Error	실행중 에러. 프로그램이 실행도중 예상치 못한 논리적 오류를 만나는 경우.

개발자가 예상하지 못한 상황에서 발생하는 Runtime Error를 **예외**라고 한다.

즉 예외처리란, 에러에 대비하기 위한 코드 작성 규칙

### 예외상황 이해하기

```
x = int(input("숫자를 입력하세요."))

# x에 0이 입력된 경우 에러가 발생함.
# -> ZeroDivisionError
y = 100 / x
print("결과는 %d입니다." % y)
```

```
ZeroDivisionError                                Traceback (most recent call last)
Untitled-1.ipynb 셀 2 line 2
      <a href='vscode-notebook-cell:Untitled-1.ipynb?jupyter-notebook#W0sdW50aXRszWQ%3
———> <a href='vscode-notebook-cell:Untitled-1.ipynb?jupyter-notebook#W0sdW50aXRszWQ%3
      <a href='vscode-notebook-cell:Untitled-1.ipynb?jupyter-notebook#W0sdW50aXRszWQ%3

ZeroDivisionError: division by zero
```

```
filePath = input("파일 경로를 지정하세요")

# 파일을 저장할 수 없는 경로(ex: z:\hello.txt)가 지정된 경우 예외가 발생함
# -> FileNotFoundError
with open(filePath, "w", encoding="utf-8") as f:
    f.write("파일 쓰기 테스트 입니다.")
```

```
FileNotFoundError                                Traceback (most recent call last)
Untitled-1.ipynb 셀 3 line 3
      <a href='vscode-notebook-cell:Untitled-1.ipynb?jupyter-notebook#W2sdW50aXRszWQ%3
———> <a href='vscode-notebook-cell:Untitled-1.ipynb?jupyter-notebook#W2sdW50aXRszWQ%3
      <a href='vscode-notebook-cell:Untitled-1.ipynb?jupyter-notebook#W2sdW50aXRszWQ%3

File c:\Users\leekh\AppData\Local\Programs\Python\Python312\Lib\site-packages\IPython\
```

```

301 if file in {0, 1, 2}:
302     raise ValueError(
303         f"IPython won't let you open fd={file} by default "
304         "as it is likely to crash IPython. If you know what you are doing, "
305         "you can use builtins' open."
306     )
→ 308 return io_open(file, *args, **kwargs)

```

`FileNotFoundError: [Errno 2] No such file or directory: 'z:\\hello.txt'`

## #02. 예외처리 구문

파이썬에서 예외상황이 발생한 경우 작동하도록 하는 문법적 구조

### try~except 문

예외상황이 예상되는 코드를 `try` 블록으로 감싼다. 이 블록 안에서 예외가 발생할 경우 연결된 `except` 블록으로 프로그램의 제어가 이동한다.

이 경우에는 예외의 종류에 상관없이 예외가 발생하면 `except` 블록을 수행한다.

```

try:
    x = int(input("숫자를 입력하세요."))
    y = 100 / x
    print("결과는 %d입니다." % y)
except:
    print("에러가 발생했습니다.")

```

에러가 발생했습니다.

### 예외의 종류에 따라 다르게 대응하기

```

try:
    x = int(input("숫자를 입력하세요."))
    y = 100 / x
    print("결과는 %d입니다." % y)
except ValueError:
    print("숫자만 입력하세요.")
except ZeroDivisionError:
    print("0으로 나눌 수 없습니다.")

```

0으로 나눌 수 없습니다.

### 에러 정보를 포함하는 객체 사용하기

`except`문 뒤에 `as 객체명` 형식으로 에러객체를 전달받아 활용할 수 있다.

```

try:
    x = int(input("숫자를 입력하세요."))
    y = 100 / x

```

```

    print("결과는 %d입니다." % y)
except ValueError as verr:
    print("숫자만 입력하세요.")
    print(verr)
except ZeroDivisionError as zerr:
    print("0으로 나눌 수 없습니다.")
    print(zerr)

```

0으로 나눌 수 없습니다.  
division by zero

## 에러 발생여부에 상관 없이 무조건 실행되는 블록

```

try:
    x = int(input("숫자를 입력하세요."))
    y = 100 / x
    print("결과는 %d입니다." % y)
except ValueError as verr:
    print("숫자만 입력하세요.")
    print(verr)
except ZeroDivisionError as zerr:
    print("0으로 나눌 수 없습니다.")
    print(zerr)
finally:
    print("처리가 완료되었습니다.")

```

0으로 나눌 수 없습니다.  
division by zero  
처리가 완료되었습니다.

## 가장 간편한 사용 형태

```

try:
    x = int(input("숫자를 입력하세요."))
    y = 100 / x
    print("결과는 %d입니다." % y)
except Exception as e:
    print("에러가 발생했습니다.", e)
finally:
    print("처리가 완료되었습니다.")

```

## #03. 예외 강제 발생

### 기본 구문 형식

```
raise ValueError("똑바로 입력 안하냐?")
```

---

ValueError

Traceback (most recent call last)

c:\Users\leekh\Desktop\python+\Untitled-1.ipynb 셀 15 line 1  
→ <a href='vscode-notebook-cell:/c%3A/Users/leekh/Desktop/python%2B%2B/Untitled-1.'

ValueError: 똑바로 입력 안하냐?

## 고전적인 예외 상황 처리

```
def oldLogin(userId, userPw):  
    if not userId:  
        return -1  
  
    if not userPw:  
        return -2  
  
    # ... 로그인 처리  
    return 0
```

```
#result1 = oldLogin("", "1234")  
#result1 = oldLogin("user", "")  
result1 = oldLogin("user", "1234")  
  
if result1 == -1:  
    print("아이디를 입력하세요")  
elif result1 == -2:  
    print("비밀번호를 입력하세요")  
else:  
    print("로그인 성공")
```

로그인 성공

## raise 문을 활용한 예외 처리

```
def newLogin(userId, userPw):  
    if not userId:  
        raise Exception("아이디를 입력하세요")  
  
    if not userPw:  
        raise Exception("비밀번호를 입력하세요")  
  
    # ... 로그인 처리  
    return 0
```

```
try:  
    #newLogin("", "1234")  
    #newLogin("user", "")  
    newLogin("user", "1234")  
    print("로그인 성공")  
except Exception as e:  
    print("로그인에 실패했습니다.", e)
```

로그인 성공