
예외처리의 이해와 활용

6주차_03

한 동 대 학 교
김경미 교수

학습목표

- 예외처리 개념 이해하기
- 입력하는 값의 범위를 지정하여 처리하기
- 없는 파일을 열었을 때 어떻게 처리하는지 알아보기
- ZeroDivisionerror 를 이해하고 활용해 보기

예외 처리

- 프로그래머의 의도와 동떨어진 상황이 발생하는 것을 예외(exception)라고 함
- 예상되는 에러를 처리하도록 코딩을 추가하는 과정
- 코드를 실행하다가 문제가 발생할 수 있는 예외적인 상황을 처리하는 코드를 추가
- 예상 가능한 문제를 구체적으로 표시하여 처리

사용 예제

```
#check File IO
```

```
import sys
```

```
try:
```

```
    inf = open('myfile.txt', 'r')
```

```
    s = f.readline()
```

```
except IOError as err:
```

```
    print("I/O error: {0}".format(err))
```

사용 예제

```
# check zero division

import sys

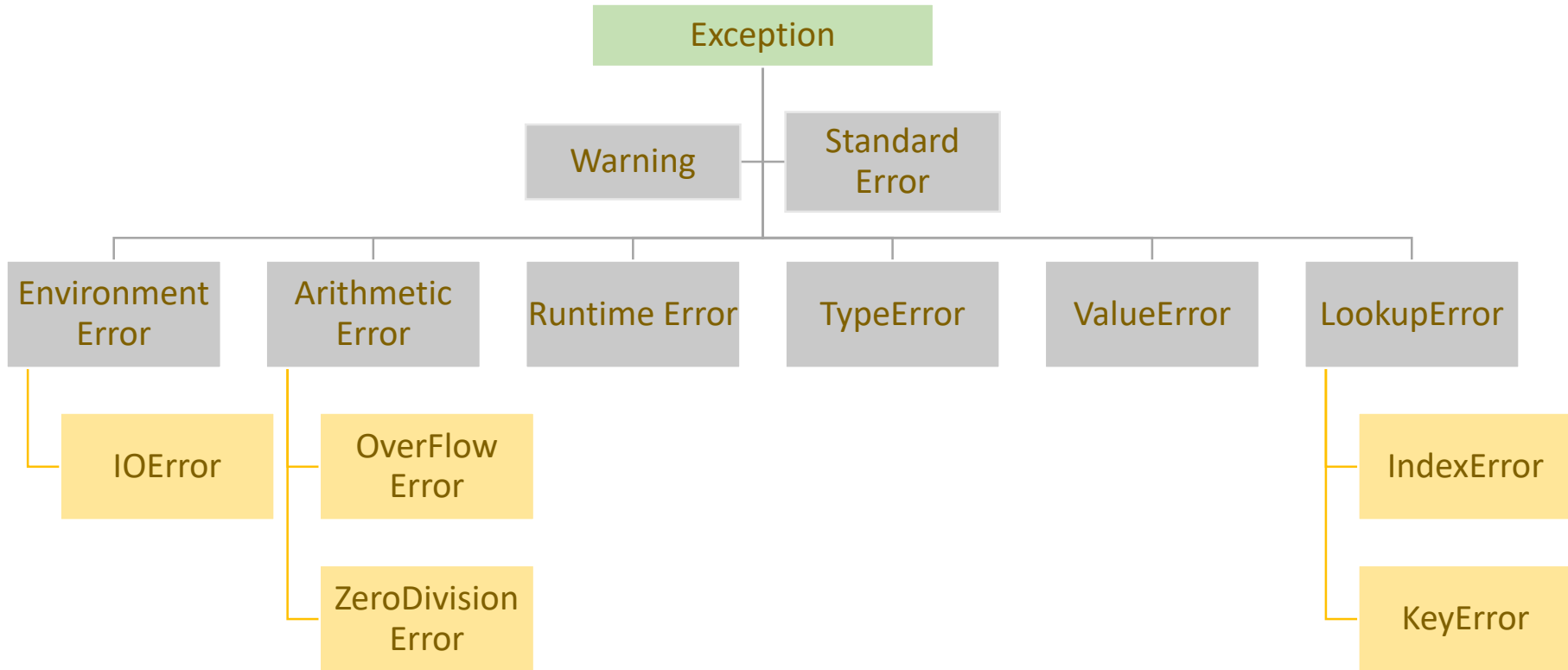
def divide(x, y):
    try:
        result = x / y

    except ZeroDivisionError:
        print("division by zero!")

    else:
        print("result is", result)

    finally:
        print("executing finally clause")
```

