

# Exception and Error Handling Methods

ฉัตรชัย เกษมทวีโชค

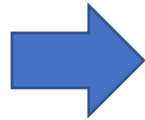
[Chatchai.kase@ku.th](mailto:Chatchai.kase@ku.th)

# ข้อผิดพลาด (Errors)

- ข้อผิดพลาดในการเขียนโปรแกรมและการทำงานของโปรแกรมคอมพิวเตอร์สามารถแบ่งออกเป็น 3 ประเภท คือ
  - Syntax Error** คือข้อผิดพลาดที่เกิดระหว่างการเขียนโปรแกรม ที่รูปแบบคำสั่งไม่ถูกต้องหลักภาษาโปรแกรม (Syntax)
  - Runtime Error** คือข้อผิดพลาดระหว่างการทำงานโปรแกรม ที่ทำให้โปรแกรมทำงานไม่สมบูรณ์หรือหยุดก่อนการทำงานเสร็จสิ้น
  - Logic Error** คือข้อผิดพลาดที่เกิดจากผลลัพธ์ของการทำงานโปรแกรมไม่ถูกต้อง โปรแกรมทำงานได้เสร็จสิ้น แต่ผลลัพธ์ไม่ตรงกับความต้องการหรือข้อกำหนดของโปรแกรม

```
a = 8  
b = 10  
c = a b
```

Syntax Error

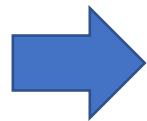


```
File "<ipython-input-8-3b3ffcedf995>", line 3  
    c = a b  
        ^  
SyntaxError: invalid syntax
```

# ข้อผิดพลาด (Errors)

- **Out of Memory Error** เป็นข้อผิดพลาดเกิดขึ้นในหน่วยความจำของคอมพิวเตอร์ไม่เพียงพอต่อข้อมูลที่ต้องการใช้งานในโปรแกรม (หน่วยความจำน้อยเกินไป) หรืออาจจะมีการตั้งค่าหน่วยความจำที่ไม่เพียงพอต่อการใช้งาน
  - `OutOfMemoryError` ([Source](#))
- **Recursion Error** เป็นข้อผิดพลาดในการเรียกฟังก์ชัน Recursive function ต่อเนื่องกันไปเรื่อย ๆ จนกระทั่ง Stack เต็ม

```
def recursion():  
    return recursion()  
  
recursion()
```

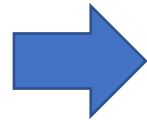


```
...  
<ipython-input-2-5395140f7f05> in recursion()  
      1 def recursion():  
----> 2     return recursion()  
  
RecursionError: maximum recursion depth exceeded
```

# ข้อผิดพลาด (Errors)

- Indentation Error เป็นข้อผิดพลาดที่เกิดจากการไม่ได้ indent (TAB) คำสั่งให้เหมาะสมกับตำแหน่งคำสั่งในโปรแกรม

```
for i in range(10):  
    print('Hello world')
```

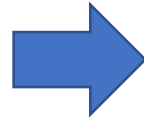


```
File "<ipython-input-6-628f419d2da8>", line 2  
    print('Hello world')  
      ^  
IndentationError: expected an indented block
```

# Exceptions

- Exceptions เป็นข้อผิดพลาดประเภท Run-time Error คือข้อผิดพลาดจะปรากฏเมื่อโปรแกรมกำลังทำงานแล้วเกิดข้อผิดพลาดในการทำงาน
  - Type Error คือการกำหนดค่าตัวแปรผิดพลาด ทำให้โปรแกรมทำงานไม่ได้

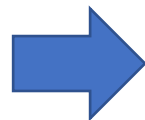
```
a = 2  
b = 'DataCamp'  
a + b
```



```
<ipython-input-7-86a706a0ffdf> in <module>  
    1 a = 2  
    2 b = 'DataCamp'  
----> 3 a + b  
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

- Zero Division Error คือการหารด้วยค่าศูนย์

```
100 / 0
```



```
<ipython-input-43-e9e866a10e2a> in <module>  
----> 1 100 / 0  
ZeroDivisionError: division by zero
```

# Exception Handling Code

**try:**

