

w07-Lab1

Errors and Exceptions

Assembled for 204111
by Kittipitch Kuptavanich

Exceptions

- ในบางกรณีโปรแกรมที่ทำงานอยู่เกิดความผิดพลาด (Run Time Error) เนื่องจากตรรกะในการเขียนโปรแกรม ในระดับที่เป็นเหตุให้โปรแกรมต้องหยุดการทำงาน
- เราเรียกข้อผิดพลาดชนิดนี้ว่า Exception
 - (โดยคำศัพท์แล้ว คำว่า Exception หมายถึงข้อยกเว้น)

```
>>> 10 * (1/0)
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
ZeroDivisionError: division by zero
```

Exception
Type

Exception
Details

Syntax Errors

- Syntax Error หรือ Parsing Error น่าจะเป็นความผิดพลาดที่พบได้บ่อยที่สุดในช่วงฝึกเขียนภาษา Python

```
>>> while True print('Hello world')
File "<stdin>", line 1, in ?
    while True print('Hello world')
                ^
```

SyntaxError: invalid syntax

- บรรทัดที่เกิดปัญหาจะถูกแสดงผลพร้อมลูกศรเล็ก ๆ ระบุจุดแรกที่พบข้อผิดพลาด
 - ในกรณีนี้ระบุข้อผิดพลาดที่ฟังก์ชัน `print()` เนื่องจากควรมี Colon : หลัง Condition ของ `while`

Exception Types

- Python มี Built-in Exception Types หลายชนิด (นอกจากนี้เรายังสามารถกำหนด Exception ชนิดใหม่ ๆ เพิ่มได้เอง)
- Exception มาตรฐานที่พบบ่อยได้แก่

ImportError

```
>>> import whatever
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    import whatever
ImportError: No module named 'whatever'
```

ImportError

Exception Types [2]

IndexError

```
>>> a = "hello"
>>> a[5]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    a[5]
IndexError: string index out of range
```

NameError

```
>>> x + 4
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    x + 4
NameError: name 'x' is not defined
```

5

Handling Exceptions

- เราสามารถออกแบบโปรแกรมที่เขียนให้ตรวจจับ (Catch) จัดการ (Handle) Exception เกิดขึ้นได้
 - เพื่อแก้ไขความผิดพลาดที่เกิดขึ้น
 - หรือเพื่อจบการทำงานของโปรแกรมอย่างเหมาะสม (โดยไม่ Crash)

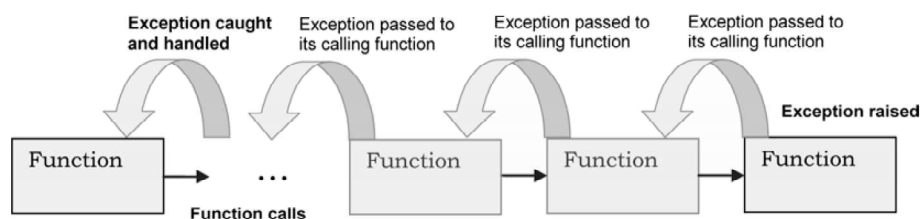


FIGURE 8-9 The Propagation of Exceptions

7

Exception Types [3]

TypeError

```
>>> '2' + 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    '2' + 2
TypeError: Can't convert 'int' object to str implicitly
```

ValueError

```
>>> import math
>>> math.sqrt(-6)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
    math.sqrt(-6)
ValueError: math domain error
```

6

Handling Exceptions [2]

- เราใช้คำสั่ง **try** และ **except** เพื่อการทำ Exception Handling

```
08 import math
09
10 x = float(input("enter a number: "))
11 try:
12     print("Sqrt of ", x, "is:", math.sqrt(x))
13 except ValueError:
14     print("Sqrt of a negative number is not supported")
15     print("Using absolute value instead")
16     print("Sqrt of ", -x, "is:", math.sqrt(-x))
17
```

Exception
Handling

Might raise
ValueError

8

Handling Exceptions [3]

- Another Example

```
19 def readInt():
20     while True:
21         val = input('Enter an integer: ')
22         try:
23             val = int(val)
24             return val
25         except ValueError:
26             print(val, 'is not an integer')
```

- Loop จนกว่า User จะป้อนค่า Input ที่ถูกต้อง

References

- <https://docs.python.org/3/tutorial/errors.html>
- <https://docs.python.org/3/library/exceptions.html>
- <https://wiki.python.org/moin/HandlingExceptions>
- Guttag, John V. *Introduction to Computation and Programming Using Python, Revised*
- Miller & Ranum , *Problem Solving with Algorithms and Data Structures Using Python*