예외처리 구성



Try: Exception 1

```
import sys

while True:
    try:
        x = int(input("Please enter an integer: "))
        break

    except ValueError:
        print("Oops! That was no valid number. Try again...")
```

```
>>>
Please enter an integer: 5
>>> ===== RESTART =====
>>>
Please enter an integer: as
Oops! That was no valid number. Try again...
Please enter an integer: v
Oops! That was no valid number. Try again...
Please enter an integer: 5.4
Oops! That was no valid number. Try again...
Please enter an integer: 09
>>>
```

Try: Exception 2

```
import sys

try:
    x = 1/0

except ZeroDivisionError as err:
    print('Handling run-time error:', err)
```

```
>>>
Handling run-time error: division by zero
>>>
```


Except with Multiple Exceptions

```
import sys

try:
    f = open('myfile.txt', 'r')
    s = f.readline()
    i = int(s.strip())
except IOError as err:
    print("I/O error: {0}".format(err))
except ValueError:
    print("Could not convert data to an integer.")
except:
    print("Unexpected error:", sys.exc_info()[0])
```

```
>>>
I/O error: [Errno 2] No such file or directory: 'myfile.txt'
>>>
```

Try-Finally 사용

```
try:
    fh = open("t.txt", "w")
    try:
        fh.write("This is my test file for exception handling!!\n")
        fh.write("=" * 35)
    finally:
        print("Going to close the file")
        fh.close()
        fh = open("t.txt", "r")
        print(fh.readlines())
        fh.close()

except IOError:
    print("Error: can't find file or read data")
```

```
>>>
Going to close the file
['This is my test file for exception handling!!\n', '=====
=====']
>>>
```

Assertion(1)

- 원하는 범위의 값을 입력 받고 싶을 때, `assert()`문장을 넣으면 그 범위의 값이 들어오지 않았을 때 에러 메시지를 띄우고 프로그램이 중단 됨
- 에러 처리를 해서 진행하지 않고, 에러 처리가 힘들거나 중요한 값이라서 중단해야 할 때 사용 함

Assertion(2)

```
#assert
```

```
def KToFahrenheit(Temperature):
```

```
    assert (Temperature >= 0), print("Colder than absolute zero!")
```

```
    return ((Temperature-273)*1.8)+32
```

```
print(KToFahrenheit(273))
```

```
print(KToFahrenheit(505.78))
```

```
print(KToFahrenheit(-5))
```

```
>>>
32.0
451.00399999999996
Colder than absolute zero!
Traceback (most recent call last):
  File "C:/Users/user/Desktop/eh.py", line 7, in <module>
    print(KelvinToFahrenheit(-5))
  File "C:/Users/user/Desktop/eh.py", line 2, in KelvinToFahrenheit
    assert (Temperature >= 0),print("Colder than absolute zero!")
AssertionError: None
>>> |
```

Assertion(3)

```
#assert
```

```
num = int(input('Enter a positive number: '))  
assert(num > 0), print('Only positive numbers are allowed!')  
  
print("number is ", num )
```

```
>>>  
Enter a positive number: 7  
number is 7  
>>> ===== RESTART =====  
>>>  
Enter a positive number: -3  
Only positive numbers are allowed!  
Traceback (most recent call last):  
  File "C:/Users/user/Desktop/eh.py", line 12, in <module>  
    assert(num > 0), print('Only positive numbers are allowed!')  
AssertionError: None  
>>> |
```

연습문제 1

- 나이를 입력 받는다
- 입력 받은 나이가 1보다 작고, 100보다 크면 오류라고 알려준다
- assert문을 사용한다

연습문제 1, 코드와 결과

#assert

```
age = int(input('Enter your age: '))
assert(age > 1 and age <= 100), print("나이는 1세에서 100세까지 만 입력 가능!")

print("age is ", age )
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise Code/turtle_test.
py =====
Enter your age: 5
age is 5
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise Code/turtle_test.
py =====
Enter your age: 102
나이는 1세에서 100세까지만 입력 가능!
Traceback (most recent call last):
  File "E:/1_Works/2017Work/KMooC강의/Exercise Code/turtle_test.py", li
ne 2, in <module>
    assert(age > 1 and age <= 100), print("나이는 1세에서 100세까지만
입력 가능!")
AssertionError: None
>>> |
```

