巨匠直播教學

APCS Python語法基礎班

例外處理

本堂教學重點

1. 程式偵錯
2. 例外處理機制
3. 拋出例外

課程內容

1. **例外處理基礎**
   - 1-1.程式偵錯
   - 1-2.例外處理機制
   - 1-3.例外處理語法
2. **例外進階處理**
   - 2-1. 例外說明
   - 2-2. 堆疊呼叫例外處理
   - 2.3. 手動拋出例外

課程內容

1. **例外處理基礎**
   - 1-1.程式偵錯
   - 1-2.例外處理機制
   - 1-3.例外處理語法

2. **例外進階處理**
   - 2-1. 例外說明
   - 2-2. 堆疊呼叫例外處理
   - 2.3. 手動拋出例外

Python 程式偵錯

- Syntax Errors 語法錯誤
  - 該行語法錯誤，Python 無法編譯執行。
  - 修正語法錯誤
- Exception 例外
  - 語法撰寫上沒有錯誤，執行時因資料邏輯或外部系統狀態發生錯誤，Python 會引發例外(Exception)
  - 如果例外沒有被處理而傳遞到Python直譯器，程式中斷執行

程式範例

exception1.py - D:\Pytho...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 1
nums = [1, 3, 5, 7, 9]
print(num1 * num3)
print(num1 / num2)
print(num1/(num2-1))
print(nums[5])
```

Ln: 4 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
==== RESTART: D:\PythonJunior\Examples\Ch7\exception1.py ===
Traceback (most recent call last):
  File "D:\PythonJunior\Examples\Ch7\exception1.py", line 4,
in <module>
    print(num1 * num3)
NameError: name 'num3' is not defined
>>>
```

Ln: 9 Col: 4

程式範例

*exception1.py - D:\Pyth...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 1
nums = [1, 3, 5, 7, 9]
#print(num1 * num3)
print(num1 / num2)
print(num1/(num2-1))
print(nums[5])
```

Ln: 8 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
==== RESTART: D:\PythonJunior\Examples\Ch7\exception1.py ===
10.0
Traceback (most recent call last):
  File "D:\PythonJunior\Examples\Ch7\exception1.py", line 6,
in <module>
    print(num1/(num2-1))
ZeroDivisionError: division by zero
>>>
```

Ln: 16 Col: 4

程式範例

exception1.py - D:\Pytho...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 1
nums = [1, 3, 5, 7, 9]
#print(num1 * num3)
print(num1 / num2)
#print(num1/(num2-1))
print(nums[5])
```

Ln: 8 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
==== RESTART: D:\PythonJunior\Examples\Ch7\exception1.py ===
10.0
Traceback (most recent call last):
  File "D:\PythonJunior\Examples\Ch7\exception1.py", line 7,
in <module>
    print(nums[5])
IndexError: list index out of range
>>>
```

Ln: 27 Col: 4

Python 內建例外類別

```
BaseException
 +-- SystemExit
 +-- KeyboardInterrupt
 +-- GeneratorExit
 +-- Exception
      +-- StopIteration
      +-- ArithmeticError
      |    +-- FloatingPointError
      |    +-- OverflowError
      |    +-- ZeroDivisionError
      +-- AssertionError
      +-- AttributeError
      +-- BufferError
      +-- EnvironmentError
      |    +-- IOError
      |    +-- OSError
      |         +-- WindowsError (Window
      |         +-- VMSError (VMS)
      +-- EOFError
      +-- ImportError
      +-- LookupError
      |    +-- IndexError
      |    +-- KeyError
      +-- MemoryError
      +-- NameError
      |    +-- UnboundLocalError
      +-- ReferenceError
      +-- RuntimeError
      |    +-- NotImplementedError
      +-- SyntaxError
      |    +-- IndentationError
      |         +-- TabError
      +-- SystemError
      +-- TypeError
      +-- ValueError
      |    +-- UnicodeError
      |         +-- UnicodeDecodeError
      |         +-- UnicodeEncodeError
      |         +-- UnicodeTranslateError
      +-- Warning
           +-- DeprecationWarning
           +-- PendingDeprecationWarning
           +-- RuntimeWarning
           +-- SyntaxWarning
           +-- UserWarning
           +-- FutureWarning
           +-- ImportWarning
           +-- UnicodeWarning
           +-- BytesWarning
```

