
예외 처리

예외 처리

❖ 예외

- 프로그램 실행 중 발생한 에러 --> 프로그램 실행 종료(디폴트)

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

Traceback (most recent call last):

File , line 2, in <module>

score = int(str)

ValueError: invalid literal for int() with base 10: '89점'

예외 처리

❖ 예외 처리

- 예외 발생을 감지하고 복구하는 방법

try:

실행할 명령

except 예외 as 변수:

오류 처리문

else:

예외가 발생하지 않을 때의 처리

예외 처리

❖ 예외 처리

```
str = "89점"
try:
    score = int(str)
    print(score)
except:
    print("예외가 발생했습니다.")

print("작업완료")
```

예외가 발생했습니다.
작업완료

예외 처리

❖ 예외 처리

```
while True:
    str = input("점수를 입력하세요 : ")
    try:
        score = int(str)
        print("입력한 점수 :", score)
        break
    except:
        print("점수 형식이 잘못되었습니다.")

print("작업완료")
```

점수를 입력하세요 : 만점
점수 형식이 잘못되었습니다.
점수를 입력하세요 : 99
입력한 점수 : 99
작업완료

예외 처리

❖ 예외의 종류

- `NameError`
- `ValueError`
- `ZeroDivisionError`
- `IndexError`
- `TypeError` 등

예외 처리

❖ 예외의 종류

```
str = "89"  
try:  
    score = int(str)  
    print(score)  
    a = str[5]  
except ValueError:  
    print("점수의 형식이 잘못되었습니다.")  
except IndexError:  
    print("첨자 범위를 벗어났습니다.")  
  
print("작업완료")
```

```
89  
첨자 범위를 벗어났습니다.  
작업완료
```

예외 처리

❖ 예외의 종류

```
str = "89"  
try:  
    score = int(str)  
    print(score)  
    a = str[5]  
except (ValueError, IndexError):  
    print("점수의 형식이나 첨자가 잘못되었습니다.")  
  
print("작업완료")
```

```
89  
점수의 형식이나 첨자가 잘못되었습니다.  
작업완료
```


예외 처리

❖ 예외의 종류

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

print("작업완료")
```

```
invalid literal for int() with base 10: '89점'
작업완료
```

예외 처리

❖ raise

- 개발자에 의해 임의로 예외를 발생시킴

```
def calcsun(n):  
    if n < 0 :  
        raise ValueError  
  
    total = 0  
    for i in range(n+1):  
        total += i  
    return total  
  
try:  
    print("~10 =", calcsun(10))  
    print("~-5 =", calcsun(-5))  
except ValueError:  
    print('입력값이 잘못되었습니다.')
```

~10 = 55

입력값이 잘못되었습니다.

자원의 정리

❖ finally

- 예외 발생 여부와 상관없이 항상 호출
- 작업의 마무리 작업(cleanup) 수행

자원의 정리

❖ finally

```
try:
    print("네트워크 접속")
    a = 2/0
    print("네트워크 통신 수행")
finally:
    print("접속 해제")

print("작업 완료")
```

네트워크 접속

접속 해제

Traceback (most recent call last):

File, line 3, in <module>

a = 2/0

ZeroDivisionError: division by zero

자원의 정리

❖ assert

- assert 조건, 메시지
 - 조건이 True이면 통과,
 - False이면 메시지를 가지는 예외 발생

```
score = 128
assert score <= 100, "점수는 100 이하여야 합니다."
print(score)
```

Traceback (most recent call last):

File, line 2, in <module>

assert score <= 100, "점수는 100 이하여야 합니다."

AssertionError: 점수는 100 이하여야 합니다.