예외 처리

이번 장에서는 파이썬에서 예외란 무엇인지를 알아보고 예외 처리 방법에 대하여 공부합니다.

목차

❑ 예외 (exception)

- 문법적으로는 문제가 없지만 실행 중에 발생하는 오류
- 예외가 발생하면 실행이 중단됨

예)

```
str = "89점"
score = int(str)
print(score)
```

```
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    score=int(str)
ValueError: invalid literal for int() with base 10: '89점'
```

ValueError (문자열 안에 숫자와 문자가 있어 정수 변환이 안됨)

❑ 예외 처리

- 예외 발생 시에 프로그램 실행을 중단시키지 않으면서 처리함

❑ 문법

```
try :
    실행하는 명령들
except 예외명 as 변수 :
    예외가 발생할 때 수행할 명령들
else :
    예외가 발생하지 않을 때 수행할 명령들
```

try 블록의 명령을 실행하는 도중에

- 예외가 발생하면: try 블록의 남은 부분은 건너뛰고, except 블록을 실행
- 예외가 발생하지 않으면: else 블록을 실행

❑ 앞의 프로그램을 예외처리 한 경우

```
str = “89점”
try :
    score = int(str)
except :
    print( “예외가 발생” )
else :
    print(score)
```

- 자주 발생하는 예외를 미리 정의하고 고유한 이름을 붙여 놓음

- 대표적인 예외
  - NameError : 지역 이름이나 전역 이름이 발견되지 않음
  - ValueError : 타입은 맞지만 값의 형식이 잘못됨
  - ZeroDivisionError : 0으로 나눔
  - IndexError : 첨자(index)가 범위를 벗어남
  - TypeError : 타입이 맞지 않음

- Python 내장 예외 (built-in exception)
  - https://docs.python.org/ko/3/library/exceptions.html

❑ 발생하는 예외에 따라 다른 명령을 수행하고 싶을 때

- except 에 예외 이름을 명시하고, 해당 명령을 수행함

```
str = "89점"
try :
    score = int(str)
except ValueError :                # ValueError 예외 발생 시 수행
    print( "점수의 형식이 잘못됨" )
except IndexError :                # IndexError 예외 발생시 수행
    print( "첨자가 범위를 벗어남" )
else :
    print(score)
```

❑ 두 개 이상의 예외를 동시에 처리할 때는 괄호로 묶어 지정함

```
str = "89점"
try :
    score = int(str)
except (ValueError, IndexError) :  # ValueError 또는 IndexError 발생 시 수행
    print( "점수의 형식이나 첨자가 잘못됨" )
else :
    print(score)
```

- Except 블록의 예외 이름 다음에 as 키워드로 예외 객체를 변수로 받으면 이 객체를 통해 에러 메시지를 구할 수 있음

```
str = "89점"
try :
    score = int(str)
except ValueError as e :
    print(e)
else :
    print(score)
```

- raise 명령은 일부러 예외를 발생시킬 때 사용

```
def calsum(n) :
    if (n < 0) :
        raise ValueError          # 예외 이름 없이 raise 만 사용해도 됨
    sum = 0
    for a in range(n+1) :
        sum += a
    return sum

try :
    print(calsum(10))
    print(calsum(-5))
except :
    print( "입력값이 잘못됨" )
```

예) 사용자가 정의한 예외 발생 : Exception 사용

```
text = input()
if text.isdigit() == False :
    raise Exception("입력 문자열이 숫자가 아닙니다.")
```

예)

```
text = input()
if text.isdigit() == False :
    try :
        raise Exception("입력 문자열이 숫자가 아닙니다.")
    except Exception :
        print("예외가 일어났습니다.")
```

- **Finally 블록은 예외 발생 여부와 상관없이 반드시 실행해야 할 명령을 지정할 때 사용**
  - 일반적으로 파일이나 통신 채널과 같은 컴퓨터 자원을 정리할 때 사용됨

```
str = "89점"
try :
    score = int(str)
except ValueError as e :
    print(e)
else :
    print(score)
finally :
    print( "무조건 수행합니다" )
```

❑ 다음 프로그램의 출력 결과를 적고, 설명하시오

```
def some_function():
    print("1~10 사이의 수를 입력하세요:")
    num = int(input())
    if num < 1 or num > 10 :
        raise Exception("유효하지 않은 숫자입니다.: {0}".format(num))
    else :
        print("입력한 수는 {0}입니다.".format(num))

try:
    some_function()
except Exception as err:
    print("예외가 발생했습니다. {0}".format(err))
```

❑ 다음 프로그램의 출력 결과를 적고, 설명하시오

```
def divide(x, y):
    try:
        result = x / y
    except ZeroDivisionError:
        print("division by zero!")
    else:
        print("result is", result)
    finally:
        print("executing finally clause")

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