Python 中常見的內建例外

例外名稱	描述
ArithmeticError	用於數值計算出現的所有錯誤的基礎類別。
OverflowError	當計算超過數字類型的最大或最小範圍限制時引發
FloatingPointError	浮點數資料計算失敗時引發。
ZeroDivisionError	對所有數字類型進行除法計算失敗時引發。
IndexError	列表越界引發的例外
ValueError	搜索列表中不存在的值或資料轉型錯誤引發的例外
KeyError	使用字典中不存在的關鍵字引發的例外
NameError	使用不存在的變數名稱引發的例外
AttributeError	呼叫不存在的方法引發的例外

Python 中常見的內建例外

例外名稱	描述
ImportError	匯入模組出錯引發的例外
IOError	I/O 操作引發的例外，如開啟檔案出錯等
EOFError	遇到檔案尾端引發的例外
TabError	敘述區塊縮排不正確引發的例外
TypeError	嘗試對指定資料類型進行無效的操作或功能時引發

Q：連連看- 例外發生的原因

1) ZeroDivisionError
2) FileNotFoundError
3) ValueError
4) TypeError
5) IndexError
6) KeyError
7) NameError
8) EOFError
9) IOError
10) TabError

a) 運算的資料型態不一致
b) 檔案發生輸出入錯誤
c) 索引超過範圍
d) 縮排不一致
e) int()函式傳入資料型態不合法
f) 變數名稱不存在
g) 除法計算時除數為0
h) 字典Key不存在
i) 檔案已達尾端
j) 檔案找不到

例外處理機制

- 例外處理機制
  - 例外並不全然是程式的邏輯錯誤
    - 例如：開啟檔案時檔案不存在、網路連線中斷、資料庫無回應
    - 當外部因素消失，程式恢復正常運作
  - 使用例外處理保護程式不至於中斷
    - 捕捉這些例外，並撰寫相關的處理流程

EAFP 設計原則

- EAFP 設計原則 It's Easier to Ask Forgiveness than Permission
  - 傳統程式採用LBYL原則(Look Before You Leap)
    - 事先分析各種可能情況，不同情況進行不同流程
    - 程式中撰寫大量的狀態檢查的判斷式
    - 程式碼變得很長且難懂
  - Python主張不作狀態檢查，直接撰寫主要流程
    - 發生問題再用例外機制處理
    - 程式更簡潔、優雅
  - 例外處理十分重要

例外處理

- 例外處理
  - try / except : 捕捉發生的例外及其處理流程。
  - try / else : 未發生例外時執行流程。
  - try / finally: 不論有沒有發生例外都會處理的流程。
- 例外丟出
  - raise : 在程式碼中手動觸發或重丟捕捉到的例外。

例外處理語法

- try / except：捕捉並處理 Python 引發的例外狀況。

try:
    受例外機制保護的程式區塊
    區塊內若產生例外狀況會停止繼續向下執行，
    跳到 except 區塊。

except:
    try 區塊產生例外時執行此區塊。

程式範例

try1.py - D:\PythonJunior\Exampl...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1, 3, 5, 7, 9]
try:
    print("output")
    print(num1 * num3)
    print(num1 / (num2))
    print(nums[100])
except:
    print("產生例外")
print('程式結束')
```

Ln: 12 Col: 0

try1.py - D:\PythonJunior\Exampl...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1, 3, 5, 7, 9]
try:
    print("output")
    #print(num1 * num3)
    print(num1 / (num2))
    print(nums[100])
except:
    print("產生例外")
print('程式結束')
```

