

Unit 38.

☀ Status	완료
👤 담당자	
📅 마감일	
📅 완료일	@2022년 12월 1일

Unit 38. 예외 처리 사용하기

예외(exception)란 코드를 실행하는 중에 발생한 에러이다

38.1 try except로 사용하기

예외처리를 하려면 try에 실행할 코드를 넣고 except에 예외가 발생했을 때 처리하는 코드를 넣는다.

```
try:
    실행할 코드
except:
    예외가 발생했을 때 처리하는 코드
```

```
try:
    x = int(input('나눌 숫자를 입력하세요: '))
    y = 10 / x
    print(y)
except: # 예외가 발생했을 때 실행됨
    print('예외가 발생했습니다.')
```

try의 코드에서 에러가 발생했을 때만 except의 코드가 실행된다.

38.1.1 특정 예외만 처리하기

except에 예외 이름을 지정해서 특정 예외가 발생했을 때만 코드가 실행되도록 하기

```
try:
    실행할 코드
except 예외이름:
    예외가 발생했을 때 처리하는 코드
```

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

try:
    index, x = map(int, input('인덱스와 나눌 숫자를 입력하세요: ').split())
    print(y[index] / x)
except ZeroDivisionError: # 숫자를 0으로 나눠서 에러가 발생했을 때 실행됨
    print('숫자를 0으로 나눌 수 없습니다.')
except IndexError: # 범위를 벗어난 인덱스에 접근하여 에러가 발생했을 때 실행됨
    print('잘못된 인덱스입니다.')
```

38.1.2 예외의 에러 메시지 받아오기

except에서 as 뒤에 변수를 지정하면 발생한 예외의 에러 메시지를 받아올 수 있다.

```
try:
    실행할 코드
except 예외 as 변수:
    예외가 발생했을 때 처리하는 코드
```

보통 변수 이름을 예외(exception)의 e를 따서 e로 짓는다

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

try:
```

```

index, x = map(int, input('인덱스와 나눌 숫자를 입력하세요: ').split())
print(y[index] / x)
except ZeroDivisionError as e:          # as 뒤에 변수를 지정하면 에러를 받아옴
    print('숫자를 0으로 나눌 수 없습니다.', e)    # e에 저장된 에러 메시지 출력
except IndexError as e:
    print('잘못된 인덱스입니다.', e)

```

모든 예외의 에러 메시지를 출력하고 싶으면 except에 Exception을 지정하고 as 뒤에 변수를 넣으면 된다.

```

except Exception as e:    # 모든 예외의 에러 메시지를 출력할 때는 Exception을 사용
    print('예외가 발생했습니다.', e)

```

38.2 else와 finally 사용하기

else는 except 바로 다음에 와야 하며 except를 생략할 수 없다.

```

try:
    실행할 코드
except:
    예외가 발생했을 때 처리하는 코드
else:
    예외가 발생하지 않았을 때 실행할 코드

```

```

try:
    x = int(input('나눌 숫자를 입력하세요: '))
    y = 10 / x
except ZeroDivisionError:    # 숫자를 0으로 나눠서 예외가 발생했을 때 실행됨
    print('숫자를 0으로 나눌 수 없습니다.')
else:                        # try의 코드에서 예외가 발생하지 않았을 때 실행됨
    print(y)

```

만약 2를 입력한다면 예외가 발생하지 않으므로 else의 코드가 실행된다.

0을 입력해서 예외가 발생한다면 except의 코드만 실행된다.

38.2.1 예외와는 상관없이 항상 코드 실행하기

- 예외 발생 여부와 상관없이 항상 코드를 실행하는 finally
- finally는 except와 else를 생략할 수 있다.

```

try:
    실행할 코드
except:
    예외가 발생했을 때 처리하는 코드
else:
    예외가 발생하지 않았을 때 실행할 코드
finally:
    예외 발생 여부와 상관없이 항상 실행할 코드

```

```

try:
    x = int(input('나눌 숫자를 입력하세요: '))
    y = 10 / x
except ZeroDivisionError:    # 숫자를 0으로 나눠서 예외가 발생했을 때 실행됨
    print('숫자를 0으로 나눌 수 없습니다.')
else:                        # try의 코드에서 예외가 발생하지 않았을 때 실행됨
    print(y)
finally:                     # 예외 발생 여부와 상관

```

38.3 예외 발생시키기

예외를 발생시킬 때는 raise에 예외를 지정하고 에러 메시지를 넣는다.

```

try:
    x = int(input('3의 배수를 입력하세요: '))
    if x % 3 != 0:                # x가 3의 배수가 아니면
        raise Exception('3의 배수가 아닙니다.')    # 예외를 발생시킴
    print(x)
except Exception as e:           # 예외가 발생했을 때 실행됨
    print('예외가 발생했습니다.', e)

```

38.3.1 raise의 처리 과정

함수 안에서 raise를 사용하지만 함수 안에는 try except가 없는 상태

```
def three_multiple():
    x = int(input('3의 배수를 입력하세요: '))
    if x % 3 != 0:
        raise Exception('3의 배수가 아닙니다.')
    print(x)

try:
    three_multiple()
except Exception as e:
    print('예외가 발생했습니다.', e)
```

x가 3의 배수가 아니면
예외를 발생시킴
현재 함수 안에는 except가 없으므로
예외를 상위 코드 블록으로 넘김

하위 코드 블록에서 예외가 발생해도 실행됨