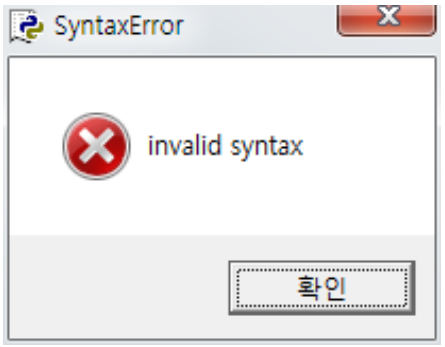
오늘의 강의 목표

- Exception에 대한 이해
- Python의 exception handling 방식에 대한 이해

Errors

- Syntax Errors

```
while True
    print("Hello World")
```



코드를 이해할 수 없음 ➔ 실행 거부

- Exceptions

```
a = 0
b = 10 / a
```

코드 실행 중 예외 상황 발생 ➔ Exception

```
Traceback (most recent call last):
  File "C:/Python34/t.py", line 2, in <module>
    b = 10 / a
ZeroDivisionError: division by zero
```

More Exceptions

- Invalid variable name

```
>>> age = 37
>>> print(aze)
Traceback (most recent call last):
  File "<pyshell#19>", line 1, in <module>
    print(aze)
NameError: name 'aze' is not defined
```

- Type issue

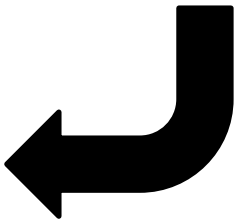
```
>>> age = 37
>>> print(age + "years old")
Traceback (most recent call last):
  File "<pyshell#23>", line 1, in <module>
    print(age + "years old")
TypeError: unsupported operand type(s) for +: 'int' and
'str'
```