Ln: 6 Col: 5

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
>>>
======= RESTART: D:\PythonJunior\
Examples\Ch7\try1.py ======
output
產生例外
程式結束
>>>
======= RESTART: D:\PythonJunior\
Examples\Ch7\try1.py ======
output
產生例外
程式結束
>>>
======= RESTART: D:\PythonJunior\
Examples\Ch7\try1.py ======
output
產生例外
程式結束
>>>
======= RESTART: D:\PythonJunior\
Examples\Ch7\try1.py ======
output
程式結束
>>>
```

Ln: 52 Col: 4

try1.py - D:\PythonJunior\Exampl...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1, 3, 5, 7, 9]
try:
    print("output")
    #print(num1 * num3)
    #print(num1 / (num2))
    print(nums[100])
except:
    print("產生例外")
print('程式結束')
```

Ln: 12 Col: 0

try1.py - D:\PythonJunior\Exampl...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1, 3, 5, 7, 9]
try:
    print("output")
    #print(num1 * num3)
    #print(num1 / (num2))
    #print(nums[100])
except:
    print("產生例外")
print('程式結束')
```

Ln: 12 Col: 0

例外處理語法

- try / except：捕捉並處理Python 引發的特定例外狀況。
  - 宣告要處理的例外類別
    - 逐一比對例外型態
    - 發生的例外未被宣告時，程式仍會中斷

```
try:
    受例外機制保護的程式區塊
except  例外類別一:
    try 區塊產生例外類別一時執行
except  例外類別二:
    try 區塊產生例外類別二時執行
```

程式範例

try2.py - D:\PythonJunior\Examples\Ch7\try2.py (3.8.1)

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    print(num1/num2)
    print(num1*num3)
    print(nums[100])
except ZeroDivisionError:
    print('Error發生，除以0')
except NameError:
    print('Error發生，使用沒有宣告過的變數')
```

Ln: 12 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
>>>
= RESTART: D:\PythonJunior\Examples\
Ch7\try2.py
Error發生，除以0
>>>
= RESTART: D:\PythonJunior\Examples\
Ch7\try2.py
Error發生，使用沒有宣告過的變數
>>>
= RESTART: D:\PythonJunior\Examples\
Ch7\try2.py
Traceback (most recent call last):
  File "D:\PythonJunior\Examples\Ch7
\try2.py", line 7, in <module>
    print(nums[100])
IndexError: list index out of range
>>>
```

Ln: 64 Col: 4

try2.py - D:\PythonJunior\Examples\Ch7\try2.py (3.8.1)

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    #print(num1/num2)
    print(num1*num3)
    print(nums[100])
except ZeroDivisionError:
    print('Error發生，除以0')
except NameError:
    print('Error發生，使用沒有宣告過的變數')
```

Ln: 12 Col: 0

try2.py - D:\PythonJunior\Examples\Ch7\try2.py (3.8.1)

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    #print(num1/num2)
    #print(num1*num3)
    print(nums[100])
except ZeroDivisionError:
    print('Error發生，除以0')
except NameError:
    print('Error發生，使用沒有宣告過的變數')
```

Ln: 12 Col: 0

程式範例

try3.py - D:\PythonJunior\Examples\Ch7\try3.py (3.8.1)

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    print(num1/num2)
    print(num1*num3)
    print(nums[100])
except ZeroDivisionError:
    print('Error發生，除以0')
except NameError:
    print('Error發生，使用沒有宣告過的變數')
except:
    print('Error發生')
```

Ln: 14 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
>>>
======= RESTART: D:\PythonJunior\
Examples\Ch7\try3.py ======
Error發生，除以0
>>>
======= RESTART: D:\PythonJunior\
Examples\Ch7\try3.py ======
Error發生，使用沒有宣告過的變數
>>>
======= RESTART: D:\PythonJunior\
Examples\Ch7\try3.py ======
Error發生
>>>
```

Ln: 12 Col: 4

try3.py - D:\PythonJunior\Examples\Ch7\try3.py (3.8.1)

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    #print(num1/num2)
    print(num1*num3)
    print(nums[100])
except ZeroDivisionError:
    print('Error發生，除以0')
except NameError:
    print('Error發生，使用沒有宣告過的變數')
except:
    print('Error發生')
```

Ln: 14 Col: 0

try3.py - D:\PythonJunior\Examples\Ch7\try3.py (3.8.1)