연습문제 2

- 나이를 입력 받는다
- 입력 받은 나이가 1보다 작고, 100보다 크면 다시 입력 받도록 코딩한다
- while문 사용한다

연습문제 2, 코드와 결과

```
age = int(input('Enter your age: '))
```

```
while age < 1 or age > 100 :
```

```
    print("처리할 수 없는 숫자가 입력 되었습니다, 다시 입력해 주세요")
```

```
    age = int(input('Enter your age: '))
```

```
print("입력된 나이는 ", age, "세 입니다")
```

```
>>>
```

```
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise Code/  
py =====
```

```
Enter your age: 12
```

```
입력된 나이는 12 세 입니다
```

```
>>>
```

```
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise Code/  
py =====
```

```
Enter your age: 102
```

```
처리할 수 없는 숫자가 입력 되었습니다, 다시 입력해 주세요
```

```
Enter your age: 99
```

```
입력된 나이는 99 세 입니다
```

```
>>> |
```

연습문제 3

- 나이를 입력 받는다
- 숫자가 아니고, 이십오세등의 문자로 입력되면 다시 입력 받도록 코딩한다
- Try문을 사용한다

연습문제 3, 코드와 결과

```
import sys

while True:
    try:
        age = int(input('Enter your age: '))
        break

    except ValueError:
        print("처리할 수 없는 문자가 입력되었습니다! 정수로 입력하세요")
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC강의/Exercise Code,
py =====
Enter your age: 이십오세
처리할 수 없는 글자가 입력되었습니다! 정수로 입력하세요
Enter your age: 25
>>>
```

연습문제 4

- 더치페이를 계산해주는 프로그램을 작성한다
- 금액과 인원을 입력받아 각자 내야 하는 금액을 출력한다
- 인원이 0일때, 정수가 아닌 인원이 입력되면 오류라고 알려준다

연습문제 4 코드

```
import sys

try:
    money = int(input("금액을 입력하세요: "))
    people = int(input("인원을 입력하세요: "))

    pay = money/people

    print("각자 ", str(pay), " 원 씩 나누어 내세요")

except ZeroDivisionError as err:
    print(err, ': 인원은 1명 이상이어야 합니다 ')

except ValueError as err:
    print(err, ': 정수만 입력 가능합니다 ')
```

연습문제 4 결과

```
>>>
>>> ===== RESTART =====
>>>
금액을 입력하세요: 20000
인원을 입력하세요: 4
각자 5000.0 원 씩 나누어 내세요
>>> ===== RESTART =====
>>>
금액을 입력하세요: 50000
인원을 입력하세요: 0
division by zero : 인원은 1명 이상이어야 합니다
>>> ===== RESTART =====
>>>
금액을 입력하세요: 40000
인원을 입력하세요: 네명
invalid literal for int() with base 10: '네명' : 정수만 입력 가능합니다
>>>
```

강의 요약

- **예외처리 개념 이해하기**
 - 의도되지 않은 상황(예외)을 처리하는 코드를 추가하는 것
 - try, except, finally 구문 활용
- **입력하는 값의 범위를 지정하여 처리하기**
 - assert(원하는 범위 지정)
- **없는 파일을 열었을 때 어떻게 처리하는지 알아보기**
 - IOError
- **ZeroDivisionerror 를 이해하고 활용해 보기**

오지 선다

- 예외처리 구문에 속하는 명령어를 모두 표시하시오
 - try
 - traceback
 - except
 - finally
 - Assert
 - if

단답형

- 다음 `assert()`문에 대해 설명하시오

```
age = input('Enter your age: ')
age = int(age)
assert(age > 1 and age <= 100), print("나이는 1세에서 100세까지!")
```

- `age > 1 and age <= 100` 값인 경우에 허용하고, 그렇지 않은 값이 입력되면 “나이는 1세에서 100세까지 ” 라고 출력한다