Built-in Exceptions

- ZeroDivisionError, NameError, TypeError, …

https://docs.python.org/3.4/library/exceptions.html#exception-hierarchy

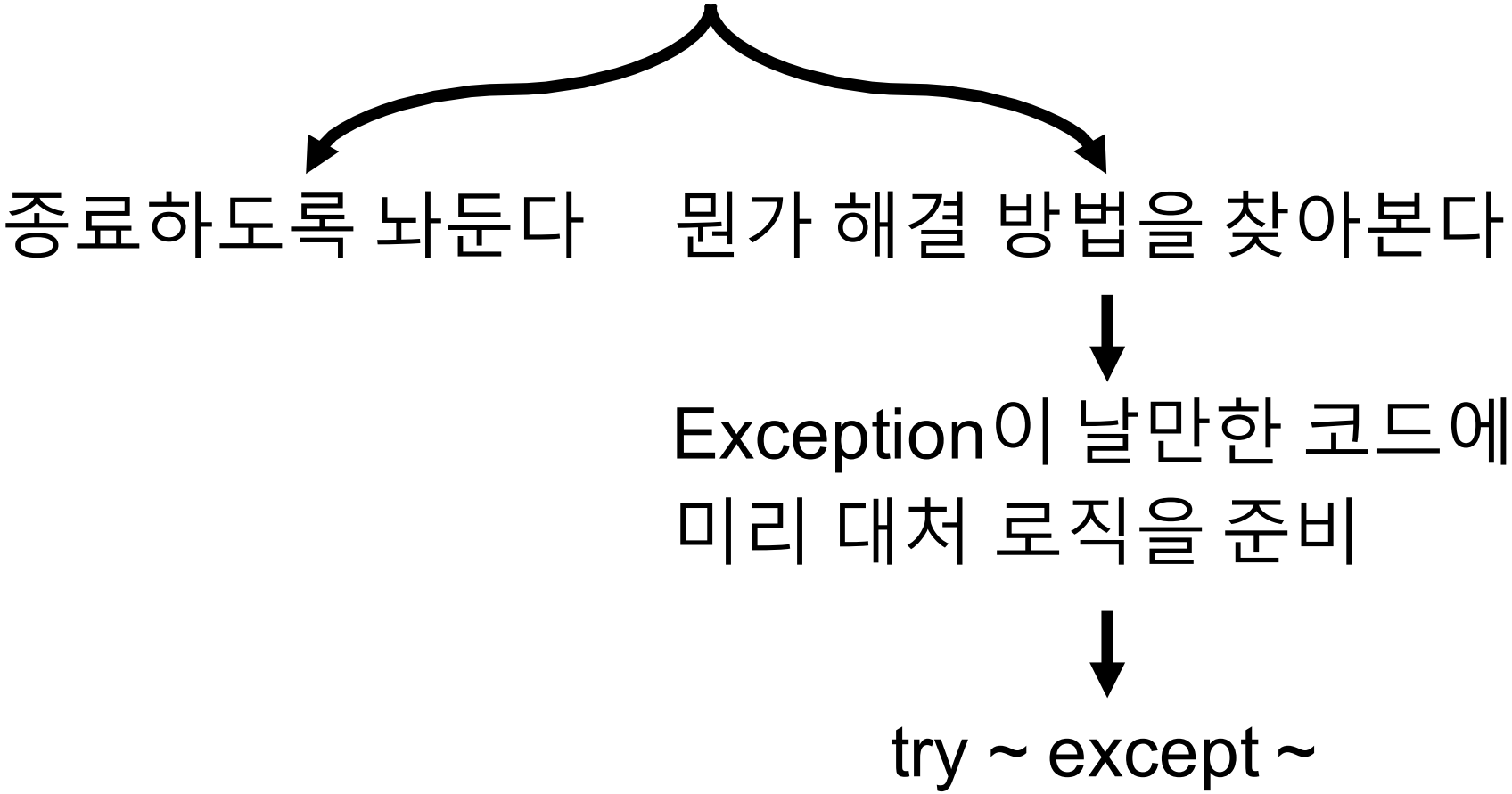
```
BaseException
 +-- SystemExit
 +-- KeyboardInterrupt
 +-- GeneratorExit
 +-- Exception
      +-- StopIteration
      +-- ArithmeticError
      |    +-- FloatingPointError
      |    +-- OverflowError
      |    +-- ZeroDivisionError
      +-- AssertionError
      +-- AttributeError
      +-- BufferError
      +-- EOFError
      +-- ImportError
      +-- LookupError
      |    +-- IndexError
      |    +-- KeyError
      +-- MemoryError
      +-- NameError
      |    +-- UnboundLocalError
      +-- OSError
      |    +-- BlockingIOError
      |    +-- ChildProcessError
      |    +-- ConnectionError
      |    |    +-- BrokenPipeError
      |    |    +-- ConnectionAbortedError
      |    |    +-- ConnectionRefusedError
      |    |    +-- ConnectionResetError
      |    +-- FileExistsError
      |    +-- FileNotFoundError
      |    +-- InterruptedError
      |    +-- IsADirectoryError
      |    +-- NotADirectoryError
      |    +-- PermissionError
      |    +-- ProcessLookupError
      |    +-- TimeoutError
      +-- ReferenceError
      +-- RuntimeError
      |    +-- NotImplementedError
      +-- SyntaxError
      |    +-- IndentationError
      |         +-- TabError
      +-- SystemError
      +-- TypeError
      +-- ValueError
      |    +-- UnicodeError
      |         +-- UnicodeDecodeError
      |         +-- UnicodeEncodeError
      |         +-- UnicodeTranslateError
      +-- Warning
           +-- DeprecationWarning
           +-- PendingDeprecationWarning
           +-- RuntimeWarning
           +-- SyntaxWarning
           +-- UserWarning
           +-- FutureWarning
           +-- ImportWarning
           +-- UnicodeWarning
           +-- BytesWarning
           +-- ResourceWarning
```



Exception Hierarchy

Exception 발생에 대처하는 자세

Exception이 발생하면 프로그램이 종료



try ~ except ~ else

- Exception 발생시 특정 코드 실행

```
filename = input("Enter the file name: ")
try:
    f = open(filename)
except FileNotFoundError:
    print("No such file")
else:
    for line in f:
        print(line)
```

Exception 발생시 실행

Exception 발생하지 않은 경우 실행 (Optional)

```
Enter the file name: thisfile
No such file
```

try ~ except ~ else

- (A) 코드에서 특정 Exception *Exception* 발생시
  (B) 코드 실행, 아니면 (C) 코드 실행

```
try:
    … (A)
except Exception:
    … (B)
else:
    … (C)
```

try ~ except ~ else

- (A) 코드에서 임의의 Exception 발생시 (B) 코드 실행, 아니면 (C) 코드 실행

```
try:
    … (A)
except:
    … (B)
else:
    … (C)
```

try ~ except ~ else

- (A) 코드에서 여러 개의 Exception중 하나라도 발생시 (B) 코드 실행 처리, 아니면 (C) 코드 실행

```
try:
    … (A)
except (Exception1, Exception2, …, ExceptionN):
    … (B)
else:
    … (C)
```

finally

- (D) 코드는 무조건 실행. 심지어 (B)나 (C)에서 return을 해도 실행

```
try:
    … (A)
except Exception:
    … (B)
else:
    … (C)
finally:
    … (D)
```

Advanced Topics

- User-defined Exceptions

Questions