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    #print(num1/num2)
    #print(num1*num3)
    print(nums[100])
except ZeroDivisionError:
    print('Error發生，除以0')
except NameError:
    print('Error發生，使用沒有宣告過的變數')
except:
    print('Error發生')
```

Ln: 14 Col: 0

程式範例

try3-1.py - D:\APCSClass1\Examples\Ch7\try3-1.py (3.8.1)

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    print(num1/num2)
    print(num1*num3)
    print(nums[100])
except ZeroDivisionError:
    print('Error發生，除以0')
except (NameError,IndexError):
    print('發生NameError或IndexError')
```

Ln: 12 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
>>>
======= RESTART: D:\APCSClass1\Examples\Ch7\try3-1.py ======
Error發生，除以0
>>>
= RESTART: D:\APCSClass1\Examples\Ch7\try3-1.py
發生NameError或IndexError
>>>
= RESTART: D:\APCSClass1\Examples\Ch7\try3-1.py
發生NameError或IndexError
>>>
```

Ln: 12 Col: 4

try3-1.py - D:\APCSClass1\Examples\Ch7\try3-1.py (3.8.1)

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    #print(num1/num2)
    print(num1*num3)
    print(nums[100])
except ZeroDivisionError:
    print('Error發生，除以0')
except (NameError,IndexError):
    print('發生NameError或IndexError')
```

Ln: 12 Col: 0

try3-1.py - D:\APCSClass1\Examples\Ch7\try3-1.py (3.8.1)

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    #print(num1/num2)
    #print(num1*num3)
    print(nums[100])
except ZeroDivisionError:
    print('Error發生，除以0')
except (NameError,IndexError):
    print('發生NameError或IndexError')
```

Ln: 12 Col: 0

try else

- else 區塊代表 try 區塊沒有產生例外時執行的區塊。

- 只有在try 區塊內的程式碼是被保護的
  - except / else 區塊內程式碼產生例外狀況時仍會造成程式中斷。
  - 可使用巢狀try結構保護

程式範例

try4.py - D:\PythonJunior\Exam...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    print(num1/(num2+1))
    print(num1*num2)
    print(nums[4])
except:
    print('Error發生')
else:
    print('Error沒發生')
```

Ln: 7 Col: 14

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
>>>
= RESTART: D:\PythonJunior\
Examples\Ch7\try4.py
10.0
0
9
Error沒發生
>>>
```

Ln: 35 Col: 4

程式範例

try5.py - D:\PythonJunior\Exam...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    print(num1/(num2+1))
    print(num1*num2)
except:
    print('Error發生')
else:
    print(nums[100])
    print('Error沒發生')
```

Ln: 12 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
Ch7\try5.py
10.0
0
Traceback (most recent call last):
  File "D:\PythonJunior\Examples\Ch7
\try5.py", line 10, in <module>
    print(nums[100])
IndexError: list index out of range
>>>
```

Ln: 49 Col: 4

try finally

- finally 區塊代表不論 try 區塊執行後一定要執行的區塊。
  - 不論有沒有產生例外，例外有沒有被處理都會執行
  - 程式中產生的例外沒有被處理而導致中斷，finally 區塊會在中斷前執行。

程式範例

try6.py - D:\PythonJunior\Examples\Ch7\try6.py (3.8.1)

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    print(num1/num2)
    print(num1*num3)
    print(nums[100])
except ZeroDivisionError:
    print('Error發生，除以0')
except NameError:
    print('Error發生，使用沒有宣告過的變數')
except IndexError:
    print('Error發生，索引值超出範圍')
except:
    print('Error發生')
finally:
    print('結束')
```

Ln: 17 Col: 15

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
>>>
======= RESTART: D:\PythonJunior
\Examples\Ch7\try6.py ======
Error發生，除以0
結束
>>>
```

Ln: 11 Col: 4

程式範例

try7.py - D:\PythonJunior\Ex...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    print(num1/(num2+1))
    print(num1*num2)
    print(nums[4])
except:
    print('Error發生')
else:
    print('Error沒發生')
finally:
    print('結束')
```

