

[Python] 파이썬 기초 12 - 예외처리 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2020-10-24 오후 7:19

URL: <https://continuous-development.tistory.com/72?category=736681>

Python

[Python] 파이썬 기초 12 - 예외처리

2020. 8. 19. 00:16 수정 삭제 공개

<예외처리 구문>

예외처리는 프로그램상에 어떠한 이유로 에러가 나서 프로그램이 강제 종료되는 것을 막아준다. 이러한 예외처리는 프로그램의 안정성에 기여한다.

예외처리 구문에는

try : 에러가 발생할 가능성이 있는 코드

except : 에러가 발생했을 때 작동하는 구문

else : 에러가 발생되지 않았을 때 작동하는 구문

finally : 위의 로직이 실행된 다음 최종적으로 실행되는 구문 (에러의 발생 유무를 떠나 나온다.)

예외 처리 구문

try : 에러가 발생 할 가능성이 있는 코드

except : 발생된 에러를 잡기위한 객체 정의

except : 발생된 에러를 잡기위한 객체 정의

else : 에러가 발생되지 않을 때 실행되는 블록 (else 는 반드시 try 와 쌍을 이룬다.)

finally : 무조건 실행 (예외의 발생 유무와 상관없이 실행 된다.)

```
def userInput():
```

```
    try:
```

```
        age = int(input("본인의 나이를 입력하세요 :"))
```

```
    except ValueError: # Value Error라는 뜻이다.
```

```
        print("숫자값을 입력해주세요.")
```

```
    else:
```

```
        print('Result - ', age)
```

```
    finally:
```

```
        print('by.hib')
```

위의 코드는 int 형태의 입력을 받고 있다. 만약 이때 문자형이나 특수문자를 입력한다면 에러를 내뱉게 되고 except구문으로 이동하게 된다. 만약 정상적인 int형의 숫자가 온다면 except구문을 건너뛰고 else 구문으로 이동하게 된다.

```
userInput()
```

```
↓ 본인의 나이를 입력하세요 : /  
숫자값을 입력해주세요.  
by.hib
```

위의 결과 값은 에러 부분으로 이동하여 예외처리를 한 결과값이다.

```
↓ 본인의 나이를 입력하세요 : 29  
Result - 29
```

위의 값은 정상적인 로직을 실행했을 때의 결과 값이다.

#에러 처리 종류에 따른 에러 처리

```

def exceptionFunc(list_data):
    try:
        sum = list_data[0] * list_data[1] * list_data[2] * list_data[3]
        if sum < 0:
            raise Exception("User Define Exception")
    except IndexError as err: # index에 대한 error
        print("4가지의 숫자를 입력하세요")
        print(str(err))
    # 여기서는 IndexError 로만 except를 잡아놔서 IndexError 에러가 아닌상태에서는 에러를 잡지 못한다.
    # 위에서는 사용자가 직접에러 처리를 했다. 그래서 이 아래구문을 통해 에러를 잡아줘야 한다.
    # 그래서 이 except 구문이 없으면 에러를 잡지 못한다. 에러의 종류가 다르기 때문이다.
    except Exception as e:
        print(str(e))
    else:
        print("합은 {} 입니다.".format(sum))
    finally:
        print("계산 끝~")

```

위의 구문에서는 첫 번째 except에서 IndexError에 관한 에러만 예외처리할 수 있다. IndexError가 아닌 경우에는 에러를 잡지 못해 프로그램상의 에러가 떠 프로그램이 종료되게 된다.

그래서 except를 하나 더 넣어 모든 경우의 에러를 Exception을 통해 잡게끔 한다.

```

list01 = [1, 2, 3, -600]
exceptionFunc(list01)

```

try 다른 위치

```

# 매개변수로 넘겨받은 각 첨자변지의 값에 제곱한 결과를 출력하려고 한다.
# 예외 발생을 확인하고 예외 처리 구문을 추가하여 정상적인 흐름의 함수 호출이 되도록
# 만들어 본다
def listExcepFunc(list):
    list_Squared = []
    try:
        for i in list:
            list_Squared.append(i ** 2)
    except:
        print("숫자가 아닌 값이 들어가 있습니다")
    else:
        print(list_Squared)
    finally:
        print('by.hib~')

def listExcepFunc2(list):
    for i in list:
        try:
            print("raw - ", i)
            squared = i**2
            print("squared - ", squared)
        except TypeError as tr:
            print("숫자가 아닌 값이 들어있습니다.")
    print("end function")

```

try 이는 for문 안에서도 사용 가능하다. 그래서 해당 for 문을 돌다가 에러가 나는 부분이 있으면 except부분을 출력하고 다시 for문을 돌린다.

```

list01 = [1, 2, 3, -600]
exceptionFunc(list01)
print("-----")
|
list02 = [10, 20, 25, 'num', 40, 50]
listExcepFunc(list02)
print("-----")
listExcepFunc2(list02)

```

```
-----
숫자가 아닌 값이 들어가 있습니다
by.hib~
-----
raw - 10
squared - 100
raw - 20
squared - 400
raw - 25
squared - 625
raw - num
숫자가 아닌 값이 들어있습니다.
raw - 40
squared - 1600
raw - 50
squared - 2500
end function
```

결괏값이 이렇게 나온다.

'Python' 카테고리의 다른 글

[Python] 파이썬 기초 14 - 아주 기초적인 pandas 사용법과 예제

[Python] 파이썬 기초 13 - 파이썬을 통한 파일 입출력 사용법

[Python] 파이썬 기초 12 - 예외처리

[Python] 파이썬 기초 11 - 객체의 4대 특성 (상속화, 캡슐화, 다형성, 추상화)

[Python] 파이썬 기초 10 - 클래스에 대한 정의와 사용법

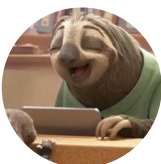
[Python] 파이썬 기초 9 - 패키지와 모듈에 대한 정의와 다양한 함수 형태

python 예외처리

파이썬 except

파이썬 예외

파이썬 예외처리



나무늘보스

혼자 끄적끄적하는 블로그 입니다.

