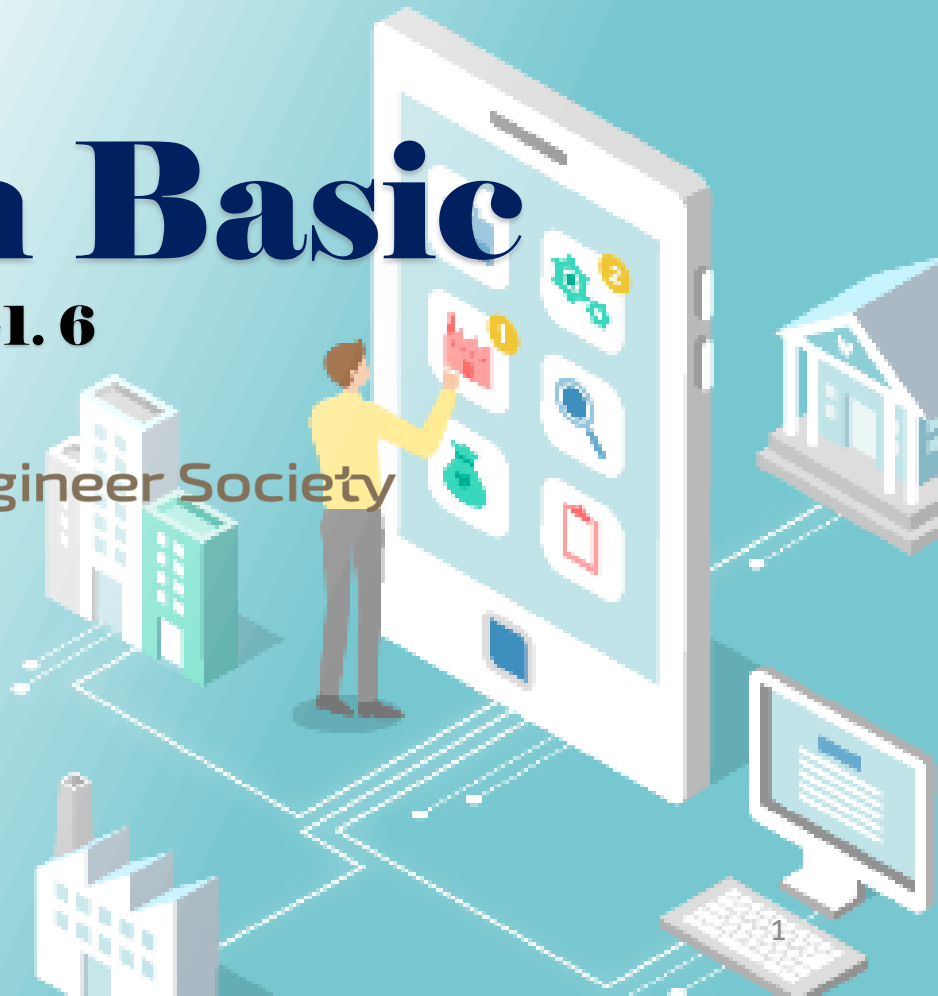
Python Basic

2021. 6

Soft Engineer Society



예외 처리

- 예외 처리

```
try:
    실행할 명령
except [(처리할 예외의 종류, ...) [as 예외를 받을 변수명]]:
    오류 처리문
[except ...]
[else:
    예외가 발생하지 않았을 때의 처리문]
[finally:
    반드시 실행할 문장]
```

► 주요 예외

함수	설명
SyntaxError	문법에 맞지 않는 구문을 작성함.
NameError	존재하지 않는 요소를 호출함.
TypeError	타입이 맞지 않음.
ValueError	타입은 맞지만 값의 형식이 잘못됨.
ZeroDivisionError	0으로 나눔.
IndexError	컬렉션의 범위를 벗어난 요소를 호출함.

► 예외 예문

```
try:
    a = int("a")
    print(a)
except:
    print("An Exception has been occured.")
print("The End.")
```

```
try:
    a = int("9999")
except:
    print("An Exception has been occured.")
else:
    print("Finished without any Exception.")
finally:
    print("The End.")
```

► 예외 예문

```
try:
    a = str(9999)
    print(a[5])
except ValueError:
    print("ValueError has been occured")
except IndexError:
    print("IndexError has been occured")
finally:
    print("The End")
```

```
try:
    a = str(9999)
    print(a[5])
except (ValueError, IndexError) as e:
    print("An Exception has been occured")
    print(e)
finally:
    print("The End")
```

raise

- raise
  - 고의적으로 예외를 발생시킴.

```
def out(n):
    if n == 0:
        raise ValueError
    print(n)

try:
    out(1)
    out(0)
except ValueError:
    print("Do not input 0.")
```

사용자 정의 예외

- 사용자 정의 예외
  - Exception 클래스를 상속받은 예외 처리 클래스.

```
class myException(Exception):
    def __init__(self):
        super().__init__("My Exception has been occured.")

try:
    raise myException
except myException as e:
    print(e)
```

assert

- assert
  - 문장을 검사함.
  - 문장이 잘못되었을 경우 AssertionError를 발생시킴.

```
assert 검사할 문장[, 출력할 메시지]
```

```
a = "b"
assert a == "a", "Variable 'a' is not 'a'"
print(a)
```