Ln: 14 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
>>>
======= RESTART: D:\PythonJunior
\Examples\Ch7\try7.py ======
10.0
0
9
Error沒發生
結束
>>>
```

Ln: 10 Col: 4

程式範例

try8.py - D:\PythonJunior\Exa...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    print(num1/(num2+1))
    print(num1*num2)
except:
    print('Error發生')
else:
    print(nums[100])
    print('Error沒發生')
finally:
    print('結束')
```

Ln: 13 Col: 15

Python 3.8.1 Shell

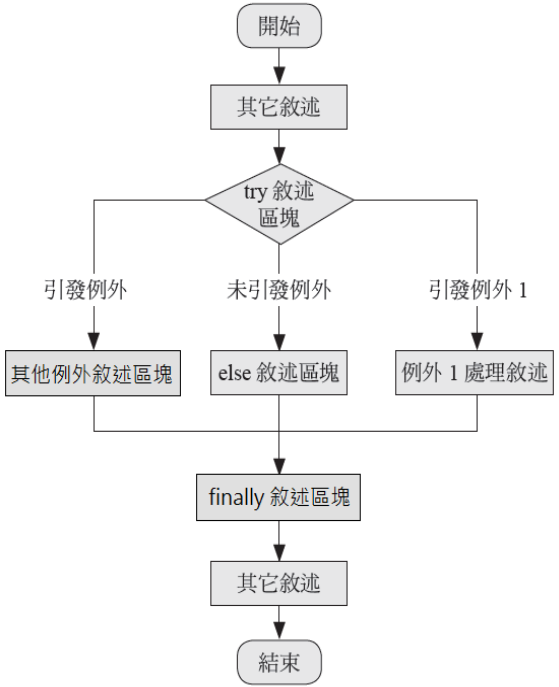
File Edit Shell Debug Options Window Help

```
mples\Ch7\try8.py ======
10.0
0
結束
Traceback (most recent call last):
  File "D:\PythonJunior\Examples\Ch7
\try8.py", line 10, in <module>
    print(nums[100])
IndexError: list index out of range
>>>
```

Ln: 12 Col: 4

例外處理語法總結

```
其他敘述
try:
    保護的程式區段
    .....
except 例外1 :
    例外1發生時執行的程式
except :
    其他例外發生時執行的程式
    #未指定例外型態，捕捉所有例外物件
else:
    若try程式區段沒產生例外執行的程式
finally:
    不管有沒有發生例外都會執行的程式
其他敘述
```



Q：下列關於例外處理的描述何者正確?

a) try 結構中，try範圍內發生錯誤，一定會執行except區段內容

b) try 結構中，try範圍內沒有錯誤，except區段內容不會執行

c) try 結構中，不論有沒有發生錯誤，else區段內容一定會執行

d) try 結構中，不論有沒有發生錯誤，finally區段內容一定會執行

Q：下列哪個選項可修正程式碼執行時發生的錯誤？

```
prices = [1.00, '3.50', 5.05]
total = 0

for item in prices:
    total += item

print(total)
```

a) total = total + item

b) total += int(item)

c) total += str(item)

d) total += float(item)

Q：下列程式碼執行結果為何？

```
proLang = {1: 'Java', 2: 'Python', 3: 'C'}

for i in range(0,3):
    print(proLang[i])
```

a) Java
   Python
   C

b) 發生KeyError

c) 發生NameError

d) 發生IndexError

課程內容

1. **例外處理基礎**
   - 1-1.程式偵錯
   - 1-2.例外處理機制
   - 1-3.例外處理語法

2. **例外進階處理**
   - 2-1. 例外說明
   - 2-2. 堆疊呼叫例外處理
   - 2.3. 手動拋出例外

例外說明

- 取得例外物件資訊
  - 使except捕捉到例外後，將例外物件指定給變數
  - 用 type(), str() 取得物件類別與訊息message

```
except ValueError as e：
        print(type(e), str(e))
```

  - 多個例外有相同處理邏輯，括號中用 ', ' 隔開指定同名變數

```
except (ValueError, ZeroDivisionError) as e：
        print(type(e), str(e))
```

  - 任何例外

```
except Exception as e：
        print(type(e), str(e))
```

程式範例

try9.py - D:\APCSClass1\Examples\Ch7\try9.p...

