

17강

예외 처리

17-1 예외 처리

● 예외 처리 개요

- ✓ 예외(exception) 란 프로그램에 예상하지 못한 상황이 발생한 것이다.

프로그래밍의 예외는 크게 예측 가능한 예외와 예측 불가능한 예외로 나눌 수 있다.

- ✓ 예측 가능한 예외 : 발생 여부를 개발자가 사전에 인지할 수 있는 예외이다.

개발자는 예외를 예측하여 명시적으로 예외가 발생할 때는 어떻게 대응하라고 할 수 있다.

대표적으로 사용자 입력란에 값이 잘못 들어갔다면, if문을 사용하여 사용자에게 잘못 입력하였다고 응답하는 방법이 있다. 매우 쉽게 대응할 수 있다.

- ✓ 예측 불가능한 예외 : 대표적으로 매우 많은 파일을 처리할 때 문제가 발생할 수 있다.

예측 불가능한 예외가 발생했을 경우, 인터프리터가 자동으로 이것이 예외라고 사용자에게 알려 준다. 대부분은 이러한 예외가 발생하면서 프로그램이 종료되므로 적절한 조치가 필요하다.

17-1 예외 처리

● 예외 처리 구문

- ✓ 파이썬 예외 처리의 기본 문법은 try -except문이다.

try:

예외 발생 가능 코드

except 예외 타입:

예외 발생 시 실행되는 코드

- ✓ 사용 방법은 아래와 같다.

```
for i in range(5):
```

```
    try:
```

```
        result = 10/i
```

```
        print(result)
```

```
    except ZeroDivisionError:
```

```
        print('숫자는 0으로 나눌 수 없습니다')
```

[결과]

숫자는 0으로 나눌 수 없습니다

10.0

5.0

3.3333333333333335

2.5

17-1 예외 처리

● 예외 처리 구문

- ✓ 예외 처리는 여러 개 이상 처리가 가능하다.
- ✓ 예외와 상관없이 특정한 코드를 실행하려면 `finally`를 사용하면 된다.

```
y = [10,20,30]

try:
    index, x = map(int, input('input index and number: ').split())
    print(y[index] / x)

except ZeroDivisionError as e:
    print('숫자는 0으로 나눌 수 없습니다',e)
except IndexError as e:
    print('wrong index',e)
```

[결과]

```
input index and number: 4 0
wrong index list index out of range
```

예외	내용
IndexError	리스트의 인덱스 범위를 넘어갈 때
NameError	존재하지 않는 변수를 호출할 때
ZeroDivisionError	0으로 숫자를 나눌 때
ValueError	변환할 수 없는 문자나 숫자를 변환할 때
FileNotFoundError	존재하지 않는 파일을 호출할 때

17-1 예외 처리

● 예외 처리 구문

- ✓ try-except-else문은 if-else문과 비슷한데, 해당 예외가 발생하지 않을 경우 수행할 코드를 else문에 작성하면 된다.
- ✓ try-except-finally문에서 finally문은 try-except문 안에 있는 수행 코드가 아무런 문제 없이 종료되었을 경우, 최종으로 호출하는 코드이다.

```
for i in range(5):  
    try:  
        result = 10 / i  
    except ZeroDivisionError:  
        print('숫자는 0으로 나눌 수 없습니다')  
    else:  
        print('resule:',i)  
    finally:  
        print('항상 실행')
```

[결과]

```
숫자는 0으로 나눌 수 없습니다  
항상 실행  
resule: 1  
항상 실행  
resule: 2  
항상 실행  
resule: 3  
항상 실행  
resule: 4  
항상 실행
```

17-1 예외 처리

● 예외 처리 구문

- ✓ raise문은 try-except문과 달리 필요할 때 예외를 강제로 발생시킬 때 사용하는 코드이다.

```
idata = input('정수를 입력하세요:')  
if not idata.isdigit():  
    raise ValueError('정수를 입력하지 않았습니다.')  
print(int(idata))
```

[결과]

정수를 입력하세요:hi

Traceback (most recent call last):

File "C:/Users//exceptEx.py", in <module>

raise ValueError('정수를 입력하지 않았습니다.')

ValueError: 정수를 입력하지 않았습니다.

17-1 예외 처리

● 커스텀 예외 생성

- ✓ 커스텀 예외를 생성하는 방법은 Exception을 상속받아서 새로운 클래스를 만들면 된다.
- ✓ `__init__` 메서드에서 부모 클래스의 `__init__` 메서드를 호출하면서 에러 메시지를 넣어주면 된다.

```
class NotThreeMError(Exception):
    def __init__(self):
        super().__init__('입력된 값이 3의 배수가 아닙니다.')

def t_multiple():
    try:
        x = int(input('3배수를 입력하세요: '))
        if x % 3 != 0:
            raise NotThreeMError
        print(x)
    except NotThreeMError as e:
        print('예외 발생!!!',e)

t_multiple()
```

[결과]

3배수를 입력하세요: 5

예외 발생!!! 입력된 값이 3의 배수가 아닙니다.