```
f = open("demofile.txt")
```

```
f.write("Lorum Ipsum")
```

**except FileNotFoundError:**

```
print("File is not found")
```

**except:**

```
print("Something went wrong with the file")
```

**else:**

```
print("All things is good.")
```

**finally:**

```
f.close()
```

**Try:** คำสั่งที่เริ่มต้นการตรวจสอบความผิดพลาดของชุดคำสั่งที่จะทำงานหลังคำสั่งนี้

**Except:** การประกาศชื่อประเภท exception ที่ต้องการดักจับแล้วจะดำเนินการ

**Else:** เป็นการประกาศชุดคำสั่งจะทำงานเมื่อไม่มี exception ใดเกิดขึ้น

**Finally:** เป็นการประกาศชุดคำสั่งที่ต้องทำงานเสมอทุกกรณีไม่ว่าจะมีข้อผิดพลาด exception หรือไม่ก็ได้

# Built-in Exceptions

- KeyboardInterrupt เป็นข้อผิดพลาดกดคีย์บอร์ด Ctrl+C หรือ Ctrl+Z เพื่อหยุดการทำงานของโปรแกรม
- Zero Division เป็นข้อผิดพลาดเมื่อการหารด้วยศูนย์

**try:**

```
inp = input()
print ('Press Ctrl+C or Interrupt the Kernel:')
```

**except KeyboardInterrupt:**

```
print ('Caught KeyboardInterrupt')
```

**else:**

```
print ('No exception occurred')
```

**try:**

```
a = 100 / 0
```

```
print (a)
```

**except ZeroDivisionError:**

```
print ("Zero Division Exception Raised." )
```

**else:**

```
print ("Success, no error!")
```

# Built-in Exceptions

- Overflow Error เป็นข้อผิดพลาดมีการกำหนดค่าที่เกินกว่าขอบเขตที่ตัวแปรหรือโปรแกรมจะรับได้
- Assertion Error เป็นข้อผิดพลาดที่คำสั่ง assert สร้างขึ้นเมื่อ condition มีค่าความจริงเป็นเท็จ (FALSE)

try:

```
import math
```

```
print(math.exp(1000))
```

except OverflowError:

```
print ("Overflow Exception Raised.")
```

else:

```
print ("Success, no error!")
```

try:

```
a = 100
```

```
b = "DataCamp"
```

```
assert a == b
```

except AssertionError:

```
print ("Assertion Exception Raised.")
```

else:

```
print ("Success, no error!")
```



# Built-in Exceptions

- Value Error เป็นข้อผิดพลาดที่ค่าในตัวแปรไม่สามารถใช้งานกับฟังก์ชันที่เรียกใช้ได้
- Lookup Error เป็นข้อผิดพลาดที่คำสั่งที่มีอ้างอิงตำแหน่งข้อมูลใน List, Set, Tuple, Dictionary ที่ไม่ถูกต้อง
  - Key Error
  - Index Error

```
try:
```

```
    print (float('Hello'))
```

```
except ValueError:
```

```
    print("ValueError: could not convert string to float ")
```

```
else:
```

```
    print ('Success, no error!')
```

```
try:
```

```
    a = {1:'a', 2:'b', 3:'c'}
```

```
    print (a[4])
```

```
except LookupError:
```

```
    print ("Key Error Exception Raised.")
```

```
else:
```

```
    print ("Success, no error!")
```

# Built-in Exceptions

- Name Error เป็นข้อผิดพลาดที่มีการเรียกใช้ค่าในตัวแปรที่ไม่มีอยู่ในโปรแกรม
- Type Error เป็นข้อผิดพลาดที่ชนิดข้อมูลของตัวแปรไม่สอดคล้องกับคำสั่งที่ใช้งาน หรือค่าในตัวแปรไม่เหมาะสมต่อการคำนวณ

```
try:
```

```
    print (ans)
```

```
except NameError:
```

```
    print ("NameError: name 'ans' is not defined")
```

```
else:
```

```
    print ("Success, no error!")
```

```
try:
```

```
    a = 5
```

```
    b = "DataCamp"
```

```
    c = a + b
```

```
except TypeError:
```

```
    print ('TypeError Exception Raised')
```

```
else:
```

```
    print ('Success, no error!')
```

# Raise an exception

- ผู้พัฒนาโปรแกรมไพธอนสามารถสร้าง exception ขึ้นเพื่อแสดงข้อผิดพลาดในการทำงานโปรแกรมตามเงื่อนไขที่กำหนดไว้ โดยใช้คำสั่ง raise

```
x = -1  
if x < 0:  
    raise Exception("Sorry, no numbers below zero")
```

สร้าง Exception ใหม่

```
x = "hello"  
if not type(x) is int:  
    raise TypeError("Only integers are allowed")
```

ใช้ built-in Exception เดิมมาเปลี่ยนข้อความ