File Edit Format Run Options Window Help

```
num1 = 10
num2 = 0
nums = [1,3,5,7,9]
try:
    print(num1/num2)
    print(num1*num3)
    print(nums[100])
except Exception as ex:
    print("例外類別:", type(ex))
    print("例外訊息:", str(ex))
finally:
    print('結束')
```

Ln: 13 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

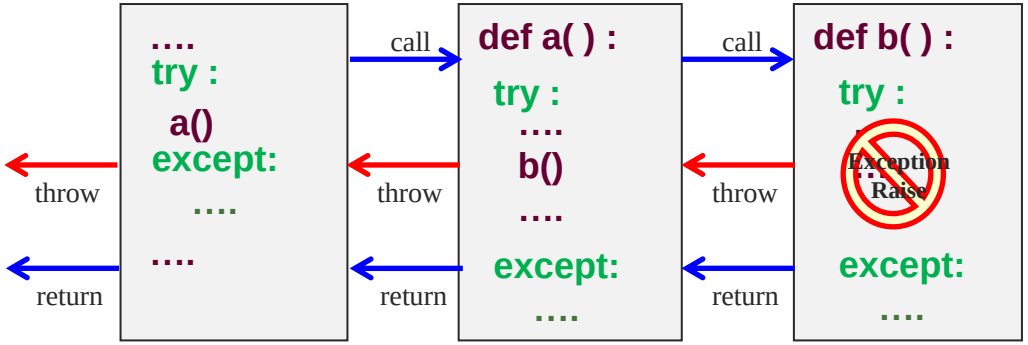
```
>>>
= RESTART: D:\APCSClass1\Examples\Ch7\try9.py
例外類別: <class 'ZeroDivisionError'>
例外訊息: division by zero
結束
>>>
= RESTART: D:\APCSClass1\Examples\Ch7\try9.py
例外類別: <class 'NameError'>
例外訊息: name 'num3' is not defined
結束
>>>
= RESTART: D:\APCSClass1\Examples\Ch7\try9.py
例外類別: <class 'IndexError'>
例外訊息: list index out of range
結束
>>>
```

Ln: 16 Col: 4

例外說明

- sys.exc_info()方法取得進階的例外資訊
  - 傳回一個Tuple物件
  - 包括三個內容
    - 例外的類型、例外訊息以及traceback物件
  - trackback 物件需使用標準庫 traceback 模組取得其資訊
    - print_tb(traceback)
    - print_exception(cls, value, traceback)
    - print_exc(cls, value, traceback)
    - format_exc()

函式/方法呼叫堆疊與例外處理



程式範例

exception2.py - D:\PythonJunior\Examples\Ch7\exception2.p...

File Edit Format Run Options Window Help

```
import sys
import traceback

def b():
    print('執行b()')
    num1 = 10
    num2 = 0
    try:
        print(num1/num2)
    except:
        cls, msg, trace = sys.exc_info()
        print(cls)
        print(msg)
        traceback.print_tb(trace)

    print('b()結束')

def a():
    print('執行a()')
    b()
    print('a()結束')


print('執行main()')
a()
print('main()結束')
```

Ln: 27 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
>>>
==== RESTART: D:\PythonJunior\Examples\Ch7\exception2.py ===
執行main()
執行a()
執行b()
<class 'ZeroDivisionError'>
division by zero
  File "D:\PythonJunior\Examples\Ch7\exception2.py", line 9, in b
    print(num1/num2)
b()結束
a()結束
main()結束
>>>
```

Ln: 15 Col: 4

程式範例

exception3.py - D:\PythonJunior\Examples\Ch7\except...

File Edit Format Run Options Window Help

