

2021학년도 2학기 언어와 컴퓨터

제7강 예외 처리

박수지

서울대학교 인문대학 언어학과

2021년 9월 29일 수요일

오늘의 목표

- 1 `def` 문을 사용하여 함수를 정의할 수 있다.
- 2 `return` 키워드의 작동 방식을 이해할 수 있다.
- 3 `try-except` 문으로 예외를 처리할 수 있다.
- 4 `raise` 키워드로 예외를 일으킬 수 있다.

의문

반환값을 여러 개 가질 수 있는가?

```
1 def example0():  
2     return 0  
3     return 1  
4  
5 print(example0())
```

return의 역할

실행을 멈추고 함수가 호출된 곳으로 돌아간다.

```
1 def example1():  
2     print('출력됩니다.')  
3     return  
4     print('출력될까요?')  
5  
6 example1()
```

```
1 def example2():  
2     for i in range(3):  
3         print(f'{i}번째 출력')  
4         return  
5  
6 example2()
```

if 절 내의 return

이후의 내용은 else 절 내에 있는 것과 같아진다.

```
1 def example3(n):  
2     if n % 2:  
3         print('홀수입니다.')  
4         return  
5     else:  
6         print('짝수입니다.')  
7         print('언제 출력될까요?')  
8  
9 example3(2)  
10 example3(3)
```

```
1 def example4(n):  
2     if n % 2:  
3         return '홀'  
4     return '짝'  
5  
6 print(example4(4))  
7 print(example4(5))
```

사실 가장 간단한 홀짝 판정

```
1 def isodd(n):  
2     return bool(n % 2)
```

예외 처리 구문

try/except

형식

```
1 try:  
2     <예외가 발생할 가능성이 있는 코드>  
3 except:  
4     <예외가 발생했을 때 실행할 코드>
```

예외 처리 구문

예시: 예외 유형별로 처리하기

```
1 def myint(n):
2     try:
3         # n의 자료형과 값에 따라 예외가 발생할 수 있는 코드
4         return int(n)
5     except ValueError: # int('2.0')
6         print('올바른 정수 문자열을 입력하세요.')
7     except TypeError: # int([2])
8         print('올바른 자료형을 입력하세요.')
9     finally:
10        print('수고하셨습니다.')
```

예외 처리 구문

예시: 모든 예외를 한꺼번에 처리하기

```
1 def myint(n):  
2     try:  
3         return int(n)  
4     except: # 임의의 예외  
5         print('어쨌든 무엇인가가 잘못되었습니다.')
```

실행

```
>>> myint(2)  
2  
>>> myint('2.0')  
어쨌든 무엇인가가 잘못되었습니다.  
>>> myint([2])  
어쨌든 무엇인가가 잘못되었습니다.
```


예외 발생 구문

try/except

형식

```
1 | raise <예외 유형>(예외 메시지)
```

예외 유형 `KeyError`, `TypeError`, `ValueError`, `ZeroDivisionError`,
...

예외 발생 구문

예시: 홀수가 아니면 에러를 일으키는 함수

```
1 def odd_only(n):  
2     if not isinstance(n, int):  
3         raise TypeError('정수를 주세요!!')  
4     elif n % 2:  
5         print('좋아요!!')  
6     else:  
7         raise ValueError('홀수를 주세요!!')
```

`isinstance(5, int)` 5의 자료형이 int인지 판정하는 함수 호출

예외 발생 구문

예시: 홀수가 아니면 에러를 일으키는 함수

실행

```
>>> odd_only(3)
```

```
좋아요!!
```

```
>>> odd_only(3.0)
```

```
...
```

```
TypeError: 정수를 주세요!!
```

```
>>> odd_only(4)
```

```
...
```

```
ValueError: 홀수를 주세요!!
```

요약

List comprehension

```
[<표현식> for <반복자> in <열> (if <조건>)]
```

str.join()

```
<구분자>.join(<열>)
```

모듈에서 객체 가져오기

```
from <모듈> import <객체>
```

함수 정의하기

```
def 문
```

예외 처리하기

```
try-except 문
```

더 생각해 볼 것

- `zip()` 함수에 대하여 알아보자.

다음 시간에 할 것

- 정규표현식(regular expressions)
- SLP3 Ch.2 읽어 오기
<https://web.stanford.edu/~jurafsky/slp3/2.pdf>