```
import sys
import traceback

def b():
    print('執行b()')
    num1 = 10
    num2 = 0
    print(num1/num2)
    print('b()結束')

def a():
    print('執行a()')
    b()
    print('a()結束')


print('執行main()')
try:
    a()
except:
    cls, msg, trace = sys.exc_info()
    print(cls)
    print(msg)
    traceback.print_tb(trace)
print('main()結束')
```

Ln: 26 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
>>>
====== RESTART: D:\PythonJunior\Examples\Ch7\exception3.py ======
執行main()
執行a()
執行b()
<class 'ZeroDivisionError'>
division by zero
  File "D:\PythonJunior\Examples\Ch7\exception3.py", line 19, in
<module>
    a()
  File "D:\PythonJunior\Examples\Ch7\exception3.py", line 13, in
a
    b()
  File "D:\PythonJunior\Examples\Ch7\exception3.py", line 8, in b
    print(num1/num2)
main()結束
>>>
```

Ln: 29 Col: 4

手動拋出例外

- 自行拋出的例外，例外機制也可以處理
  - 拋出例外語法
    - 拋出自行建構例外

raise 異常類別

    - 拋出包含指定訊息例外

raise 異常類別(message)

重拋出例外

- 拋出例外語法
  - 重拋補捉到的例外

raise 異常物件 / raise

  - 拋出一個新建例外，鏈接到所補捉到的例外

raise 異常型別名或物件 from 異常物件

程式範例

try10.py - D:\PythonJunior\Examples\C...

File Edit Format Run Options Window Help

```
def getResult(s):
    if 60<=score<=100:
        return '及格'
    elif(0<=score<60):
        return '不及格'
    else:
        raise OverflowError

score=int(input('輸入成績:'))
try:
    res = getResult(score)
except OverflowError:
    print("成績數值錯誤")
else:
    print("考試結果:", res)
```

Ln: 16 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
>>>
= RESTART: D:\PythonJunior\
Examples\Ch7\try10.py
輸入成績:90
考試結果: 及格
>>>
= RESTART: D:\PythonJunior\
Examples\Ch7\try10.py
輸入成績:55
考試結果: 不及格
>>>
= RESTART: D:\PythonJunior\
Examples\Ch7\try10.py
輸入成績:-50
成績數值錯誤
>>>
```

Ln: 24 Col: 4

程式範例

try10-1.py - D:\APCSClass1\Examples\Ch7\try10-1.py (3.8.1)

File Edit Format Run Options Window Help

```
def getResult(s):
    if 60<=score<=100:
        return '及格'
    elif(0<=score<60):
        return '不及格'
    else:
        raise OverflowError("成績數值錯誤")

score=int(input('輸入成績:'))
try:
    res = getResult(score)
except Exception as e:
    print(e)
else:
    print("考試結果:", res)
```

Ln: 16 Col: 0

Python 3.8.1 Shell

File Edit Shell Debug Options Window Help

```
>>>
= RESTART: D:\APCSClass1\
Examples\Ch7\try10-1.py
輸入成績:80
考試結果: 及格
>>>
= RESTART: D:\APCSClass1\
Examples\Ch7\try10-1.py
輸入成績:50
考試結果: 不及格
>>>
= RESTART: D:\APCSClass1\
Examples\Ch7\try10-1.py
輸入成績:110
成績數值錯誤
>>>
```

Ln: 13 Col: 4

Q：程式碼中加入哪個選項可將補捉到的例外重拋？

```
try :
    x = 1/0
except Exception as e:
    print(e)
    # 加入程式
```

a) raise as e

b) raise

c) raise ValueError(e)

d) raise ValueError('Error Division by Zero')

Q：哪個程式片段可在例外區段中捕捉多個例外？

```
def division(a, b):
    return a / b
try:
    division(4,0)
    division("3","4")
# 加入程式
    print("exception caught %s" %e)
```

a) except (ZeroDivisionError | TypeError) as e:

b) except (ZeroDivisionError, TypeError) from e:

c) except (ZeroDivisionError, TypeError) as e:

d) except ZeroDivisionError, TypeError as e:

練習：BMI 計算

- 修改 BMI 程式，加入例外處理機制避免下列問題造成程式中斷
  - 輸入非數值的身高體重
  - 身高為負數或0
  - 體重為負數或0
  - …

巨匠直播教學

www.pcschoolonline.com.tw